ENRKA-1.1

IDC SUBSYSTEM
FUNCT DIAGNOSTIC

EP-ENRKA-DL-1.1
1 OF 3 JUL 1986

digital
MADE IN USA

ENRKA-1.1

IDC SUBSYSTEM
FUNCT DIAGNOSTIC

EP-ENRKA-DL-1.1
2 OF 3 JUL 1986

digital
MADE IN USA

ENRKA-1.1

IDC SUBSYSTEM
FUNCT DIAGNOSTIC

EP-ENRKA-DL-1.1
3 OF 3 JUL 1986

digital
MADE IN USA

VAX DIAGNOSTIC

Product ID: ZZ-ENRKA - 1.1

1 ABSTRACT

This program contains a set of tests which will verify the functional integrity of the IDC controller and its attacted disk Drives. This is a diagnostic supervisor based program which can be run in standalone mode (Level 3) only using direct I/O. The program test flow is designed to begin with simplified operations and then progressing to more sofisticated operations.

2 HARDWARE REQUIREMENTS

2.1 Equipment

VAX 11/730 CPU
MS730 with 256 KB of memory
Integrated disk controller (IDC)
Up to four RL02's or up to 3 RL02's and one R80 disk drive connected to the IDC.

2.2 Media

In order to protect user data, the program requires that IDC disk cartridges have a volume name of 'SCRATCH' in either ODS1 or ODS2 format, installed on all drives selected. If the volume name 'SCRATCH' is not found, or if an error occurs reading the home block, the program will prompt the operator regarding whether to continue testing on that unit. When R80 disk drives are selected for testing all write operations default to the FE maintence cylinders which are not used for customer data. The only test that will use other areas of the media is the drive activity test and to accomplish this event flag 20 must be set by the operator before the diagnostic is started. Otherwise this test will also default to the FE cylinders

3 SOFTWARE REQUIREMENTS

Before running this diagnostic the IDC should have successfully passed the IDC micro diagnostics ENXXX and ENXXX

4 PREREQUISITES

This program expects all elements in the CPU cluster to be fully operational. The diagnostic supervisor must be loaded and is the operator interface.

5 LOADING AND STARTING

SAMPLE DEVICE ATTACH SEQUENCE

```
DS> ATTACH DW780 HUB DW0          ! ATTACH UNIBUS ADAPTER (SEE NOTE)
DS> ATTACH RB730 DW0 DQA          ! ATTACH IDC
DS> ATTACH R80 DQA DQA0           ! ATTACH AN R80...DON'T ATTATCH AN RM80
DS> ATTACH RL02 DQA DQA1          ! ATTACH AN RL02
DS> SEL DQA0,DQA1                 ! SELECT DRIVES FOR TESTING
DS> RUN ENRKA                     ! LOAD AND START DIAGNOSTIC
```

6 PROGRAM DESCRIPTION

6.1 Testing Sequence

All devices selected will be tested in any given test before proceeding to the next test. In other words all drives will be tested in test 10 before proceeding to test 11 and so on.

6.2 Program Size

ENRKA has a current size of 57.5 kbytes or 112 pages of memory

6.3 Program Run Time Dynamics

Many buffers utilized by this program are internal to itself, but in the case of the maximun data transfer test the program is required to obtain 32 additional pages of memory for large DMA transfers.

6.4 Performance Reports

Summary reports are not included in this program.

6.5 Catastrophic Hardware Failures

Servere hardware failures from other necessary system components will be signalled by the diagnostic supervisor or through this program with a system error message followed by an abort sequence, while servere failures on the drives selected for testing will cause the program to print a device error message and abort.

6.6 Event Flag Usage

All event flags refer to test number 43 which is the subsystem disk activity test. These event flags are present so the user may have a tool to somewhat vary the operation of the drive under test for debugging purposes. The event flags are as follows:

```
        Event flag 20 - normally reset
                When set causes the activity test to use
                the entire portion of the media not just
                the FE cylinders.
                ** This flag will destroy customer data **

        Event flag 10 - normally reset
                when set directs the activity test not to
                perform read header operations, thus the IDC
                must 'memorize' current locations for RL02
                drives.

        Event flag 11 - normally reset
                When set directs the activity test to delete
                write and write check operations.

        Event flag 12 - normally reset
                When set directs the activity test to delete
                write check operations

        Event flag 13 - normally reset
                When set directs the activity test to delete
                read operations.
```

6.7 Execution Time

Run Time for the default tests is approximiately 7 minutes per device selected for testing.

6.8 Section Descriptions

This program contains 2 test subsections each of which can be selected for execution via the /SECTION switch in the start command line.

These test sections are:

```
A.   Default - Runs all default tests.
           /SECTION switch not necessary.
C.   Manual - Runs only manual intervention tests
D.   All - Run all sections of the diagnostic
```

6.9 ERROR REPORTING

The following is a sample error message and how to control error reporting through the Diagnostic supervisor.

```
------------------------->SECTION ONE<----------------------------

********  ENRKA IDC FUNCTIONAL DIAGNOSTIC - 1.0  ********
 PASS 1  TEST 16  SUBTEST 0  ERROR 5   10-OCT-1981 11:02:26.43
 HARD ERROR WHILE TESTING DQA0:   ERROR OCCURED EXECUTING SEEK COMMAND
------------------------->SECTION TWO<----------------------------

DRIVE READY FAILED TO SET

REGISTER                DATA              MNEMONICS

IDCCSR  [40F26200]:     00010086(X)       ; ATN0,CONT__ERR=NONE,DRIVE=0(D),DRDY,FUNCTION=SEEK
IDCBAR  [40F26204]:     00000000(X)       ; BUS ADDRESS=0000(X)
IDCCNT  [40F26208]:     00000000(X)       ; BYTE COUNT =000(X)
IDCDAR  [40F2620C]:     0000EE85(X)       ; DIFFERENCE=477(D),HEAD=UPPER,DIR=FORWARD,MK0
IDCMPR  [40F26210]:     0000009D(X)       ; RL02,HEAD=UPPER,CO,BH,STATE=SPINNING DOWN
IDCPOS  [40F26214]:     00000000(X)       ; ECC POSITION = 0(D)
IDCPAT  [40F26218]:     00000000(X)       ; ECC PATTERN = 0(X)
------------------------->SECTION THREE<--------------------------

PREVIOUS CYLINDER WAS:    017(D)
DIFFERENCE WORD WAS:      447(D)
DESTINATION CYLN WAS:     494(D)
```

Setting IE1 flag in the supervisor deletes sections 1, 2, and 3
Setting IE2 flag in the supervisor deletes sections 2 and 3
Setting IE3 flag in the supervisor deletes section 3

7 PROGRAM SOURCES

THE FOLLOWING FILES ARE THOSE THAT ARE REQUIRED FOR FUTURE PROGRAM MAINTENANCE AND SUBSEQUENT RE-RELEASE:

1. ENRKA0.MAR
2. ENRKA1.MAR
3. ENRKA2.MAR
4. ENRKA3.MAR
5. ENRKA.RNO
6. ENRKA.MAR (.MLB)

8 MAINTENANCE HISTORY

V1.0 March 1982

Initial Release V1.1 FEBRUARY 1986
CHANGE TEST 3 DATA MASK

9 TEST DESCRIPTIONS

```
        <IDC PRIMARY ADDRESS TEST>
;++
;
; TEST DESCRIPTION:
;
; THE PURPOSE OF THIS TEST IS TO INSURE PROPER COMMUNICATION
; BETWEEN THE CPU AND THE IDC CONTROLLER.  THIS IS ACCOMPLISHED
; BY READING EACH OF THE IDC REGISTERS.
;
; TEST STEPS:
;
; 1. PROBE IDC REGISTER
; 2. RETURN TO STEP 1. UNTIL ALL IDC REGISTERS HAVE BEEN ACCESSED
;
; DEBUG:
;
; IF THIS TEST FAILS PROBLEM IS LOCATED IN IDC ADDRESS DECODE LOGIC
;
;--

        <IDC SECONDARY ADDRESS TEST>

;++
;
; TEST DESCRIPTION:
;
; THE PURPOSE OF THIS TEST IS TO INSURE PROPER COMMUNICATION
; BETWEEN THE CPU AND THE IDC VIA ITS UNIBUS OR SECONDARY ADDRESS.
; THE ONLY REGISTER AVAILABLE ON THE UNIBUS IS THE LOWER 16 BITS
; OF THE IDCCSR.
;
; TEST STEPS:
;
; 1. ISSUE ADAPTER INIT
; 2. CLEAR UBA STATUS
; 3. PROBE IDC REGISTER
; 4. RETURN TO STEP 2. UNTIL ALL IDC REGISTERS HAVE BEEN ACCESSED
;
; DEBUG:
;
; IF THIS TEST FAILS PROBLEM IS LOCATED IN IDC ADDRESS DECODE LOGIC
; OR WITH UBA ADAPTER LOGIC.
;
;--
```

```
        <IDCCSR REG DATA TEST>

;++
;
; TEST DESCRIPTION:
;
; THIS TEST WILL CHECK THE DATA INTEGRITY OF THE CONTROL AND STATUS
; REGISTER ON A BIT BY BIT BASIS.  REGISTER WILL BE TESTED FOR HOLDING
; ALL ONES AND ALL ZEROS. THE IDCCSR IS ALSO TESTED WITH A FLOATING ONES
; AND A FLOATING ZEROS PATTERN.  ONCE THE DATA INTEGRITY HAS BEEN
; VERIFIED THE TEST CHECKS THAT THE IDCCSR CAN BE LOADED USING A READ
; MODIFY WRITE INSTRUCTION.
;
; TEST STEPS:
;
; A. SUBTEST 1
;       1. WRITE IDCCSR WITH ONES
;       2. READ IDCCSR
;       3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
; B. SUBTEST 2
;       1. WRITE IDCCSR WITH ZEROS
;       2. READ IDCCSR
;       3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
; C. SUBTEST 3
;       1. WRITE IDCCSR USING FLOATING ONES PATTERN
;       2. READ IDCCSR
;       3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
;       4. RETURN TO STEP 1. UNTIL ALL BITS HAVE BEEN TESTED
; D. SUBTEST 4
;       1. WRITE IDCCSR WITH FLOATING ZERO PATTERN
;       2. READ IDCCSR
;       3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
;       4. RETURN TO STEP 1. UNTIL ALL BITS HAVE BEEN TESTED
; E. SUBTEST 5
;       1. LOAD IDCCSR WITH ONES USING A BIT SET INSTRUCTION
;       2. READ IDCCSR
;       3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
;
; DEBUG:
;
; IF THIS TEST FAILS THE IDCCSR REGISTER LOGIC SHOULD BE CHECKED
; AND ALSO THE REGISTER DECODE LOGIC.
;
;--
```

```
        <IDCBAR REG DATA TEST>

;++
;
; TEST DESCRIPTION:
;
; THIS TEST WILL CHECK THE DATA INTEGRITY OF THE BUS ADDRESS
; REGISTER ON A BIT BY BIT BASIS.  REGISTER WILL BE TESTED FOR HOLDING
; ALL ONES AND ALL ZEROS. THE IDCBAR IS ALSO TESTED WITH A FLOATING ONES
; AND A FLOATING ZEROS PATTERN.  ONCE THE DATA INTEGRITY HAS BEEN
; VERIFIED THE TEST CHECKS THAT THE IDCBAR CAN BE LOADED USING A READ
; MODIFY WRITE INSTRUCTION.
;
; TEST STEPS:
;
; A. SUBTEST 1
;       1. WRITE IDCBAR WITH ONES
;       2. READ IDCBAR
;       3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
; B. SUBTEST 2
;       1. WRITE IDCBAR WITH ZEROS
;       2. READ IDCBAR
;       3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
; C. SUBTEST 3
;       1. WRITE IDCBAR USING FLOATING ONES PATTERN
;       2. READ IDCBAR
;       3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
;       4. RETURN TO STEP 1. UNTIL ALL BITS HAVE BEEN TESTED
; D. SUBTEST 4
;       1. WRITE IDCBAR WITH FLOATING ZERO PATTERN
;       2. READ IDCBAR
;       3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
;       4. RETURN TO STEP 1. UNTIL ALL BITS HAVE BEEN TESTED
; E. SUBTEST 5
;       1. LOAD IDCBAR WITH ONES USING A BIT SET INSTRUCTION
;       2. READ IDCBAR
;       3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
;
; DEBUG:
;
; IF THIS TEST FAILS THE IDCBAR REGISTER LOGIC SHOULD BE CHECKED
; AND ALSO THE REGISTER DECODE LOGIC.
;
;--
```

```
        <IDCCNT REG DATA TEST>

;++
;
; TEST DESCRIPTION:
;
; THIS TEST WILL CHECK THE DATA INTEGRITY OF THE BYTE COUNT
; REGISTER ON A BIT BY BIT BASIS.  REGISTER WILL BE TESTED FOR HOLDING
; ALL ONES AND ALL ZEROS. THE IDCCNT IS ALSO TESTED WITH A FLOATING ONES
; AND A FLOATING ZEROS PATTERN.  ONCE THE DATA INTEGRITY HAS BEEN
; VERIFIED THE TEST CHECKS THAT THE IDCCNT CAN BE LOADED USING A READ
; MODIFY WRITE INSTRUCTION.
;
; TEST STEPS:
;
; A. SUBTEST 1
;       1. WRITE IDCCNT WITH ONES
;       2. READ IDCCNT
;       3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
; B. SUBTEST 2
;       1. WRITE IDCCNT WITH ZEROS
;       2. READ IDCCNT
;       3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
; C. SUBTEST 3
;       1. WRITE IDCCNT USING FLOATING ONES PATTERN
;       2. READ IDCCNT
;       3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
;       4. RETURN TO STEP 1. UNTIL ALL BITS HAVE BEEN TESTED
; D. SUBTEST 4
;       1. WRITE IDCCNT WITH FLOATING ZERO PATTERN
;       2. READ IDCCNT
;       3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
;       4. RETURN TO STEP 1. UNTIL ALL BITS HAVE BEEN TESTED
; E. SUBTEST 5
;       1. LOAD IDCCNT WITH ONES USING A BIT SET INSTRUCTION
;       2. READ IDCCNT
;       3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
;
; DEBUG:
;
; IF THIS TEST FAILS THE IDCCNT REGISTER LOGIC SHOULD BE CHECKED
; AND ALSO THE REGISTER DECODE LOGIC.
;
;--
```

```
        <IDCDAR REG DATA TEST>

;++
;
; TEST DESCRIPTION:
;
; THIS TEST WILL CHECK THE DATA INTEGRITY OF THE DISK ADDRESS
; REGISTER ON A BIT BY BIT BASIS.  REGISTER WILL BE TESTED FOR HOLDING
; ALL ONES AND ALL ZEROS. THE IDCDAR IS ALSO TESTED WITH A FLOATING ONES
; AND A FLOATING ZEROS PATTERN.  ONCE THE DATA INTEGRITY HAS BEEN
; VERIFIED THE TEST CHECKS THAT THE IDCDAR CAN BE LOADED USING A READ
; MODIFY WRITE INSTRUCTION. THE SUBTESTS THREE AND FOUR USE THE MULTI
; PURPOSE REGISTER TO WRITE THE IDCDAR REGISTER SO THAT THE IDC
; MICRO CODE WILL NOT PERFORM ANY MODIFICATIONS ON THE DATA
;
; TEST STEPS:
;
; A. SUBTEST 1
;       1. WRITE IDCDAR WITH ONES
;       2. READ IDCDAR
;       3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
; B. SUBTEST 2
;       1. WRITE IDCDAR WITH ZEROS
;       2. READ IDCDAR
;       3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
; C. SUBTEST 3
;       1. WRITE IDCMPR USING FLOATING ONES PATTERN
;       2. READ IDCDAR
;       3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
;       4. RETURN TO STEP 1. UNTIL ALL BITS HAVE BEEN TESTED
; D. SUBTEST 4
;       1. WRITE IDCMPR WITH FLOATING ZERO PATTERN
;       2. READ IDCDAR
;       3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
;       4. RETURN TO STEP 1. UNTIL ALL BITS HAVE BEEN TESTED
; E. SUBTEST 5
;       1. LOAD IDCDAR WITH ONES USING A BIT SET INSTRUCTION
;       2. READ IDCDAR
;       3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
;
; DEBUG:
;
; IF THIS TEST FAILS THE IDCDAR REGISTER LOGIC SHOULD BE CHECKED
; AND ALSO THE REGISTER DECODE LOGIC.
;
;--
```

```
        <GET STATUS FUNCTION TEST>

;++
;
; TEST DESCRIPTION:
;
; THIS TEST WILL EXECUTE A GET STATUS COMMAND (WITH THE RST BIT
; SET IN THE IDCMPR REGISTER IF DRIVE IS AND RL02) AND CHECK THAT
; NO ERRORS OCCUR. THEN THE IDCMPR REGISTER CONTAINING THE DRIVE
; STATUS IS CHECKED FOR THE PROPER STATE.
;
; TEST STEPS:
;
; 1.    INITIALIZE SUBSYSTEM
; 2.    LOAD IDCMPR REGISTER FOR GET STATUS FUNCTION
; 3.    LOAD IDCCSR REGISTER AND EXECUTE GET STATUS COMMAND
; 4.    WAIT FOR COMMAND COMPLETION AND CHECK FOR ANY ERRORS -
;       REPORT ERROR TO OPERATOR
; 5.    CHECK DRIVE STATUS VIA THE IDCMPR REGISTER - REPORT ANY DISCREPANCY
;
; DEBUG:
;
;--

        <NO-OP FUNCTION TEST>

;++
;
; TEST DESCRIPTION:
;
; THIS TEST WILL VERIFY THE FUNCTIONALITY OF THE NO OP COMMAND BY
; LOADING THE IDCBAR, IDCCNT  AND IDCDAR REGISTERS WITH DATA.
; THEN EXECUTE THE NO OP COMMAND AND VERIFY THAT NO ERRORS OCCUR
; AND THE IDCBAR, IDCCNT AND IDCDAR REGISTERS REMAIN UNCHANGED.
;
; TEST STEPS:
;
; 1.    INITIALIZE SUBSYTEM
; 2.    LOAD IDCBAR, IDCCNT AND IDCDAR REGISTERS
; 3.    EXECUTE NO OP CMD
; 4.    WAIT FOR CRDY - IF NOT SET REPORT ERROR
; 5.    TEST FOR ANY ERRORS - REPORT ERROR IF ANY
; 6.    CHECK IDCBAR FOR ANY MODIFICATION - REPORT ERROR IF ANY
; 7.    CHECK IDCDAR FOR ANY MODIFICATION - REPORT ERROR IF ANY
; 8.    CHECK IDCCNT FOR ANY MODIFICATION - REPORT ERROR IF ANY
;
; DEBUG:
;
;       COMMAND DECODE LOGIC
;
;--
```

```
        <INTERRUPT TEST>

;++
;
; TEST DESCRIPTION:
;
; THIS TEST WILL VERIFY THAT THE IDC CONTROLLER IS ABLE TO COMPLETE
; THE INTERRUPT SEQUENCE PROPERLY.  THIS IS DONE BY EXECUTING A NO OP
; COMMAND WITH THE INTERRUPT ENABLE BIT (IE) SET IN THE CONTROL AND
; STATUS REGISTER.  AFTER INTERRUPT HAS OCCURED CHECK PROPER STATUS.
;
; TEST STEPS:
;
; 1.    INITIALIZE SUBSYSTEM
; 2.    CLEAR INTERRUPT STATUS AND ENABLE INTERRUPTS
; 3.    EXECUTE NO OP COMMAND
; 4.    WAIT FOR INTERRUPT
; 5.    IF INTERRUPT DID NOT OCCUR REPORT ERROR
; 6.    CHECK FOR ANY ERRORS FOLLOWING INTERRUPT
;       REPORT ANY FAILURES
; 7.    DISABLE INTERRUPTS
;
; DEBUG:
;
; IDC INTERRUPT LOGIC
;
;--
```

```
      <READ HEADER AND HEADER CRC TEST>

;++
;
; TEST DESCRIPTION:
;
; THIS TEST WILL VERIFY THE READ HEADER COMMAND BY EXECUTING 40 READ
; HEADER COMMANDS.  AFTER EACH READ HEADER CONTROLLER AND DRIVE STATUS
; ARE CHECKED FOR ANY ERRORS. HEADER WORDS ARE CHECKED FOR PROPER
; FORMAT AND THE HEADER WORD IS ALSO VALIDATED.
;
; TEST STEPS:
;
; 1.     INITIALIZE SUBSYSTEM
; 2.     EXECUTE READ HEADER COMMAND
; 3.     CHECK FOR ANY STATUS ERRORS
; 4.     TEST HDR WORD #1 ALWAYS YIELDS SAME CYLINDER
; 5.     TEST HDR WORD #1 ALWAYS YIELDS SAME HEAD
; 6.     TEST HDR WORD #2 IS EQUAL TO ZERO
; 7.     CALCULATE EXPECTED CRC WORD
; 8.     TEST THAT ACTUAL CRC IN HDR WORD #3 = EXPECTED CRC
;
; DEBUG:
; SINCE WE DO NOT KNOW THAT THE HEADER CRC CHECKING LOGIC IN THE
; CONTROLLER IS WORKING THE SOFTWARE CHECKING OF THE CRC SHOULD
; CONFIRM THE ABILITY OF THE SUBSYSTEM TO READ GOOD HEADERS.
;
;
;--
```

```
        <RECAL FUNCTION TEST - R80>

;++
;
; TEST DESCRIPTION:
;
; THIS TEST IS UTILIZED FOR THE R80 DISK DRIVE ONLY. THE PURPOSE
; OF THE TEST IS TO EXECTUTE A RECAL (SEEK TO CYLINDER 0) AND
; CHECK FOR ANY COMPORMISING CONTROLLER/DRIVE ERRORS.
;
; TEST STEPS:
;
; 1.    INITIALIZE SUBSYSTEM
; 2.    EXECUTE RECAL FUNCTION
; 3.    REPORT ANY ANY POSITIONING, DRIVE, OR CONTROLLER ERRORS
;
; DEBUG:
;
;       NONE
;
;--

        <SEEK ERROR TEST - R80>

;++
;
; TEST DESCRIPTION:
;
; THIS TEST IS UTILIZED FOR THE R80 DISK DRIVE ONLY. THE PURPOSE
; OF THIS TEST IS TO ISSUE A SEEK COMMAND TO THE R80 DISK DRIVE
; USING CYLINDER 561 WHICH IS AN ILLEGAL CYLINDER.  THIS WILL CAUSE
; THE R80 TO GENERATE A SKE ERROR IN THE DRIVE STATUS WORD. THEN
; EXECUTE A RECAL FUNCTION AND CHECK THAT THE DRIVE RECOVERS PROPERLY.
;
; TEST STEPS:
;
; 1.    INITIALIZE SUBSYSTEM
; 2.    ISSUE SEEK TO CYLINDER 561
; 3.    CHECK DRIVE STATUS IS CORRECT (SKE)
; 4.    EXECUTE RECAL FUNCTION
; 5.    REPORT ANY POSITIONING, DRIVE, OR CONTROLLER ERRORS
;
; DEBUG:
;
;       NONE
;
;--
```

```
        <BASIC SEEK FUNCTION TEST>

;++
;
; TEST DESCRIPTION:
;
; THIS TEST CHECKS THE BASIC POSITIONING LOGIC IN THE SUBSYSTEM
; BUT WITH NO DRIVE MOVEMENT.  A READ HEADER COMMAND IS EXECUTED
; AND THE CURRENT POSITION OF THE HEADS IS SAVED.  SEEK COMMANDS
; ARE THEN ISSUED WITH A DIFFERENCE OF ZERO TO BOTH HEADS IN BOTH
; DIRECTIONS.  AFTER EACH SEEK COMMAND A READ HEADER IS ISSUED
; ONCE AGAIN.  THE HEADER READ BACK MUST INDICATE NO CHANGE IN
; THE CURRENT POSITIONING OF THE HEADS AND THE PROPER HEAD MUST
; BE SELECTED.  THE IDCMPR REGISTER CONTAINING DRIVE STATUS IS ALSO
; CHECKED TO INSURE THAT IT INDICATES THAT THE PROPER HEAD WAS
; SELECTED.
;
;
; TEST STEPS:
;
; 1.      INITIALIZE SUBSYSTEM
; 2.      EXECUTE READ HEADER CMD
; 3.      CHECK FOR ANY ERRORS
; 4.      SAVE CYLINDER ADDRESS FROM HEADER WORD
; 5.      EXECUTE SEEK USING DATA FROM ARGUMENT TABLE
; 6.      CHECK FOR ANY ERRORS
; 7.      CHECK FOR PROPER DRIVE STATUS
; 8.      EXECUTE READ HEADER COMMAND
; 9.      CHECK FOR ANY ERRORS
; 10.     CHECK HEADER WORD SHOWS NO DRIVE MOVEMENT TOOK PLACE
; 11.     CHECK HEAD SELECT BIT IN HEADER WORD IS SAME AS HEAD
;         USED IN SEEK COMMAND
; 12.     CHECK HEAD SELECT BIT IN DRIVE STATUS WORD IS SAME AS
;         HEAD USED IN SEEK COMMAND.
; 13.     RETURN TO STEP ONE UNTIL ARGUMENT TABLE EXHAUSTED
;
; DEBUG:
;
;         DIFFERENCE REGISTER
;         COUNT LOGIC
;
;--
```

```
        <HEAD PROXIMITY TEST - RL02>

;++
;
; TEST DESCRIPTION:
;
;       THIS TEST WILL CHECK HEAD PROXIMITY BY EXECUTING ZERO
;       DIFFERENCE SEEKS TO BOTH HEADS AND VERIFY THAT THEY
;       ARE ON THE SAME CYLINDER.
;
; TEST STEPS:
;
;       1.  INITIALIZE SUBSYSTEM
;       2.  EXECUTE ZERO DIFFERENCE SEEK TO HEAD 0
;       3.  RECORD CURRENT POSITION OF HEADS
;       4.  EXECUTE ZERO DIFFERENCE SEEK TO HEAD 1
;       5.  GET CURRENT POSITION OF HEADS
;       6.  VERIFY THAT BOTH HEADS ARE FOLLOWING SAVE TRACK
;
; DEBUG
;
;       CHECK HEAD ALIGNMENT
;
;--
```

```
        <DIFFERENCE OF ONE SEEK TEST>

;++
;
; TEST DESCRIPTION:
;
; THIS TEST CHECKS THE POSITIONING LOGIC IN THE SUBSYSTEM
; WITH MINIMUM DRIVE MOVEMENT.  A READ HEADER COMMAND IS EXECUTED
; AND THE CURRENT POSITION OF THE HEADS IS SAVED.  SEEK COMMANDS
; ARE THEN ISSUED WITH A DIFFERENCE OF ONE IN BOTH
; DIRECTIONS.  AFTER EACH SEEK COMMAND A READ HEADER IS ISSUED
; ONCE AGAIN.  THE HEADER READ BACK MUST INDICATE A CHANGE IN
; THE CURRENT POSITIONING OF THE HEADS AND THE PROPER HEAD MUST
; BE SELECTED.
;
; NOTE: IF DRIVE IS AN RL02 THEN BOTH HEADS ARE TESTED HERE.
;
; TEST STEPS:
;
; 1.      INITIALIZE SUBSYSTEM
; 2.      EXECUTE READ HEADER CMD
; 3.      CHECK FOR ANY ERRORS
; 4.      SAVE CYLINDER ADDRESS FROM HEADER WORD
; 5.      EXECUTE SEEK USING DATA FROM ARGUMENT TABLE
; 6.      CHECK FOR ANY ERRORS
; 7.      CHECK THAT DRIVE READY IS RESET
; 8.      CHECK THAT DRIVE STATUS INDICATES DRIVE IS SEEKING
; 9.      WAIT FOR DRIVE READY
; 10.     GET DRIVE STATUS AND CHECK THAT DRIVE IS LOCKED ON TRACK
; 11.     EXECUTE READ HEADER COMMAND
; 12.     CHECK FOR ANY ERRORS
; 13.     CHECK THAT HEADER WORD #1 INDICATES DRIVE SEEKED TO RIGHT CYLINDER
; 14.     RETURN TO STEP ONE UNTIL ARGUMENT TABLE EXHAUSTED
;
; DEBUG:
;
;         DIFFERENCE REGISTER
;         COUNT LOGIC
;         TRACK CROSSING DETECTION
;
;--
```

```
        <INCREMENTAL FORWARD AND REVERSE SEEK TEST>

;++
;
; TEST DESCRIPTION:
;
; TEST ONE TRACK SEEKS WITH BOTH HEADS ACROSS ALL TRACKS.  ALSO
; INNER GUARD BAND DETECTION WILL BE CHECKED WHEN LAST CYLINDER
; IS REACHED IF THE DRIVE UNDER TEST IS AN RL02.
;
; TEST STEPS:
;
; A.  SUBTEST 1
;         1.  INITIALIZE SUBSYSTEM
;         2.  POSITION HEADS TO CYLINDER 0
;         3.  CHECK FOR ANY ERRORS
;         4.  EXECUTE +1 SEEK CMD
;         5.  CHECK FOR ANY ERRORS
;         6.  VERIFY SEEK WITH READ HEADER
;         7.  RETURN TO STEP 4. UNTIL LAST CYLINDER IS REACHED
;
; B.  SUBTEST 2
;         1.  INITIALIZE SUBSYSTEM
;         2.  POSITION HEADS TO LAST CYLINDER
;         3.  CHECK FOR ERRORS
;         4.  EXECUTE ANOTHER +1 SEEK CMD
;         5.  CHECKK FOR ERRORS
;         6.  VERIIFY DRIVE HAS REPOSITIONED TO LAST EVEN TRACK (510
;             FOR RL02) OR THAT DRIVE POSITION DID NOT CHANGE (R80)
;
; C.  SUBTEST 3
;         1.  INITIALIZE SUBSYSTEM
;         2.  POSITION HEADS TO LAST CYLINDER
;         3.  CHECK FOR ANY ERRORS
;         4.  EXECUTE -1 SEEK CMD
;         5.  CHECK FOR ANY ERRORS
;         6.  VERIFY SEEK WITH READ HEADER
;         7.  RETURN TO STEP 4. UNTIL CYLINDER 0 IS REACHED
;
; DEBUG:
;
;         DIFFERENCE LOGIC
;         TRACK CROSSING LOGIC
;
;--
```

```
        <DIFFERENCE WORD TRANSMISSION TEST>

;++
;
; TEST DESCRIPTION:
;
; THIS TEST WILL VERIFY THE TRANSMISSION AND DECODING OF THE DIFFERENCE
; WORD.  EACH SEEK WILL START AT CYLINDER ZERO AND RETURN THERE WITH A
; SEEK IN THE OPPOSIITE DIRECTION.  BOTH HEADS ARE TESTED IN THIS MANNER
; IF THE DRIVE CURRENTLY UNDER TEST IS AN RL02.
; PATTERNS USED FOR THE DIFFERENCE WORD ARE FLOATING ONES, FLOATING ZEROS,
; GROWING ONES AND GROWING ZEROS.
;
; TEST STEPS:
;
; 1.    INITIALIZE SUBSYSTEM
; 2.    GET CURRENT DRIVE POSITION
; 3.    IF AT CYLN 0 GO TO STEP 5
; 4.    SEEK HEADS BACK TO CYLN 0
;       CHECK FOR ANY ERRORS
; 5.    GET CYLINDER DESTINATION FROM CYLINDER LIST
; 6.    EXECUTE SEEK AND CHECK FOR ERRORS
; 7.    REVERSE DIRECTION AND SEEK BACK TO CYLN ZERO
; 8.    CHECK FOR ANY ERRORS
; 9.    IF NOT AT CYLN ZERO GO TO STEP 4
;10.    GET NEXT CYLN FOR LIST AND GO TO STEP 5
;
; DEBUG:
;
;       DIFFERENCE LOGIC
;       COUNT LOGIC
;       VELOCITY ROM
;
;--
```

```
        <CONVERGING - DIVERGING SEEK TEST>

;++
;
; TEST DESCRIPTION:
;
; THIS TEST WILL EXERCISE THE IDC DISK DRIVE WITH CONVERGING SEEKS
; AND THEN WITH DIVERGING SEEKS. THIS IS A MEANS OF CHECKING ALL
; POSSIBLE DIFFERENCE WORD COMBINATIONS.
;
;
; TEST STEPS:
;
;       1.  INITIALIZE SUBSYSTEM
;       2.  GET CURRENT DRIVE POSITION
;       3.  IF ALREADY AT CYLINDER 0 GO TO STEP 6.
;       4.  SEEK HEADS BACK TO CYLINDER 0
;       5.  VERIFY SEEK IF NOT AT CYLINDER 0 EXIT TEST
;       6.  START CONVERGING SEEKS
;       7.  WHEN UPPER LIMIT AND LOWER CROSS OVER -
;           THEN START DIVERGING SEEKS.
;       8.  CONTINUE UNTIL CURRENT CYLN RETURN TO ZERO
;
; DEBUG:
;
;       DIFFERNECE LOGIC
;       COUNT LOGIC
;       VELOCITY ROM
;
;--
```

```
        <RANDOM RECAL TEST - R80>

;++
;
; TEST DESCRIPTION:
;
; THIS TEST IS DESIGNED TO EXERCISE THE RECAL FUNCTION FOR THE R80
; DISK DRIVE BY SEEKING TO RANDOM CYLINDERS AND EXECUTING A RECAL
; FUNCTION (SEEK TO 0) FROM THAT CYLINDER.
;
; TEST STEPS:
;
;       1.  INITIALIZE SUBSYSTEM
;       2.  SEEK HEADS TO RANDOM CYLINDER
;       3.  REPORT ANY ERRORS
;       4.  EXECUTE RECAL FUNCTION (SEEK TO ZERO)
;       5.  REPORT ANY ERRORS
;       6.  CONTINUE UNTIL 64 RECALS HAVE BEEN COMPLETED
;
; DEBUG:
;
;       DIFFERNECE LOGIC
;       GUARD BAND DETECTION LOGIC
;       VELOCITY ROM
;
;--
```

```
	<RANDOM SEEK TEST>

;++
;
; TEST DESCRIPTION:
;
; THIS TEST EXERCISES THE POSITIONING LOGIC IN THE DRIVE
; UNDER TEST WITH RANDOM SEEKS. 1024 SEEKS ARE PERFORMED
; DURING THIS TEST. IF THE TEST DRIVE IS AN RL02 THEN
; RANDOM TRACKS ARE ALSO USED.
;
; TEST STEPS:
;
;	1.  INITIALIZE SUBSYSTEM
;	2.  SEEK HEADS TO RANDOM CYLINDER
;	3.  REPORT ANY ERRORS
;	4.  CONTINUE UNTIL 1024 SEEKS HAVE BEEN COMPLETED
;
; DEBUG:
;
;	DIFFERNECE LOGIC
;	TRACK COUNTING LOGIC
;	VELOCITY ROM
;
;--
```

```
        <HEADER NOT FOUND TEST>

;++
;
; TEST DESCRIPTION:
;
; THIS TEST WILL CHECK THAT A HEADER NOT FOUND ERROR CAN BE DETECTED
; USING A READ DATA COMMAND WITH THE IDCDAR REGISTER SPECIFING AN
; ILLEGAL ADDRESS.
;
; TEST STEPS:
;
; A.  SUBTEST 1
;
;       1.  INITIALIZE SUBSYSTEM
;       2.  GET CURRENT POSITION OF DRIVE
;       3.  IF AT CYLN ZERO GO TO SUBTEST 2
;       4.  EXECUTE SEEK BACK TO ZERO
;       5.  IF NOT AT CYLN ZERO NOW EXIT TEST
;
; B.  SUBTEST 2
;
;       1.  SET UP IDCDAR REGISTER FOR SECTO 40
;       2.  EXECUTE READ DATA COMMAND
;       3.  CHECK FOR ANY UNEXPECTED ERRORS
;       4.  CHECK THAT HEADER NOT FOUND IS SET IN IDCCSR
;
; DEBUG:
;
;       HEADER COMPARASIN LOGIC
;
;--
```

```
        <NON-EXISTANT MEMORY ERROR TEST>

;++
;
; TEST DESCRIPTION:
;
; THIS TEST WILL VERIFY THAT A NON-EXISTANT MEMORY ERROR WILL SET
; BY MAPPING THE UBA SUCH THAT A TRANSFER WILL OCCUR ACROSS AN
; UNMAPPED PAGE BOUNDRY THUS CAUSING A NON-EXISTANT MEMORY FAILURE.
; A NON-EXISTANAT MEMORY FAILURE SHOULD OCCUR USING ALL BUS ADDRESS BITS
; 9 THRU 17.
;
; TEST STEPS:
;
; A.  SUBTEST 1
;
;         1.  INITIALIZE SUBSYSTEM
;         2.  GET CURRENT DRIVE POSITION
;         3.  IF AT CYLN ZERO GO TO SUBTEST 2
;         4.  EXECUTE SEEK BACK TO ZERO
;         5.  IF NOT AT CYLN ZERO NOW EXIT TEST
;
; B.  SUBTEST 2
;
;         1.  INITIALIZE SUBSYSTEM
;         2.  MAP UBA FOR AN ILLEGAL ADDRESS
;         3.  EXECUTE A WRITE DATA COMMAND
;         4.  CHECK FOR ANY UNEXPECTED ERRORS
;         5.  CHECK THAT NON-EXISTANT MEMORY ERROR IS SET IN IDCCSR
;         6.  RETURN TO STEP 1 UNTIL BA BITS 9 THRU 17 HAVE BEEN TESTED.
;
; DEBUG:
;
;         BUS SLAVE SYNC TIME OUT LOGIC
;         BUS ADDRESS REGISTER
;         BUS ADDRESS DRIVERS
;
;--
```

```
        <LOOK AHEAD REGISTER TEST - R80>

;++
;
; TEST DESCRIPTION:
;
; THIS TEST VERIFIES THAT THE LOOK AHEAD SECTOR BITS IN
; R80 DRIVE STATUS WORD ARE COUNTING CORRECTLY.
;
; TEST STEPS:
;
; 1.  INITIALIZE SUBSYSTEM
; 2.  PERFORM GET STATUS FUNCTION
; 3.  RETRUN TO STEP 2 UNTILL COUNTER LOOKAHEAD REACHES 0
; 4.  DISPLAY ERROR IF PROGRAM TIMES OUT WAITING FOR
;     SECTOR 0.
; 5.  PERFORM CONTINIOUS GET STATUS FUNCTION WHILE MONITORING
;     THE COUNT OF THE LOOKAHEAD BITS
; 6.  REPORT ERROR IF COUNT IS NOT THE EXPECTED VALUE.
;
; DEBUG:
;
;       DRIVE SECTOR COUNTER
;
;--
```

```
        <HEAD SELECT LOGIC TEST - R80>

;++
;
; TEST DESCRIPTION:
;
; THIS TEST WILL VERIFY THAT ALL HEADS IN THE R80 DRIVE CAN BE
; SELECTED USING A SEEK COMMAND. ALL HEADS ARE TESTED BY EXECUTING
; SEEKS AND VERIFYING HEAD SELECTED WITH A READ HEADER COMMAND.
;
; TEST STEPS:
;
;       1.  INITIALIZE SUBSYSTEM
;       2.  SEEK HEADS TO CYLINDER ONE
;       3.  REPORT ANY ERRORS ESPECIALLY HEAD SELECT ERRORS
;       6.  CONTINUE UNTIL ALL HEADS HAVE BEEN TESTED
;
; DEBUG:
;
;       HEAD SELECT LOGIC
;
;--
```

```
        <BASIC DATA TRANSFER TEST>

;++
;
; TEST DESCRIPTION:
;
; THIS TEST WILL CHECK THAT READ WRITE LOOP CAPABILITY OF THE IDC
; SUBSYSTEM BY EXECUTING A ONE SECTOR WRITE DATA COMMAND.
; AND THEN EXECUTE A ONE SECTOR READ DATA COMMAND. VERIFY THAT THE
; DATA READ BACK IS CORRECT.
;
; TEST STEPS:
;
; A.  SUBTEST 1
;
;       1.  INITIALIZE SUBSYSTEM
;       2.  EXECUTE SEEK TO TEST CYLINDER
;       3.  IF ERROR THEN ESCAPE TEST
;
; B.  SUBTEST 2
;
;       1.  INITIALIZE SUBSYSTEM
;       2.  EXECUTE WRITE DATA COMMAND
;       3.  CHECK FOR ANY ERRORS
;
; C.  SUBTEST 3
;
;       1.  INITIALIZE SUBSYSTEM
;       2.  EXECUTE READ DATA COMMAND
;       3.  CHECK FOR ANY ERRORS
;       4.  VERIFY DATA IS CORRECT - REPORT ANY DESCREPENCIES
;
; DEBUG:
;
;       PHASE LOCK LOOP CKT
;       RL02 R/W MODULE
;       UPPER OR LOWER HEAD
;
;--
```

```
     <WRITE CHECK ERROR TEST>

;++
;
; TEST DESCRIPTION:
;
; THIS STEST WILL CHECK THAT WRITE CHECK ERROR WILL SET BY WRITING
; A PATTERN ON THE DISK, THEN CHANGE ONE WORD IN MEMORY AND EXECUTE
; A WRITE CHECK COMMAND.
;
; TEST STEPS:
;
; A.  SUBTEST 1
;
;      1. INITIALIZE SUBSYSTEM
;      2.  EXECUTE SEEK TO TEST CYLINDER
;      3.  IF ERROR THEN ESCAPE TEST
;
; B.  SUBTEST 2
;
;      1. INITIALIZE SUBSYSTEM
;      2. EXECUTE A WRITE DATA COMMAND OF ONE SECTOR
;      3. CHECK FOR ANY ERRORS
;
; C.     SUBTEST 3
;
;      1. INITIALIZE SUBSYSTEM
;      2. MODIFY A WORD IN THE WRITE BUFFER
;      3. EXECUTE A WRITE CHECK COMMAND
;      4. CHECK FOR ANY UNEXPECTED ERRORS
;      5. CHECK THAT WRITE CHECK ERROR DID SET IN IDCCSR
;
; DEBUG:
;
;      WRITE CHECK LOGIC
;
;--
```

```
        <WRITE CHECK DATA TEST>

;++
;
; TEST DESCRIPTION:
;
; CHECK THAT ABILITY OF THE IDC SUBSYSTEM TO WRITE CHECK DATA ERROR FREE
; BY EXECUTING A ONE SECTOR WRITE DATA COMMAND.  THEN READ THE DATA BACK
; AND CHECK THAT IT IS CORRECT.  THEN EXECUTE A WRITE CHECK COMMAND AND
; CHECK THAT NO ERROR OCCURS.
;
; TEST STEPS:
;
; A.   SUBTEST 1
;
;       1. INITIALIZE SUBSYSTEM
;       2.  EXECUTE SEEK TO TEST CYLINDER
;       3.  IF ERROR THEN ESCAPE TEST
;
; B.   SUBTEST 2
;
;       1. INITIALIZE SUBSYSTEM
;       2. EXECUTE WRITE DATA COMMAND
;       3. CHECK FOR ANY ERRORS
;
; C.   SUBTEST 3
;
;       1. INITIALIZE SUBSYSTEM
;       2. EXECUTE READ DATA COMMAND
;       3. CHECK FOR ANY ERRORS
;       4. VERIFY DATA WAS WRITTEN CORRECTLY
;
; D.   SUBTEST 4
;
;       1. INITIAZLIZE SUBSYSTEM
;       2. EXECUTE WRITE CHECK COMMAND
;       3. CHECK FOR ANY ERRORS
;
; DEBUG:
;
;       WRITE CHECK LOGIC
;
;--
```

```
        <PARTIAL SECTOR DATA TRANSFER TEST>

;++
;
; TEST DESCRIPTION:
;
; THIS TEST WILL VERIFY THAT WITH A BYTE COUNT OF LESS THAN ONE SECTOR
; (1/2 SECTOR IS USED HERE) A WRITE DATA COMMAND WILL CAUSE A SECTOR TO
; ZERO FILLED.  ALSO CHECK THAT A PARTIAL SECTOR READ WILL ONLY TRANSFER
; THAT NUMBER OF WORDS SPECIFIED, AND THAT A PARTIAL SECTOR WRITE CHECK
; CAN BE PERFORMED ERROR FREE.
;
; TEST STEPS:
;
; A.  SUBTEST 1
;
;       1. INITIALIZE SUBSYSTEM
;       2.  EXECUTE SEEK TO TEST CYLINDER
;       3.  IF ERROR THEN ESCAPE TEST
;
; B.  SUBTEST 2
;
;       1. INITIALIZE SUBSYSTEM
;       2. EXECUTE A FULL SECTOR WRITE DATA COMMAND
;       3. CHECK FOR ANY ERRORS
;       4. EXECUTE A PARTIAL SECTOR WRITE COMMAND
;       5. CHECK FOR ANY ERRORS
;
; C.  SUBTEST 3
;
;       1. INITIALIZE SUBSYSTEM
;       2. EXECUTE A FULL SECTOR READ
;       3. CHECK FOR ANY ERRORS
;       4. VERIFY DATA AND CHECK THAT SECTOR WAS ZERO FILLED
;
; D. SUBTEST 4
;
;       1. INITIALIZE SUBSYSTEM
;       2. EXECUTE A PARTIAL READ DATA CMD
;       3. CHECK FOR ANY ERRORS
;       4. VERIFY 1ST HALF WORDS OF READ BUFFER IS CORRECT
;       5. VERIFY THAT TOP HALF OF READ BUFFER REMAINED UNCHANGED
;
; E.  SUBTEST 5
;
;       1. INITIALIZE SUBSYSTEM
;       2. EXECUTE A PARTIAL SECTOR WRITE CHECK COMMAND
;       3. CHECK FOR ANY ERRORS
;
; DEBUG:
;
;       SECTOR FORMAT CONTROL
;
;--
```

```
        <MULTI SECTOR DATA TRANSFER TEST>

;++
;
; TEST DESCRIPTION:
;
; THIS TEST WILL CHECK THAT THE IDC SUBSYSTEM CAN PERFORM DATA
; TRANSFERS GREATER THANONE SECTOR BY EXECUTING A WRITE, READ, AND A
; WRITE CHECK OF TWO SECTORS. ALL ERRORS ARE CHECKED
; AND THE IDCDAR REGISTER IS CHECKED FOR PROPER INCREMENT.
;
;
; TEST STEPS:
;
; A.  SUBTEST 1
;
;        1.  INITIALIZE SUBSYSTEM
;        2.  POSITION DRIVE TO TEST CYLINDER
;        3.  IF ERROR THEN ESCAPE TEST
;
; B.  SUBTEST 2
;
;        1.  INITIALIZE SUBSYSTEM
;        2.  EXECUTE WRITE DATA COMMAND (2 SECTORS)
;        3.  CHECK FOR ANY ERRORS
;        4.  CHECK THAT THE IDCDAR REG INCREMENTED CORRECTLY
;
; C.  SUBTEST 3
;
;        1.  INITIALIZE SUBSYSTEM
;        2.  EXECUTE READ DATA COMMAND (2 SECTORS)
;        3.  CHECK FOR ANY ERRORS
;        4.  CHECK THAT THE IDCDAR REG INCREMENTED CORRECTLY
;        5.  VERIFY DATA IS CORRECT - REPORT ANY DESCREPENCIES
;
; D.  SUBTEST 4
;
;        1. INITIALIZE SUBSYSTEM
;        2. EXECUTE WRITE CHECK COMMAND (2 SECTORS)
;        3. CHECK FOR ANY ERRORS
;
; DEBUG:
;
;        SECTOR FORMAT CONTROL
;        MICRO SEQUENCER
;
;--
```

```
        &lt;SECTOR ADDRESSING TEST&gt;

;++
;
; TEST DESCRIPTION:
;
; THIS TEST WILL CHECK THAT THE SECTOR BITS CAN COMPARE CORRECTLY
; BY WRITING ALL SECTORS OF A TRACK.  DATA FOR EACH SECTOR WILL
; BE ITS OWN ADDRESS.  EACH SECTOR IS THEN READ TO VERIFY THE SECTOR
; CONTAINS THE CORRECT DATA.  IF DATA IS NOT CORRECT THEN SECTOR BITS
; ARE NOT COMPARING CORRECTLY.
;
; TEST STEPS:
;
; A.  SUBTEST 1
;
;       1.  INITIALIZE SUBSYSTEM
;       2.  POSITION DRIVE TO TEST CYLINDER
;       3.  IF ERROR THEN ESCAPE TEST
;
; B.  SUBTEST 2
;
;       1.  INITIALIZE SUBSYSTEM
;       2.  EXECUTE WRITE DATA COMMAND
;       3.  CHECK FOR ANY ERRORS
;       4.  CHECK THAT IDCDAR INCREMENTED CORRECTLY
;       5.  GOTO TO STEP 2 UNTIL ALL SECTORS WRITTEN
;
; C.  SUBTEST 3
;
;       1.  INITIALIZE SUBSYSTEM
;       2.  EXECUTE READ DATA COMMAND
;       3.  CHECK FOR ANY ERRORS
;       4.  CHECK THAT IDCDAR REGISTER INCREMENTED CORRECTLY
;       5.  VERIFY DATA IS CORRECT - REPORT ANY DESCREPENCIES
;       6.  GOTO STEP 2 UNTIL ALL SECTORS READ
;
; DEBUG:
;
;       HEADER FOUND LOGIC
;
;--
```

```
        <MAXIMUM DATA TRANSFER TEST>

;++
;
; TEST DESCRIPTION:
;
; THIS TEST WILL VERIFY THAT THE IDC SUBSYSTEM IS CAPABLE OF DOING
; WRITE, READ, AND WRITE CHECK DATA FUNCTIONS WITH A MAXIMUM WORD
; COUNT.
;
; TEST STEPS:
;
; A.  SUBTEST 1
;
;       1. INITIALIZE SUBSYSTEM
;       2.  POSITION DRIVE TO TEST CYLINDER
;       3.  IF ERROR THEN ESCAPE TEST
;
; B.  SUBTEST 2
;
;       1. INITIALIZE SUBSYSTEM
;       2. EXECUTE WRITE DATA COMMAND
;       3. CHECK FOR ANY ERRORS
;
; C.  SUBTEST 3
;
;       1. INITIALIZE SUBSYSTEM
;       2. EXECUTE READ DATA COMMAND
;       3. CHECK FOR ANY ERRORS
;       4. VERIFY DATA WAS WRITTEN CORRECTLY
;
; D.  SUBTEST 4
;
;       1. INITIAZLIZE SUBSYSTEM
;       2. EXECUTE WRITE CHECK COMMAND
;       3. CHECK FOR ANY ERRORS
;
; DEBUG:
;
;--
```

```
        <READ DATA WITHOUT HEADER COMPARE TEST>

;++
;
; TEST DESCRIPTION:
;
; THIS TEST WILL VERIFY THAT DATA CAN BE RECOVERED USING A READ WITHOUT
; HEADER COMPARE FUNCTION.  FIRST A WRITE DATA COMMAND IS DONE TO SECTOR
; THREE WITH A RANDOM DATA PATTERN. NEXT DO A WRITE DATA COMMAND TO
; SECTOR ONE. WHEN THIS FUNCTION COMPLETES THE HEADS WILL BE ONE SECTOR AWAY
; FROM SECTOR THREE. EXECUTE A READ WITH NO HEADER COMPARE FUNCTION
; WITH THE DISK ADDRESS REGISTER POINTING TO AN UNSELECTED TRACK ADDRESS.
; AND VERIFY THE THE DATA COME FROM SECTOR THREE ERROR FREE.
;
; TEST STEPS:
;
; A.  SUBTEST 1
;
;       1.  INITIALIZE SUBSYSTEM
;       2.  POSITION DRIVE TO TEST CYLINDER
;       3.  IF ERROR THEN ESCAPE TEST
;
; B.  SUBTEST 2
;
;       1.  INITIALIZE SUBSYSTEM
;       2.  EXECUTE WRITE DATA COMMAND TO SECTOR 3 USE RANDOM DATA.
;       3.  CHECK FOR ANY ERRORS
;
; C.  SUBTEST 3
;
;       1.  INITIALIZE SUBSYSTEM
;       2.  WRITE DATA TO SECTOR 1
;       3.  CHECK FOR ANY ERRORS
;       4.  EXECUTE READ WITH NO HEADER COMPARE FUNCTION
;       5.  CHECK FOR ANY ERRORS
;       6.  VERIFY IDCDAR UPDATED CORRECTLY
;       7.  VERIFY DATA READ IS CORRECT
;       8.  REPORT ANY DESCREPENCIES
;
; DEBUG:
;
;       SECTOR FORMAT CONTROL
;
;--
```

```
	&lt;OVERLAP SEEK TEST&gt;
;++
;
; TEST DESCRIPTION:
;
; THIS TEST CHECKS THAT OVERLAPPED SEEK OPERATIONS CAN BE
; PERFORMED BY THE IDC. THIS TEST IS ONLY RUN WHEN AT LEAST
; TWO DRIVES ARE SELECTED FOR TESTING.
;
; TEST STEPS:
;
;	1.  INITIALIZE SUBSYSTEM
;	2.  POSITION DRIVES TO CYLINDER 0
;	3.  POSITION SECONDARY DRIVE TO IT MIDPOINT
;	    AND DO NOT WAIT FOR DRIVE READY.
;	4.  SEEK DRIVE UNDER TEST TO ITS LAST CYLINDER
;	5.  VERIFY THAT DRIVE UNDER TEST COMPLETED ITS
;	    SEEK WITHOUT ERRORS (INTERFERENCE FROM OTHER DRIVE).
;
; DEBUG:
;
;--
```

```
        <DMA BOUNDRY TEST>

;++
;
; TEST DESCRIPTION:
;
; TEST THAT THE IDC IS CAPABLE OF HANDLING ODD BYTE
; COUNTS, ODD ADDRESS WHILE PROCESSING PARTIAL SECTOR
; SECTOR TRANSFERS THAT CROSS PAGE BOUNDRIES.
;
; TEST STEPS:                                          ;
;
; A.  SUBTEST 1
;
;       1.  INITIALIZE SUBSYSTEM
;       2.  EXECUTE SEEK TO TEST CYLINDER
;       3.  IF ERROR THEN ESCAPE TEST
;
; B.  SUBTEST 2
;
;       1.  INITIALIZE SUBSYSTEM
;       2.  EXECUTE WRITE DATA COMMAND
;       3.  CHECK FOR ANY ERRORS
;
; C.  SUBTEST 3
;
;       1.  INITIALIZE SUBSYSTEM
;       2.  EXECUTE READ DATA COMMAND
;       3.  CHECK FOR ANY ERRORS
;       4.  VERIFY DATA IS CORRECT - REPORT ANY DESCREPENCIES
;
; DEBUG:
;
; IDC MODULE
;
;--
```

```
        <WRITE FORMAT FUNCTION TEST - R80>

;++
;
; TEST DESCRIPTION:
;
; THIS TEST WILL EXERCISE THE THE WRITE FORMAT FUNCTION.
; ALL CYLINDERS ON THE TEST CYLINDER ARE FORMATTED AND VERIFY
; BY READING ALL HEADERS BACK SEQUENTIALLY.
;
; TEST STEPS:
;
; A.  SUBTEST 1
;
;       1. INITIALIZE SUBSYSTEM
;       2.  EXECUTE SEEK TO TEST CYLINDER
;       3.  IF ERROR THEN ESCAPE TEST
;
; B.  SUBTEST 2
;
;       1. INITIALIZE SUBSYSTEM
;       2. BUILD HEADERS FOR TRACK
;       3. FORMAT TRACK
;       4. CHECK FOR ERRORS
;       5. VERIFY CORRECTNESS OF HEADERS
;
; DEBUG:
;
;
;--
```

```
        <MANUAL SKIP SECTOR TEST - R80>

;++
;
; TEST DESCRIPTION:
;
; THIS TEST WILL VERIFY THE OPERATION OF A MANUAL SKIP
; SECTOR RECOVERY OPERATION.
;
; TEST STEPS:
;
; A.  SUBTEST 1
;
;       1. INITIALIZE SUBSYSTEM
;       2.  EXECUTE SEEK TO TEST CYLINDER
;       3.  IF ERROR THEN ESCAPE TEST
;
; B.  SUBTEST 2
;
;       1. INITIALIZE SUBSYSTEM
;       2. BUILD HEADERS FOR TRACK
;       3. FORMAT TRACK WITH SECTOR 0 MARKED AS A SKIP SECTOR
;       4. CHECK FOR ERRORS
;
; B.  SUBTEST 3
;       1. INIT SUBSYSTEM
;       2. WRITE SECTORS 0 AND 1 WITH SKIP SECTORS TURNED OFF
;          DATA = ^X3333 FOR SECTOR 0 AND ^XCCCC FOR SECTOR 1
;       3. READ DATA (SECTOR 0) WITH ASSI ASSERTED AND EXPECT SSE TO SET
;       4. RESTART READ BY SETTING SSEI AND CLEARING CRDY
;       5. VERIFY OPERATION COMPLETED ERROR FREE
;       6. CHECK DATA READ BACK WAS SECTOR 1 DATA (^XCCCC)
;
; C.  SUBTEST 4
;       1. INIT SUBSYSTEM
;       2. WRITE CHECK SECTOR 0 AND EXPECT SSE TO ASSERT
;       3. RESTART OPERTION BY SETTING SSEI AND CLEARING CRDY
;       4. VERIFY OPERATION COMPLETED ERROR FREE
;
; DEBUG:
;
;
;--
```

```
        <AUTOMATIC SKIP SECTOR TEST - R80>

;++
;
; TEST DESCRIPTION:
;
; THIS TEST WILL VERIFY THE AUTOMATIC SKIP SECTOR FUNCTION
; OF THE IDC BY FORMATTING A TRACK WITH SECTOR 0 AS A SKIP
; SECTOR. PERFORM WRITE, READ, AND WRITECHECK OPERATIONS
; TO SECTOR 0 AND CHECK FOR ANY ERRORS. ALSO VERIFY THAT
; AT COMMAND COMPLETION THE IDCDAR IS POINTING TO SECTOR 2.
;
; TEST STEPS:
;
; A.  SUBTEST 1
;
;       1. INITIALIZE SUBSYSTEM
;       2.  EXECUTE SEEK TO TEST CYLINDER
;       3.  IF ERROR THEN ESCAPE TEST
;
; B.  SUBTEST 2
;
;       1. INITIALIZE SUBSYSTEM
;       2. BUILD HEADERS FOR TRACK
;       3. FORMAT TRACK
;       4. CHECK FOR ERRORS
;
; DEBUG:
;
;
;--
```

```
        <ADDRESS RECALL TEST - RL02>

;++
;
; TEST DESCRIPTION:
;
; THIS TEST WILL EXECUTE 512 PSEUDO RANDOM SEEKS.  THE SEEKS ARE PSEUDO
; RANDOM BECAUSE EACH TIME THE TEST IS RUN IT WILL ALWAYS USE THE
; SAME 512 SEEK DESTINATIONS.  ALL SEEKS ARE DONE WITH NO INTERVEANING
; READ HEADERS.  THIS REQUIRES THE IDC TO KEEP TRACK OF THE CURRENT
; CYLINDER THE HEADS ARE POSITIONED ON BEFORE CALCULATING THE DIFFERENCE
; WORD FOR THE NEXT SEEK.  THIS IS THE ONLY TEST THAT USES THIS MODE
; OF OPERATION.
;
; TEST STEPS:
;
;       1.  INITIALIZE SUBSYSTEM
;       2.  CALCULATE RANDOM ADDRESS
;       3.  EXECUTE SEEK TO DESTINATION
;       4.  RETURN TO STEP 2 UNTIL 512 SEEKS HAVE BEEN DONE
;       5.  READ HEADER TO GET CURRENT POSITION OF HEADS
;       6.  IF DRIVE NO POSITION ON CYLINER XYZ REPORT ERROR
;
; DEBUG:
;
;       IDC MODULE
;
; ASSUMPTIONS:
;
;       RL02 SEEK LOGIC IS WORKING PROPERLY.
;       ANY SEEK ERROR WILL CAUSE TEST TO ABORT.
;
;--
```

```
        <SUBSYSTEM ACTIVITY TEST>

;++
;
; TEST DESCRIPTION:
;
; THIS TEST IS INTENDED TO EXERCISE THE IDC SUBSYSTEM USING
; RANDOM DISK ADDRESSES FOR SEEKS, WRITES, WRITECHECKS, AND
; READS.  IT IS INTENDED AS A TOOL FOR MAXIMIZING FUNCTIONS
; IN A STANDALONE ENVIRONMENT.
;
; TEST STEPS:
;
;       1.  INITIALIZE SUBSYSTEM
;       2.  EXECUTE READ HEADER/SEEK/READ HEADER
;           AND CHECK FOR ERRORS
;       3.  EXECUTE WRITE AND CHECK FOR ERRORS
;       4.  EXECUTE WRITECHECK AND CHECK FOR ERRORS
;       5.  EXECUTE READ AND CHECK FOR ERRORS
;
; DEBUG:
;
;--
```

```
        <DRIVE INTERFACE TEST>

;++
;
; TEST DESCRIPTION:
;
; THIS TEST WILL ASK THE OPERATOR TO UNLOAD THE CURRENT DRIVE UNDER
; TEST.  THEN WHEN THE HEADS ARE UNLOADED TO OPEN THE COVER (IF RL02)
; AND SET THE WRITE LOCK SWITCH.  A GET STATUS COMMAND IS EXECUTED
; AND PROPER DRIVE STATUS IS VERIFIED.
; NEXT ASK THE OPERATOR TO CLOSE THE COVER (RL02) AND RESET THE WRITE
; LOCK SWITCH ON THE DRIVE UNDER TEST.  THEN EXECUTE A GET STATUS COMMAND
; AND CHECK FOR PROPER DRIVE STATUS.
;
; TEST STEPS:
;
;       1.  INITIALIZE SUBSYSTEM
;       2.  ASK OPERATOR TO UNLOAD DRIVE
;       3.  WAIT FOR OPERATOR TO HIT RETURN
;       4.  EXECUTE GET STATUS FUNCTION
;       5.  VERIFY DRIVE STATUS
;       6.  ASK OPERATOR TO CLOSE COVER AND RESET WRITE LOCK
;       7.  WAIT FOR OPERATOR TO HIT RETURN
;       8.  EXECUTE GET STATUS FUNCTION
;       9.  VERIFY DRIVE STATUS
;
; DEBUG:
;
;       DRIVE LOGIC MODULE
;       COVER LATCH (RL02)
;       SWITCH PANEL
;
;--
```

```
        <HEAD LOADING TEST - RL02>

;++
;
; TEST DESCRIPTION:
;
; THIS TEST WILL INSTRUCT THE OPERATOR TO LOAD THE HEADS ON THE UNIT
; CURRENTLY UNDER TEST.  THE OCCURANCE OF ALL DRIVE STATES WILL BE
; CHECK AND ALSO THE DRIVE STATUS WITHIN EACH STATE WILL BE CHECKED.
; ONCE THE HEADS ARE LOADED DRIVE READY BIT IS CHECKED IN THE CONTROL
; AND STATUS REGISTER.  THEN A DRIVE RESET IS DONE AND VOLUME CHECK
; BIT IN THE DRIVE STATUS WORD IS CHECKED FOR THE RESET STATE.
;
; TEST STEPS:
;
;       1.  INIT SUBSYSTEM
;       2.  TELL OPERATOR TO LOAD HEADS ON UUT
;       3.  DONT WAIT FOR ANY RETURN FROM OPERATOR
;       4.  WAIT FOR DRIVE STATE 1 (SPIN UP)
;       5.  TEST DRIVE STATUS IN SPIN UP STATE
;       6.  WAIT FOR DRIVE STATE 2 (BRUSH CYCLE)
;       7.  TEST DRIVE STATUS IN STATE 2
;       8.  WAIT FOR BRUSH CYCLE TO START (BH BIT RESET)
;       9   WAIT FOR BRUSH CYCLE TO COMPLETE (BH BIT SET)
;       10. CHECK STATE 3 DRIVE STATUS (LOAD HEADS)
;       11. WAIT FOR DRIVE STATE 4 (SEEK)
;       12. CHECK DRIVE STATE 4 STATUS
;       13. WAIT FOR DRIVE STATE 5 (LOCK ON TRACK)
;       14. CHECK STATE 5 DRIVE STATUS
;       15. CHECK DRIVE READY BIT IN IDCCSR IS SET
;       16. EXECUTE DRIVE RESET
;       17. CHECK THAT VOLUME CHECK BIT IN DRIVE STATUS IS CLEAR
;
; DEBUG:
;
;       DRIVE LOGIC MODULE
;       BRUSH MOTOR
;       BRUSH SWITCH
;       AC SERVO MODULE
;
;--
```

```
        <HEAD UNLOADING TEST - RL02>

;++
;
; TEST DESCRITPION:
;
; THIS TEST WILL REQUEST THE OPERATOR TO UNLOAD THE DRIVE CURRENTLY
; UNDER TEST.  THEN WE WILL TEST THE DRIVE STATUS DURING THE SPIN
; DOWN AND HEAD UNLOADING STATES AND WAIT FOR DRIVE TO STOP IN THE
; LOAD CARTRIDGE STATE.
;
; TEST STEPS:
;
;       1.  INITIALIZE SUBSYSTEM
;       2.  TELL OPERATOR TO UNLOAD DRIVE UNDER TEST
;       3.  DO NOT WAIT FOR RETURN FROM OPERATOR
;       4.  WAIT FOR DRIVE TO PROGRESS FROM STATE 5 TO STATE 6
;       5.  TEST DRIVE STATUS IN STATE 6
;       6.  WAIT FOR DRIVE TO PROGRESS FROM STATE 6 TO STATE 7
;       7.  TEST DRIVE STATUS IN STATE 7
;       8.  WAIT FOR DRIVE TO PROGRESS FROM STATE 7 TO STATE 0
;
; DEBUG:
;
;       DRIVE LOGIC MODULE
;       AC SERVO
;
;--
```

```
        <DRIVE SELECT TEST>

;++
;
; TEST DESCRIPTION:
;
; THIS TEST WILL ASK THE OPERATOR TO REMOVE ALL DRIVE PLUGS FROM THE
; DRIVES.  TELL OPERATOR TO INSERT DRIVE PLUG #0 IN UUT AND VERIFY
; THAT GETTING A DRIVE STATUS FROM THAT UNIT IS POSSIBLE WITHOUT ERROR.
; THEN DO A GET STATUS COMMAND FOR EACH OF THE REMAINING DRIVE PLUGS AND
; VERIFY THAT OPERATION INCOMPLETE ERROR IS SET IN THE IDCCSR REGISTER.
;
; TEST STEPS:
;
;       1.  INITIALIZE SUBSYSTEM
;       2.  TELL OPERATOR TO REMOVE ALL DRIVE PLUGS
;       3.  TELL OPERATOR TO INSERT PLUG IN UUT
;       4.  EXECUTE GET STATUS CMD - CHECK FOR ANY ERRORS
;       5.  EXECUTE GET STATUS CMD FOR REMAING ADDRESS PLUGS
;       6.  VERIFY THE OPI ERROR SET IN IDCCSR REGISTER
;       7.  RETURN TO STEP 3 UNTIL ALL PLUGS HAVE BEEN TESTED
;
; DEBUG:
;
;       DRIVE SELECT LOGIC
;
;--
```

```
        <WRITE PROTECTION TEST>

;++
;
; TEST DESCRIPTION:
;
; THIS TEST WILL WRITE A PATTERN IN SECTOR 0 THEN ASK THE OPERATOR
; TO SET THE WRITE LOCK SWITCH ON THE UUT.  EXECUTE ANOTHER WRITE
; DATA CMD TO THE SAME SECTOR AND VERIFY THAT WRITE GATE ERROR AND
; DRIVE ERROR ARE SET.  EXECUTE A DRIVE RESET AND VERIFY THAT ERRORS
; ARE RESET THEN EXECUTE A READ DATA CMD AND VERIFY THAT THE DATA
; WAS NOT DISTURBED WHILE THE WRITE LOCK SWITCH WAS SET.
;
; TEST STEPS:
;
; A.  SUBTEST 1
;
;       1.  INITIALIZE SUBSYSTEM
;       2.  POSITION DRIVE TO TEST CYLINDER
;       3.  IF ERROR THEN ESCAPE TEST
;
; B.  SUBTEST 2
;
;       1.  INITIALIZE SUBSYSTEM
;       2.  EXECUTE WRITE DATA COMMAND
;       3.  CHECK FOR ANY ERRORS
;
; C.  SUBTEST 3
;
;       1.  INITIALIZE SUBSYSTEM
;       2.  TELL OPERATOR TO SET WRITE LOCK SWITCH ON UUT
;       3.  EXECUTE WRITE DATA CMD
;       4.  VERIFY THAT WRITE GATE AND DRIVE ERROR ARE SET
;       5.  EXECUTE DRIVE RESET FUNCTION
;       6.  VERIFY THAT WRITE GATE ERROR IS RESET
;       7.  INIT SUBSYSTEM
;       8.  EXEUCTE READ DATA CMD - CHECK FOR ERRORS
;       9.  VERIFY THAT DATA WAS NOT DISTURBED
;       10. TELL OPERATOR TO RESET WRITE LOCK SWITCH
;
; DEBUG:
;
;       SWITCH PANEL
;
;--
```

! Object Module Synopsis !

Module Name	Ident	Bytes	File	Creation Date	Creator
ENRKA0	1.1	22802	1DUA9:[BOWSER.TAPE]ENRKA0.OBJ;1	4-FEB-1986 15:56	VAX/VMS Macro V04-00
ENRKA1	1.1	8931	1DUA9:[BOWSER.TAPE]ENRKA1.OBJ;1	4-FEB-1986 15:58	VAX/VMS Macro V04-00
ENRKA2	1.1	9874	1DUA9:[BOWSER.TAPE]ENRKA2.OBJ;1	4-FEB-1986 16:00	VAX/VMS Macro V04-00
ENRKA3	1.1	12309	1DUA9:[BOWSER.TAPE]ENRKA3.OBJ;1	4-FEB-1986 16:03	VAX/VMS Macro V04-00

! Program Section Synopsis !

Psect Name	Module Name	Base	End	Length	Align	Attributes
$HEADER		00000200	0000027E	0000007F (127.)	PAGE 9	NOPIC,USR,CON,REL,LCL,NOSHR,NOEXE, RD,NOWRT,NOVEC
	ENRKA0	00000200	0000027E	0000007F (127.)	PAGE 9	
$TSTCNT		00000280	00000283	00000004 (4.)	LONG 2	NOPIC,USR,OVR,REL,LCL,NOSHR,NOEXE, RD,NOWRT,NOVEC
	ENRKA1	00000280	00000283	00000004 (4.)	LONG 2	
	ENRKA2	00000280	00000283	00000004 (4.)	LONG 2	
	ENRKA3	00000280	00000283	00000004 (4.)	LONG 2	
ARGLIST		00000284	000005E3	00000360 (864.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD,NOWRT,NOVEC
	ENRKA1	00000284	00000397	00000114 (276.)	LONG 2	
	ENRKA2	00000398	000004FF	00000168 (360.)	LONG 2	
	ENRKA3	00000500	000005E3	000000E4 (228.)	LONG 2	
BUFFERS		00000600	00000FFF	00000A00 (2560.)	PAGE 9	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD, WRT,NOVEC
	ENRKA0	00000600	00000FFF	00000A00 (2560.)	PAGE 9	
CLEANUP		00001000	00001078	00000079 (121.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD, WRT,NOVEC
	ENRKA0	00001000	00001078	00000079 (121.)	LONG 2	
DISPATCH		0000107C	0000149B	00000420 (1056.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR,NOEXE, RD,NOWRT,NOVEC
	ENRKA1	0000107C	00001213	00000198 (408.)	LONG 2	
	ENRKA2	00001214	00001363	00000150 (336.)	LONG 2	
	ENRKA3	00001364	0000149B	00000138 (312.)	LONG 2	
DISPATCH_X		0000149C	000014B3	00000018 (24.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR,NOEXE, RD,NOWRT,NOVEC
	ENRKA0	0000149C	000014B3	00000018 (24.)	LONG 2	
GLOBAL_DATA		000014B4	00001B65	000006B2 (1714.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD, WRT,NOVEC
	ENRKA0	000014B4	00001B65	000006B2 (1714.)	LONG 2	
GLOBAL_TEXT		00001B68	00004C3F	000030D8 (12504.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR,NOEXE, RD,NOWRT,NOVEC
	ENRKA0	00001B68	00004C3F	000030D8 (12504.)	LONG 2	
INITIALIZE		00004E00	0000504B	0000024C (588.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD, WRT,NOVEC
	ENRKA0	00004E00	0000504B	0000024C (588.)	LONG 2	
SUBROUTINES		0000504C	0000646C	00001421 (5153.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD, WRT,NOVEC
	ENRKA0	0000504C	0000646C	00001421 (5153.)	LONG 2	
SUMMARY		00006470	0000647A	0000000B (11.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD, WRT,NOVEC
	ENRKA0	00006470	0000647A	0000000B (11.)	LONG 2	
TEST_001		0000647C	000064F7	0000007C (124.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD,NOWRT,NOVEC
	ENRKA1	0000647C	000064F7	0000007C (124.)	LONG 2	
TEST_002		000064F8	000065FD	00000106 (262.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD,NOWRT,NOVEC
	ENRKA1	000064F8	000065FD	00000106 (262.)	LONG 2	
TEST_003		00006600	00006877	00000278 (632.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD,NOWRT,NOVEC
	ENRKA1	00006600	00006877	00000278 (632.)	LONG 2	
TEST_004		00006878	00006A86	0000020F (527.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD,NOWRT,NOVEC

Psect Name	Module Name	Base	End	Length	Align	Attributes
TEST_004		00006878	00006A86	0000020F (527.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD,NOWRT,NOVEC
	ENRKA1	00006878	00006A86	0000020F (527.)	LONG 2	
TEST_005		00006A88	00006C92	0000020B (523.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD,NOWRT,NOVEC
	ENRKA1	00006A88	00006C92	0000020B (523.)	LONG 2	
TEST_006		00006C94	00006EAF	0000021C (540.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD,NOWRT,NOVEC
	ENRKA1	00006C94	00006EAF	0000021C (540.)	LONG 2	
TEST_007		00006EB0	00006F9A	000000EB (235.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD,NOWRT,NOVEC
	ENRKA1	00006EB0	00006F9A	000000EB (235.)	LONG 2	
TEST_008		00006F9C	000070F7	0000015C (348.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD,NOWRT,NOVEC
	ENRKA1	00006F9C	00C070F7	0000015C (348.)	LONG 2	
TEST_009		000070F8	000072B8	000001C1 (449.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD,NOWRT,NOVEC
	ENRKA1	000070F8	000072B8	000001C1 (449.)	LONG 2	
TEST_010		000072BC	0000743E	00000183 (387.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD,NOWRT,NOVEC
	ENRKA1	000072BC	0000743E	00000183 (387.)	LONG 2	
TEST_011		00007440	00007508	000000C9 (201.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD,NOWRT,NOVEC
	ENRKA1	00007440	00007508	000000C9 (201.)	LONG 2	
TEST_012		0000750C	0000762D	00000122 (290.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD,NOWRT,NOVEC
	ENRKA1	0000750C	0000762D	00000122 (290.)	LONG 2	
TEST_013		00007630	000078B8	00000289 (649.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD,NOWRT,NOVEC
	ENRKA1	00007630	000078B8	00000289 (649.)	LONG 2	
TEST_014		000078BC	00007A85	000001CA (458.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD,NOWRT,NOVEC
	ENRKA1	000078BC	00007A85	000001CA (458.)	LONG 2	
TEST_015		00007A88	00007E22	0000039B (923.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD,NOWRT,NOVEC
	ENRKA1	00007A88	00007E22	0000039B (923.)	LONG 2	
TEST_016		00007E24	0000829C	00000479 (1145.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD,NOWRT,NOVEC
	ENRKA1	00007E24	0000829C	00000479 (1145.)	LONG 2	
TEST_017		000082A0	000084C5	00000226 (550.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD,NOWRT,NOVEC
	ENRKA1	000082A0	000084C5	00000226 (550.)	LONG 2	
TEST_018		000084C8	00008710	00000249 (585.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD,NOWRT,NOVEC
	ENRKA2	000084C8	00008710	00000249 (585.)	LONG 2	
TEST_019		00008714	00008850	0000013D (317.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD,NOWRT,NOVEC
	ENRKA2	00008714	00008850	0000013D (317.)	LONG 2	
TEST_020		00008854	00008953	00000100 (256.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD,NOWRT,NOVEC
	ENRKA2	00008854	00008953	00000100 (256.)	LONG 2	
TEST_021		00008954	00008C4A	000002F7 (759.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD,NOWRT,NOVEC
	ENRKA2	00008954	00008C4A	000002F7 (759.)	LONG 2	
TEST_022		00008C4C	00008E78	0000022D (557.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD,NOWRT,NOVEC
	ENRKA2	00008C4C	00008E78	0000022D (557.)	LONG 2	
TEST_023		00008E7C	000090A7	0000022C (556.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD,NOWRT,NOVEC
	ENRKA2	00008E7C	000090A7	0000022C (556.)	LONG 2	
TEST_024		000090A8	00009193	000000EC (236.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD,NOWRT,NOVEC
	ENRKA2	000090A8	00009193	000000EC (236.)	LONG 2	
TEST_025		00009194	0000950A	00000377 (887.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD,NOWRT,NOVEC
	ENRKA2	00009194	0000950A	00000377 (887.)	LONG 2	
TEST_026		0000950C	0000976A	0000025F (607.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD,NOWRT,NOVEC
	ENRKA2	0000950C	0000976A	0000025F (607.)	LONG 2	
TEST_027		0000976C	000099F8	0000028D (653.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD,NOWRT,NOVEC
	ENRKA2	0000976C	000099F8	0000028D (653.)	LONG 2	
TEST_028		000099FC	00009E02	00000407 (1031.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD,NOWRT,NOVEC
	ENRKA2	000099FC	00009E02	00000407 (1031.)	LONG 2	
TEST_029		00009E04	0000A1E9	000003E6 (998.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD,NOWRT,NOVEC

```
Psect Name   Module Name      Base     End       Length            Align              Attributes
----------   -----------      ----     ---       ------            -----              ----------
TEST_029                   00009E04 0000A1E9 000003E6 (      998.) LONG 2 NOPIC,USR,CON,REL,LCL,NOSHR,  EXE,  RD,NOWRT,NOVEC
             ENRKA2        00009E04 0000A1E9 000003E6 (      998.) LONG 2
TEST_030                   0000A1EC 0000A4E8 000002FD (      765.) LONG 2 NOPIC,USR,CON,REL,LCL,NOSHR,  EXE,  RD,NOWRT,NOVEC
             ENRKA2        0000A1EC 0000A4E8 000002FD (      765.) LONG 2
TEST_031                   0000A4EC 0000A8B2 000003C7 (      967.) LONG 2 NOPIC,USR,CON,REL,LCL,NOSHR,  EXE,  RD,NOWRT,NOVEC
             ENRKA2        0000A4EC 0000A8B2 000003C7 (      967.) LONG 2
TEST_032                   0000A8B4 0000ABE6 00000333 (      819.) LONG 2 NOPIC,USR,CON,REL,LCL,NOSHR,  EXE,  RD,NOWRT,NOVEC
             ENRKA3        0000A8B4 0000ABE6 00000333 (      819.) LONG 2
TEST_033                   0000ABE8 0000AD9E 000001B7 (      439.) LONG 2 NOPIC,USR,CON,REL,LCL,NOSHR,  EXE,  RD,NOWRT,NOVEC
             ENRKA3        0000ABE8 0000AD9E 000001B7 (      439.) LONG 2
TEST_034                   0000ADA0 0000B020 00000281 (      641.) LONG 2 NOPIC,USR,CON,REL,LCL,NOSHR,  EXE,  RD,NOWRT,NOVEC
             ENRKA3        0000ADA0 0000B020 00000281 (      641.) LONG 2
TEST_035                   0000B024 0000B239 00000216 (      534.) LONG 2 NOPIC,USR,CON,REL,LCL,NOSHR,  EXE,  RD,NOWRT,NOVEC
             ENRKA3        0000B024 0000B239 00000216 (      534.) LONG 2
TEST_036                   0000B23C 0000B73C 00000501 (     1281.) LONG 2 NOPIC,USR,CON,REL,LCL,NOSHR,  EXE,  RD,NOWRT,NOVEC
             ENRKA3        0000B23C 0000B73C 00000501 (     1281.) LONG 2
TEST_037                   0000B740 0000BCA0 00000561 (     1377.) LONG 2 NOPIC,USR,CON,REL,LCL,NOSHR,  EXE,  RD,NOWRT,NOVEC
             ENRKA3        0000B740 0000BCA0 00000561 (     1377.) LONG 2
TEST_038                   0000BCA4 0000BE43 000001A0 (      416.) LONG 2 NOPIC,USR,CON,REL,LCL,NOSHR,  EXE,  RD,NOWRT,NOVEC
             ENRKA3        0000BCA4 0000BE43 000001A0 (      416.) LONG 2
TEST_039                   0000BE44 0000C0FE 000002BB (      699.) LONG 2 NOPIC,USR,CON,REL,LCL,NOSHR,  EXE,  RD,NOWRT,NOVEC
             ENRKA3        0000BE44 0000C0FE 000002BB (      699.) LONG 2
TEST_040                   0000C100 0000C35C 0000025D (      605.) LONG 2 NOPIC,USR,CON,REL,LCL,NOSHR,  EXE,  RD,NOWRT,NOVEC
             ENRKA3        0000C100 0000C35C 0000025D (      605.) LONG 2
TEST_041                   0000C360 0000CBA5 00000846 (     2118.) LONG 2 NOPIC,USR,CON,REL,LCL,NOSHR,  EXE,  RD,NOWRT,NOVEC
             ENRKA3        0000C360 0000CBA5 00000846 (     2118.) LONG 2
TEST_042                   0000CBA8 0000CF40 00000399 (      921.) LONG 2 NOPIC,USR,CON,REL,LCL,NOSHR,  EXE,  RD,NOWRT,NOVEC
             ENRKA3        0000CBA8 0000CF40 00000399 (      921.) LONG 2
TEST_043                   0000CF44 0000D196 00000253 (      595.) LONG 2 NOPIC,USR,CON,REL,LCL,NOSHR,  EXE,  RD,NOWRT,NOVEC
             ENRKA3        0000CF44 0000D196 00000253 (      595.) LONG 2
TEST_044                   0000D198 0000D6BF 00000528 (     1320.) LONG 2 NOPIC,USR,CON,REL,LCL,NOSHR,  EXE,  RD,NOWRT,NOVEC
             ENRKA3        0000D198 0000D6BF 00000528 (     1320.) LONG 2

                                +-------------------------+
                                ! Symbol Cross Reference !
                                +-------------------------+

Symbol              Value          Defined By     Referenced By ...
------              -----          ----------     -----------------
$ENV                00000003        ENRKA0
$GS_SETMAP          00006073-R      ENRKA0           ENRKA2          ENRKA3
$MO                 00000001        ENRKA0
ABORT_FLAG          00001508-R      ENRKA0
ACT_NAME            00001500-R      ENRKA0
ALL                 00000002        ENRKA0
BAD_DATA            00001658-R      ENRKA0
BASE_ADDR           000014C8-R      ENRKA0           ENRKA1          ENRKA3
BDDAT               000015BA-R      ENRKA0
BUFFER1             00000600-R      ENRKA0           ENRKA2
BUFFER2             00000A00-R      ENRKA0           ENRKA2
BUFF_INUSE          00001558-R      ENRKA0
```

Symbol	Value	Defined By	Referenced By ...		
BUILD_HDRS	00005F98-R	ENRKA0	ENRKA3		
CALC_DIFF	00006044-R	ENRKA0			
CHANGE_DAR$A	00006264-R	ENRKA0	ENRKA2	ENRKA3	
CHAR_BUFFER	00001674-R	ENRKA0	ENRKA3		
CHECK_BADBLK	00006430-R	ENRKA0	ENRKA3		
CHECK_ECC	000061E6-R	ENRKA0	ENRKA3		
CHECK_FATAL	00005C98-R	ENRKA0			
CMPDATA	00005792-R	ENRKA0	ENRKA2	ENRKA3	
COM_BIT16	00001C57-R	ENRKA0			
COM_BIT32	00001BA1-R	ENRKA0			
CONFIG	000015D0-R	ENRKA0	ENRKA1	ENRKA2	ENRKA3
CRC_TABLE	000016DC-R	ENRKA0	ENRKA1		
CSR_BITS	00001D32-R	ENRKA0	ENRKA1	ENRKA2	ENRKA3
CURR_CMD	000015B0-R	ENRKA0	ENRKA3		
CURR_CYLN	00001588-R	ENRKA0	ENRKA1	ENRKA2	ENRKA3
CURR_CYL_HD	00001590-R	ENRKA0	ENRKA2		
CURR_FLAGS	000014DC-R	ENRKA0	ENRKA3		
CURR_HD	0000158C-R	ENRKA0			
CURR_SECTOR	000015A4-R	ENRKA0			
CVT_BRUSH	00002244-R	ENRKA0			
CVT_DIR_TABLE	00001B5A-R	ENRKA0			
CVT_DRVTYP	00001AD6-R	ENRKA0			
CVT_ERROR0	000021AE-R	ENRKA0			
CVT_ERROR1	000021B3-R	ENRKA0			
CVT_ERROR2	000021C8-R	ENRKA0			
CVT_ERROR3	000021DC-R	ENRKA0			
CVT_ERROR4	000021E4-R	ENRKA0			
CVT_ERROR5	000021EE-R	ENRKA0			
CVT_ERROR6	000021FC-R	ENRKA0			
CVT_ERROR7	00002211-R	ENRKA0			
CVT_ERR_TABLE	00001B2A-R	ENRKA0			
CVT_FORWARD	000022A8-R	ENRKA0			
CVT_FUN_TABLE	00001AE2-R	ENRKA0			
CVT_GETSTAT	0000216B-R	ENRKA0			
CVT_LDHEADS	00002250-R	ENRKA0			
CVT_LOAD	00002229-R	ENRKA0			
CVT_LOCKED	00002266-R	ENRKA0			
CVT_LOWER	0000229A-R	ENRKA0			
CVT_NOOP	00002159-R	ENRKA0			
CVT_RDNHC	0000219C-R	ENRKA0			
CVT_READ	00002192-R	ENRKA0			
CVT_READH	0000217B-R	ENRKA0			
CVT_REVERSE	000022A0-R	ENRKA0			
CVT_SEEK	00002176-R	ENRKA0			
CVT_SK	0000225E-R	ENRKA0			
CVT_SPINDN	00002286-R	ENRKA0			
CVT_SPINUP	00002238-R	ENRKA0			
CVT_STA_TABLE	00001B06-R	ENRKA0			
CVT_SUR_TABLE	00001B4E-R	ENRKA0			
CVT_UNLOAD	00002276-R	ENRKA0			
CVT_UPPER	00002294-R	ENRKA0			

Symbol	Value	Defined By	Referenced By ...		
CVT_WRITE	00002187-R	ENRKA0			
CVT_WRTCHK	0000215F-R	ENRKA0			
CYLN	00001584-R	ENRKA0			
CYLN_LIST	0000171C-R	ENRKA0	ENRKA1		
CYLN_LIST_X	00001766-R	ENRKA0	ENRKA1		
DAR_DMA02_BITS	00001DFC-R	ENRKA0			
DAR_DMA80_BITS	00001E2D-R	ENRKA0			
DAR_GS02_BITS	00001F12-R	ENRKA0			
DAR_MNEM	00001570-R	ENRKA0	ENRKA2	ENRKA3	
DAR_RCL80_BITS	00001ED8-R	ENRKA0			
DAR_SK02_BITS	00001E5C-R	ENRKA0			
DAR_SK80_BITS	00001E9F-R	ENRKA0			
DEFAULT	00000000	ENRKA0			
DELTA_TIME	00001538-R	ENRKA0	ENRKA3		
DERR_FLAG	00001568-R	ENRKA0			
DEST_CYLN	00001598-R	ENRKA0	ENRKA1		
DEV	000014BC-R	ENRKA0			
DEV_NAME	000014F8-R	ENRKA0	ENRKA3		
DEV_REG	00001574-R	ENRKA0			
DIFF_WORD	000015B4-R	ENRKA0	ENRKA1		
DISK_SECTION	00001560-R	ENRKA0			
DRIVE_INIT$A	000057F1-R	ENRKA0			
DRIVE_TBL	000014B4-R	ENRKA0	ENRKA1	ENRKA2	ENRKA3
DS$ABORT	00010020	ENRKA0	ENRKA1		
DS$ASKADR	00010090	ENRKA0			
DS$ASKDATA	00010080	ENRKA0			
DS$ASKLGCL	00010098	ENRKA0			
DS$ASKSTR	000100A0	ENRKA0	ENRKA3		
DS$ASKVLD	00010088	ENRKA0			
DS$ATTACH	000101A8	ENRKA0			
DS$BGNSUB	00010030	ENRKA0	ENRKA1	ENRKA2	ENRKA3
DS$BRANCH	000100A8	ENRKA0	ENRKA1	ENRKA2	ENRKA3
DS$BREAK	00010058	ENRKA0	ENRKA1	ENRKA2	ENRKA3
DS$CANWAIT	00010070	ENRKA0			
DS$CHANNEL	00010180	ENRKA0	ENRKA1		
DS$CKLOOP	00010040	ENRKA0	ENRKA1	ENRKA2	ENRKA3
DS$CLRVEC	00010168	ENRKA0			
DS$CNTRLC	00010078	ENRKA0			
DS$CVTREG	000100B0	ENRKA0			
DS$ENDPASS	00010010	ENRKA0			
DS$ENDSUB	00010038	ENRKA0	ENRKA1	ENRKA2	ENRKA3
DS$ERRDEV	000100C8	ENRKA0			
DS$ERRHARD	000100D0	ENRKA0	ENRKA1	ENRKA2	ENRKA3
DS$ERRPREP	00010108	ENRKA0			
DS$ERRSOFT	000100D8	ENRKA0			
DS$ERRSYS	000100C0	ENRKA0	ENRKA1		
DS$ESCAPE	00010050	ENRKA0	ENRKA2	ENRKA3	
DS$FREEDBGSYM	000101B8	ENRKA0			
DS$GETBUF	00010120	ENRKA0	ENRKA2		
DS$GETMEM	00010130	ENRKA0			
DS$GETTERM	000101C8	ENRKA0			

Symbol	Value	Defined By	Referenced By ...
DS$GPHARD	00010018	ENRKA0	
DS$HELP	000101B0	ENRKA0	
DS$INITSCB	00010170	ENRKA0	
DS$INLOOP	00010048	ENRKA0	
DS$K_BBC	0000001D	ENRKA0	
DS$K_BBCC	00000021	ENRKA0	
DS$K_BBCCI	00000023	ENRKA0	
DS$K_BBCS	0000001F	ENRKA0	
DS$K_BBS	0000001C	ENRKA0	
DS$K_BBSC	00000020	ENRKA0	
DS$K_BBSS	0000001E	ENRKA0	
DS$K_BBSSI	00000022	ENRKA0	
DS$K_BCC_M	0000001B	ENRKA0	
DS$K_BCS_M	0000001A	ENRKA0	
DS$K_BEQLU_B	00000005	ENRKA0	
DS$K_BEQLU_M	00000011	ENRKA0	
DS$K_BEQL_B	00000004	ENRKA0	
DS$K_BEQL_M	00000010	ENRKA0	
DS$K_BERROR	00000026	ENRKA0	
DS$K_BGEQU_B	00000007	ENRKA0	
DS$K_BGEQU_M	00000013	ENRKA0	
DS$K_BGEQ_B	00000006	ENRKA0	
DS$K_BGEQ_M	00000012	ENRKA0	
DS$K_BGTRU_B	00000009	ENRKA0	
DS$K_BGTRU_M	00000015	ENRKA0	
DS$K_BGTR_B	00000008	ENRKA0	
DS$K_BGTR_M	00000014	ENRKA0	
DS$K_BLBC	00000025	ENRKA0	
DS$K_BLBS	00000024	ENRKA0	
DS$K_BLEQU_B	00000003	ENRKA0	
DS$K_BLEQU_M	0000000F	ENRKA0	
DS$K_BLEQ_B	00000002	ENRKA0	
DS$K_BLEQ_M	0000000E	ENRKA0	
DS$K_BLSSU_B	00000001	ENRKA0	
DS$K_BLSSU_M	0000000D	ENRKA0	
DS$K_BLSS_B	00000000	ENRKA0	
DS$K_BLSS_M	0000000C	ENRKA0	
DS$K_BNEQU_B	0000000B	ENRKA0	
DS$K_BNEQU_M	00000017	ENRKA0	
DS$K_BNEQ_B	0000000A	ENRKA0	
DS$K_BNEQ_M	00000016	ENRKA0	
DS$K_BNERROR	00000027	ENRKA0	
DS$K_BVC_M	00000019	ENRKA0	
DS$K_BVS_M	00000018	ENRKA0	
DS$LOAD	00010198	ENRKA0	
DS$LOADPCS	000101C0	ENRKA0	
DS$MAPDBGBLOCK	00010118	ENRKA0	
DS$MEMSIZE	000101D8	ENRKA0	
DS$MMOFF	00010158	ENRKA0	
DS$MMON	00010150	ENRKA0	
DS$MOVPHY	00010148	ENRKA0	

Symbol	Value	Defined By	Referenced By ...		
DS$MOVVRT	00010140	ENRKA0			
DS$PARSE	000100B8	ENRKA0			
DS$PRINTB	000100E0	ENRKA0	ENRKA2		
DS$PRINTF	000100F0	ENRKA0	ENRKA3		
DS$PRINTS	000100F8	ENRKA0			
DS$PRINTSIG	00010100	ENRKA0			
DS$PRINTX	000100E8	ENRKA0			
DS$PROBE	000101A0	ENRKA0	ENRKA1		
DS$RELBUF	00010128	ENRKA0	ENRKA2		
DS$RELMEM	00010138	ENRKA0			
DS$SERVICE_DISPATCHER	000101D0	ENRKA0			
DS$SETIPL	00010178	ENRKA0			
DS$SETMAP	00010188	ENRKA0			
DS$SETPRIEXV	00010110	ENRKA0			
DS$SETVEC	00010160	ENRKA0			
DS$SHOCHAN	00010190	ENRKA0			
DS$SUMMARY	00010028	ENRKA0			
DS$WAITMS	00010060	ENRKA0	ENRKA2	ENRKA3	
DS$WAITUS	00010068	ENRKA0	ENRKA3		
DUMP_REG	00005175-R	ENRKA0			
E.BA	00001614-R	ENRKA0	ENRKA1		
E.BC	00001618-R	ENRKA0	ENRKA1		
E.BR	0000163C-R	ENRKA0	ENRKA1		
E.CS	00001610-R	ENRKA0	ENRKA1	ENRKA2	ENRKA3
E.DA	0000161C-R	ENRKA0	ENRKA1		
E.DAT	00001638-R	ENRKA0			
E.HDR	00001630-R	ENRKA0			
E.MP	00001620-R	ENRKA0	ENRKA1	ENRKA3	
E.MP1	00001624-R	ENRKA0			
E.MP2	00001628-R	ENRKA0			
E.MP3	0000162C-R	ENRKA0			
E.VEC	00001640-R	ENRKA0	ENRKA1		
ERR_CNT	0000166C-R	ENRKA0			
ERR_MSG	0000156C-R	ENRKA0	ENRKA1	ENRKA2	ENRKA3
EVENT_FLAGS	000014D8-R	ENRKA0	ENRKA3		
FMT_ATTACH	0000231E-R	ENRKA0			
FMT_AVG_SEEK	00002505-R	ENRKA0			
FMT_BA	0000266C-R	ENRKA0			
FMT_BAD	000022CF-R	ENRKA0			
FMT_BADBLK	00002A04-R	ENRKA0	ENRKA3		
FMT_BC	0000268F-R	ENRKA0			
FMT_CRLF	000022B0-R	ENRKA0			
FMT_CRLF1	000022B5-R	ENRKA0			
FMT_CS	00002649-R	ENRKA0			
FMT_DA	000026B2-R	ENRKA0			
FMT_DATA	00002938-R	ENRKA0			
FMT_DRVTYP	000022F4-R	ENRKA0			
FMT_ERRCOUNT	00002953-R	ENRKA0			
FMT_EXT_1A	00002815-R	ENRKA0			
FMT_EXT_1B	00002838-R	ENRKA0			
FMT_EXT_1C	00002858-R	ENRKA0			

Symbol	Value	Defined By	Referenced By ...		
FMT_EXT_2	0000287A-R	ENRKA0			
FMT_GOOD	000022B8-R	ENRKA0			
FMT_HDR	000025D8-R	ENRKA0			
FMT_HDR_A	0000289B-R	ENRKA0			
FMT_HEADER1	0000290E-R	ENRKA0			
FMT_HMB_ERROR	000025B7-R	ENRKA0			
FMT_MAX_SEEK	000024D5-R	ENRKA0			
FMT_MP	000026D5-R	ENRKA0			
FMT_MP1_02	0000273E-R	ENRKA0			
FMT_MP1_80	00002769-R	ENRKA0			
FMT_MP2_02	00002794-R	ENRKA0			
FMT_MP2_80	000027BF-R	ENRKA0			
FMT_MP3	000027EB-R	ENRKA0			
FMT_NYI	00002535-R	ENRKA0			
FMT_ONE_SEEK	000024A9-R	ENRKA0			
FMT_PAT	0000271B-R	ENRKA0			
FMT_POS	000026F8-R	ENRKA0			
FMT_SEEK1	0000236A-R	ENRKA0			
FMT_SEEK1_SP	000023F0-R	ENRKA0			
FMT_SEEK2	00002394-R	ENRKA0			
FMT_SEEK2_SP	0000242B-R	ENRKA0			
FMT_SEEK3	000023C2-R	ENRKA0			
FMT_SEEK3_SP	0000246A-R	ENRKA0			
FMT_SPINDLE	0000297A-R	ENRKA0			
FMT_SPINDLE_SP	00002998-R	ENRKA0			
FMT_SP_TIME	000029DB-R	ENRKA0			
FMT_VOL	0000255C-R	ENRKA0			
FMT_WDCNT	000028F3-R	ENRKA0			
FMT_XOR	000022E5-R	ENRKA0			
GDDAT	000015B8-R	ENRKA0			
GET_LUN$A	000062B8-R	ENRKA0	ENRKA1	ENRKA2	ENRKA3
GET_POSITION	00005EFC-R	ENRKA0	ENRKA1	ENRKA2	ENRKA3
GET_RANDOM	000061A5-R	ENRKA0	ENRKA2	ENRKA3	
GET_SECLUN$A	00006303-R	ENRKA0	ENRKA3		
GOOD_DATA	00001644-R	ENRKA0			
HARD_TABLE	0000150C-R	ENRKA0			
HDR02_BITS	000020F8-R	ENRKA0			
HDR1_02_BITS	0000209C-R	ENRKA0			
HDR1_80_BITS	000020BD-R	ENRKA0			
HDR2_80_BITS	000020DC-R	ENRKA0			
HDR80_BITS	00002129-R	ENRKA0			
HMB_ERR_FLAG	00001530-R	ENRKA0			
HOLD_CS	00001554-R	ENRKA0			
HOLD_DA	000015AC-R	ENRKA0			
HP$A_DEPENDENT	00000032	ENRKA0			
HP$A_DEVICE	00000018	ENRKA0			
HP$A_DVA	0000001C	ENRKA0			
HP$A_LINK	00000020	ENRKA0			
HP$B_DRIVE	0000000B	ENRKA0			
HP$B_FLAGS	0000000A	ENRKA0			
HP$M_ALLOC	00000001	ENRKA0			

Symbol	Value	Defined By	Referenced By ...		
HP$M_NAME	00000004	ENRKA0			
HP$M_WASALL	00000002	ENRKA0			
HP$Q_DEVICE	00000000	ENRKA0			
HP$T_DEVICE	0000000C	ENRKA0			
HP$T_TYPE	00000026	ENRKA0			
HP$V_ALLOC	00000000	ENRKA0			
HP$V_NAME	00000002	ENRKA0			
HP$V_WASALL	00000001	ENRKA0			
HP$W_SIZE	00000008	ENRKA0			
HP$W_VECTOR	00000024	ENRKA0			
IDCBAR_BITS	00001DBB-R	ENRKA0			
IDCCNT_BITS	00001DDC-R	ENRKA0			
IDCCSR_BITS	00001CAD-R	ENRKA0			
IDCDSINT_ERR	00002FD7-R	ENRKA0			
IDCENINT_ERR	00002FB5-R	ENRKA0			
IDCGETBUF	00002FFA-R	ENRKA0	ENRKA2		
IDCMPR02_BITS	00001F38-R	ENRKA0			
IDCMPR80_BITS	00001FD3-R	ENRKA0			
IDCPAT_BITS	0000207A-R	ENRKA0			
IDCPOS_BITS	00002059-R	ENRKA0			
IDC_ADRFAIL	00002F02-R	ENRKA0	ENRKA1		
IDC_ISR$A	00006344-R	ENRKA0			
IDC_SADRFAIL	00002F2D-R	ENRKA0	ENRKA1		
IDC_STAT$L_1	000014C0-R	ENRKA0	ENRKA1		
IDC_STAT$W_2	000014C4-R	ENRKA0	ENRKA1		
IDC_STAT$W_VR	000014C6-R	ENRKA0	ENRKA1		
IDC_STATUS	000014C0-R	ENRKA0			
INH_PNT	000015BC-R	ENRKA0			
ITERAT	000015C4-R	ENRKA0			
L$UNIT	000014E0-R	ENRKA0	ENRKA1	ENRKA2	ENRKA3
LUN0_TBL	000017A0-R	ENRKA0			
LUN1_TBL	00001824-R	ENRKA0			
LUN2_TBL	000018A8-R	ENRKA0			
LUN3_TBL	0000192C-R	ENRKA0			
LUN_TBL	00001790-R	ENRKA0			
MANUAL	00000001	ENRKA0			
MBSF	00000E00-R	ENRKA0			
MBSF_BUFFER	00001788-R	ENRKA0			
MPR02_BITS	00001F89-R	ENRKA0			
MPR80_BITS	00002011-R	ENRKA0			
MS100_DAT_DST	0000476C-R	ENRKA0	ENRKA3		
MS103_SK_ONE	0000479A-R	ENRKA0			
MS104_SK_MAX	000047C2-R	ENRKA0			
MS105_SK_AVG	000047EE-R	ENRKA0			
MS106_SP_TIM	0000481A-R	ENRKA0			
MS107A_SEEKERR	00004866-R	ENRKA0	ENRKA1		
MS107B_RECAL	0000488C-R	ENRKA0	ENRKA1		
MS107_RECAL	00004837-R	ENRKA0	ENRKA1		
MS108_SECTNZ	000048B2-R	ENRKA0	ENRKA2		
MS109_BADCNT	000048E3-R	ENRKA0	ENRKA2		
MS110_UEHNF31	00004917-R	ENRKA0			

Symbol	Value	Defined By	Referenced By ...		
MS111_EHNF31	0000494F-R	ENRKA0			
MS112_ACC31	00004989-R	ENRKA0			
MS113_UWRTERR	000049A9-R	ENRKA0	ENRKA3		
MS114A_UWCKERR	00004A0E-R	ENRKA0	ENRKA3		
MS114_URDERR	000049DD-R	ENRKA0	ENRKA3		
MS115A_WKRST	00004A85-R	ENRKA0	ENRKA3		
MS115_RDRST	00004A46-R	ENRKA0	ENRKA3		
MS116A_AUTOCMP	00004AEF-R	ENRKA0	ENRKA3		
MS116_SKPCMP	00004AC1-R	ENRKA0	ENRKA3		
MS117_AUTORD	00004B5B-R	ENRKA0	ENRKA3		
MS117_AUTOWCK	00004B9A-R	ENRKA0	ENRKA3		
MS117_AUTOWRT	00004B22-R	ENRKA0	ENRKA3		
MS118_SEEKIDC	00004BD2-R	ENRKA0	ENRKA3		
MS119_IDCPOS	00004C0D-R	ENRKA0	ENRKA3		
MS44A_INC_CRC	00003B47-R	ENRKA0			
MS45A_HD_PROX	00003B95-R	ENRKA0	ENRKA1		
MS83A_ST4_DE	000043CE-R	ENRKA0	ENRKA3		
MS94A_NO_OTH	00004637-R	ENRKA0			
MSG10_RG_INIT	000032FF-R	ENRKA0			
MSG11_GS_CRDY	0000332A-R	ENRKA0			
MSG12_GS_DRDY	0000335E-R	ENRKA0			
MSG13_INI_ERR	00003378-R	ENRKA0			
MSG14_SYSINIT	000033A6-R	ENRKA0	ENRKA1	ENRKA2	ENRKA3
MSG15_NOOP	000033D1-R	ENRKA0	ENRKA1		
MSG16A_WRTCHK	00003422-R	ENRKA0	ENRKA2		
MSG16B_WRTCHK	0000344B-R	ENRKA0	ENRKA2		
MSG16_WRTCHK	000033F6-R	ENRKA0	ENRKA2	ENRKA3	
MSG17_GETSTAT	00003480-R	ENRKA0	ENRKA1	ENRKA2	ENRKA3
MSG18B_SEEK	000034D0-R	ENRKA0	ENRKA3		
MSG18_SEEK	000034AB-R	ENRKA0			
MSG19A_READH	0000351E-R	ENRKA0	ENRKA1		
MSG19_READH	000034F2-R	ENRKA0	ENRKA1	ENRKA2	ENRKA3
MSG1_ADR_ERR	0000317B-R	ENRKA0			
MSG20A_FORMAT	000035E6-R	ENRKA0	ENRKA3		
MSG20A_WRITE	000035B3-R	ENRKA0	ENRKA2		
MSG20B_WRITE	0000360A-R	ENRKA0	ENRKA2		
MSG20C_WRITE	00003642-R	ENRKA0	ENRKA3		
MSG20_WRITE	0000355A-R	ENRKA0	ENRKA2	ENRKA3	
MSG20_WRT_BND	00003585-R	ENRKA0	ENRKA3		
MSG21A_READ	0000369E-R	ENRKA0	ENRKA2		
MSG21A_READH	000036D0-R	ENRKA0			
MSG21B_READ	000036FD-R	ENRKA0	ENRKA2		
MSG21_READ	00003674-R	ENRKA0	ENRKA2	ENRKA3	
MSG22_READNHC	00003734-R	ENRKA0	ENRKA3		
MSG23_CS_INC	0000376C-R	ENRKA0	ENRKA1	ENRKA2	ENRKA3
MSG24A_BC_MOD	000037AF-R	ENRKA0	ENRKA1		
MSG24_BA_MOD	00003792-R	ENRKA0	ENRKA1		
MSG25_DA_MOD	000037CC-R	ENRKA0	ENRKA1		
MSG26_MP_MOD	000037E9-R	ENRKA0			
MSG27_MP_INC	00003806-R	ENRKA0	ENRKA1		
MSG28_DA_INC	00003827-R	ENRKA0	ENRKA3		

Symbol	Value	Defined By	Referenced By ...	
MSG29_BA_INC	00003847-R	ENRKA0		
MSG2_RG_ZERO	0000319D-R	ENRKA0	ENRKA1	
MSG30A_CRDY	00003885-R	ENRKA0	ENRKA3	
MSG30_CRDY	00003866-R	ENRKA0	ENRKA2	
MSG31_DRDY	000038BC-R	ENRKA0		
MSG32_CERR	000038D6-R	ENRKA0		
MSG33A_UXEATTN	00003905-R	ENRKA0		
MSG33B_NOATTN	00003920-R	ENRKA0		
MSG33_DERR	000038F0-R	ENRKA0		
MSG34_GS_OPI	0000393E-R	ENRKA0		
MSG35_OPI_TO	0000397B-R	ENRKA0	ENRKA2	
MSG36_OPI_RLY	0000399F-R	ENRKA0	ENRKA2	
MSG37_OPI_LAT	000039CB-R	ENRKA0	ENRKA2	
MSG38_NOOP_IE	000039F5-R	ENRKA0	ENRKA1	
MSG39A_INT	00003A43-R	ENRKA0		
MSG39_NO_INT	00003A1F-R	ENRKA0	ENRKA1	
MSG3_RG_ONE	000031B7-R	ENRKA0	ENRKA1	
MSG40_BAD_BR	00003A75-R	ENRKA0	ENRKA1	
MSG41_BAD_VEC	00003A9E-R	ENRKA0	ENRKA1	
MSG42_RPT_CYL	00003AC5-R	ENRKA0	ENRKA1	
MSG43_MP2_NZ	00003AFD-R	ENRKA0		
MSG44_BAD_CRC	00003B1E-R	ENRKA0	ENRKA1	
MSG45_SK_ZERO	00003B75-R	ENRKA0	ENRKA1	
MSG46_POS_ERR	00003BD0-R	ENRKA0	ENRKA1	
MSG47_DS_INC	00003BEA-R	ENRKA0	ENRKA1	ENRKA3
MSG48_SK_ONE	00003C01-R	ENRKA0	ENRKA1	
MSG49_DRDY_NC	00003C1B-R	ENRKA0	ENRKA1	
MSG4_RG_FLT1	000031D1-R	ENRKA0	ENRKA1	
MSG50_SK_OUTR	00003C35-R	ENRKA0		
MSG51_SK_INNR	00003C59-R	ENRKA0	ENRKA1	
MSG52_RD_HNF	00003C7D-R	ENRKA0	ENRKA2	
MSG53_WRT_NXM	00003CAA-R	ENRKA0	ENRKA2	
MSG54_DA_UPD	00003CDB-R	ENRKA0	ENRKA2	ENRKA3
MSG55_DAT_BND	00003D43-R	ENRKA0	ENRKA3	
MSG55_DAT_CP1	00003D0E-R	ENRKA0	ENRKA2	
MSG55_HDRCMP	00003D77-R	ENRKA0		
MSG56_DAT_UNQ	00003DA8-R	ENRKA0		
MSG57_WCK_FRC	00003DEF-R	ENRKA0	ENRKA2	
MSG58_PAR_WRT	00003E19-R	ENRKA0	ENRKA2	
MSG59_PAR_RD	00003E4C-R	ENRKA0	ENRKA2	
MSG5_RG_FLT0	000031F4-R	ENRKA0	ENRKA1	
MSG60_WRT_PRD	00003E7E-R	ENRKA0	ENRKA2	
MSG61_DAT_PAR	00003EB6-R	ENRKA0	ENRKA2	
MSG62_DC_PAR	00003EF6-R	ENRKA0	ENRKA2	
MSG63_PRD_TMW	00003F30-R	ENRKA0	ENRKA2	
MSG64_WCK_PAR	00003F5D-R	ENRKA0	ENRKA2	
MSG65_DAT_CP2	00003F90-R	ENRKA0	ENRKA2	
MSG66_HNF_LT	00003FCA-R	ENRKA0		
MSG67_DAT_ES	00003FFF-R	ENRKA0	ENRKA2	
MSG68_DAT_MAX	00004047-R	ENRKA0	ENRKA2	
MSG69_DAT_NHC	0000408A-R	ENRKA0	ENRKA3	

Symbol	Value	Defined By	Referenced By ...
MSG6_CS_FAIL	00003217-R	ENRKA0	ENRKA1
MSG70A_NFSEC31	000040EB-R	ENRKA0	
MSG70_FND_S39	000040C7-R	ENRKA0	
MSG71_RD_BKT	00004107-R	ENRKA0	
MSG72_DAT_BKT	0000414B-R	ENRKA0	
MSG73_GS_UND	00004189-R	ENRKA0	ENRKA3
MSG74_DRDY_NC	000041B5-R	ENRKA0	ENRKA3
MSG75_NO_ST1	000041E0-R	ENRKA0	ENRKA3
MSG76_ST1_BS	0000420D-R	ENRKA0	ENRKA3
MSG77_NO_ST2	00004239-R	ENRKA0	ENRKA3
MSG77_NO_ST3	00004266-R	ENRKA0	ENRKA3
MSG78_ST2_BS	00004293-R	ENRKA0	ENRKA3
MSG79_BH_NC	000042C3-R	ENRKA0	ENRKA3
MSG7A_BC_FAIL	00003261-R	ENRKA0	ENRKA1
MSG7_BA_FAIL	0000323C-R	ENRKA0	ENRKA1
MSG80_BH_NF	000042FD-R	ENRKA0	ENRKA3
MSG81_ST3_BS	00004338-R	ENRKA0	ENRKA3
MSG82_NO_ST4	00004367-R	ENRKA0	ENRKA3
MSG83_ST4_BS	00004394-R	ENRKA0	ENRKA3
MSG84_NO_ST5	00004407-R	ENRKA0	ENRKA3
MSG85_ST5_BS	00004434-R	ENRKA0	ENRKA3
MSG86_DRDY_NS	00004468-R	ENRKA0	ENRKA3
MSG87_VC_NC	000044A0-R	ENRKA0	ENRKA3
MSG88_NO_ST6	000044D9-R	ENRKA0	ENRKA3
MSG89_ST6_BS	00004506-R	ENRKA0	ENRKA3
MSG8A_MP_FAIL	000032AB-R	ENRKA0	
MSG8_DA_FAIL	00003286-R	ENRKA0	ENRKA1
MSG90_NO_ST7	0000453A-R	ENRKA0	ENRKA3
MSG91_ST7_BS	00004567-R	ENRKA0	ENRKA3
MSG92_NO_ST0	00004595-R	ENRKA0	ENRKA3
MSG93_RELV_UP	000045C2-R	ENRKA0	ENRKA3
MSG94_UXE_DSE	000045FB-R	ENRKA0	
MSG95_DSE_NS	00004662-R	ENRKA0	
MSG96_DSE_NC	0000469B-R	ENRKA0	
MSG97_GS_NXD	000046CF-R	ENRKA0	ENRKA3
MSG98_WRT_WL	00004701-R	ENRKA0	ENRKA3
MSG99_RST_WGE	0000472F-R	ENRKA0	ENRKA3
MSG9_RG_RMW	000032D0-R	ENRKA0	ENRKA1
MSG_ABORT	000030A2-R	ENRKA0	
MSG_AVG	00003076-R	ENRKA0	
MSG_BLANK	00003099-R	ENRKA0	
MSG_DRVFATAL	000030F2-R	ENRKA0	
MSG_ECCFAIL	00003143-R	ENRKA0	
MSG_ECCPASS	0000311E-R	ENRKA0	
MSG_INNER	00003090-R	ENRKA0	
MSG_MAX	0000306E-R	ENRKA0	
MSG_MIDDLE	00003087-R	ENRKA0	
MSG_ONE	0000306A-R	ENRKA0	
MSG_OPERATTCH	000030C3-R	ENRKA0	
MSG_OUTER	0000307E-R	ENRKA0	
MSG_R80	00003066-R	ENRKA0	

Symbol	Value	Defined By	Referenced By ...		
MSG_RL02	00003061-R	ENRKA0			
MSG_TRACE	00003041-R	ENRKA0			
NEW_CYLN	000015A0-R	ENRKA0	ENRKA1		
OLD_CYLN	0000159C-R	ENRKA0			
OPR10_INS_ID	00002CB2-R	ENRKA0			
OPR11_ORIGIN	00002D26-R	ENRKA0	ENRKA3		
OPR12_WRTLOCK	00002D8D-R	ENRKA0	ENRKA3		
OPR13A_RST_FLT	00002DD1-R	ENRKA0	ENRKA3		
OPR13_RST_WL	00002DAE-R	ENRKA0	ENRKA3		
OPR14_ROTATE	00002DF1-R	ENRKA0			
OPR15_SWAP	00002E4C-R	ENRKA0			
OPR16_SWAP_BK	00002E73-R	ENRKA0			
OPR1_UNLOAD	00002A4F-R	ENRKA0	ENRKA3		
OPR2_CLOSE	00002ADD-R	ENRKA0	ENRKA3		
OPR3_START	00002B1A-R	ENRKA0	ENRKA3		
OPR4_MKRDY	00002B47-R	ENRKA0			
OPR5_UNLOAD	00002B7B-R	ENRKA0	ENRKA3		
OPR6_REMOVE	00002BAA-R	ENRKA0	ENRKA3		
OPR7_INSERT	00002BD7-R	ENRKA0	ENRKA3		
OPR8_SETUP	00002BF8-R	ENRKA0	ENRKA3		
OPR9_DUPPLUG	00002C28-R	ENRKA0			
OPR_AVAIL	00002C9F-R	ENRKA0			
OPR_FAULT	00002ECB-R	ENRKA0			
OPR_RETURN	00002A39-R	ENRKA0	ENRKA3		
P$ATTN	000014F0-R	ENRKA0			
P$UNIT	000014E8-R	ENRKA0	ENRKA1	ENRKA2	ENRKA3
PHY_RD_BUF	00001520-R	ENRKA0	ENRKA2		
PHY_WRT_BUF	00001510-R	ENRKA0	ENRKA2		
PNT_FLAG	000015C0-R	ENRKA0			
POSITION_HDS	00005E0A-R	ENRKA0	ENRKA1	ENRKA2	ENRKA3
POS_TYPE	00001580-R	ENRKA0			
PREV_CYLN	00001594-R	ENRKA0			
PRINT_BA	00005278-R	ENRKA0			
PRINT_BC	000052C2-R	ENRKA0			
PRINT_CS	0000522B-R	ENRKA0			
PRINT_DATA_ERR	00005704-R	ENRKA0	ENRKA2	ENRKA3	
PRINT_DA_02	0000530C-R	ENRKA0			
PRINT_DA_80	0000539A-R	ENRKA0			
PRINT_EXT_1	000056A6-R	ENRKA0			
PRINT_EXT_2	000056F2-R	ENRKA0			
PRINT_MP1_02	0000554E-R	ENRKA0			
PRINT_MP1_80	0000559B-R	ENRKA0			
PRINT_MP2_02	000055E6-R	ENRKA0			
PRINT_MP2_80	00005626-R	ENRKA0			
PRINT_MP3	00005666-R	ENRKA0			
PRINT_MP_02	00005422-R	ENRKA0			
PRINT_MP_80	00005470-R	ENRKA0			
PRINT_PAT	00005504-R	ENRKA0			
PRINT_POS	000054BA-R	ENRKA0			
PRINT_REG	00005123-R	ENRKA0	ENRKA1	ENRKA2	ENRKA3
PRINT_REG_HDR	0000521D-R	ENRKA0			

Symbol	Value	Defined By	Referenced By ...		
R80_HDR_TMP	00001AC7-R	ENRKA0			
R80_TBL	00001A34-R	ENRKA0			
RD_BAD_BLK	000063A5-R	ENRKA0	ENRKA3		
RD_HOME_BLK	00006153-R	ENRKA0			
RD_IDCREGS	0000593B-R	ENRKA0	ENRKA1	ENRKA2	ENRKA3
READ_BUFFER	00001778-R	ENRKA0	ENRKA2	ENRKA3	
READ_HEADERS	00005FF1-R	ENRKA0			
REG_ERRPT	0000504C-R	ENRKA0	ENRKA1	ENRKA2	ENRKA3
RESTORE_DRIVE	00005BAC-R	ENRKA0			
RGLIST	0000155C-R	ENRKA0	ENRKA1	ENRKA3	
RL02_HDR_TMP	00001AB8-R	ENRKA0			
RL02_TBL	000019B0-R	ENRKA0			
SECD_ADDR	000014CC-R	ENRKA0	ENRKA1		
SECTOR	00001578-R	ENRKA0			
SEED	000015A8-R	ENRKA0	ENRKA3		
SEEK_TYPE	00001564-R	ENRKA0			
SKIP_BUFFER	00001780-R	ENRKA0	ENRKA3		
START_CMD	000059B2-R	ENRKA0	ENRKA1	ENRKA2	ENRKA3
STATUS_CHK	00005A6E-R	ENRKA0	ENRKA1	ENRKA2	ENRKA3
STEP_HDS	00005CFD-R	ENRKA0	ENRKA1		
SUBSYS_INIT	0000586F-R	ENRKA0	ENRKA1	ENRKA2	ENRKA3
SURFACE	0000157C-R	ENRKA0			
SYS$ALLOC	00010238	ENRKA0			
SYS$ASCTIM	00010248	ENRKA0			
SYS$ASSIGN	00010250	ENRKA0			
SYS$BINTIM	00010258	ENRKA0			
SYS$CANCEL	00010260	ENRKA0			
SYS$CANTIM	00010268	ENRKA0	ENRKA3		
SYS$CLOSE	000105B8	ENRKA0			
SYS$CLREF	00010298	ENRKA0			
SYS$CONNECT	000105C0	ENRKA0			
SYS$DALLOC	000102D8	ENRKA0			
SYS$DASSGN	000102E0	ENRKA0			
SYS$DISCONNECT	000105D0	ENRKA0			
SYS$FAO	00010350	ENRKA0			
SYS$FAOL	00010358	ENRKA0			
SYS$GET	00010580	ENRKA0			
SYS$GETCHN	000104C8	ENRKA0			
SYS$GETTIM	00010378	ENRKA0			
SYS$LKWSET	000103A0	ENRKA0			
SYS$NUMTIM	000103B8	ENRKA0			
SYS$OPEN	00010608	ENRKA0			
SYS$QIO	000103C8	ENRKA0			
SYS$QIOW	00010200	ENRKA0			
SYS$READ	00010590	ENRKA0			
SYS$READEF	000103D0	ENRKA0	ENRKA3		
SYS$SETAST	000103F8	ENRKA0			
SYS$SETEF	00010400	ENRKA0			
SYS$SETIMR	00010420	ENRKA0	ENRKA3		
SYS$SETPRI	00010428	ENRKA0			
SYS$SETPRT	00010430	ENRKA0			

Symbol	Value	Defined By	Referenced By ...		
SYS$SETRWM	00010438	ENRKA0			
SYS$SETSWM	00010448	ENRKA0			
SYS$ULKPAG	00010460	ENRKA0			
SYS$ULWSET	00010468	ENRKA0			
SYS$UNWIND	00010470	ENRKA0			
SYS$WAITFR	00010478	ENRKA0			
SYS$WFLAND	00010488	ENRKA0			
SYS$WFLOR	00010490	ENRKA0			
SYS_TIME	000014D0-R	ENRKA0			
S_DRIVE_TBL	000014B8-R	ENRKA0	ENRKA3		
S_L$UNIT	000014E4-R	ENRKA0			
S_P$ATTN	000014F4-R	ENRKA0			
S_P$UNIT	000014EC-R	ENRKA0	ENRKA3		
T.BA	000015D8-R	ENRKA0	ENRKA1		
T.BC	000015DC-R	ENRKA0	ENRKA1		
T.BR	00001608-R	ENRKA0	ENRKA1		
T.CS	000015D4-R	ENRKA0	ENRKA1	ENRKA2	ENRKA3
T.DA	000015E0-R	ENRKA0	ENRKA1	ENRKA2	ENRKA3
T.DAT	00001604-R	ENRKA0			
T.HDR	000015F4-R	ENRKA0	ENRKA1		
T.MP	000015E4-R	ENRKA0	ENRKA1	ENRKA2	ENRKA3
T.MP1	000015E8-R	ENRKA0	ENRKA1		
T.MP2	000015EC-R	ENRKA0	ENRKA1		
T.MP3	000015F0-R	ENRKA0	ENRKA1		
T.PAT	000015FC-R	ENRKA0			
T.POS	00001600-R	ENRKA0			
T.VEC	0000160C-R	ENRKA0	ENRKA1		
T16_S1	00007F45-R	ENRKA1			
T16_S2	0000802F-R	ENRKA1			
T16_S3	000081A6-R	ENRKA1			
T21_S1	000089AC-R	ENRKA2			
T21_S2	00008A6E-R	ENRKA2			
T22_S1	00008CC2-R	ENRKA2			
T22_S2	00008D59-R	ENRKA2			
T25_S1	000092AB-R	ENRKA2			
T25_S2	00009335-R	ENRKA2			
T25_S3	0000941A-R	ENRKA2			
T26_S1	00009591-R	ENRKA2			
T26_S2	0000961A-R	ENRKA2			
T26_S3	00009687-R	ENRKA2			
T27_S1	000097F4-R	ENRKA2			
T27_S2	0000987D-R	ENRKA2			
T27_S3	000098E3-R	ENRKA2			
T27_S4	00009988-R	ENRKA2			
T28_S1	00009A7A-R	ENRKA2			
T28_S2	00009B03-R	ENRKA2			
T28_S3	00009BD3-R	ENRKA2			
T28_S4	00009C8A-R	ENRKA2			
T28_S5	00009D92-R	ENRKA2			
T29_S1	00009F31-R	ENRKA2			
T29_S2	00009FBB-R	ENRKA2			

Symbol	Value	Defined By	Referenced By ...
T29_S3	0000A093-R	ENRKA2	
T29_S4	0000A179-R	ENRKA2	
T30_S1	0000A262-R	ENRKA2	
T30_S2	0000A2EB-R	ENRKA2	
T30_S3	0000A3C1-R	ENRKA2	
T31_S1	0000A69C-R	ENRKA2	
T31_S2	0000A726-R	ENRKA2	
T31_S3	0000A78C-R	ENRKA2	
T31_S4	0000A831-R	ENRKA2	
T32_S1	0000A936-R	ENRKA3	
T32_S2	0000A9BF-R	ENRKA3	
T32_S3	0000AA63-R	ENRKA3	
T34_S1	0000AE13-R	ENRKA3	
T34_S2	0000AE9C-R	ENRKA3	
T35_S1	0000B0B9-R	ENRKA3	
T35_S2	0000B142-R	ENRKA3	
T36_S1	0000B2CD-R	ENRKA3	
T36_S2	0000B356-R	ENRKA3	
T36_S3	0000B3CE-R	ENRKA3	
T36_S4	0000B5E7-R	ENRKA3	
T37_S1	0000B7D5-R	ENRKA3	
T37_S2	0000B85E-R	ENRKA3	
T37_S3	0000B8D6-R	ENRKA3	
T37_S4	0000BA6A-R	ENRKA3	
T37_S5	0000BBA0-R	ENRKA3	
T3_S1	00006627-R	ENRKA1	
T3_S2	00006696-R	ENRKA1	
T3_S3	000066FA-R	ENRKA1	
T3_S4	00006775-R	ENRKA1	
T3_S5	000067F8-R	ENRKA1	
T44_S1	0000D252-R	ENRKA3	
T44_S2	0000D2DB-R	ENRKA3	
T44_S3	0000D391-R	ENRKA3	
T4_S1	0000689F-R	ENRKA1	
T4_S2	00006901-R	ENRKA1	
T4_S3	00006954-R	ENRKA1	
T4_S4	000069B4-R	ENRKA1	
T4_S5	00006A1C-R	ENRKA1	
T5_S1	00006AAF-R	ENRKA1	
T5_S2	00006B0F-R	ENRKA1	
T5_S3	00006B62-R	ENRKA1	
T5_S4	00006BC2-R	ENRKA1	
T5_S5	00006C2A-R	ENRKA1	
T6_S1	00006CBB-R	ENRKA1	
T6_S2	00006D2A-R	ENRKA1	
T6_S3	00006D7D-R	ENRKA1	
T6_S4	00006DDD-R	ENRKA1	
T6_S5	00006E45-R	ENRKA1	
TEMP	000015CC-R	ENRKA0	
TEST_001	0000649C-R	ENRKA1	
TEST_001_X	000064F0-R	ENRKA1	

Symbol	Value	Defined By	Referenced By ...
TEST_002	0000651C-R	ENRKA1	
TEST_002_X	000065F6-R	ENRKA1	
TEST_003	0000661C-R	ENRKA1	
TEST_003_X	00006870-R	ENRKA1	
TEST_004	00006894-R	ENRKA1	
TEST_004_X	00006A7F-R	ENRKA1	
TEST_005	00006AA4-R	ENRKA1	
TEST_005_X	00006C8B-R	ENRKA1	
TEST_006	00006CB0-R	ENRKA1	
TEST_006_X	00006EA8-R	ENRKA1	
TEST_007	00006EF0-R	ENRKA1	
TEST_007_X	00006F93-R	ENRKA1	
TEST_008	00006FD8-R	ENRKA1	
TEST_008_X	000070F0-R	ENRKA1	
TEST_009	00007130-R	ENRKA1	
TEST_009_X	000072B1-R	ENRKA1	
TEST_010	00007304-R	ENRKA1	
TEST_010_X	00007437-R	ENRKA1	
TEST_011	00007480-R	ENRKA1	
TEST_011_X	00007501-R	ENRKA1	
TEST_012	00007548-R	ENRKA1	
TEST_012_X	00007626-R	ENRKA1	
TEST_013	00007710-R	ENRKA1	
TEST_013_X	000078B1-R	ENRKA1	
TEST_014	00007900-R	ENRKA1	
TEST_014_X	00007A7E-R	ENRKA1	
TEST_015	00007B6C-R	ENRKA1	
TEST_015_X	00007E1B-R	ENRKA1	
TEST_016	00007F14-R	ENRKA1	
TEST_016_X	00008295-R	ENRKA1	
TEST_017	00008388-R	ENRKA1	
TEST_017_X	000084BE-R	ENRKA1	
TEST_018	000085B0-R	ENRKA2	
TEST_018_X	00008709-R	ENRKA2	
TEST_019	00008754-R	ENRKA2	
TEST_019_X	00008849-R	ENRKA2	
TEST_020	0000888C-R	ENRKA2	
TEST_020_X	0000894C-R	ENRKA2	
TEST_021	00008990-R	ENRKA2	
TEST_021_X	00008C43-R	ENRKA2	
TEST_022	00008C94-R	ENRKA2	
TEST_022_X	00008E71-R	ENRKA2	
TEST_023	00008EC4-R	ENRKA2	
TEST_023_X	000090A0-R	ENRKA2	
TEST_024	000090EC-R	ENRKA2	
TEST_024_X	0000918C-R	ENRKA2	
TEST_025	00009274-R	ENRKA2	
TEST_025_X	00009503-R	ENRKA2	
TEST_026	0000954C-R	ENRKA2	
TEST_026_X	00009763-R	ENRKA2	
TEST_027	000097A8-R	ENRKA2	

Symbol	Value	Defined By	Referenced By ...	
TEST_027_X	000099F1-R	ENRKA2		
TEST_028	00009A44-R	ENRKA2		
TEST_028_X	00009DFB-R	ENRKA2		
TEST_029	00009EEC-R	ENRKA2		
TEST_029_X	0000A1E2-R	ENRKA2		
TEST_030	0000A22C-R	ENRKA2		
TEST_030_X	0000A4E1-R	ENRKA2		
TEST_031	0000A5D0-R	ENRKA2		
TEST_031_X	0000A8AB-R	ENRKA2		
TEST_032	0000A900-R	ENRKA3		
TEST_032_X	0000ABDF-R	ENRKA3		
TEST_033	0000AC20-R	ENRKA3		
TEST_033_X	0000AD97-R	ENRKA3		
TEST_034	0000ADD8-R	ENRKA3		
TEST_034_X	0000B019-R	ENRKA3		
TEST_035	0000B06C-R	ENRKA3		
TEST_035_X	0000B232-R	ENRKA3		
TEST_036	0000B280-R	ENRKA3		
TEST_036_X	0000B735-R	ENRKA3		
TEST_037	0000B788-R	ENRKA3		
TEST_037_X	0000BC99-R	ENRKA3		
TEST_038	0000BCE8-R	ENRKA3		
TEST_038_X	0000BE3C-R	ENRKA3		
TEST_039	0000BE84-R	ENRKA3		
TEST_039_X	0000C0F7-R	ENRKA3		
TEST_040	0000C13C-R	ENRKA3		
TEST_040_X	0000C355-R	ENRKA3		
TEST_041	0000C3A0-R	ENRKA3		
TEST_041_X	0000CB9E-R	ENRKA3		
TEST_042	0000CBEC-R	ENRKA3		
TEST_042_X	0000CF39-R	ENRKA3		
TEST_043	0000CF7C-R	ENRKA3		
TEST_043_X	0000D18F-R	ENRKA3		
TEST_044	0000D1D4-R	ENRKA3		
TEST_044_X	0000D6B8-R	ENRKA3		
TIME_300MS	00001548-R	ENRKA0		
TIME_800MS	00001540-R	ENRKA0		
TIME_LEFT	00001550-R	ENRKA0		
TIME_OUT	00001534-R	ENRKA0	ENRKA3	
T_DEVICE	00001B8B-R	ENRKA0		
UBACLEAR_ERR	00002F90-R	ENRKA0	ENRKA1	
UBAINIT_ERR	00002FA0-R	ENRKA0	ENRKA1	
UBAMAP_ERR	00002F6E-R	ENRKA0		
UBAPURGE_ERR	00002F80-R	ENRKA0		
UNBINIT_ERR	00002F5B-R	ENRKA0		
VIR_RD_BUF	00001528-R	ENRKA0	ENRKA2	
VIR_WRT_BUF	00001518-R	ENRKA0	ENRKA2	
VOL_NAME	00001768-R	ENRKA0		
WORD_COUNT	00001670-R	ENRKA0	ENRKA2	ENRKA3
WORK	000015C8-R	ENRKA0	ENRKA1	ENRKA2
WRITE_BUFFER	00001770-R	ENRKA0	ENRKA2	ENRKA3

```
Symbol              Value       Defined By     Referenced By ...
------              -----       ----------     -----------------


          Key for special characters above:
                  +--------------------+
                  ! *  - Undefined     !
                  ! U  - Universal     !
                  ! R  - Relocatable   !
                  ! X  - External      !
                  +--------------------+
```

```
                                        +-----------------+
                                        ! Image Synopsis !
                                        +-----------------+

Virtual memory allocated:                00000200 0000D7FF 0000D600 (54784. bytes, 107. pages)
Stack size:                                     0. pages
Image binary virtual block limits:              1.      107. (   107. blocks)
Image name and identification:           ENRKA 1.1
Number of files:                                4.
Number of modules:                              4.
Number of program sections:                    60.
Number of global symbols:                     777.
Number of cross references:                  1089.
Number of image sections:                       1.
Image type:                              SYSTEM.
Map format:                              DEFAULT WITH CROSS REFERENCE in file $1$DUA9:[BOWSER.TAPE]ENRKA.MAP;1
Estimated map length:                    170. blocks
                                      +----------------------+
                                      ! Link Run Statistics !
                                      +----------------------+

Performance Indicators                   Page Faults  CPU Time        Elapsed Time
----------------------                   -----------  --------        ------------
    Command processing:                           73  00:00:00.43     00:00:00.91
    Pass 1:                                      167  00:00:01.80     00:00:03.30
    Allocation/Relocation:                         7  00:00:00.18     00:00:00.56
    Pass 2:                                      113  00:00:04.31     00:00:08.03
    Map data after object module synopsis:       119  00:00:03.24     00:00:06.43
    Symbol table output:                           0  00:00:00.03     00:00:00.19
Total run values:                                479  00:00:09.99     00:00:19.42

Using a working set limited to 1006 pages and 114 pages of data storage (excluding image)

Total number object records read (both passes):  806
    of which 0 were in libraries and 8 were DEBUG data records containing 1520 bytes

Number of modules extracted explicitly             = 0
    with 0 extracted to resolve undefined symbols

0 library searches were for symbols not in the library searched

A total of 0 global symbol table records was written

LINK/SYSTEM:%X200/EXECU=ENRKA/MAP=ENRKA/CROSS/CONTIG ENRKA0,ENRKA1,ENRKA2,ENRKA3
```

Table of contents

```
    0000
    0000                .TITLE  ENRKA0  IDC DISK FUNCTIONAL DIAGNOSTIC GLOBAL FILE
    0000                .IDENT  /1.1/
    0000                .LIST   MEB
    0000                .NLIST  CND
00000000                .PSECT  HEADER, PAGE, NOWRT
    0000                .DEFAULT DISPLACEMENT, WORD
    0000
    0000
    0000        ;++
    0000        ; COPYRIGHT (C) 1982, 1986
    0000        ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS 01754
    0000        ;
    0000        ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON  A
    0000        ; SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLU-
    0000        ; SION OF THE ABOVE COPYRIGHT NOTICE.  THIS  SOFTWARE,  OR  ANY
    0000        ; OTHER  COPIES  THEREOF, MAY NOT BE PROVIDED OR OTHERWISE MADE
    0000        ; AVAILABLE TO ANY OTHER PERSON EXCEPT FOR USE ON  SUCH  SYSTEM
    0000        ; AND  TO  ONE WHO AGREES TO THESE LICENSE TERMS.  TITLE TO AND
    0000        ; OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN DEC.
    0000        ;
    0000        ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT
    0000        ; NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL
    0000        ; EQUIPMENT CORPORATION.
    0000        ;
    0000        ; DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF
    0000        ; ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
    0000        ;
    0000        ;++
    0000        ; FACILITY: VAX-11 DIAGNOSTIC LIBRARY
    0000        ;
    0000        ; ABSTRACT: THIS SECTION CONTAINS GLOBAL DATA AND GLOBAL SUBROUTINES.
    0000        ;
    0000        ; ENVIRONMENT: VAX-11 DIAGNOSTIC SUPERVISOR
    0000        ;
    0000        ;--
    0000        ;++
    0000        ; AUTHOR: JOHN L. WALSH
    0000        ;
    0000        ; CREATION DATE:  MARCH 1982
    0000        ;
    0000        ; MODIFIED BY:
    0000        ;       V1.1 NO CHANGES TO THIS MODULE
    0000        ;
    0000        ;--
```

```
                  0000          .SBTTL  DECLARATIONS
                  0000     ;++
                  0000     ; INCLUDE FILES
                  0000     ;--
                  0000
                  0000          .LIBRARY /disk$diag:[DIAGLIB.LIBRARY.BLISS]DIAG.MLB/
                  0000          .LIBRARY /SYS$LIBRARY:LIB.MLB/
                  0000          .LIBRARY /disk$diag:[BOWSER.TAPE]ENRKA.MLB/
                  0000     ;++
                  0000     ; MACROS
                  0000     ;--
                  0000
                  0000     ;++
                  0000     ; EQUATED SYMBOLS
                  0000     ;--
                  0000
                  0000     ;
                  0000     ; CREATE SYMBOLS FOR THE PROGRAM/SUPERVISOR INTERFACE.
                  0000     ;
                  0000          $DS_BGNMOD      <SEP_REPAIR>      ; LEVEL 3, SYSTEM ENVIRONMENT
                  0000     ;;;  Macro-32 Diagnostic macro library Version 8.2 "DIAG.MLB (233)"
                  0000
                  0000          ENRKADEF                          ; DEFINE OWN LIBRARY
                  0000     ;;;  ENRKA MACRO LIBRARY VERSION      V1.1
                  0000          $DS_QADEF       ;GLOBAL
                  0000          $DS_DSSDEF      GLOBAL            ; SUPERVISOR SERVICE ENTRY VECTORS
                  0000          $DS_BITDEF                        ; MNEMONIC BIT DEFINITIONS
                  0000          $DS_DSADEF                        ; OFFSETS IN SUPERVISOR/APT MAILBOX
                  0000          $DS_CHCDEF                        ; CHANNEL FUNCTION DEFINITIONS
                  0000          $DS_CHIDEF                        ; INTERRUPT DEFINITIONS
                  0000          $DS_CHMDEF                        ; ADAPTER MAPPING DEFINITIONS
                  0000          $DS_ERRDEF                        ; ERROR CALL PARAMETER OFFSETS
                  0000          $DS_DEFDEL                        ; DELETE MACRO EXPANSIONS
                  0000          $DS_HPODEF      GLOBAL            ; DEVICE PTABLE OFFSETS
                  0000          .SAVE   LOCAL_BLOCK
                  0000          .PSECT  $ABS$,ABS
        00000000  0610          .=0
                  0000          .IIF    NB,HP$Q_DEVICE, HP$Q_DEVICE::
        00000008  0000          .IIF    NB,.BLKQ,       .BLKQ   1
                  0008          .IIF    NB,HP$W_SIZE,   HP$W_SIZE::
        0000000A  0008          .IIF    NB,.BLKW,       .BLKW   1
                  000A          .IIF    NB,HP$B_FLAGS,  HP$B_FLAGS::
        0000000B  000A          .IIF    NB,.BLKB,       .BLKB   1
                  000B          .IIF    NB,HP$B_DRIVE,  HP$B_DRIVE::
        0000000C  000B          .IIF    NB,.BLKB,       .BLKB   1
                  000C          .IIF    NB,HP$T_DEVICE, HP$T_DEVICE::
        00000018  000C          .IIF    NB,.BLKB,       .BLKB   12
                  0018          .IIF    NB,HP$A_DEVICE, HP$A_DEVICE::
        0000001C  0018          .IIF    NB,.BLKL,       .BLKL   1
                  001C          .IIF    NB,HP$A_DVA,    HP$A_DVA::
        00000020  001C          .IIF    NB,.BLKL,       .BLKL   1
                  0020          .IIF    NB,HP$A_LINK,   HP$A_LINK::
        00000024  0020          .IIF    NB,.BLKL,       .BLKL   1
                  0024          .IIF    NB,HP$W_VECTOR, HP$W_VECTOR::
        00000026  0024          .IIF    NB,.BLKW,       .BLKW   1
                  0026          .IIF    NB,HP$T_TYPE,   HP$T_TYPE::
```

```
00000032  0026          .IIF    NB,.BLKB,          .BLKB   12
          0032          .IIF    NB,HP$A_DEPENDENT,         HP$A_DEPENDENT::
  00000000              .RESTORE
          0000     79   $DS_RB730_DEF                      ; IDC PTABLE OFFSETS
          0000          .SAVE   LOCAL_BLOCK
          0000          .PSECT  $ABS$,ABS
00000032  0610          .=HP$A_DEPENDENT
          0032          .IIF    NB,HP$L_RB730_CSR,         HP$L_RB730_CSR:
          0032          .IIF    NB,RB730$L_CSR, RB730$L_CSR:
00000036  0032          .IIF    NB,.BLKL,          .BLKL   1
          0036          .IIF    NB,HP$B_RB730_BR,          HP$B_RB730_BR:
          0036          .IIF    NB,RB730$B_BR,  RB730$B_BR:
00000037  0036          .IIF    NB,.BLKB,          .BLKB   1
          0037          .IIF    NB,HP$K_RB730_LEN,         HP$K_RB730_LEN:
          0037          .IIF    NB,RB730$K_LEN, RB730$K_LEN:
  00000000              .RESTORE
          0000     80   $HMBDEF                            ; HOME BLOCK OFFSETS
          0000          .SAVE   LOCAL_BLOCK
          0000          .PSECT  $ABS$,ABS
00000000  0610          .=0
          0000          .IIF    NB,HMB$W_IBMAPSIZE,        HMB$W_IBMAPSIZE:
00000002  0000          .IIF    NB,.BLKW,          .BLKW
          0002          .IIF    NB,HMB$L_IBMAPLBN,         HMB$L_IBMAPLBN:
00000006  0002          .IIF    NB,.BLKL,          .BLKL
          0006          .IIF    NB,HMB$W_MAXFILES,         HMB$W_MAXFILES:
00000008  0006          .IIF    NB,.BLKW,          .BLKW
          0008          .IIF    NB,HMB$W_CLUSTER,          HMB$W_CLUSTER:
0000000A  0008          .IIF    NB,.BLKW,          .BLKW
          000A          .IIF    NB,HMB$W_DEVTYPE,          HMB$W_DEVTYPE:
0000000C  000A          .IIF    NB,.BLKW,          .BLKW
          000C          .IIF    NB,HMB$W_STRUCLEV,         HMB$W_STRUCLEV:
0000000E  000C          .IIF    NB,.BLKW,          .BLKW
          000E          .IIF    NB,HMB$T_VOLNAME,          HMB$T_VOLNAME:
0000001A  000E          .IIF    NB,.BLKB,          .BLKB   12
0000001E  001A                             .BLKB   4
          001E          .IIF    NB,HMB$W_VOLOWNER,         HMB$W_VOLOWNER:
00000020  001E          .IIF    NB,.BLKW,          .BLKW
          0020          .IIF    NB,HMB$W_PROTECT,          HMB$W_PROTECT:
00000022  0020          .IIF    NB,.BLKW,          .BLKW
          0022          .IIF    NB,HMB$W_VOLCHAR,          HMB$W_VOLCHAR:
00000024  0022          .IIF    NB,.BLKW,          .BLKW
          0024          .IIF    NB,HMB$W_FILEPROT,         HMB$W_FILEPROT:
00000026  0024          .IIF    NB,.BLKW,          .BLKW
0000002C  0026                             .BLKB   6
          002C          .IIF    NB,HMB$B_WINDOW,           HMB$B_WINDOW:
0000002D  002C          .IIF    NB,.BLKB,          .BLKB
          002D          .IIF    NB,HMB$B_EXTEND,           HMB$B_EXTEND:
0000002E  002D          .IIF    NB,.BLKB,          .BLKB
          002E          .IIF    NB,HMB$B_LRU_LIM,          HMB$B_LRU_LIM:
0000002F  002E          .IIF    NB,.BLKB,          .BLKB
0000003A  002F                             .BLKB   11
          003A          .IIF    NB,HMB$W_CHECKSUM1,        HMB$W_CHECKSUM1:
0000003C  003A          .IIF    NB,.BLKW,          .BLKW
          003C          .IIF    NB,HMB$T_CREDATE,          HMB$T_CREDATE:
0000004A  003C          .IIF    NB,.BLKB,          .BLKB   14
000001D8  004A                             .BLKB   398
```

```
             01D8                .IIF    NB,HMB$T_VOLNAME2,      HMB$T_VOLNAME2:
    000001E4 01D8                .IIF    NB,.BLKB,        .BLKB  12
             01E4                .IIF    NB,HMB$T_OWNERNAME,     HMB$T_OWNERNAME:
    000001F0 01E4                .IIF    NB,.BLKB,        .BLKB  12
             01F0                .IIF    NB,HMB$T_FORMAT,        HMB$T_FORMAT:
    000001FC 01F0                .IIF    NB,.BLKB,        .BLKB  12
    000001FE 01FC                                .BLKB   2
             01FE                .IIF    NB,HMB$W_CHECKSUM2,     HMB$W_CHECKSUM2:
    00000200 01FE                .IIF    NB,.BLKW,        .BLKW
         00000000                .RESTORE
             0000     81         $DS_PARDEF                      ; PARAMETER DEFINITIONS
             0000     82         DRV_TBL                         ; DRIVE TABLE OFFSETS
             0000     83         .LIST   MEB
```

```
                                  0000     85          .SBTTL  PROGRAM HEADER DATA BLOCK
                                  0000     86 ;++
                                  0000     87 ; THE PROGRAM HEADER DATA BLOCK CONTAINS THE PARAMETERS WHICH
                                  0000     88 ; ALLOW THE DIAGNOSTIC SUPERVISOR TO CONTROL THE PROGRAM.
                                  0000     89 ; THE DIAGNOSTIC SUPERVISOR LOOKS FOR THE HEADER INFORMATION
                                  0000     90 ; BEGINING AT VIRTUAL ADDRESS 200(HEX).
                                  0000     91 ;
                                  0000     92 ;--
                                  0000     93
                                  0000     94          $DS_HEADER  <ENRKA - IDC DISK FUNCTIONAL DIAGNOSTIC>,-
                                  0000     95                  REV     = 1,              - ; MAJOR REVISION
                                  0000     96                  UPDATE  = 1,              - ; MINOR REVISION
                                  0000     97                  NUNIT   = 4                 ; MAX # OF SELECTABLE UNITS
                                  0000                         .SAVE
                              00000000                         .PSECT  $HEADER, PAGE, NOEXE, NOWRT
                        00000058' 0000     L$L_HEADLENGTH: .LONG    A_HEADEND-.      ; LENGTH OF THE HEADER DATA BLOCK
                        00000007  0004     L$L_ENVIRON:    .LONG    <ENV$_SUPER@ENV$V_SUPER>+$ENV  ; PROGRAM ENVIRONMENT
                        00000058' 0008     L$A_NAME:       .ADDRESS T_NAME           ; PROGRAM NAME TEXT ADDRESS
                        00000001  000C     L$L_REV:        .LONG    1                ; PROGRAM REVISION LEVEL
                        00000001  0010     L$L_UPDATE:     .LONG    1                ; DIAGNOSTIC ENGINEERING PATCH ORDER
                        00000000' 0014     L$A_LASTAD:     .ADDRESS LASTAD           ; FIRST FREE LOCATION AFTER PROGRAM
                        00000000' 0018     L$A_DTP:        .ADDRESS DISPATCH         ; TEST DISPATCH TABLE POINTER
                        0000002D' 001C     L$A_DEVP:       .ADDRESS AL_DEVTYP        ; DEVICE TYPE LIST POINTER
                        00000004  0020     L$L_UNIT:       .LONG    4                ; NUMBER OF UNITS THAT CAN BE TESTED
                        000000C0' 0024     L$A_DREG:       .ADDRESS DEV_REG          ; DEVICE REGISTER CONTENTS TABLE POINTER
                        0000001C  0028                     .LONG    28               ; Ptable length for 4.05 compatability
00000000'00000000'00000000'00000000' 002C                  .LONG    0[4]             ; UNUSED OFFSETS
                        00000000' 003C     L$A_ICP:        .ADDRESS INITIALIZE       ; INITIALIZATION CODE POINTER
                        00000000' 0040     L$A_CCP:        .ADDRESS CLEAN_UP         ; CLEAN-UP CODE POINTER
                        00000000' 0044     L$A_REPP:       .ADDRESS SUMMARY          ; SUMMARY REPORT CODE POINTER
                        00000000' 0048     L$A_STATAB:     .ADDRESS 0                ; STATISTIC TABLE POINTER
                        00000000  004C     L$L_ERRTYP:     .LONG    0                ; # OF TYPES OF $ERRSOFT & $ERRPREP
                        00000013' 0050     L$A_SECNAM:     .ADDRESS SECTION          ; LIST OF SECTION NAME ADDRESSES
                        00000000' 0054     L$A_TSTCNT:     .ADDRESS L_TSTCNT         ; POINTER TO NUMBER OF TESTS
                                  0058     A_HEADEND:
43 44 49 20 2D 20 41 4B 52 4E 45 00' 0058  T_NAME:         .ASCIC  \ENRKA - IDC DISK FUNCTIONAL DIAGNOSTIC\
49 54 43 4E 55 46 20 4B 53 49 44 20  0064
53 4F 4E 47 41 49 44 20 4C 41 4E 4F  0070
                            43 49 54  007C
                                  26  0058
                              00000000                     .PSECT  _LAST, PAGE
                                  0000     LASTAD:
                              00000000                     .PSECT  $TSTCNT, NOEXE, NOWRT, OVR, LONG
                                  0000     L_TSTCNT:
                              00000000                     .RESTORE
```

```
                              0000     99          .SBTTL  DISPATCH TABLE
                              0000    100 ;++
                              0000    101 ; FUNCTIONAL DESCRIPTION::
                              0000    102 ;
                              0000    103 ;         THE DISPATCH TABLE IS A COLLECTION OF ADDRESSES GENERATED AT THE
                              0000    104 ;         BEGINNING OF EACH TEST AND GROUPED TOGETHER INTO A CONTIGUOUS
                              0000    105 ;         LIST BY THE LINKER.  THIS IS DONE BY DEFINING A PSECT CALLED
                              0000    106 ;         DISPATCH.
                              0000    107 ;
                              0000    108 ;--
                              0000    109
                              0000    110 $DS_DISPATCH
                              0000                        .SAVE
                          00000000                        .PSECT  DISPATCH, LONG, NOWRT, NOEXE
                              0000            DISPATCH:
                          00000000                        .PSECT  DISPATCH_X, LONG, NOWRT, NOEXE
00000000'00000000'00000000'00000000' 0000                 .LONG   0[6]
                  00000000'00000000' 0010
                          00000000                        .RESTORE
```

```
                0000         .SBTTL  REGISTER DEFINITIONS
                0000
                0000    ;++
                0000    ;        DEFINE ALL OF THE IDC REGISTER OFFSETS AND
                0000    ;        REGISTER BIT MNEMONICS.
                0000    ;--
                0000         $DEFINI IDC_REG, LOCAL, 0        ; SET-UP TO DEFINE REGISTER BITS
                0000         .SAVE   LOCAL_BLOCK
                0000         .PSECT  $ABS$,ABS
       00000000 0610         .=0
                0000
                0000    ;++
                0000    ;        DEFINE VALUES FOR CONTROL AND STATUS REGISTER
                0000    ;--
                0000         $EQU    IDCCSR, 0                ; IDCCSR REGISTER OFFSET
                0000
                0000         $VIELD  CSR, 0, <-               ; SET UP TO DEFINE CSR BITS
                0000                 <DRDY,,M>,-              ; DRIVE READY BIT (0)
                0000                 <  F0,,M>,-              ; FUNCTION BIT 0, BIT (1)
                0000                 <  F1,,M>,-              ; FUNCTION BIT 1, BIT (2)
                0000                 <  F2,,M>,-              ; FUNCTION BIT 2, BIT (3)
                0000                 <    ,, >,-              ; THIS BIT UNUSED, BIT (4)
                0000                 <    ,, >,-              ; THIS BIT UNUSED, BIT (5)
                0000                 <  IE,,M>,-              ; INTERRUPT ENABLE, BIT (6)
                0000                 <CRDY,,M>,-              ; CONTROLLER READY, BIT (7)
                0000                 < DS0,,M>,-              ; DRIVE SELECT BIT 0, BIT (8)
                0000                 < DS1,,M>,-              ; DRIVE SELECT BIT 1, BIT (9)
                0000                 < OPI,,M>,-              ; OPERATION INCOMPLETE, BIT (10)
                0000                 < DCK,,M>,-              ; DATA CHECK, BIT (11)
                0000                 < DLT,,M>,-              ; DATA LATE, BIT (12)
                0000                 < NXM,,M>,-              ; NON-EXISTANT MEMORY, BIT (13)
                0000                 <  DE,,M>,-              ; DRIVE ERROR, BIT (14)
                0000                 < ERR,,M>,-              ; COMPOSITE ERROR, BIT (15)
                0000                 <ATN0,,M>,-              ; ATTENTION BIT 0, BIT (16)
                0000                 <ATN1,,M>,-              ; ATTENTION BIT 1, BIT (17)
                0000                 <ATN2,,M>,-              ; ATTENTION BIT 2, BIT (18)
                0000                 <ATN3,,M>,-              ; ATTENTION BIT 3, BIT (19)
                0000                 <ECS0,,M>,-              ; ECC STATUS BIT 0, BIT (20)
                0000                 <ECS1,,M>,-              ; ECC STATUS BIT 1, BIT (21)
                0000                 <SSEI,,M>,-              ; SKIP SECTOR ERROR INHIBIT, BIT (22)
                0000                 < SSE,,M>,-              ; SKIP SECTOR ERROR, BIT (23)
                0000                 <  IR,,M>,-              ; INTERRUPT REQUEST, BIT (24)
                0000                 < MTN,,M>,-              ; MAINT MODE, BIT (25)
                0000                 < R80,,M>,-              ; R80 IDENTIFIER BIT, BIT (26)
                0000                 <ASSI,,M>,-              ; AUTO SKIP SECTOR INHIBIT, BIT (27)
                0000                 < TOI,,M>,-              ; TIME OUT INHIBIT, BIT (28)
                0000                 < FMT,,M>,-              ; WRITE R80 FORMAT, BIT (29)
                0000                         >
                0000    ;++
                0000    ;        DEFINE SOME OF THE BITS IN THE CONTROL AND
                0000    ;        STATUS REGISTER THAT HAVE MORE THEN ONE MNEMONIC.
                0000    ;--
                0000
                0000         $VIELD  CSR, 11, <-              ; DEFINE DUPLICATE BITS IN CSR
                0000                 <HCRC,,M>,-              ; HEADER CRC ERROR, BIT (11)
                0000                 < HNF,,M>,-              ; HEADER NOT FOUND, BIT (12)
```

```
0000                         >
0000         ;++
0000         ;       DEFINE REGISTER OFFSET FOR BUS ADDRESS REGISTER
0000         ;       AND ASSOCIATED BIT MNEMONICS.
0000         ;--
0000
0000                 $EQU    IDCBAR, 4               ; DEFEINE IDCBAR REGISTER OFFSET
0000
0000                 $VIELD  BAR, 0, <-              ; SETUP TO DEFINE BAR BITS
0000                         <BITS,16,M>,-           ; 16 BUS ADDRESS BITS
0000                         >
0000
0000         ;++
0000         ;       DEFINE REGISTERS OFFSET FOR BYTE COUNT REGISTER
0000         ;       AND ASSCOCIATED BIT MNEMONICS.
0000         ;--
0000
0000                 $EQU    IDCCNT, 8               ; DEFINE IDCCNT REGISTER OFFSET
0000
0000                 $VIELD  CNT, 0, <-              ; SET UP TO DEFINE CNT BITS
0000                         <BITS,16,M>,-           ; 16 BYTE COUNT BITS
0000                         >
0000
0000         ;++
0000         ;       DEFINE REGISTER OFFSET FOR DISK ADDRESS REGISTER
0000         ;       AND ASSOCIATED BIT MNEMONICS.
0000         ;--
0000
0000                 $EQU    IDCDAR, 12              ; DEFINE IDCDAR REGISTER OFFSET
0000
0000                 $VIELD  DAR, 0, <-              ; SET UP TO DEFINE DAR BITS
0000                         <SEC,8,M>,-             ; DESIRED SECTOR, BITS (0 - 7)
0000                         <TRK,8,M>,-             ; DESIRED TRACK, BITS (8 - 15)
0000                         <CYL,16,M>,-            ; DESIRED CYLINDER, BIT (16 - 31)
0000                         >
0000
0000         ;++
0000         ;       DEFINE REGISTER OFFSET FOR MULTI PURPOSE REGISTER
0000         ;       AND BIT MNEMONICS USED WHEN ISSUING A GET STATUS
0000         ;       COMMAND TO AN RL02.
0000         ;--
0000
0000                 $EQU    IDCMPR, 16              ; DEFINE IDCMAP REGISTER OFFSET
0000
0000                 $VIELD  MPR_GS, 0, <-           ; SET UP TO DEFINE MPR GET STATUS BITS
0000                         <MARK,,M>,-             ; MARKER BIT, BIT (0)
0000                         <  GS,,M>,-             ; GET STATUS FLAG, BIT (1)
0000                         <    ,, >,-             ; NOT USED, BIT (2)
0000                         < RST,,M>,-             ; DRIVE RESET FLAG, BIT (3)
0000                         >
0000
0000         ;++
0000         ;       DEFINE RL02 STATUS WORD FOUND IN THE MULTI
0000         ;       PURPOSE REGISTER.
0000         ;--
0000
```

```
    0000          $VIELD  MPR_02, 0, <-             ; SETUP TO DEFINE MPR RL02 STATUS BITS
    0000                  < STA,,M>,-               ; STATUS BIT A, BIT (0)
    0000                  < STB,,M>,-               ; STATUS BIT B, BIT (1)
    0000                  < STC,,M>,-               ; STATUS BIT C, BIT (2)
    0000                  <  BH,,M>,-               ; BRUSHES HOME, BIT (3)
    0000                  <  HO,,M>,-               ; HEADS OUT, BIT (4)
    0000                  <  CO,,M>,-               ; COVER OPEN, BIT (5)
    0000                  <  HS,,M>,-               ; HEAD SELECT, BIT (6)
    0000                  <  DT,,M>,-               ; RL02 DRIVE TYPE BIT (7)
    0000                  < DSE,,M>,-               ; DRIVE SELECT ERROR, BIT (8)
    0000                  <  VC,,M>,-               ; VOLUME CHECK, BIT (9)
    0000                  < WGE,,M>,-               ; WRITE GATE ERROR, BIT (10)
    0000                  < SPE,,M>,-               ; SPEED LOSS ERROR, BIT (11)
    0000                  <SKTO,,M>,-               ; SEEK TIMEOUT ERROR, BIT (12)
    0000                  <  WL,,M>,-               ; WRITE LOCKED, BIT (13)
    0000                  < HCE,,M>,-               ; HEAD CURRENT ERROR, BIT (14)
    0000                  < WDE,,M>,-               ; WRITE DATA ERROR, BIT (15)
    0000                          >
    0000
    0000      ;++
    0000      ;   DEFINE THE R80 STATUS WORD FOUND IN THE MULTI
    0000      ;   PURPOSE REGISTER.
    0000      ;--
    0000
    0000          $VIELD  MPR_80, 0, <-             ; SETUPO TO DEFINE R80 STATUS BITS
    0000                  <SECT,5,M>,-              ; SECTOR LOOKAHEAD BITS (0 - 4)
    0000                  <    ,3, >,-              ; UNUSED BITS (5 - 7)
    0000                  < FLT,,M>,-               ; DRIVE FAULT, BIT (8)
    0000                  <PLGV,,M>,-               ; DRIVE PLUG VALID, BIT (9)
    0000                  < SKE,,M>,-               ; SEEK ERROR, BIT (10)
    0000                  <ONCY,,M>,-               ; DRIVE ON CYLINDER, BIT (11)
    0000                  <DRDY,,M>,-               ; DRIVE READY, BIT (12)
    0000                  < WTP,,M>,-               ; WRITE PROTECTED, BIT (13)
    0000                          >
    0000
    0000      ;++
    0000      ;   DEFINE REGISTER BITS AND REGISTER OFFSETS FOR
    0000      ;   ECC REGISTERS.
    0000      ;--
    0000
    0000          $EQU    IDCPOS, 20                ; DEFINE OFFSET FOR ECC POSITION REG
    0000
    0000          $VIELD  POS, 0, <-                ; DEFINE REGISTER BITS
    0000                  <BITS,13,M>,-             ;
    0000                          >
    0000
    0000          $EQU    IDCPAT, 24                ; DEFINE OFFSET FOR ECC PATTERN REG
    0000
    0000          $VIELD  PAT, 0, <-                ; DEFINE REGISTER BITS
    0000                  <BITS,11,M>,-             ;
    0000                          >
    0000
    0000          $DEFEND IDC_REG,        LOCAL     ; END OF REGISTER DEFINITION
00000000          .RESTORE
```

```
                 0000    275           .SBTTL  HEADER BIT DEFINITIONS
                 0000    276
                 0000    277 ;++
                 0000    278 ;         DEFINE HEADER WORD LAYOUT FOR THE RL02.
                 0000    279 ;-
                 0000    280
                 0000    281           $DEFINI HDR_LAYOUT, LOCAL, 0
                 0000                  .SAVE   LOCAL_BLOCK
                 0000                  .PSECT  $ABS$,ABS
        00000000 0610                  .=0
                 0000    282           $VIELD  HDR_02, 0, <-                 ; SETUP TO DEFINE RL02 HEADER LAYOUT
                 0000    283                   <SEC,6,M>,-                   ; CURRENT SECTOR, BITS (0 - 5)
                 0000    284                   <TRK,1,M>,-                   ; CURRENT TRACK, BIT (6)
                 0000    285                   <CYL,9,M>,-                   ; CURRENT CYLINDER, BITS (7 - 15)
                 0000    286                           >
                 0000    287
                 0000    288 ;++
                 0000    289 ;         DEFINE HEADER LAYOUT FOR THE R80
                 0000    290 ;--
                 0000    291
                 0000    292           $VIELD  HDR_80, 0, <-                ; SETUP TO DEFINE R80 HEADER LAYOUT
                 0000    293                   <CYL,10,M>,-                 ; CURRENT CYLINDER BITS (0 - 9)
                 0000    294                   <    ,2, >,-                 ; UNUSED BITS BITS (10 - 11)
                 0000    295                   < FMT,1,M>,-                 ; FORMAT 16 BIT (12)
                 0000    296                   < SSF,1,M>,-                 ; SKIP SECTOR FLAG BIT (13)
                 0000    297                   <  UF,1,M>,-                 ; USER BAD SECTOR FLAG BIT (14)
                 0000    298                   <  MF,1,M>,-                 ; MANUFACTURING BAD SECTOR FLAG BIT (15)
                 0000    299                   <SEC,5,M>,-                  ; CURRENT SECTOR BIT (16 - 20)
                 0000    300                   <    ,3,M>,-                 ; UNUSED BITS (21 -23)
                 0000    301                   <TRK,4,M>,-                  ; CURRENT TRACK BITS (24 - 27)
                 0000    302                           >
                 0000    303           $DEFEND HDR_LAYOUT,     LOCAL   ; END OF HEADER LAYOUT DEFINITIONS
             00000000                  .RESTORE
                 0000    304
```

```
                0000          .SBTTL  MISC PROGRAM SYMBOL DEFINITIONS
                0000
                0000    ;++
                0000    ;     THIS SECTION DEFINES THOSE SYMBOLS WHICH ARE USED
                0000    ;     THROUGHOUT THE DIAGNOSTIC TO CONTROL PROGRAM
                0000    ;     EXECUTION AND CURRENT STATUS.
                0000    ;--
                0000
                0000          $DEFINI ENRKA_SYM, LOCAL, 0      ; SETUP TO DEFINE ENRKA SYMBOLS
                0000          .SAVE   LOCAL_BLOCK
                0000          .PSECT  $ABS$,ABS
       00000000 0610          .=0
                0000
                0000          $VIELD  PRT, 0, <-               ; SET UP TO DEFINE REGISTER PRINT
                0000                                  -        ; FLAGS FOUND IN LOCATION 'RGLIST'
                0000                  <MPR_1,,M>,-             ;
                0000                  <MPR_2,,M>,-             ;
                0000                  <MPR_3,,M>,-             ;
                0000                  <EXT_1,,M>,-             ;
                0000                  <EXT_2,,M>,-             ;
                0000                  <EXT_3,,M>,-             ;
                0000                  < NONE,,M>,-             ; DONT DUMP REGISTERS
                0000                          >
                0000          $VIELD  PRT, 0, <-               ; DEFINE PRINT ALL FLAG
                0000                  <  ALL,3,M>,-            ; PRINT ALL MPR REGISTERS
                0000                          >
                0000
                0000          $VIELD  NOCHK, 0, <-             ; SET UP TO DEFINE CONTROL FLAGS
                0000                                  -        ; FOR STATUS CHECKING
                0000                  <CRDY,,M>,-              ; NO WAIT FOR CONTROLLER READY
                0000                  <CERR,,M>,-              ; NO CONTROLLER ERROR CHECKING
                0000                  <DRDY,,M>,-              ; NO DRIVE READY CHECKING
                0000                  <DERR,,M>,-              ; NO DRIVE ERROR CHECKING
                0000                  <DSTA,,M>,-              ; NO DRIVE STATUS CHECKING
                0000                  < CRC,,M>,-              ; NO CRC CHECKING
                0000                  < FTL,,M>,-              ; DONT CHECK FOR FATAL ERRORS
                0000                  <ATTN,,M>,-              ; DONT CHECK FOR DRIVE ATTENTION
                0000                          >
                0000
                0000          $VIELD  RECAL, 0, <-             ; DEFINE RECAL FLAG
                0000                  <INPROG,,M>,-            ; IN PROGRESS
                0000                          >
                0000
                0000          $VIELD  HDR, 5, <-               ; DEFINE HEADER ERROR OCCURED BIT
                0000                  < ERR,,M>,-              ; HDR ERROR OCCURED
                0000                          >
                0000
                0000          $VIELD  DEL, 10, <-              ; DEFINE EVENT FLAG USAGE
                0000                  <READH,,>,-              ; EVENT FLAG 10 = DELETE READ HEADERS
                0000                  <WRIT,,>,-               ; EVENT FLAG 11 = DELETE WRITE DATA
                0000                  <WRCK,,>,-               ; EVENT FLAG 12 = DELETE WRITE CHECK
                0000                  <READ,,>,-               ; EVENT FLAG 13 = DELETE READ DATA
                0000                  <    ,6,>,-              ; UNUSED FLAGS
                0000                  <PROT,,>,-               ; EVENT FLAG 20 = DELETE CUSTOMER DATA PRO -
cont> TECTION
                0000                          >
                0000
```

```
0000    ;++
0000    ;               DEFINE IDC COMMAND CODES
0000    ;--
0000
0000                    $EQU    NO_OP,          0       ; NO OPERATION
0000                    $EQU    WRTCHK,         2       ; WRITE CHECK DATA
0000                    $EQU    GETSTAT,        4       ; GET STATUS
0000                    $EQU    SEEK,           6       ; SEEK
0000                    $EQU    READH,          8       ; READ HEADER
0000                    $EQU    WRITE,          10      ; WRITE DATA
0000                    $EQU    READ,           12      ; READ DATA
0000                    $EQU    RD_NHC,         14      ; READ DATA WITH NO HEADER COMPARE
0000                    $EQU    RECAL,          6       ; R80 RECAL = SEEK TO CYLN 0
0000
0000    ;++
0000    ;               DEFINE DRIVE TYPES FOR EACH DISK.
0000    ;--
0000
0000                    $EQU    RL02$T,         1       ; RL02 DISK
0000                    $EQU    R80$T,          2       ; R80 DISK
0000                    $EQU    NUMENT,         33      ; NUMBER OF DRIVE TABLE ENTRIES
0000
0000    ;++
0000    ;               DEFINE PARAMETERS USED IN THE $DS_BGNDATA TABLES
0000    ;--
0000
0000                    $EQU    LUN0,   0               ; LOGICAL UNIT 0
0000                    $EQU    LUN1,   1               ; LOGICAL UNIT 1
0000                    $EQU    LUN2,   2               ; LOGICAL UNIT 2
0000                    $EQU    LUN3,   3               ; LOGICAL UNIT 3
0000                    $EQU    HEAD_0, 0               ; USE HEAD 0
0000                    $EQU    HEAD_1, 1               ; USE HEAD 1
0000                    $EQU    ONE,    0               ; ONE CYLINDER SEEK TIMING
0000                    $EQU    MAX,    1               ; MAXIMUM SEEK TIMING
0000                    $EQU    AVG,    2               ; AVERAGE SEEK TIMING
0000                    $EQU    OUTER,  0               ; OUTER SECTION OF DISK
0000                    $EQU    MIDDLE, 1               ; MIDDLE SECTION OF DISK
0000                    $EQU    INNER,  2               ; INNER SECTION OF DISK
0000                    $EQU    BLANK,  3               ; SECTION OF DISK  = DONT CARE
0000                    $EQU    OFST_UNIT,      4       ; OFFSET FOR LOGICAL UNIT
0000                    $EQU    OFST_DRIV,      8       ; OFFSET FOR DRIVE TYPE
0000                    $EQU    OFST_SKHD,      12      ; OFFSET FOR SEEK HEAD
0000                    $EQU    OFST_DTHD,      12      ; OFFSET FOR DATA TRANSFER HEAD
0000                    $EQU    BUFLEN, 100             ; BUFFER LENGTH FOR CVTREG MACRO
0000
0000    ;++
0000    ;               DEFINE OFFSETS FOR HEADER TEMPLATE
0000    ;--
0000
0000                    $EQU    OFST_CYLPOS,    0       ; OFFSET FOR CYLINDER POSITION
0000                    $EQU    OFST_CYLSIZ,    4       ; OFFSET FOR CYLINDER SIZE
0000                    $EQU    OFST_TRKPOS,    5       ; OFFSET FOR TRACK POSITION
0000                    $EQU    OFST_TRKSIZ,    9       ; OFFSET FOR TRACK SIZE
0000                    $EQU    OFST_SECPOS,    10      ; OFFSET FOR SECTOR POSITION
0000                    $EQU    OFST_SECSIZ,    14      ; OFFSET FOR SECTOR SIZE
0000
```

```
             0000   415          $DEFEND ENRKA_SYM        LOCAL   ; END OF SYMBOL DEFINITIONS
         00000000                .RESTORE
             0000   416
```

```
                        0000    418             .SBTTL  GLOBAL DATA
                    00000000    419             .PSECT  GLOBAL_DATA,LONG
                        0000    420 ;++
                        0000    421 ; GLOBAL DATA
                        0000    422 ; ALL DYNAMICALLY MODIFIED DATA IS PLACED IN THIS SECTION.
                        0000    423 ; THIS IS THE ONLY SECTION WHICH IS NORMALLY WRITE ENABLED.
                        0000    424 ;--
                        0000    425
              00000000  0000    426 DRIVE_TBL::      .LONG   0                    ; ADDRESS OF DRIVE INFORMATION
              00000000  0004    427 S_DRIVE_TBL::    .LONG   0                    ; ADDRESS OF SECONDARY DRIVE INFO
              00000000  0008    428 DEV::            .LONG   0                    ; DEVICE COUNTER
                        000C    429 IDC_STATUS::
              00000000  000C    430 IDC_STAT$L_1::   .LONG   0                    ; STORAGE FOR INTERRUPT STATUS WD 1
                  0000  0010    431 IDC_STAT$W_2::   .WORD   0                    ; STATUS WORD 2
                  0000  0012    432 IDC_STAT$W_VR::  .WORD   0                    ; VECTOR REC'VD
              00000000  0014    433 BASE_ADDR::      .LONG   0                    ; IDC UNIBUS BASE ADDRESS
              00FFFB86  0018    434 SECD_ADDR::      .LONG   ^XFFFB86             ; IDC SECONDARY UNIBUS ADDRESS
     00000000 00000000  001C    435 SYS_TIME::       .QUAD   0                    ; 64 BIT SYSTEM TIME STORAGE
              00000000  0024    436 EVENT_FLAGS::    .LONG   0                    ; STORAGE FOR STATIC EVENT FLAGS
              00000000  0028    437 CURR_FLAGS::     .LONG   0                    ; STORAGE FOR DYNAMIC EVENT FLAGS
              00000000  002C    438 L$UNIT::         .LONG   0                    ; LOGICAL UNIT #
              00000000  0030    439 S_L$UNIT::       .LONG   0                    ; SECONDARY LOGICAL UNIT #
              00000000  0034    440 P$UNIT::         .LONG   0                    ; PHYSICAL UNIT#
              00000000  0038    441 S_P$UNIT::       .LONG   0                    ; SECONDARY PHYSICAL UNIT#
              00000000  003C    442 P$ATTN::         .LONG   0                    ; PHYSICAL ATTENTION BIT
              00000000  0040    443 S_P$ATTN::       .LONG   0                    ; SECONDARY PHYSICAL ATTENTION BIT
     00000000 00000000  0044    444 DEV_NAME::       .QUAD   0                    ; POINTER TO DEVICE MNMENONIC
     00000000 00000000  004C    445 ACT_NAME::       .QUAD   0                    ; SAVE VOLUME NAME ON PACK
              00000000  0054    446 ABORT_FLAG::     .LONG   0                    ; FLAG FOR ABORTING TEST
              00000000  0058    447 HARD_TABLE::     .LONG   0                    ; HARDWARE P-TABLE ADDRESS
     00000000 00000000  005C    448 PHY_WRT_BUF::    .QUAD   0                    ; QUAD WORD FOR GET BUF PHYSICAL
     00000000 00000000  0064    449 VIR_WRT_BUF::    .QUAD   0                    ; QUAD WORD FOR GET BUF VIRTUAL
     00000000 00000000  006C    450 PHY_RD_BUF::     .QUAD   0                    ; QUAD WORD FOR GET BUF PHYSICAL
     00000000 00000000  0074    451 VIR_RD_BUF::     .QUAD   0                    ; QUAD WORD FOR GET BUF VIRTUAL
              00000000  007C    452 HMB_ERR_FLAG::   .LONG   0                    ; HOME BLOCK READ ERROR FLAG
              00000000  0080    453 TIME_OUT::       .LONG   0                    ; HOLDS EVENT FLAGS FOR SETIMR CALLS
     00000000 00000000  0084    454 DELTA_TIME::     .QUAD   0                    ; HOLDS TIME CONSTANT FOR SETIMR CALLS
     FFFFFFFF FF85EE00  008C    455 TIME_800MS::     .LONG -10*1000*800,-1        ; 800 MILLI SEC QUAD WORD
     FFFFFFFF FFD23940  0094    456 TIME_300MS::     .LONG -10*1000*300,-1        ; 300 MILLI SEC QUAD WORD
              00000000  009C    457 TIME_LEFT::      .LONG   0                    ; LEFT OVER TIME FROM $DS_WAIT CALLS
              00000000  00A0    458 HOLD_CS::        .LONG   0                    ; STORAGE FOR CS WRITE DATA
              00000000  00A4    459 BUFF_INUSE::     .LONG   0                    ; ADDRESS OF BUFF FOR DATA COMPARE RTN
              00000000  00A8    460 RGLIST::         .LONG   0                    ; LIST OF REGISTERS TO PRINT
              00000000  00AC    461 DISK_SECTION::   .LONG   0                    ; ASCII OUTER, INNER, ETC.
              00000000  00B0    462 SEEK_TYPE::      .LONG   0                    ; ASCII ONE, MAX, AVG, ETC.
              00000000  00B4    463 DERR_FLAG::      .LONG   0                    ; DRIVE ERROR FLAG LONG WORD
              00000000  00B8    464 ERR_MSG::        .LONG   0                    ; HOLDS ERROR MESSAGE FOR SUPERV CALLS
              00000000  00BC    465 DAR_MNEM::       .LONG   0                    ; HOLDS ADDRESS OF DAR BIT MNEMONICS
              00000000  00C0    466 DEV_REG::        .LONG   0                    ; POINTER FOR DEVICE REGISTER TABLE
              00000000  00C4    467 SECTOR::         .LONG   0                    ; SECTOR ADDRESS
              00000000  00C8    468 SURFACE::        .LONG   0                    ; HEAD ADDRESS
              00000000  00CC    469 POS_TYPE::       .LONG   0                    ; DEFINES R80 POSITION AS SEEK OR RECAL
              00000000  00D0    470 CYLN::           .LONG   0                    ; TRANSFER CYLINDER
              00000000  00D4    471 CURR_CYLN::      .LONG   0                    ; CURRENT CYLINDER
              00000000  00D8    472 CURR_HD::        .LONG   0                    ; SAVES CURRENT HEAD
              00000000  00DC    473 CURR_CYL_HD::    .LONG   0                    ; SAVES CYL AND HEAD TOGETHER
```

```
                  00000000  00E0  PREV_CYLN::      .LONG   0               ; SAVE PREVIOUS CYLINDER
                  00000000  00E4  DEST_CYLN::      .LONG   0               ; HOLDS THE DESTINATION CYLINDER
                  00000000  00E8  OLD_CYLN::       .LONG   0               ; SAVES A CYLINDER FOR ERROR LOOPING
                  00000000  00EC  NEW_CYLN::       .LONG   0               ; SAVES NEXT CYLN TO SEEK TO
                  00000000  00F0  CURR_SECTOR::    .LONG   0               ; HOLDS CURRENT SECTOR
                  00000000  00F4  SEED::           .LONG   0               ; SEED FOR RANDOM NUMBER GENERATOR
                  00000000  00F8  HOLD_DA::        .LONG   0               ; STORAGE FOR DISK ADDRESS REG
                  00000000  00FC  CURR_CMD::       .LONG   0               ; SAVES CURRENT COMMAND IN PROGRESS
                  00000000  0100  DIFF_WORD::      .LONG   0               ; SAVES CURRENT DIFFERNECE WORD
                      0000  0104  GDDAT::          .WORD   0               ; SAVES EXPECTED (GOOD) DATA
                      0000  0106  BDDAT::          .WORD   0               ; SAVES DATA AS READ
                  00000000  0108  INH_PNT::        .LONG   0               ; INHIBIT START HEADER WARNING
                  00000000  010C  PNT_FLAG::       .LONG   0               ; INHIBIT EXIT TEST PRINT OUT
                  00000000  0110  ITERAT::         .LONG   0               ; USED FOR TEST ITERATIONS
                  00000000  0114  WORK::           .LONG   0               ; TEMP WORK LOCATION
                  00000000  0118  TEMP::           .LONG   0               ; TEMP WORK LOCATION
                  00000000  011C  CONFIG::         .LONG   0               ; PROGRAM CONFIGURATION FLAGS
                  00000000  0120  T.CS::           .LONG   0               ; CONTROL AND STATUS REGISTER
                  00000000  0124  T.BA::           .LONG   0               ; BUS ADDRESS REGISTER
                  00000000  0128  T.BC::           .LONG   0               ; BYTE COUNT REGISTER
                  00000000  012C  T.DA::           .LONG   0               ; DESIRED TRACK SECTOR REGISTER
                  00000000  0130  T.MP::           .LONG   0               ; SAVES MULTI PURPOSE REG
                  00000000  0134  T.MP1::          .LONG   0               ; SAVES HDR WORD #1
                  00000000  0138  T.MP2::          .LONG   0               ; SAVES HDR WORD #2
                  00000000  013C  T.MP3::          .LONG   0               ; SAVES HDR WORD #3
         00000000 00000000  0140  T.HDR::          .QUAD   0               ; SAVE HEADER
                  00000000  0148  T.PAT::          .LONG   0               ; ECC PATTERN REGISTER
                  00000000  014C  T.POS::          .LONG   0               ; ECC POSITION REGISTER
                  00000000  0150  T.DAT::          .LONG   0               ; SAVES DATA READ FROM REGS.
                  00000000  0154  T.BR::           .LONG   0               ; SAVES RECIEVED BR LEVEL FROM INTERRUPT
                  00000000  0158  T.VEC::          .LONG   0               ; SAVES RECIEVED VECTOR
                  00000000  015C  E.CS::           .LONG   0               ; CONTROL AND STATUS REGISTER
                  00000000  0160  E.BA::           .LONG   0               ; BUS ADDRESS REGISTER
                  00000000  0164  E.BC::           .LONG   0               ; BYTE COUNT REGISTER
                  00000000  0168  E.DA::           .LONG   0               ; DESIRED TRACK SECTOR REGISTER
                  00000000  016C  E.MP::           .LONG   0               ; EXPECTED MULTI PURPOSE REG
                  00000000  0170  E.MP1::          .LONG   0               ; EXPECTED HDR WD #1
                  00000000  0174  E.MP2::          .LONG   0               ; EXPECTED HDR WD #2
                  00000000  0178  E.MP3::          .LONG   0               ; EXPECTED HDR WD #3
         00000000 00000000  017C  E.HDR::          .QUAD   0               ; EXPECTED HEADER
                  00000000  0184  E.DAT::          .LONG   0               ; SAVES EXPECTED DATA
                  00000000  0188  E.BR::           .LONG   0               ; EXPECTED BR LEVEL
                  00000000  018C  E.VEC::          .LONG   0               ; EXPECTED DEVICE VECTOR
                  000001A4  0190  GOOD_DATA::      .BLKL   5
                  000001B8  01A4  BAD_DATA::       .BLKL   5
                  00000000  01B8  ERR_CNT::        .LONG   0
                  00000000  01BC  WORD_COUNT::     .LONG   0
                            01C0
                            01C0  ;++
                            01C0  ; BUFFER FOR $DS_CVTREG_L MACRO
                            01C0  ;--
                            01C0
                  00000225  01C0  CHAR_BUFFER::    .BLKB   BUFLEN+1
                            0225
                            0225  ;++
                            0225  ; TABLE FOR CRC INSTRUCTION BASED ON CRC 16 120001(O)
```

```
                                 0225  ; THIS 16 LONGWORD TABLE IS A REQUIREMENT FOR THE CRC INSTRUCTION
                                 0225  ; THIS TABLE WAS DERVIVED FROM THE THE COMMON RUN TIME LIBRARY
                                 0225  ; LIB$CRC_TABLE.  THE ROUTINE WAS CALLED FROM THE INIT CODE BY
                                 0225  ; THE FOLLOWING
                                 0225  ;                       PUSHAL  CRC_TABLE
                                 0225  ;                       PUSHAL  POLY                    ; WHERE POLY WAS = ^O120001
                                 0225  ;                       CALLS   #2,LIB$CRC_TABLE
                                 0225  ;
                                 0225  ; THIS CODE HAS SUBSEQUENTLY BEEN REMOVED FROM THE INIT SECTION.
                                 0225  ; NOTE THAT A WELL FORMED TABLE HAS ENTRY [0] = 0
                                 0225  ; AND ENTRY [8] EQUAL TO THE POLYNOMIAL
                                 0225  ;--
                                 0225
                                 0225  .ALIGN  LONG
                      00000000   0228  CRC_TABLE::            .LONG   0                       ; ENTRY [0]
                      0000CC01   022C                         .LONG   ^XCC01
                      0000D801   0230                         .LONG   ^XD801
                      00001400   0234                         .LONG   ^X1400
                      0000F001   0238                         .LONG   ^XF001
                      00003C00   023C                         .LONG   ^X3C00
                      00002800   0240                         .LONG   ^X2800
                      0000E401   0244                         .LONG   ^XE401
                      0000A001   0248                         .LONG   ^O120001                ; ENTRY [8]
                      00006C00   024C                         .LONG   ^X6C00
                      00007800   0250                         .LONG   ^X7800
                      0000B401   0254                         .LONG   ^XB401
                      00005000   0258                         .LONG   ^X5000
                      00009C01   025C                         .LONG   ^X9C01
                      00008801   0260                         .LONG   ^X8801
                      00004400   0264                         .LONG   ^X4400
                                 0268
                                 0268  ;++
                                 0268  ; TABLE OF CYLINDERS TO BE USED IN THE  DIFFERENCE WORD TRANSMISSION
                                 0268  ; TEST. IT IS MADE UP OF FLOATING AND GROWING ONES AND ZEROS
                                 0268  ;--
                                 0268
                                 0268  CYLN_LIST::
          0008 0004 0002 0001    0268                 .WORD   ^X0001,^X0002,^X0004,^X0008
          0080 0040 0020 0010    0270                 .WORD   ^X0010,^X0020,^X0040,^X0080
     01FB 01FD 01FE 0200 0100    0278                 .WORD   ^X0100,^X200,^X01FE,^X01FD,^X01FB
          01BF 01DF 01EF 01F7    0282                 .WORD   ^X01F7,^X01EF,^X01DF,^X01BF
          0003 0001 00FF 017F    028A                 .WORD   ^X017F,^X00FF,^X0001,^X0003
          003F 001F 000F 0007    0292                 .WORD   ^X0007,^X000F,^X001F,^X003F
          01FE 01FF 00FF 007F    029A                 .WORD   ^X007F,^X00FF,^X01FF,^X01FE
          01E0 01F0 01F8 01FC    02A2                 .WORD   ^X01FC,^X01F8,^X01F0,^X01E0
          0000 0100 0180 01C0    02AA                 .WORD   ^X01C0,^X0180,^X0100,^X0000
                         0000    02B2  CYLN_LIST_X::  .WORD   0
                                 02B4
                                 02B4  ;++
                                 02B4  ; EXPECTED PACK NAME DURING PROTECTION CHECKING
                                 02B4  ;__
                                 02B4
  48 43 54 41 52 43 53 00'       02B4  VOL_NAME::             .ASCIC  ?SCRATCH?
                           07    02B4
                                 02BC
                                 02BC  ;++
```

```
                     02BC   ;THE NPR DATA BUFFERS ARE DEFINED HERE
                     02BC   ;--
                     02BC
           00000000' 02BC   WRITE_BUFFER::  .ADDRESS        BUFFER1
           00000400' 02C0                   .LONG   BUFFER1+1024
                     02C4
           00000400' 02C4   READ_BUFFER::   .ADDRESS        BUFFER2
           00000800' 02C8                   .LONG   BUFFER2+1024
                     02CC
           00000200' 02CC   SKIP_BUFFER::   .ADDRESS        BUFFER1+512
           00000400' 02D0                   .LONG   BUFFER1+1024
                     02D4
           00000800' 02D4   MBSF_BUFFER::   .ADDRESS        MBSF
           00000C00' 02D8                   .LONG   MBSF+1024
                     02DC   .SAVE
             00000000       .PSECT  BUFFERS PAGE                    ; ALIGN BUFFERS ON PAGE BOUNDRIES
                     0000
           00000400  0000   BUFFER1::       .BLKB   1024
           00000800  0400   BUFFER2::       .BLKB   1024
           00000A00  0800   MBSF::          .BLKB   512
             000002DC       .RESTORE
```

```
                          02DC     ;++
                          02DC     ;       DEVICE INFORMATION TABLES
                          02DC     ;--
                          02DC     LUN_TBL::
                000002EC' 02DC             .ADDRESS        LUN0_TBL
                00000370' 02E0             .ADDRESS        LUN1_TBL
                000003F4' 02E4             .ADDRESS        LUN2_TBL
                00000478' 02E8             .ADDRESS        LUN3_TBL
                          02EC
                          02EC     LUN0_TBL::
                00000370  02EC             .BLKL   NUMENT
                          0370
                          0370     LUN1_TBL::
                000003F4  0370             .BLKL   NUMENT
                          03F4
                          03F4     LUN2_TBL::
                00000478  03F4             .BLKL   NUMENT
                          0478
                          0478     LUN3_TBL::
                000004FC  0478             .BLKL   NUMENT
                          04FC
                          04FC     RL02_TBL::
                00000001  04FC             .LONG   RL02$T              ; DRIVE TYPE
                00000000  0500             .LONG   0                   ; STORAGE FOR PHYSICAL UNIT NUMBER
                000001FF  0504             .LONG   511                 ; LAST CYLINDER
                00000001  0508             .LONG   1                   ; LAST TRACK
                00000027  050C             .LONG   39                  ; LAST SECTOR
                00000155  0510             .LONG   341                 ; TEST CYLINDER
                000001FE  0514             .LONG   510                 ; CYLINDER FOR INNER GUARD BAND TEST
                000014F9' 0518             .ADDRESS MSG_RL02           ; ADDRESS OF DEV MNEMONIC
       00000000 00000000  051C             .QUAD   0                   ; ADDRESS OF DEVICE NAME DESCRIPTOR (DQxx)
                00000100  0524             .LONG   256                 ; BYTES PER SECTOR
                00002800  0528             .LONG   <256*40>            ; BYTES PER TRACK
                00000000  052C             .LONG   0                   ; BYTE COUNT FOR FORMAT COMMAND
                00000003  0530             .LONG   <MPR_GS$M_MARK!-    ; BITS REQUIRED FOR GET STATUS CMD
                          0534                     MPR_GS$M_GS>        ;
                0000009D  0534             .LONG   ^X9D                ; NORMAL STATUS (DT,HO,BH,STC,STA)
                0000009C  0538             .LONG   ^X9C                ; NORMAL SEEK STATUS (DT,HO,BH,STC)
                00000421' 053C             .ADDRESS MPR02_BITS         ; ADR OF MPR BIT MNEMONICS
                00000590' 0540             .ADDRESS HDR02_BITS         ; ADDRESS OF HDR BIT MNEMONICS
                00000604' 0544             .ADDRESS RL02_HDR_TMP       ; ADDRESS OF HEADER TEMPLATE
                0000DD00  0548             .LONG   ^XDD00              ; DRIVE ERROR MASK (WDE,HCE,SKTO,SPE,WGE,DSE)
                0000000B  054C             .LONG   <MPR_GS$M_MARK!-    ; DRIVE RESET MASK
                          0550                     MPR_GS$M_GS!-       ;
                          0550                     MPR_GS$M_RST>       ;
                00000000  0550             .LONG   0                   ; HEADER DONT CARE BITS BIT MASK
                00000000  0554             .LONG   0                   ; CSR DRIVE TYPE BIT
                00000002  0558             .LONG   2                   ; SECTOR FOR DISK LABEL
                000020A8  055C             .LONG   ^X20A8              ; MANUAL STATUS (WL!DT!CO!BH)
                00000000  0560             .LONG   0                   ; DRIVE STATUS DONT CARE BITS
                0000249D  0564             .LONG   ^X249D              ; WRITE PROT STATUS (WL!WGE!DT!HO!BH!STC!STA)
                000000DE  0568             .LONG   222                 ; CYLINDER FOR ADDRESS RECALL TEST
                00000992  056C             .LONG   2450                ; LO RPM TIMING RANGE 24.5 MS
                000009F6  0570             .LONG   2550                ; HI RPM TIMING RANGE 25.5 MS
                000005DC  0574             .LONG   1500                ; ONE CYLINDER TIMING SPEC 15 MS
                00002710  0578             .LONG   10000               ; MAX CYLINDER TIMING LIMIT 100 MS
```

```
         0000157C  057C                  .LONG    5500              ; ACCESS TIMING SPEC 55 MS
                   0580
                   0580      R80_TBL::
         00000002  0580                  .LONG    R80$T             ; DRIVE TYPE
         00000000  0584                  .LONG    0                 ; STORAGE FOR PHYSICAL UNIT NUMBER
         00000230  0588                  .LONG    560               ; LAST CYLINDER
         0000000D  058C                  .LONG    13                ; LAST TRACK
         0000001E  0590                  .LONG    30                ; LAST SECTOR
         00000230  0594                  .LONG    560               ; TEST CYLINDER
         00000231  0598                  .LONG    561               ; CYLINDER FOR INNER GUARD BAND TEST
         000014FE' 059C                  .ADDRESS MSG_R80           ; ADDRESS OF DEV MNEMONIC
00000000 00000000  05A0                  .QUAD    0                 ; ADDRESS OF DEVICE NAME DESXRIPTOR (DQxx)
         00000200  05A8                  .LONG    512               ; BYTES PER SECTOR
         00003E00  05AC                  .LONG    <512*31>          ; BYTES PER TRACK
         00000080  05B0                  .LONG    <4*32>            ; BYTE COUNT FOR FORMAT COMMAND
         00000000  05B4                  .LONG    0                 ; BITS REQUIRED FOR GET STATUS CMD
         00001A00  05B8                  .LONG    ^X1A00            ; NORMAL STATUS (DRDY,ONCY,PLGV)
         00001200  05BC                  .LONG    ^X1200            ; NORMAL SEEK STATUS MASK (DRDY,PLGV)
         000004A9' 05C0                  .ADDRESS MPR80_BITS        ; ADR OF MPR REG MNEMONICS
         000005C1' 05C4                  .ADDRESS HDR80_BITS        ; ADDRESS OF HDR BIT MNEMONICS
         00000613' 05C8                  .ADDRESS R80_HDR_TMP       ; ADDRESS OF HEADER TEMPLATE
         00000500  05CC                  .LONG    <MPR_80$M_FLT!MPR_80$M_SKE> ; DRIVE ERROR MASK
         00000000  05D0                  .LONG    0                 ; DRIVE RESET MASK
         0000F000  05D4                  .LONG    <HDR_80$M_FMT!-   ; HEADER WORD DONT CARE BITS
                   05D8                           HDR_80$M_MF!-     ;
                   05D8                           HDR_80$M_UF!-     ;
                   05D8                           HDR_80$M_SSF>     ;
         04000000  05D8                  .LONG    CSR$M_R80         ; CSR DRIVE TYPE BIT
         00000001  05DC                  .LONG    1                 ; SECTOR FOR DISK LABEL
         00000A00  05E0                  .LONG    ^XA00             ; MANUAL STATUS (PLGV!ONCY)
         0000C0FF  05E4                  .LONG    <MPR_80$M_SECT!-  ; DRIVE STATUS DONT CARE MASK
                   05E8                           BIT15!BIT14!BIT7!BIT6!BIT5>
         00003B00  05E8                  .LONG    ^X3B00            ; WRITE PROT STATUS (WTP!DRDY!ONCY!PLGV!FLT)
         00000023  05EC                  .LONG    35                ; CYLINDER FOR ADDRESS RECALL TEST
         00000652  05F0                  .LONG    1618              ; LOW RANGE FOR RPM SPEC 16.18 MS
         000006A6  05F4                  .LONG    1702              ; HI RNAGE FOR RPM SPEC 17.02 MS
         00000258  05F8                  .LONG    600               ; ONE CYLINCER TIMING SPEC 6 MS
         00001388  05FC                  .LONG    5000              ; MAX SEEK TIMING SPEC 50 MS
         000009C4  0600                  .LONG    2500              ; ACCESS TIMING SPEC 25 MS
                   0604
                   0604      ;++
                   0604      ;           HEADER TEMPLATES CONTAINS STARTING BIT AND SIZE
                   0604      ;           OF EACH FIELD IN THE HEADER
                   0604      ;--
                   0604
                   0604      RL02_HDR_TMP::
         00000007  0604                  .LONG    HDR_02$V_CYL
               09  0608                  .BYTE    HDR_02$S_CYL
         00000006  0609                  .LONG    HDR_02$V_TRK
               01  060D                  .BYTE    HDR_02$S_TRK
         00000000  060E                  .LONG    HDR_02$V_SEC
               06  0612                  .BYTE    HDR_02$S_SEC
                   0613
                   0613      R80_HDR_TMP::
         00000000  0613                  .LONG    HDR_80$V_CYL
               0A  0617                  .BYTE    HDR_80$S_CYL
```

```
00000018  0618   719          .LONG   HDR_80$V_TRK
      04  061C   720          .BYTE   HDR_80$S_TRK
00000010  061D   721          .LONG   HDR_80$V_SEC
      05  0621   722          .BYTE   HDR_80$S_SEC
```

```
                   0622              .SBTTL  CONVERT REGISTER TABLES
                   0622
                   0622      CVT_DRVTYP::
         00000002  0622              .LONG   2                                ; NUMBER OF ENTRIES
         000014F9' 0626              .ADDRESS          MSG_RL02               ; RL02 = 0
         000014FE' 062A              .ADDRESS          MSG_R80                ; R80 = 1
                   062E
                   062E      CVT_FUN_TABLE::
         00000008  062E              .LONG   8                                ; NUMBER OF ENTRIES
         000005F1' 0632              .ADDRESS          CVT_NOOP               ; NOOP = 0
         000005F7' 0636              .ADDRESS          CVT_WRTCHK             ; WRITE CHECK = 1
         00000603' 063A              .ADDRESS          CVT_GETSTAT            ; GET STATUS = 2
         0000060E' 063E              .ADDRESS          CVT_SEEK               ; SEEK = 3
         00000613' 0642              .ADDRESS          CVT_READH              ; READ HEADER = 4
         0000061F' 0646              .ADDRESS          CVT_WRITE              ; WRITE DATA = 5
         0000062A' 064A              .ADDRESS          CVT_READ               ; READ DATA = 6
         00000634' 064E              .ADDRESS          CVT_RDNHC              ; READ NO HDR CHK = 7
                   0652
                   0652      CVT_STA_TABLE::
         00000008  0652              .LONG   8                                ; NUMBER OF ENTRIES
         000006C1' 0656              .ADDRESS          CVT_LOAD               ; LOAD CARTIDGE = 0
         000006D0' 065A              .ADDRESS          CVT_SPINUP             ; SPIN UP = 1
         000006DC' 065E              .ADDRESS          CVT_BRUSH              ; BRUSH CYCLE = 2
         000006E8' 0662              .ADDRESS          CVT_LDHEADS            ; LOADING HEADS = 3
         000006F6' 0666              .ADDRESS          CVT_SK                 ; DRIVE SEEKING = 4
         000006FE' 066A              .ADDRESS          CVT_LOCKED             ; DRV LOCKED ON TRK = 5
         0000070E' 066E              .ADDRESS          CVT_UNLOAD             ; DRV UNLOADING HEADS = 6
         0000071E' 0672              .ADDRESS          CVT_SPINDN             ; SPIN DOWN = 7
                   0676
                   0676      CVT_ERR_TABLE::
         00000008  0676              .LONG   8                                ; NUMBER OF ENTRIES
         00000646' 067A              .ADDRESS          CVT_ERROR0             ; NO ERRORS = 0
         0000064B' 067E              .ADDRESS          CVT_ERROR1             ; OPI = 1
         00000660' 0682              .ADDRESS          CVT_ERROR2             ; DCK OR WRT CHK ERR = 2
         00000674' 0686              .ADDRESS          CVT_ERROR3             ; HCRC = 3
         0000067C' 068A              .ADDRESS          CVT_ERROR4             ; DATA LATE = 4
         00000686' 068E              .ADDRESS          CVT_ERROR5             ; HDR NOT FOUND = 5
         00000694' 0692              .ADDRESS          CVT_ERROR6             ; DLT AND DCK = 6
         000006A9' 0696              .ADDRESS          CVT_ERROR7             ; HNF AND HCRC = 7
                   069A
                   069A      CVT_SUR_TABLE::
         00000002  069A              .LONG   2                                ; NUMBER OF ENTRIES
         0000072C' 069E              .ADDRESS          CVT_UPPER              ; HEAD 0 OR UPPER SURFACE
         00000732' 06A2              .ADDRESS          CVT_LOWER              ; HEAD 1 OR LOWER SURFACE
                   06A6
                   06A6      CVT_DIR_TABLE::
         00000002  06A6              .LONG   2                                ; NUMBER OF ENTRIES
         00000738' 06AA              .ADDRESS          CVT_REVERSE            ; HEADS TRAVELING REVERSE
         00000740' 06AE              .ADDRESS          CVT_FORWARD            ; HEADS TRAVELING FORWARD
```

```
                  06B2          .SBTTL  OFFSET DEFINITIONS
                  06B2     ;++
                  06B2     ;        THIS SECTION DEFINES OFFSETS USED
                  06B2     ;        WITHIN THE PROGRAM.
                  06B2     ;--
                  06B2              DRV_TBL_DEF
                  06B2     ;+
                  06B2     ; OFFSET DEFINITIONS FOR DEVICE PARAMETER TABLE
                  06B2     ;-
        00000000  06B2     DEF=     OFST_DRVTYP      ; OFFSET TO DRIVE TYPE PARM
        00000004  06B2     DEF=     OFST_PHYU        ; OFFSET TO PHYSICAL UNIT NUMBER
        00000008  06B2     DEF=     OFST_LSTCYL      ; OFFSET TO LAST CYLINDER PARM
        0000000C  06B2     DEF=     OFST_LSTTRK      ; OFFSET TO LAST TRACK PARM
        00000010  06B2     DEF=     OFST_LSTSEC      ; OFFSET TO LAST SECTOR PARM
        00000014  06B2     DEF=     OFST_TSTCYL      ; OFFSET TO TEST CYLINDER
        00000018  06B2     DEF=     OFST_IGBCYL      ; OFFSET TO CYLINDER USED FOR GUARD BAND TESTING
        0000001C  06B2     DEF=     OFST_DEVMEM      ; OFFSET TO DEVICE MNEMONICS
        00000020  06B2     DEF=     OFST_DEVNAM      ; OFFSET TO DEVICE NAME
        00000028  06B2     DEF=     OFST_SECLEN      ; OFFSET TO SECTOR BYTE COUNT
        0000002C  06B2     DEF=     OFST_TRKLEN      ; OFFSET TO TRACK BYTE COUNT
        00000030  06B2     DEF=     OFST_FMTLEN      ; OFFSET TO TRACK BYTE COUNT PARM
        00000034  06B2     DEF=     OFST_GTSCMD      ; OFFSET TO GET STATUS COMMAND BITS (MRP REG)
        00000038  06B2     DEF=     OFST_NMLSTS      ; OFFSET TO EXPECTED NORMAL DRIVE STATUS BITS
        0000003C  06B2     DEF=     OFST_SEKSTS      ; OFFSET TO EXPECTED SEEK STATUS
        00000040  06B2     DEF=     OFST_MPBITS      ; OFFSET TO ADDRESS OF MPR REG BIT MNEMONICS
        00000044  06B2     DEF=     OFST_HDBITS      ; OFFSET TO ADDRESS OF HDR BIT MNEMONICS
        00000048  06B2     DEF=     OFST_HDRTMP      ; OFFSET TO HEADER TEMPLATE
        0000004C  06B2     DEF=     OFST_DRVERR      ; DRIVE ERROR BIT MASK
        00000050  06B2     DEF=     OFST_DRVRST      ; DRIVE RESET MASK
        00000054  06B2     DEF=     OFST_HDRDNC      ; HEADER WORD DONT CARE BITS BIT MASK
        00000058  06B2     DEF=     OFST_CSRTYP      ; DRIVE TYPE BIT IN CSR REG
        0000005C  06B2     DEF=     OFST_HMBSEC      ; SECTOR ADDRESS OF HOME BLOCK
        00000060  06B2     DEF=     OFST_OP1STS      ; DRIVE STATUS FOR MANUAL INTERVENTION
        00000064  06B2     DEF=     OFST_STSDNC      ; DRIVE STATUS DONT CARE BIT MASK
        00000068  06B2     DEF=     OFST_WTPSTS      ; DRIVE STATUS FOR WRITE PROTECTION TEST
        0000006C  06B2     DEF=     OFST_RANCYL      ; FINAL CYLINDER FOR ADDRESS RECALL TEST
        00000070  06B2     DEF=     OFST_LOWRPM      ; LOW RANGE FOR ROTATION TIMING
        00000074  06B2     DEF=     OFST_HIGRPM      ; HI RANGE FOR ROTATION TIMING
        00000078  06B2     DEF=     OFST_ONETIM      ; SPEC FOR ONE CYLINDER SEEK TIMING
        0000007C  06B2     DEF=     OFST_MAXTIM      ; SPEC FOR MAX SEEK TIMING
        00000080  06B2     DEF=     OFST_AVGTIM      ; SPEC FOR AVERAGE SEEK TIMING
```

```
                          06B2   781          .SBTTL  PROGRAM TEXT SECTION.
                          06B2   782
                          06B2   783 ;++
                          06B2   784 ; FUNCTIONAL DESCRIPTION::
                          06B2   785 ;
                          06B2   786 ;        THIS SECTION CONTAINS ALL THE COMMON CHARACTER STRING TYPE DATA
                          06B2   787 ;        ENTRIES.
                          06B2   788 ;
                          06B2   789 ;--
                          06B2   790
                      00000000   791          .PSECT  GLOBAL_TEXT, LONG, NOEXE, NOWRT
                          0000   792
                          0000   793 ;++
                          0000   794 ; PROGRAM SECTION NAMES
                          0000   795 ;--
                          0000   796
                          0000   797          $DS_SECTION        MANUAL,ALL
                          0000       S_DEFAULT:
54 4C 55 41 46 45 44 00'  0000                          .ASCIC  \DEFAULT\
                      07  0000
                          0008       S_MANUAL:
   4C 41 55 4E 41 4D 00'  0008                          .ASCIC  \MANUAL\
                      06  0008
                          000F       S_ALL:
            4C 4C 41 00'  000F                          .ASCIC  \ALL\
                      03  000F
                          0013       SECTION:
               00000003   0013                          .LONG   3
               00000000'  0017                          .LONG   S_DEFAULT
               00000008'  001B                          .LONG   S_MANUAL
               0000000F'  001F                          .LONG   S_ALL
                          0023   798
                          0023   799
                          0023   800 ;+
                          0023   801 ; DEVICE MNEMONICS LIST.
                          0023   802 ;-
                          0023   803
                          0023   804 T_DEVICE::
                          0023   805          $DS_DEVTYP         <RL02,R80>
         32 30 4C 52 00'  0023       T_RL02:            .ASCIC  \RL02\
                      04  0023
            30 38 52 00'  0028       T_R80:             .ASCIC  \R80\
                      03  0028
                      00  002C               .BYTE   0
               00000002   002D       AL_DEVTYP:         .LONG   $$N
               00000023'  0031                          .LONG   T_RL02
               00000028'  0035                          .LONG   T_R80
                          0039   806
```

```
                                      0039   808            .SBTTL  IDC REGISTER BIT MNEMONICS
                                      0039   809
                                      0039   810 COM_BIT32::
30 33 54 49 42 2C 31 33 54 49 42 00'  0039   811            .ASCIC  /BIT31,BIT30,BIT29,BIT28,BIT27,BIT26,BIT25,BIT24,/-
38 32 54 49 42 2C 39 32 54 49 42 2C   0045
36 32 54 49 42 2C 37 32 54 49 42 2C   0051
34 32 54 49 42 2C 35 32 54 49 42 2C   005D
                                 2C   0069
2C 32 32 54 49 42 2C 33 32 54 49 42   006A   812                    /BIT23,BIT22,BIT21,BIT20,BIT19,BIT18,BIT17,BIT16,/-
2C 30 32 54 49 42 2C 31 32 54 49 42   0076
2C 38 31 54 49 42 2C 39 31 54 49 42   0082
2C 36 31 54 49 42 2C 37 31 54 49 42   008E
2C 34 31 54 49 42 2C 35 31 54 49 42   009A   813                    /BIT15,BIT14,BIT13,BIT12,BIT11,BIT10,BIT9,/-
2C 32 31 54 49 42 2C 33 31 54 49 42   00A6
2C 30 31 54 49 42 2C 31 31 54 49 42   00B2
                  2C 39 54 49 42      00BE
49 42 2C 37 54 49 42 2C 38 54 49 42   00C3   814                    /BIT8,BIT7,BIT6,BIT5,BIT4,BIT3,BIT2,BIT1,BIT0/
34 54 49 42 2C 35 54 49 42 2C 36 54   00CF
42 2C 32 54 49 42 2C 33 54 49 42 2C   00DB
         30 54 49 42 2C 31 54 49      00E7
                                 B5   0039
                                      00EF   815 COM_BIT16::
34 31 54 49 42 2C 35 31 54 49 42 00'  00EF   816            .ASCIC  /BIT15,BIT14,BIT13,BIT12,BIT11,BIT10,BIT9,/-
32 31 54 49 42 2C 33 31 54 49 42 2C   00FB
30 31 54 49 42 2C 31 31 54 49 42 2C   0107
                  2C 39 54 49 42 2C   0113
49 42 2C 37 54 49 42 2C 38 54 49 42   0119   817                    /BIT8,BIT7,BIT6,BIT5,BIT4,BIT3,BIT2,BIT1,BIT0/
34 54 49 42 2C 35 54 49 42 2C 36 54   0125
42 2C 32 54 49 42 2C 33 54 49 42 2C   0131
         30 54 49 42 2C 31 54 49      013D
                                 55   00EF
                                      0145   818 IDCCSR_BITS::
41 2C 49 4F 54 2C 54 4D 46 2C 2C 00'  0145   819            .ASCIC  /,,FMT,TOI,ASSI,DT=1@,MTN,IR,SSE,SSEI,ECS1,ECS0,/-
54 4D 2C 40 31 3D 54 44 2C 49 53 53   0151
45 53 53 2C 45 53 53 2C 52 49 2C 4E   015D
2C 30 53 43 45 2C 31 53 43 45 2C 49   0169
54 41 2C 32 4E 54 41 2C 33 4E 54 41   0175   820                    /ATN3,ATN2,ATN1,ATN0,/-
         2C 30 4E 54 41 2C 31 4E      0181
45 2C 4D 58 4E 2C 45 44 2C 52 52 45   0189   821                    /ERR,DE,NXM,ERRTYP=3@,DRIVE=2^D,/-
49 52 44 2C 40 33 3D 50 59 54 52 52   0195
         2C 44 5E 32 3D 45 56         01A1
35 54 49 42 2C 45 49 2C 59 44 52 43   01A8   822                    /CRDY,IE,BIT5,BIT4,FUNCTION=3@,DRDY/
49 54 43 4E 55 46 2C 34 54 49 42 2C   01B4
      59 44 52 44 2C 40 33 3D 4E 4F   01C0
                                 84   0145
                                      01CA   823 CSR_BITS::
41 2C 49 4F 54 2C 54 4D 46 2C 2C 00'  01CA   824            .ASCIC  /,,FMT,TOI,ASSI,R80,MTN,IR,SSE,SSEI,ECS1,ECS0,/-
2C 4E 54 4D 2C 30 38 52 2C 49 53 53   01D6
2C 49 45 53 53 2C 45 53 53 2C 52 49   01E2
      2C 30 53 43 45 2C 31 53 43 45   01EE
54 41 2C 32 4E 54 41 2C 33 4E 54 41   01F8   825                    /ATN3,ATN2,ATN1,ATN0,/-
         2C 30 4E 54 41 2C 31 4E      0204
44 2C 4D 58 4E 2C 45 44 2C 52 52 45   020C   826                    /ERR,DE,NXM,DLT-HNF,DCK-HCRC,OPI,DS1,DS0,/-
48 2D 4B 43 44 2C 46 4E 48 2D 54 4C   0218
2C 31 53 44 2C 49 50 4F 2C 43 52 43   0224
                  2C 30 53 44         0230
```

```
35 54 49 42 2C 45 49 2C 59 44 52 43  0234   827                         /CRDY,IE,BIT5,BIT4,F2,F1,F0,DRDY/
2C 31 46 2C 32 46 2C 34 54 49 42 2C  0240
               59 44 52 44 2C 30 46  024C
                                 88  01CA
                                     0253   828 IDCBAR_BITS::
2C 2C 2C 2C 2C 2C 2C 2C 2C 2C 2C 00' 0253   829             .ASCIC  /,,,,,,,,,,,,,,,BUS ADDRESS=16^X/
44 44 41 20 53 55 42 2C 2C 2C 2C 2C  025F
         58 5E 36 31 3D 53 53 45 52  026B
                                 20  0253
                                     0274   830 IDCCNT_BITS::
2C 2C 2C 2C 2C 2C 2C 2C 2C 2C 2C 00' 0274   831             .ASCIC  /,,,,,,,,,,,,,,,BYTE COUNT=16^X/
4F 43 20 45 54 59 42 2C 2C 2C 2C 2C  0280
         58 5E 36 31 3D 54 4E 55     028C
                                 1F  0274
                                     0294   832 DAR_DMA02_BITS::
2C 2C 2C 2C 2C 2C 2C 2C 2C 2C 2C 00' 0294   833             .ASCIC  /,,,,,,,,,,,,,,,CYLINDER=9^D,HEAD=1^D,SECTOR=6^D/
45 44 4E 49 4C 59 43 2C 2C 2C 2C 2C  02A0
31 3D 44 41 45 48 2C 44 5E 39 3D 52  02AC
5E 36 3D 52 4F 54 43 45 53 2C 44 5E  02B8
                                 44  02C4
                                 30  0294
                                     02C5   834 DAR_DMA80_BITS::
2C 2C 2C 2C 2C 2C 2C 2C 2C 2C 2C 00' 02C5   835             .ASCIC  /,,,,,,,,,,,,CYLINDER=10^D,HEAD=4^D,SECTOR=5^D/
31 3D 52 45 44 4E 49 4C 59 43 2C 2C  02D1
44 5E 34 3D 44 41 45 48 2C 44 5E 30  02DD
   44 5E 35 3D 52 4F 54 43 45 53 2C  02E9
                                 2E  02C5
                                     02F4   836 DAR_SK02_BITS::
2C 2C 2C 2C 2C 2C 2C 2C 2C 2C 2C 00' 02F4   837             .ASCIC  /,,,,,,,,,,,,,,,DIFF_WORD=9^D,BIT6,BIT5,HEAD=1@,BIT3,/-
4F 57 5F 46 46 49 44 2C 2C 2C 2C 2C  0300
2C 36 54 49 42 2C 44 5E 39 3D 44 52  030C
40 31 3D 44 41 45 48 2C 35 54 49 42  0318
               2C 33 54 49 42 2C     0324
2C 31 54 49 42 2C 40 31 3D 52 49 44  032A   838                         /DIR=1@,BIT1,M/
                                 4D  0336
                                 42  02F4
                                     0337   839 DAR_SK80_BITS::
2C 2C 2C 2C 2C 2C 2C 2C 2C 2C 2C 00' 0337   840             .ASCIC  /,,,,,,,,,,,,,,,CNTLTAG, HEADTAG,CYLTAG,,,,CYLINDER=10^D/
47 41 54 4C 54 4E 43 2C 2C 2C 2C 2C  0343
59 43 2C 47 41 54 44 41 45 48 20 2C  034F
49 4C 59 43 2C 2C 2C 2C 47 41 54 4C  035B
         44 5E 30 31 3D 52 45 44 4E  0367
                                 38  0337
                                     0370   841 DAR_RCL80_BITS::
2C 2C 2C 2C 2C 2C 2C 2C 2C 2C 2C 00' 0370   842             .ASCIC  /,,,,,,,,,,,,,,,CNTLTAG,HEADTAG,CYCLTAG,,,,,,RECAL,,,,,/
47 41 54 4C 54 4E 43 2C 2C 2C 2C 2C  037C
43 59 43 2C 47 41 54 44 41 45 48 2C  0388
52 2C 2C 2C 2C 2C 2C 2C 47 41 54 4C  0394
      2C 2C 2C 2C 2C 2C 4C 41 43 45  03A0
                                 39  0370
                                     03AA   843 DAR_GS02_BITS::
2C 2C 2C 2C 2C 2C 2C 2C 2C 2C 2C 00' 03AA   844             .ASCIC  /,,,,,,,,,,,,,,,,,,,,,,,,,,RST,,GS,M/
2C 2C 2C 2C 2C 2C 2C 2C 2C 2C 2C 2C  03B6
53 47 2C 2C 54 53 52 2C 2C 2C 2C 2C  03C2
                              4D 2C  03CE
                                 25  03AA
```

```
                                  03D0    845 IDCMPR02_BITS::
2C 2C 2C 2C 2C 2C 2C 2C 2C 2C 2C 00' 03D0    846         .ASCIC  /,,,,,,,,,,,,,,,,/-
               2C 2C 2C 2C 2C 03DC
53 2C 4C 57 2C 45 48 43 2C 45 44 57 03E1    847                 /WDE,CHE,WL,SKTO,SPE,WGE,VC,DSE,DT,HD_SELECT=1@,CO,/-
2C 45 47 57 2C 45 50 53 2C 4F 54 4B 03ED
44 48 2C 54 44 2C 45 53 44 2C 43 56 03F9
43 2C 40 31 3D 54 43 45 4C 45 53 5F 0405
                           2C 4F 0411
3D 45 54 41 54 53 2C 48 42 2C 4F 48 0413    848                 /HO,BH,STATE=3@/
                           40 33 041F
                              50 03D0
                                  0421    849 MPR02_BITS::
2C 2C 2C 2C 2C 2C 2C 2C 2C 2C 2C 00' 0421    850         .ASCIC  /,,,,,,,,,,,,,,,,/-
               2C 2C 2C 2C 2C 042D
53 2C 4C 57 2C 45 48 43 2C 45 44 57 0432    851                 /WDE,CHE,WL,SKTO,SPE,WGE,VC,DSE,DT,HS,CO,/-
2C 45 47 57 2C 45 50 53 2C 4F 54 4B 043E
53 48 2C 54 44 2C 45 53 44 2C 43 56 044A
                  2C 4F 43 2C 0456
54 53 2C 43 54 53 2C 48 42 2C 4F 48 045A    852                 /HO,BH,STC,STB,STA/
               41 54 53 2C 42 0466
                              49 0421
                                  046B    853 IDCMPR80_BITS::
2C 2C 2C 2C 2C 2C 2C 2C 2C 2C 2C 00' 046B    854         .ASCIC  /,,,,,,,,,,,,,,,,,WTP,DRDY,ONCY,SKE,PLGV,FLT,,,,LOOKAHEAD=5^D/
44 2C 50 54 57 2C 2C 2C 2C 2C 2C 2C 0477
45 4B 53 2C 59 43 4E 4F 2C 59 44 52 0483
2C 2C 2C 54 4C 46 2C 56 47 4C 50 2C 048F
35 3D 44 41 45 48 41 4B 4F 4F 4C 2C 049B
                           44 5E 04A7
                              3D 046B
                                  04A9    855 MPR80_BITS::
2C 2C 2C 2C 2C 2C 2C 2C 2C 2C 2C 00' 04A9    856         .ASCIC  /,,,,,,,,,,,,,,,,/-
               2C 2C 2C 2C 2C 04B5
4F 2C 59 44 52 44 2C 50 54 57 2C 2C 04BA    857                 /,,WTF,DRDY,ONCY,SKE,PLGV,FLT,,,SEC4,SEC3,SEC2,SEC1,SEC0/
56 47 4C 50 2C 45 4B 53 2C 59 43 4E 04C6
2C 34 43 45 53 2C 2C 2C 54 4C 46 2C 04D2
45 53 2C 32 43 45 53 2C 33 43 45 53 04DE
            30 43 45 53 2C 31 43 04EA
                              47 04A9
                                  04F1    858 IDCPOS_BITS::
2C 2C 2C 2C 2C 2C 2C 2C 2C 2C 2C 00' 04F1    859         .ASCIC  /,,,,,,,,,,,,,,,,,,POSITION=13^D/
49 53 4F 50 2C 2C 2C 2C 2C 2C 2C 2C 04FD
      44 5E 33 31 3D 4E 4F 49 54 0509
                              20 04F1
                                  0512    860 IDCPAT_BITS::
2C 2C 2C 2C 2C 2C 2C 2C 2C 2C 2C 00' 0512    861         .ASCIC  /,,,,,,,,,,,,,,,,,,,PATTERN=11^X/
41 50 2C 2C 2C 2C 2C 2C 2C 2C 2C 2C 051E
   58 5E 31 31 3D 4E 52 45 54 54 052A
                              21 0512
                                  0534    862 HDR1_02_BITS::
5E 39 3D 52 45 44 4E 49 4C 59 43 00' 0534    863         .ASCIC  /CYLINDER=9^D,TRACK=1@,SECTOR=6^D/
53 2C 40 31 3D 4B 43 41 52 54 2C 44 0540
      44 5E 36 3D 52 4F 54 43 45 054C
                              20 0534
                                  0555    864 HDR1_80_BITS::
46 2C 46 53 53 2C 46 55 2C 46 4D 00' 0555    865         .ASCIC  /MF,UF,SSF,FMT16,,CYLINDER=10^D/
44 4E 49 4C 59 43 2C 2C 36 31 54 4D 0561
```

```
                  44 5E 30 31 3D 52 45  056D
                                    1E  0555
                                        0574     866 HDR2_80_BITS::
34 3D 4B 43 41 52 54 2C 2C 2C 2C 00'  0574     867          .ASCIC  /,,,,TRACK=4^D,,,,SECTOR=5^D/
52 4F 54 43 45 53 2C 2C 2C 2C 44 5E  0580
                           44 5E 35 3D  058C
                                    1B  0574
                                        0590     868 HDR02_BITS::
31 3D 44 52 4F 57 2D 4F 52 45 5A 00'  0590     869          .ASCIC  /ZERO-WORD=16^D,CYLINDER=9^D,TRACK=1^D,SECTOR=6^D/
52 45 44 4E 49 4C 59 43 2C 44 5E 36  059C
31 3D 4B 43 41 52 54 2C 44 5E 39 3D  05A8
5E 36 3D 52 4F 54 43 45 53 2C 44 5E  05B4
                                    44  05C0
                                    30  0590
                                        05C1     870 HDR80_BITS::
34 3D 4B 43 41 52 54 2C 2C 2C 2C 00'  05C1     871          .ASCIC  /,,,,TRACK=4^D,,,,SECTOR=5^D,,,,,,,CYLINDER=10^D/
52 4F 54 43 45 53 2C 2C 2C 2C 44 5E  05CD
43 2C 2C 2C 2C 2C 2C 2C 44 5E 35 3D  05D9
44 5E 30 31 3D 52 45 44 4E 49 4C 59  05E5
                                    2F  05C1
                                        05F1     872
                                        05F1     873 ;++
                                        05F1     874 ; THE FOLLOWING ARE THE ASCII MESSAGES FOR THE IDC FUNCTION CODES
                                        05F1     875 ; FOR THE $DS_CVT_REG_L MACRO
                                        05F1     876 ;__
                                        05F1     877
                                        05F1     878 CVT_NOOP::
               50 4F 20 4F 4E 00'  05F1     879          .ASCIC  \NO OP\
                                    05  05F1
                                        05F7     880 CVT_WRTCHK::
4B 43 45 48 43 20 45 54 49 52 57 00'  05F7     881          .ASCIC  \WRITE CHECK\
                                    0B  05F7
                                        0603     882 CVT_GETSTAT::
   53 55 54 41 54 53 20 54 45 47 00'  0603     883          .ASCIC  \GET STATUS\
                                    0A  0603
                                        060E     884 CVT_SEEK::
                     4B 45 45 53 00'  060E     885          .ASCIC  \SEEK\
                                    04  060E
                                        0613     886 CVT_READH::
52 45 44 41 45 48 20 44 41 45 52 00'  0613     887          .ASCIC  \READ HEADER\
                                    0B  0613
                                        061F     888 CVT_WRITE::
   41 54 41 44 20 45 54 49 52 57 00'  061F     889          .ASCIC  \WRITE DATA\
                                    0A  061F
                                        062A     890 CVT_READ::
      41 54 41 44 20 44 41 45 52 00'  062A     891          .ASCIC  \READ DATA\
                                    09  062A
                                        0634     892 CVT_RDNHC::
52 44 48 20 4F 4E 20 44 41 45 52 00'  0634     893          .ASCIC  \READ NO HDR CHECK\
                  4B 43 45 48 43 20  0640
                                    11  0634
                                        0646     894
                                        0646     895 ;++
                                        0646     896 ; THE FOLLOWING ARE ASCII MESSAGES FOR THE CONTROLLER ERROR
                                        0646     897 ; BITS (BIT 10 THRU BIT 12) USED IN THE $DS_CVTREG MACRO.
                                        0646     898 ;--
```

```
                                     0646    899
                                     0646    900 CVT_ERROR0::
                  45 4E 4F 4E 00'    0646    901            .ASCIC   \NONE\
                              04     0646
                                     064B    902 CVT_ERROR1::
49 20 4E 4F 49 54 41 52 45 50 4F 00' 064B    903            .ASCIC   \OPERATION INCOMPLETE\
      45 54 45 4C 50 4D 4F 43 4E     0657
                              14     064B
                                     0660    904 CVT_ERROR2::
52 4F 20 4B 48 43 20 41 54 41 44 00' 0660    905            .ASCIC   \DATA CHK OR WRT CHK\
         4B 48 43 20 54 52 57 20     066C
                              13     0660
                                     0674    906 CVT_ERROR3::
         43 52 43 20 52 44 48 00'    0674    907            .ASCIC   \HDR CRC\
                              07     0674
                                     067C    908 CVT_ERROR4::
   45 54 41 4C 20 41 54 41 44 00'    067C    909            .ASCIC   \DATA LATE\
                              09     067C
                                     0686    910 CVT_ERROR5::
55 4F 46 20 54 4F 4E 20 52 44 48 00' 0686    911            .ASCIC   \HDR NOT FOUND\
                           44 4E     0692
                              0D     0686
                                     0694    912 CVT_ERROR6::
26 20 45 54 41 4C 20 41 54 41 44 00' 0694    913            .ASCIC   \DATA LATE & DATA CHK\
      4B 48 43 20 41 54 41 44 20     06A0
                              14     0694
                                     06A9    914 CVT_ERROR7::
55 4F 46 20 54 4F 4E 20 52 44 48 00' 06A9    915            .ASCIC   \HDR NOT FOUND & HDR CRC\
43 52 43 20 52 44 48 20 26 20 44 4E  06B5
                              17     06A9
                                     06C1    916
                                     06C1    917 ;++
                                     06C1    918 ; THE FOLLOWING ARE THE ASCII MESSAGES FOR THE RL02 DRIVE
                                     06C1    919 ; STATUS CODES USED FOR THE $DS_CVTREG_L MACRO
                                     06C1    920 ;--
                                     06C1    921
                                     06C1    922 CVT_LOAD::
49 52 54 52 41 43 20 44 41 4F 4C 00' 06C1    923            .ASCIC   \LOAD CARTRIDGE\
                        45 47 44     06CD
                              0E     06C1
                                     06D0    924 CVT_SPINUP::
50 55 20 47 4E 49 4E 4E 49 50 53 00' 06D0    925            .ASCIC   \SPINNING UP\
                              0B     06D0
                                     06DC    926 CVT_BRUSH::
45 4C 43 59 43 20 48 53 55 52 42 00' 06DC    927            .ASCIC   \BRUSH CYCLE\
                              0B     06DC
                                     06E8    928 CVT_LDHEADS::
41 45 48 20 47 4E 49 44 41 4F 4C 00' 06E8    929            .ASCIC   \LOADING HEADS\
                           53 44     06F4
                              0D     06E8
                                     06F6    930 CVT_SK::
         47 4E 49 4B 45 45 53 00'    06F6    931            .ASCIC   \SEEKING\
                              07     06F6
                                     06FE    932 CVT_LOCKED::
54 20 4E 4F 20 44 45 4B 43 4F 4C 00' 06FE    933            .ASCIC   \LOCKED ON TRACK\
                     4B 43 41 52     070A
```

```
                                  0F  06FE
                                      070E    934 CVT_UNLOAD::
48 20 47 4E 49 44 41 4F 4C 4E 55 00'  070E    935           .ASCIC  \UNLOADING HEADS\
                         53 44 41 45  071A
                                  0F  070E
                                      071E    936 CVT_SPINDN::
4F 44 20 47 4E 49 4E 4E 49 50 53 00'  071E    937           .ASCIC  \SPINNING DOWN\
                               4E 57  072A
                                  0D  071E
                                      072C    938
                                      072C    939 ;++
                                      072C    940 ; THE FOLLOWING ARE THE ASCII MESSAGES FOR THE IDC
                                      072C    941 ; HEAD SELECT BIT.
                                      072C    942 ;++
                                      072C    943
                                      072C    944 CVT_UPPER::
                   52 45 50 50 55 00' 072C    945           .ASCIC  \UPPER\
                                  05  072C
                                      0732    946 CVT_LOWER::
                   52 45 57 4F 4C 00' 0732    947           .ASCIC  \LOWER\
                                  05  0732
                                      0738    948
                                      0738    949 ;++
                                      0738    950 ; THE FOLLOWING ARE THE ASCII MESSAGES FOR THE IDC
                                      0738    951 ; HEAD DIRECTION BIT USED ON THE RL02 DRIVES.
                                      0738    952 ;++
                                      0738    953
                                      0738    954 CVT_REVERSE::
             45 53 52 45 56 45 52 00' 0738    955           .ASCIC  \REVERSE\
                                  07  0738
                                      0740    956 CVT_FORWARD::
             44 52 41 57 52 4F 46 00' 0740    957           .ASCIC  \FORWARD\
                                  07  0740
                                      0748    958
                                      0748    959 ;++
                                      0748    960 ; GLOBAL FORMATTED ASCII OUTPUT STATEMENTS
                                      0748    961 ;--
                                      0748    962
                      2F 21 2F 21 00' 0748    963 FMT_CRLF::       .ASCIC  \!/!/\
                                  04  0748
                            2F 21 00' 074D    964 FMT_CRLF1::      .ASCIC  \!/\
                                  02  074D
58 21 5F 21 3A 44 4F 4F 47 2F 21 00'  0750    965 FMT_GOOD::       .ASCIC  ?!/GOOD:!_!XL(X) ;  !AC?
   43 41 21 20 20 3B 20 29 58 28 4C   075C
                                  16  0750
4C 58 21 5F 21 3A 44 41 42 2F 21 00'  0767    966 FMT_BAD::        .ASCIC  "!/BAD:!_!XL(X) ;  !AC"
      43 41 21 20 20 3B 20 29 58 28   0773
                                  15  0767
4C 58 21 5F 21 3A 52 4F 58 2F 21 00'  077D    967 FMT_XOR::        .ASCIC  "!/XOR:!_!XL(X)"
                            29 58 28  0789
                                  0E  077D
45 50 59 54 20 4B 53 49 44 2F 21 00'  078C    968 FMT_DRVTYP::     .ASCIC  ?!/DISK TYPE = !AC FOR PHYSICAL DEVICE !AS?
50 20 52 4F 46 20 43 41 21 20 3D 20   0798
49 56 45 44 20 4C 41 43 49 53 59 48   07A4
                   53 41 21 20 45 43  07B0
                                  29  078C
```

```
49 20 45 52 41 57 52 41 48 2F 21 00' 07B6   969 FMT_ATTACH::    .ASCIC  ?!/HARWARE INSPECTION INDICATES DEVICE IS AN?-
4E 49 20 4E 4F 49 54 43 45 50 53 4E  07C2
49 56 45 44 20 53 45 54 41 43 49 44  07CE
            4E 41 20 53 49 20 45 43  07DA
54 41 52 45 50 4F 20 43 41 21 2F 21  07E2   970                         ?!/!AC OPERATOR ATTACHED AN !AC!/?
20 44 45 48 43 41 54 54 41 20 52 4F  07EE
            2F 21 43 41 21 20 4E 41  07FA
                                  4B 07B6
                                     0802   971 FMT_SEEK1::
4E 49 4C 59 43 20 45 4E 4F 2F 21 00' 0802   972            .ASCIC  ?!/ONE CYLINDER SEEK TIMING FOR DEVICE !AS?
4D 49 54 20 4B 45 45 53 20 52 45 44  080E
49 56 45 44 20 52 4F 46 20 47 4E 49  081A
                  53 41 21 20 45 43  0826
                                  29 0802
                                     082C   973 FMT_SEEK2::
43 20 4D 55 4D 49 58 41 4D 2F 21 00' 082C   974            .ASCIC  ?!/MAXIMUM CYLINDER SEEK TIMING FOR DEVICE !AS?
4B 45 45 53 20 52 45 44 4E 49 4C 59  0838
20 52 4F 46 20 47 4E 49 4D 49 54 20  0844
      53 41 21 20 45 43 49 56 45 44  0850
                                  2D 082C
                                     085A   975 FMT_SEEK3::
43 20 45 47 41 52 45 56 41 2F 21 00' 085A   976            .ASCIC  ?!/AVERAGE CYLINDER SEEK TIMING FOR DEVICE !AS?
4B 45 45 53 20 52 45 44 4E 49 4C 59  0866
20 52 4F 46 20 47 4E 49 4D 49 54 20  0872
      53 41 21 20 45 43 49 56 45 44  087E
                                  2D 085A
                                     0888   977 FMT_SEEK1_SP::
4E 49 4C 59 43 20 45 4E 4F 2F 21 00' 0888   978            .ASCIC  ?!/ONE CYLINDER TIMING MAY NOT BE GREATER THAN !UL.!2ZL MS.?
4D 20 47 4E 49 4D 49 54 20 52 45 44  0894
52 47 20 45 42 20 54 4F 4E 20 59 41  08A0
21 20 4E 41 48 54 20 52 45 54 41 45  08AC
   2E 53 4D 20 4C 5A 32 21 2E 4C 55  08B8
                                  3A 0888
                                     08C3   979 FMT_SEEK2_SP::
43 20 4D 55 4D 49 58 41 4D 2F 21 00' 08C3   980            .ASCIC  ?!/MAXIMUM CYLINDER TIMING MAY NOT BE GREATER THAN !UL.!2ZL MS.?
49 4D 49 54 20 52 45 44 4E 49 4C 59  08CF
42 20 54 4F 4E 20 59 41 4D 20 47 4E  08DB
48 54 20 52 45 54 41 45 52 47 20 45  08E7
20 4C 5A 32 21 2E 4C 55 21 20 4E 41  08F3
                        2E 53 4D     08FF
                                  3E 08C3
                                     0902   981 FMT_SEEK3_SP::
43 20 45 47 41 52 45 56 41 2F 21 00' 0902   982            .ASCIC  ?!/AVERAGE CYLINDER TIMING MAY NOT BE GREATER THAN !UL.!2ZL MS.?
49 4D 49 54 20 52 45 44 4E 49 4C 59  090E
42 20 54 4F 4E 20 59 41 4D 20 47 4E  091A
48 54 20 52 45 54 41 45 52 47 20 45  0926
20 4C 5A 32 21 2E 4C 55 21 20 4E 41  0932
                        2E 53 4D     093E
                                  3E 0902
                                     0941   983 FMT_ONE_SEEK::
4E 49 4C 59 43 20 45 4E 4F 2F 21 00' 0941   984            .ASCIC  ?!/ONE CYLINDER SEEK TIME WAS:!_!UL.!2ZL MS.?
4D 49 54 20 4B 45 45 53 20 52 45 44  094D
2E 4C 55 21 5F 21 3A 53 41 57 20 45  0959
            2E 53 4D 20 4C 5A 32 21  0965
                                  2B 0941
                                     096D   985 FMT_MAX_SEEK::
```

```
43 20 4D 55 4D 49 58 41 4D 2F 21 00' 096D   986           .ASCIC  ?!/MAXIMUM CYLINDER SEEK TIME WAS:!_!UL.!2ZL MS.?
4B 45 45 53 20 52 45 44 4E 49 4C 59  0979
5F 21 3A 53 41 57 20 45 4D 49 54 20  0985
2E 53 4D 20 4C 5A 32 21 2E 4C 55 21  0991
                                  2F  096D
                                      099D   987 FMT_AVG_SEEK::
43 20 45 47 41 52 45 56 41 2F 21 00' 099D   988           .ASCIC  ?!/AVERAGE CYLINDER SEEK TIME WAS:!_!UL.!2ZL MS.?
4B 45 45 53 20 52 45 44 4E 49 4C 59  09A9
5F 21 3A 53 41 57 20 45 4D 49 54 20  09B5
2E 53 4D 20 4C 5A 32 21 2E 4C 55 21  09C1
                                  2F  099D
45 4C 50 4D 49 20 54 4F 4E 2F 21 00' 09CD   989 FMT_NYI::       .ASCIC  \!/NOT IMPLEMENTED IN THIS DIAGNOSTIC!/\
48 54 20 4E 49 20 44 45 54 4E 45 4D  09D9
49 54 53 4F 4E 47 41 49 44 20 53 49  09E5
                            2F 21 43  09F1
                                  26  09CD
                                      09F4   990 FMT_VOL::
4F 4D 20 45 4D 55 4C 4F 56 2F 21 00' 09F4   991           .ASCIC  ?!/VOLUME MOUNTED ON !AS FAILED PROTECTION CHECK!/?-
53 41 21 20 4E 4F 20 44 45 54 4E 55  0A00
54 4F 52 50 20 44 45 4C 49 41 46 20  0A0C
4B 43 45 48 43 20 4E 4F 49 54 43 45  0A18
                               2F 21  0A24
20 3A 44 45 54 43 45 50 58 45 2F 21  0A26   992                   ?!/EXPECTED: 'SCRATCH'  ;  ACTUAL: '!AS'!/?
3B 20 20 27 48 43 54 41 52 43 53 27  0A32
21 27 20 3A 4C 41 55 54 43 41 20 20  0A3E
                   2F 21 27 53 41  0A4A
                                  5A  09F4
                                      0A4F   993 FMT_HMB_ERROR::
43 4F 4C 42 20 45 4D 4F 48 2F 21 00' 0A4F   994           .ASCIC  ?!/HOME BLOCK READ ERROR ON !AS!/?
52 4F 52 52 45 20 44 41 45 52 20 4B  0A5B
         2F 21 53 41 21 20 4E 4F 20  0A67
                                  20  0A4F
                                      0A70   995 FMT_HDR::
20 52 45 54 53 49 47 45 52 2F 21 00' 0A70   996           .ASCIC  ?!/REGISTER   ADDRESS!_CONTENTS!_BIT DEFINITIONS!/?-
43 5F 21 53 53 45 52 44 44 41 20 20  0A7C
54 49 42 5F 21 53 54 4E 45 54 4E 4F  0A88
53 4E 4F 49 54 49 4E 49 46 45 44 20  0A94
                               2F 21  0AA0
2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D  0AA2   997                   ?------------------------------------------------------------?
2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D  0AAE
2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D  0ABA
2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D  0AC6
2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D  0AD2
                            2D 2D 2D  0ADE
                                  70  0A70
                                      0AE1   998 FMT_CS::
3A 20 20 52 53 43 43 44 49 2F 21 00' 0AE1   999           .ASCIC  ?!/IDCCSR  :  [!XL]:!_!XL(X)!_; !AC?
58 21 5F 21 3A 5D 4C 58 21 5B 20 20  0AED
   43 41 21 20 3B 5F 21 29 58 28 4C  0AF9
                                  22  0AE1
                                      0B04  1000 FMT_BA::
3A 20 20 52 41 42 43 44 49 2F 21 00' 0B04  1001           .ASCIC  ?!/IDCBAR  :  [!XL]:!_!XL(X)!_; !AC?
58 21 5F 21 3A 5D 4C 58 21 5B 20 20  0B10
   43 41 21 20 3B 5F 21 29 58 28 4C  0B1C
                                  22  0B04
                                      0B27  1002 FMT_BC::
```

```
3A 20 20 54 4E 43 43 44 49 2F 21 00' 0B27  1003          .ASCIC  ?!/IDCCNT  :  [!XL]:!_!XL(X)!_; !AC?
58 21 5F 21 3A 5D 4C 58 21 5B 20 20  0B33
   43 41 21 20 3B 5F 21 29 58 28 4C  0B3F
                                 22  0B27
                                     0B4A  1004 FMT_DA::
3A 20 20 52 41 44 43 44 49 2F 21 00' 0B4A  1005          .ASCIC  ?!/IDCDAR  :  [!XL]:!_!XL(X)!_; !AC?
58 21 5F 21 3A 5D 4C 58 21 5B 20 20  0B56
   43 41 21 20 3B 5F 21 29 58 28 4C  0B62
                                 22  0B4A
                                     0B6D  1006 FMT_MP::
3A 20 20 52 50 4D 43 44 49 2F 21 00' 0B6D  1007          .ASCIC  ?!/IDCMPR  :  [!XL]:!_!XL(X)!_; !AC?
58 21 5F 21 3A 5D 4C 58 21 5B 20 20  0B79
   43 41 21 20 3B 5F 21 29 58 28 4C  0B85
                                 22  0B6D
                                     0B90  1008 FMT_POS::
3A 20 20 53 4F 50 43 44 49 2F 21 00' 0B90  1009          .ASCIC  ?!/IDCPOS  :  [!XL]:!_!XL(X)!_; !AC?
58 21 5F 21 3A 5D 4C 58 21 5B 20 20  0B9C
   43 41 21 20 3B 5F 21 29 58 28 4C  0BA8
                                 22  0B90
                                     0BB3  1010 FMT_PAT::
3A 20 20 54 41 50 43 44 49 2F 21 00' 0BB3  1011          .ASCIC  ?!/IDCPAT  :  [!XL]:!_!XL(X)!_; !AC?
58 21 5F 21 3A 5D 4C 58 21 5B 20 20  0BBF
   43 41 21 20 3B 5F 21 29 58 28 4C  0BCB
                                 22  0BB3
                                     0BD6  1012 FMT_MP1_02::
4F 57 20 52 45 44 41 45 48 2F 21 00' 0BD6  1013          .ASCIC  ?!/HEADER WORD #1 [ADDR]!_    !XW(X)!_; !AC?
5D 52 44 44 41 5B 20 31 23 20 44 52  0BE2
29 58 28 57 58 21 20 20 20 20 5F 21  0BEE
            43 41 21 20 3B 5F 21     0BFA
                                 2A  0BD6
                                     0C01  1014 FMT_MP1_80::
4F 57 20 52 45 44 41 45 48 2F 21 00' 0C01  1015          .ASCIC  ?!/HEADER WORD #1 [CYLN]!_    !XW(X)!_; !AC?
5D 4E 4C 59 43 5B 20 31 23 20 44 52  0C0D
29 58 28 57 58 21 20 20 20 20 5F 21  0C19
            43 41 21 20 3B 5F 21     0C25
                                 2A  0C01
                                     0C2C  1016 FMT_MP2_02::
4F 57 20 52 45 44 41 45 48 2F 21 00' 0C2C  1017          .ASCIC  ?!/HEADER WORD #2 [ZERO]!_    !XW(X)!_; !AC?
5D 4F 52 45 5A 5B 20 32 23 20 44 52  0C38
29 58 28 57 58 21 20 20 20 20 5F 21  0C44
            43 41 21 20 3B 5F 21     0C50
                                 2A  0C2C
                                     0C57  1018 FMT_MP2_80::
4F 57 20 52 45 44 41 45 48 2F 21 00' 0C57  1019          .ASCIC  ?!/HEADER WORD #2 [TRK-SEC]    !XW(X)!_; !AC?
53 2D 4B 52 54 5B 20 32 23 20 44 52  0C63
58 28 57 58 21 20 20 20 20 5D 43 45  0C6F
         43 41 21 20 3B 5F 21 29     0C7B
                                 2B  0C57
                                     0C83  1020 FMT_MP3::
4F 57 20 52 45 44 41 45 48 2F 21 00' 0C83  1021          .ASCIC  ?!/HEADER WORD #3 [CRC]!_    !XW(X)!_; !AC?
21 5D 43 52 43 5B 20 33 23 20 44 52  0C8F
21 29 58 28 57 58 21 20 20 20 20 5F  0C9B
               43 41 21 20 3B 5F     0CA7
                                 29  0C83
                                     0CAD  1022 FMT_EXT_1A::
20 53 55 4F 49 56 45 52 50 2F 21 00' 0CAD  1023          .ASCIC  ?!/PREVIOUS CYLINDER WAS:!_ !3ZL(D)?
```

```
53 41 57 20 52 45 44 4E 49 4C 59 43  0CB9
   29 44 28 4C 5A 33 21 20 5F 21 3A  0CC5
                                 22  0CAD
                                     0CD0   1024 FMT_EXT_1B::
43 4E 45 52 45 46 46 49 44 2F 21 00' 0CD0   1025            .ASCIC  ?!/DIFFERENCE WORD WAS:!_!4SL(D)?
21 3A 53 41 57 20 44 52 4F 57 20 45  0CDC
            29 44 28 4C 53 34 21 5F  0CE8
                                 1F  0CD0
                                     0CF0   1026 FMT_EXT_1C::
49 54 41 4E 49 54 53 45 44 2F 21 00' 0CF0   1027            .ASCIC  ?!/DESTINATION CYLN WAS:!_ !3ZL(D)?
3A 53 41 57 20 4E 4C 59 43 20 4E 4F  0CFC
      29 44 28 4C 5A 33 21 20 5F 21  0D08
                                 21  0CF0
                                     0D12   1028 FMT_EXT_2::
43 20 54 4E 45 52 52 55 43 2F 21 00' 0D12   1029            .ASCIC  ?!/CURRENT CYLINDER IS:!_ !3ZL(D)?
21 3A 53 49 20 52 45 44 4E 49 4C 59  0D1E
            29 44 28 4C 5A 33 21 20 5F  0D2A
                                 20  0D12
                                     0D33   1030 FMT_HDR_A::
50 4D 4F 43 20 41 54 41 44 2F 21 00' 0D33   1031            .ASCIC  ?!/DATA COMPARE ERROR ON READ FROM ?-
4E 4F 20 52 4F 52 52 45 20 45 52 41  0D3F
   20 4D 4F 52 46 20 44 41 45 52 20  0D4B
21 3A 52 45 44 4E 49 4C 59 43 2F 21  0D56   1032                    ?!/CYLINDER:!ZL(10)  SURFACE:!ZL(10)  SECTOR:!ZL(10)!/?
46 52 55 53 20 20 29 30 31 28 4C 5A  0D62
20 29 30 31 28 4C 5A 21 3A 45 43 41  0D6E
28 4C 5A 21 3A 52 4F 54 43 45 53 20  0D7A
                  2F 21 29 30 31     0D86
                                 57  0D33
                                     0D8B   1033 FMT_WDCNT::
4E 55 4F 43 20 44 52 4F 57 2F 21 00' 0D8B   1034            .ASCIC  ?!/WORD COUNT WAS:!ZL(10)!/?
30 31 28 4C 5A 21 3A 53 41 57 20 54  0D97
                           2F 21 29  0DA3
                                 1A  0D8B
                                     0DA6   1035 FMT_HEADER1::
5F 21 53 53 45 52 44 44 41 2F 21 00' 0DA6   1036            .ASCIC  ?!/ADDRESS!_!_GOOD!_BAD!_XOR!_WORD# (10)!/?
21 44 41 42 5F 21 44 4F 4F 47 5F 21  0DB2
20 23 44 52 4F 57 5F 21 52 4F 58 5F  0DBE
               2F 21 29 30 31 28     0DCA
                                 29  0DA6
                                     0DD0   1037 FMT_DATA::
57 58 21 5F 21 3A 4C 58 21 2F 21 00' 0DD0   1038            .ASCIC  ?!/!XL:!_!XW!_!XW!_!XW!_!ZL?
5F 21 57 58 21 5F 21 57 58 21 5F 21  0DDC
                           4C 5A 21  0DE8
                                 1A  0DD0
                                     0DEB   1039 FMT_ERRCOUNT::
20 52 45 42 4D 55 4E 2F 21 2F 21 00' 0DEB   1040            .ASCIC  ?!/!/NUMBER OF WORDS IN ERROR = !SL!/!/?
20 4E 49 20 53 44 52 4F 57 20 46 4F  0DF7
21 4C 53 21 20 3D 20 52 4F 52 52 45  0E03
                           2F 21 2F  0E0F
                                 26  0DEB
                                     0E12   1041 FMT_SPINDLE::
45 4C 44 4E 49 50 53 2F 21 2F 21 00' 0E12   1042            .ASCIC  ?!/!/SPINDLE TIMING FOR !AS !/?
20 52 4F 46 20 47 4E 49 4D 49 54 20  0E1E
               2F 21 20 53 41 21     0E2A
                                 1D  0E12
                                     0E30   1043 FMT_SPINDLE_SP::
```

```
54 20 45 4C 44 4E 49 50 53 2F 21 00' 0E30   1044           .ASCIC  ?!/SPINDLE TIMING MUST BE IN THE RANGE !UL.!2ZL MS TO !UL.!2ZL MS!/?
42 20 54 53 55 4D 20 47 4E 49 4D 49  0E3C
4E 41 52 20 45 48 54 20 4E 49 20 45  0E48
20 4C 5A 32 21 2E 4C 55 21 20 45 47  0E54
32 21 2E 4C 55 21 20 4F 54 20 53 4D  0E60
            2F 21 53 4D 20 4C 5A     0E6C
                                 42  0E30
                                     0E73   1045 FMT_SP_TIME::
54 41 54 4F 52 20 47 56 41 2F 21 00' 0E73   1046           .ASCIC  ?!/AVG ROTATIONAL SPEED WAS !UL.!2ZL MS!/?
20 44 45 45 50 53 20 4C 41 4E 4F 49  0E7F
4C 5A 32 21 2E 4C 55 21 20 53 41 57  0E8B
                  2F 21 53 4D 20     0E97
                                 28  0E73
                                     0E9C   1047 FMT_BADBLK::
4B 43 4F 4C 42 20 44 41 42 2F 21 00' 0E9C   1048           .ASCIC  ?!/BAD BLOCK FILE IS UNREADABLE OR CORRUPTED ON !AS!/?
52 4E 55 20 53 49 20 45 4C 49 46 20  0EA8
43 20 52 4F 20 45 4C 42 41 44 41 45  0EB4
20 4E 4F 20 44 45 54 50 55 52 52 4F  0EC0
                  2F 21 53 41 21     0ECC
                                 34  0E9C
                                     0ED1   1049
                                     0ED1   1050 ;++
                                     0ED1   1051 ; MANUAL SECTION MESSAGES TO THE OPERATOR
                                     0ED1   1052 ;--
                                     0ED1   1053
                                     0ED1   1054 OPR_RETURN::
54 20 3E 54 45 52 3C 20 54 49 48 00' 0ED1   1055           .ASCIC  "HIT <RET> TO CONTINUE"
      45 55 4E 49 54 4E 4F 43 20 4F  0EDD
                                 15  0ED1
                                     0EE7   1056 OPR1_UNLOAD::
45 48 20 44 41 4F 4C 4E 55 2F 21 00' 0EE7   1057           .ASCIC  "!/UNLOAD HEADS ON !AS,!/"-
21 2C 53 41 21 20 4E 4F 20 53 44 41  0EF3
                                 2F  0EFF
43 41 50 20 52 4F 46 20 54 49 41 57  0F00   1058                   "WAIT FOR PACK TO STOP AND RUN STOP SWITCH TO LIGHT!/"-
4E 41 20 50 4F 54 53 20 4F 54 20 4B  0F0C
53 20 50 4F 54 53 20 4E 55 52 20 44  0F18
47 49 4C 20 4F 54 20 48 43 54 49 57  0F24
                     2F 21 54 48     0F30
4F 4C 20 45 54 49 52 57 20 54 45 53  0F34   1059                   "SET WRITE LOCK SWITCH, IF DRIVE IS AN RL02 THEN OPEN!/"-
49 20 2C 48 43 54 49 57 53 20 4B 43  0F40
41 20 53 49 20 45 56 49 52 44 20 46  0F4C
20 4E 45 48 54 20 32 30 4C 52 20 4E  0F58
            2F 21 4E 45 50 4F        0F64
   2E 4F 53 4C 41 20 52 45 56 4F 43  0F6A   1060                   "COVER ALSO."
                                 8D  0EE7
                                     0F75   1061 OPR2_CLOSE::
56 4F 43 20 45 53 4F 4C 43 2F 21 00' 0F75   1062           .ASCIC  "!/CLOSE COVER (IF RL02) AND RESET WRITE LOCK SWITCH ON !AS, "
29 32 30 4C 52 20 46 49 28 20 52 45  0F81
57 20 54 45 53 45 52 20 44 4E 41 20  0F8D
57 53 20 4B 43 4F 4C 20 45 54 49 52  0F99
2C 53 41 21 20 4E 4F 20 48 43 54 49  0FA5
                                 20  0FB1
                                 3C  0F75
                                     0FB2   1063 OPR3_START::
44 41 4F 4C 20 48 53 55 50 2F 21 00' 0FB2   1064           .ASCIC  "!/PUSH LOAD SWITCH ON !AS, TO LOAD THE HEADS"
21 20 4E 4F 20 48 43 54 49 57 53 20  0FBE
```

```
20 44 41 4F 4C 20 4F 54 20 2C 53 41   0FCA
         53 44 41 45 48 20 45 48 54   0FD6
                                  2C   0FB2
                                       0FDF   1065 OPR4_MKRDY::
44 41 45 48 20 44 41 4F 4C 2F 21 00'  0FDF   1066          .ASCIC  "!/LOAD HEADS ON !AS, WAIT FOR READY LITE TO GO ON, "
41 57 20 2C 53 41 21 20 4E 4F 20 53   0FEB
59 44 41 45 52 20 52 4F 46 20 54 49   0FF7
20 4F 47 20 4F 54 20 45 54 49 4C 20   1003
                     20 2C 4E 4F      100F
                                  33   0FDF
                                       1013   1067 OPR5_UNLOAD::
4C 20 53 53 45 52 50 45 44 2F 21 00'  1013   1068          .ASCIC  "!/DEPRESS LOAD SWITCH ON !AS TO UNLOAD HEADS, "
4F 20 48 43 54 49 57 53 20 44 41 4F   101F
4C 4E 55 20 4F 54 20 53 41 21 20 4E   102B
   20 2C 53 44 41 45 48 20 44 41 4F   1037
                                  2E   1013
                                       1042   1069 OPR6_REMOVE::
4E 55 20 45 56 4F 4D 45 52 2F 21 00'  1042   1070          .ASCIC  "!/REMOVE UNIT SELECT PLUGS FROM ALL DRIVES, "
4C 50 20 54 43 45 4C 45 53 20 54 49   104E
4C 4C 41 20 4D 4F 52 46 20 53 47 55   105A
         20 2C 53 45 56 49 52 44 20   1066
                                  2C   1042
                                       106F   1071 OPR7_INSERT::
4E 55 20 54 52 45 53 4E 49 2F 21 00'  106F   1072          .ASCIC  "!/INSERT UNIT PLUG #!ZL IN !AS, "
4C 5A 21 23 20 47 55 4C 50 20 54 49   107B
         20 2C 53 41 21 20 4E 49 20   1087
                                  20   106F
                                       1090   1073 OPR8_SETUP::
4E 55 20 4E 52 55 54 45 52 2F 21 00'  1090   1074          .ASCIC  "!/RETURN UNIT SELECT PLUGS TO ORIGIONAL STATE, "
4C 50 20 54 43 45 4C 45 53 20 54 49   109C
49 47 49 52 4F 20 4F 54 20 53 47 55   10A8
20 2C 45 54 41 54 53 20 4C 41 4E 4F   10B4
                                  2F   1090
                                       10C0   1075 OPR9_DUPPLUG::
54 53 45 54 20 53 49 48 54 2F 21 00'  10C0   1076          .ASCIC  "!/THIS TEST REQUIRES A DUPLICATE UNIT SELECT #0 PLUG"-
20 41 20 53 45 52 49 55 51 45 52 20   10CC
4E 55 20 45 54 41 43 49 4C 50 55 44   10D8
30 23 20 54 43 45 4C 45 53 20 54 49   10E4
                  47 55 4C 50 20      10F0
41 45 4C 20 54 41 20 44 4E 41 2F 21   10F5   1077                  "!/AND AT LEAST TWO RL02 DRIVES CONNECTED TO THE"-
20 32 30 4C 52 20 4F 57 54 20 54 53   1101
45 4E 4E 4F 43 20 53 45 56 49 52 44   110D
   45 48 54 20 4F 54 20 44 45 54 43   1119
52 54 4E 4F 43 20 45 4D 41 53 2F 21   1124   1078                  "!/SAME CONTROLLER, "
            20 2C 52 45 4C 4C 4F      1130
                                  76   10C0
                                       1137   1079 OPR_AVAIL::
56 41 20 59 45 48 54 20 45 52 41 00'  1137   1080          .ASCIC  "ARE THEY AVAILABLE"
            45 4C 42 41 4C 49 41      1143
                                  12   1137
                                       114A   1081 OPR10_INS_ID::
4C 41 20 45 56 4F 4D 45 52 2F 21 00'  114A   1082          .ASCIC  "!/REMOVE ALL UNIT PLUGS FROM DRIVES AND INSTALL"-
53 47 55 4C 50 20 54 49 4E 55 20 4C   1156
53 45 56 49 52 44 20 4D 4F 52 46 20   1162
4C 4C 41 54 53 4E 49 20 44 4E 41 20   116E
55 4C 50 20 54 49 4E 55 20 41 2F 21   117A   1083                  "!/A UNIT PLUG #0 INTO !AS AND A UNIT PLUG #1"-
```

```
41 21 20 4F 54 4E 49 20 30 23 20 47  1186
54 49 4E 55 20 41 20 44 4E 41 20 53  1192
         31 23 20 47 55 4C 50 20  119E
4F 20 45 4E 4F 20 4F 54 4E 49 2F 21  11A6  1084                     "!/INTO ONE OTHER DRIVE, "
20 2C 45 56 49 52 44 20 52 45 48 54  11B2
                                 73  114A
                                     11BE  1085 OPR11_ORIGIN::
4E 49 20 45 53 41 45 4C 50 2F 21 00' 11BE  1086         .ASCIC   "!/PLEASE INSURE THAT DRIVES ARE POWERED UP, HEADS LOADED"-
52 44 20 54 41 48 54 20 45 52 55 53  11CA
57 4F 50 20 45 52 41 20 53 45 56 49  11D6
41 45 48 20 2C 50 55 20 44 45 52 45  11E2
         44 45 44 41 4F 4C 20 53 44  11EE
53 20 54 49 4E 55 20 44 4E 41 2F 21  11F7  1087                     "!/AND UNIT SELECT PLUGS ARE IN PROPER DRIVES, "
20 53 47 55 4C 50 20 54 43 45 4C 45  1203
45 50 4F 52 50 20 4E 49 20 45 52 41  120F
      20 2C 53 45 56 49 52 44 20 52  121B
                                 66  11BE
                                     1225  1088 OPR12_WRTLOCK::
45 54 49 52 57 20 54 45 53 2F 21 00' 1225  1089         .ASCIC   "!/SET WRITE LOCK SWITCH ON !AS, "
48 43 54 49 57 53 20 4B 43 4F 4C 20  1231
      20 2C 53 41 21 20 4E 4F 20  123D
                                 20  1225
                                     1246  1090 OPR13_RST_WL::
49 52 57 20 54 45 53 45 52 2F 21 00' 1246  1091         .ASCIC   "!/RESET WRITE LOCK SWITCH ON !AS, "
54 49 57 53 20 4B 43 4F 4C 20 45 54  1252
   20 2C 53 41 21 20 4E 4F 20 48 43  125E
                                 22  1246
                                     1269  1092 OPR13A_RST_FLT::
55 41 46 20 54 45 53 45 52 2F 21 00' 1269  1093         .ASCIC   "!/RESET FAULT FROM SWITCH PANEL"
54 49 57 53 20 4D 4F 52 46 20 54 4C  1275
         4C 45 4E 41 50 20 48 43  1281
                                 1F  1269
                                     1289  1094 OPR14_ROTATE::
4E 49 54 4E 4F 43 20 4F 54 2F 21 00' 1289  1095         .ASCIC   "!/TO CONTINUE OVERWRITE TESTING ROTATE PACKS ON DRIVES"-
45 54 49 52 57 52 45 56 4F 20 45 55  1295
54 4F 52 20 47 4E 49 54 53 45 54 20  12A1
4E 4F 20 53 4B 43 41 50 20 45 54 41  12AD
         53 45 56 49 52 44 20  12B9
54 53 45 54 20 52 45 44 4E 55 2F 21  12C0  1096                     "!/UNDER TEST AND START/SEC:OVERWRITE"
53 2F 54 52 41 54 53 20 44 4E 41 20  12CC
45 54 49 52 57 52 45 56 4F 3A 43 45  12D8
                                 5A  1289
                                     12E4  1097 OPR15_SWAP::
4C 50 20 45 56 4F 4D 45 52 2F 21 00' 12E4  1098         .ASCIC   "!/REMOVE PLUG #1 AND INSTALL PLUG #0, "
4E 49 20 44 4E 41 20 31 23 20 47 55  12F0
23 20 47 55 4C 50 20 4C 4C 41 54 53  12FC
                           20 2C 30  1308
                                 26  12E4
                                     130B  1099 OPR16_SWAP_BK::
4E 55 20 45 56 4F 4D 45 52 2F 21 00' 130B  1100         .ASCIC   "!/REMOVE UNIT PLUG #0 FROM THE DRIVE NOT UNDER TEST"-
46 20 30 23 20 47 55 4C 50 20 54 49  1317
56 49 52 44 20 45 48 54 20 4D 4F 52  1323
20 52 45 44 4E 55 20 54 4F 4E 20 45  132F
                     54 53 45 54  133B
48 54 20 54 55 50 20 44 4E 41 2F 21  133F  1101                     "!/AND PUT THE UNIT PLUG #1 BACK IN, "
20 47 55 4C 50 20 54 49 4E 55 20 45  134B
```

```
20 2C 4E 49 20 4B 43 41 42 20 31 23   1357
                                  57   130B
                                       1363  1102 OPR_FAULT::
54 49 4C 20 54 4C 55 41 46 2F 21 00'  1363  1103          .ASCIC  "!/FAULT LITE SHOULD BE LIT ON !AS OR THE OTHER DRIVE!/"
20 45 42 20 44 4C 55 4F 48 53 20 45   136F
4F 20 53 41 21 20 4E 4F 20 54 49 4C   137B
20 52 45 48 54 4F 20 45 48 54 20 52   1387
               2F 21 45 56 49 52 44   1393
                                  36   1363
                                       139A  1104
                                       139A  1105
                                       139A  1106 ;++
                                       139A  1107 ; PROGRAM WIDE SYSTEM ERROR MESSAGES
                                       139A  1108 ;--
                                       139A  1109
53 52 4F 53 49 56 52 45 50 55 53 00'  139A  1110 IDC_ADRFAIL::   .ASCIC  "SUPERVISORS PROBE OF DEVICE ADDRESS FAILED"
45 44 20 46 4F 20 45 42 4F 52 50 20   13A6
53 53 45 52 44 44 41 20 45 43 49 56   13B2
               44 45 4C 49 41 46 20   13BE
                                  2A   139A
53 52 4F 53 49 56 52 45 50 55 53 00'  13C5  1111 IDC_SADRFAIL::  .ASCIC  "SUPERVISORS PROBE OF SECONDARY ADDRESS FAILED"
45 53 20 46 4F 20 45 42 4F 52 50 20   13D1
52 44 44 41 20 59 52 41 44 4E 4F 43   13DD
      44 45 4C 49 41 46 20 53 53 45   13E9
                                  2D   13C5
45 53 45 52 20 53 55 42 49 4E 55 00'  13F3  1112 UNBINIT_ERR::   .ASCIC  "UNIBUS RESET ERROR"
               52 4F 52 52 45 20 54   13FF
                                  12   13F3
47 4E 49 50 50 41 4D 20 41 42 55 00'  1406  1113 UBAMAP_ERR::    .ASCIC  "UBA MAPPING ERROR"
                  52 4F 52 52 45 20   1412
                                  11   1406
45 20 45 47 52 55 50 20 41 42 55 00'  1418  1114 UBAPURGE_ERR::  .ASCIC  "UBA PURGE ERROR"
                        52 4F 52 52   1424
                                  0F   1418
45 20 52 41 45 4C 43 20 41 42 55 00'  1428  1115 UBACLEAR_ERR::  .ASCIC  "UBA CLEAR ERROR"
                        52 4F 52 52   1434
                                  0F   1428
4C 41 49 54 49 4E 49 20 41 42 55 00'  1438  1116 UBAINIT_ERR::   .ASCIC  "UBA INITIALIZE ERROR"
         52 4F 52 52 45 20 45 5A 49   1444
                                  14   1438
52 55 43 43 4F 20 52 4F 52 52 45 00'  144D  1117 IDCENINT_ERR::  .ASCIC  "ERROR OCCURED ENABLING INTERRUPTS"
20 47 4E 49 4C 42 41 4E 45 20 44 45   1459
      53 54 50 55 52 52 45 54 4E 49   1465
                                  21   144D
52 55 43 43 4F 20 52 4F 52 52 45 00'  146F  1118 IDCDSINT_ERR::  .ASCIC  "ERROR OCCURED DISABLING INTERRUPTS"
47 4E 49 4C 42 41 53 49 44 20 44 45   147B
   53 54 50 55 52 52 45 54 4E 49 20   1487
                                  22   146F
4F 20 4F 54 20 45 4C 42 41 4E 55 00'  1492  1119 IDCGETBUF::     .ASCIC  "UNABLE TO OBTAIN 20 PAGES OF MEMORY ABORTING"-
47 41 50 20 30 32 20 4E 49 41 54 42   149E
59 52 4F 4D 45 4D 20 46 4F 20 53 45   14AA
      47 4E 49 54 52 4F 42 41 20      14B6
41 54 41 44 20 4D 55 4D 49 58 41 4D   14BF  1120                   "MAXIMUM DATA TRANSFER TEST"
45 54 20 52 45 46 53 4E 41 52 54 20   14CB
                              54 53   14D7
                                  46   1492
```

```
                                     14D9   1121
                                     14D9   1122 ;++
                                     14D9   1123 ; PROGRAM WIDE DEVICE ERROR MESSAGES
                                     14D9   1124 ;--
                                     14D9   1125
4E 4C 59 43 20 52 52 55 43 2F 21 00' 14D9   1126 MSG_TRACE::      .ASCIC  "!/CURR CYLN = !ZL!_HEAD = !ZL!/"
44 41 45 48 5F 21 4C 5A 21 20 3D 20  14E5
      2F 21 4C 5A 21 20 3D 20        14F1
                                1F   14D9
                  32 30 4C 52 00'    14F9   1127 MSG_RL02::       .ASCIC  "RL02"
                                04   14F9
                     30 38 52 00'    14FE   1128 MSG_R80::        .ASCIC  "R80"
                                03   14FE
                     45 4E 4F 00'    1502   1129 MSG_ONE::        .ASCIC  /ONE/
                                03   1502
         4D 55 4D 49 58 41 4D 00'    1506   1130 MSG_MAX::        .ASCIC  /MAXIMUM/
                                07   1506
         45 47 41 52 45 56 41 00'    150E   1131 MSG_AVG::        .ASCIC  /AVERAGE/
                                07   150E
      29 20 52 45 54 55 4F 28 00'    1516   1132 MSG_OUTER::      .ASCIC  /(OUTER )/
                                08   1516
      29 45 4C 44 44 49 4D 28 00'    151F   1133 MSG_MIDDLE::     .ASCIC  /(MIDDLE)/
                                08   151F
      29 20 52 45 4E 4E 49 28 00'    1528   1134 MSG_INNER::      .ASCIC  /(INNER )/
                                08   1528
      29 2D 2D 2D 2D 2D 2D 28 00'    1531   1135 MSG_BLANK::      .ASCIC  /(------)/
                                08   1531
                                     153A   1136 MSG_ABORT::
54 4E 41 57 20 55 4F 59 20 4F 44 00' 153A   1137              .ASCIC  \DO YOU WANT TO CONTINUE THE TEST\
45 55 4E 49 54 4E 4F 43 20 4F 54 20  1546
         54 53 45 54 20 45 48 54 20  1552
                                20   153A
                                     155B   1138 MSG_OPERATTCH::
45 44 20 52 4F 54 41 52 45 50 4F 00' 155B   1139                  .ASCIC  /OPERATOR DEVICE SELECTION VS HARDWARE ID ERROR/
49 54 43 45 4C 45 53 20 45 43 49 56  1567
41 57 44 52 41 48 20 53 56 20 4E 4F  1573
   52 4F 52 52 45 20 44 49 20 45 52  157F
                                2E   155B
52 4F 52 52 45 20 4C 41 54 41 46 00' 158A   1140 MSG_DRVFATAL::   .ASCIC  /FATAL ERROR DETECTED IN DRIVE STATUS IDCMPR/
4E 49 20 44 45 54 43 45 54 45 44 20  1596
55 54 41 54 53 20 45 56 49 52 44 20  15A2
         52 50 4D 43 44 49 20 53     15AE
                                2B   158A
4F 43 20 53 41 57 20 41 54 41 44 00' 15B6   1141 MSG_ECCPASS::    .ASCIC  /DATA WAS CORRECTED BY ECC PROCESSING/
45 20 59 42 20 44 45 54 43 45 52 52  15C2
4E 49 53 53 45 43 4F 52 50 20 43 43  15CE
                                47   15DA
                                24   15B6
46 20 43 49 47 4F 4C 20 43 43 45 00' 15DB   1142 MSG_ECCFAIL::    .ASCIC  /ECC LOGIC FAILURE - CORRECTABLE ERROR FAILED TO CORRECT/
52 4F 43 20 2D 20 45 52 55 4C 49 41  15E7
52 52 45 20 45 4C 42 41 54 43 45 52  15F3
4F 54 20 44 45 4C 49 41 46 20 52 4F  15FF
         54 43 45 52 52 4F 43 20     160B
                                37   15DB
54 4F 4E 20 44 49 44 20 43 44 49 00' 1613   1143 MSG1_ADR_ERR::   .ASCIC  /IDC DID NOT RESPOND TO ADDRESSING/
20 4F 54 20 44 4E 4F 50 53 45 52 20  161F
```

```
      47 4E 49 53 53 45 52 44 44 41   162B
                                 21   1613
47 45 52 20 47 4E 49 54 49 52 57 00'  1635   1144 MSG2_RG_ZERO::  .ASCIC  /WRITING REGISTER WITH 0'S/
30 20 48 54 49 57 20 52 45 54 53 49   1641
                              53 27   164D
                                 19   1635
47 45 52 20 47 4E 49 54 49 52 57 00'  164F   1145 MSG3_RG_ONE::   .ASCIC  /WRITING REGISTER WITH 1'S/
31 20 48 54 49 57 20 52 45 54 53 49   165B
                              53 27   1667
                                 19   164F
47 45 52 20 47 4E 49 54 49 52 57 00'  1669   1146 MSG4_RG_FLT1::  .ASCIC  /WRITING REGISTER WITH FLOATING 1'S/
46 20 48 54 49 57 20 52 45 54 53 49   1675
   53 27 31 20 47 4E 49 54 41 4F 4C   1681
                                 22   1669
47 45 52 20 47 4E 49 54 49 52 57 00'  168C   1147 MSG5_RG_FLT0::  .ASCIC  /WRITING REGISTER WITH FLOATING 0'S/
46 20 48 54 49 57 20 52 45 54 53 49   1698
   53 27 30 20 47 4E 49 54 41 4F 4C   16A4
                                 22   168C
49 47 45 52 20 52 53 43 43 44 49 00'  16AF   1148 MSG6_CS_FAIL::  .ASCIC  /IDCCSR REGISTER DATA COMPARE FAILURE/
4F 43 20 41 54 41 44 20 52 45 54 53   16BB
52 55 4C 49 41 46 20 45 52 41 50 4D   16C7
                                 45   16D3
                                 24   16AF
49 47 45 52 20 52 41 42 43 44 49 00'  16D4   1149 MSG7_BA_FAIL::  .ASCIC  /IDCBAR REGISTER DATA COMPARE FAILURE/
4F 43 20 41 54 41 44 20 52 45 54 53   16E0
52 55 4C 49 41 46 20 45 52 41 50 4D   16EC
                                 45   16F8
                                 24   16D4
49 47 45 52 20 54 4E 43 43 44 49 00'  16F9   1150 MSG7A_BC_FAIL:: .ASCIC  /IDCCNT REGISTER DATA COMPARE FAILURE/
4F 43 20 41 54 41 44 20 52 45 54 53   1705
52 55 4C 49 41 46 20 45 52 41 50 4D   1711
                                 45   171D
                                 24   16F9
49 47 45 52 20 52 41 44 43 44 49 00'  171E   1151 MSG8_DA_FAIL::  .ASCIC  /IDCDAR REGISTER DATA COMPARE FAILURE/
4F 43 20 41 54 41 44 20 52 45 54 53   172A
52 55 4C 49 41 46 20 45 52 41 50 4D   1736
                                 45   1742
                                 24   171E
49 47 45 52 20 52 50 4D 43 44 49 00'  1743   1152 MSG8A_MP_FAIL:: .ASCIC  /IDCMPR REGISTER DATA COMPARE FAILURE/
4F 43 20 41 54 41 44 20 52 45 54 53   174F
52 55 4C 49 41 46 20 45 52 41 50 4D   175B
                                 45   1767
                                 24   1743
44 41 45 52 20 44 45 55 53 53 49 00'  1768   1153 MSG9_RG_RMW::   .ASCIC  /ISSUED READ-MODIFY-WRITE (BIT SET) TO REGISTER/
54 49 52 57 2D 59 46 49 44 4F 4D 2D   1774
20 29 54 45 53 20 54 49 42 28 20 45   1780
   52 45 54 53 49 47 45 52 20 4F 54   178C
                                 2E   1768
20 53 27 31 20 44 45 44 41 4F 4C 00'  1797   1154 MSG10_RG_INIT:: .ASCIC  /LOADED 1'S AND ISSUED BUS INIT TO REGISTER/
42 20 44 45 55 53 53 49 20 44 4E 41   17A3
52 20 4F 54 20 54 49 4E 49 20 53 55   17AF
                  52 45 54 53 49 47 45   17BB
                                 2A   1797
52 44 20 44 45 54 55 43 45 58 45 00'  17C2   1155 MSG11_GS_CRDY:: .ASCIC  /EXECUTED DRIVE RESET - CRDY DID NOT SET OR WAS LATE/
20 2D 20 54 45 53 45 52 20 45 56 49   17CE
54 4F 4E 20 44 49 44 20 59 44 52 43   17DA
```

```
20 53 41 57 20 52 4F 20 54 45 53 20  17E6
                     45 54 41 4C  17F2
                              33  17C2
59 44 41 45 52 20 45 56 49 52 44 00' 17F6   1156 MSG12_GS_DRDY:: .ASCIC  /DRIVE READY FAILED TO SET/
53 20 4F 54 20 44 45 4C 49 41 46 20  1802
                           54 45  180E
                              19  17F6
45 52 20 29 53 28 52 4F 52 52 45 00' 1810   1157 MSG13_INI_ERR:: .ASCIC  /ERROR(S) REMAIN PRESENT FOLLOWING DRIVE RESET/
54 4E 45 53 45 52 50 20 4E 49 41 4D  181C
44 20 47 4E 49 57 4F 4C 4C 4F 46 20  1828
      54 45 53 45 52 20 45 56 49 52  1834
                              2D  1810
52 55 43 43 4F 20 52 4F 52 52 45 00' 183E   1158 MSG14_SYSINIT:: .ASCIC  /ERROR OCCURED WHILE INITIALIZING SUBSYSTEM/
49 4E 49 20 45 4C 49 48 57 20 44 45  184A
55 53 20 47 4E 49 5A 49 4C 41 49 54  1856
               4D 45 54 53 59 53 42  1862
                              2A  183E
52 55 43 43 4F 20 52 4F 52 52 45 00' 1869   1159 MSG15_NOOP::    .ASCIC  /ERROR OCCURED EXECUTING NOOP COMMAND/
47 4E 49 54 55 43 45 58 45 20 44 45  1875
4E 41 4D 4D 4F 43 20 50 4F 4F 4E 20  1881
                              44  188D
                              24  1869
52 55 43 43 4F 20 52 4F 52 52 45 00' 188E   1160 MSG16_WRTCHK::  .ASCIC  /ERROR OCCURED EXECUTING WRITE CHECK COMMAND/
47 4E 49 54 55 43 45 58 45 20 44 45  189A
4B 43 45 48 43 20 45 54 49 52 57 20  18A6
            44 4E 41 4D 4D 4F 43 20  18B2
                              2B  188E
52 55 43 43 4F 20 52 4F 52 52 45 00' 18BA   1161 MSG16A_WRTCHK:: .ASCIC  /ERROR OCCURED WRITE CHECKING TWO SECTORS/
45 48 43 20 45 54 49 52 57 20 44 45  18C6
45 53 20 4F 57 54 20 47 4E 49 4B 43  18D2
                     53 52 4F 54 43  18DE
                              28  18BA
52 55 43 43 4F 20 52 4F 52 52 45 00' 18E3   1162 MSG16B_WRTCHK:: .ASCIC  /ERROR OCCURED WRITE CHECKING WITH MAXIMUM WORD COUNT/
45 48 43 20 45 54 49 52 57 20 44 45  18EF
4D 20 48 54 49 57 20 47 4E 49 4B 43  18FB
20 44 52 4F 57 20 4D 55 4D 49 58 41  1907
                     54 4E 55 4F 43  1913
                              34  18E3
52 55 43 43 4F 20 52 4F 52 52 45 00' 1918   1163 MSG17_GETSTAT:: .ASCIC  /ERROR OCCURED EXECUTING GET STATUS COMMAND/
47 4E 49 54 55 43 45 58 45 20 44 45  1924
20 53 55 54 41 54 53 20 54 45 47 20  1930
            44 4E 41 4D 4D 4F 43  193C
                              2A  1918
52 55 43 43 4F 20 52 4F 52 52 45 00' 1943   1164 MSG18_SEEK::    .ASCIC  /ERROR OCCURED EXECUTING SEEK COMMAND/
47 4E 49 54 55 43 45 58 45 20 44 45  194F
4E 41 4D 4D 4F 43 20 4B 45 45 53 20  195B
                              44  1967
                              24  1943
52 55 43 43 4F 20 52 4F 52 52 45 00' 1968   1165 MSG18B_SEEK::   .ASCIC  /ERROR OCCURED DURING OVERLAP SEEK/
56 4F 20 47 4E 49 52 55 44 20 44 45  1974
      4B 45 45 53 20 50 41 4C 52 45  1980
                              21  1968
52 55 43 43 4F 20 52 4F 52 52 45 00' 198A   1166 MSG19_READH::   .ASCIC  /ERROR OCCURED EXECUTING READ HEADER COMMAND/
47 4E 49 54 55 43 45 58 45 20 44 45  1996
52 45 44 41 45 48 20 44 41 45 52 20  19A2
            44 4E 41 4D 4D 4F 43 20  19AE
```

```
                                  2B  198A
52 45 44 41 45 48 20 44 41 45 52 00'  19B6   1167 MSG19A_READH::  .ASCIC  /READ HEADER CMD FAILED FOLLOWING SEEK INTO INNER GUARD BA -
cont> ND/
20 44 45 4C 49 41 46 20 44 4D 43 20   19C2
45 53 20 47 4E 49 57 4F 4C 4C 4F 46   19CE
45 4E 4E 49 20 4F 54 4E 49 20 4B 45   19DA
44 4E 41 42 20 44 52 41 55 47 20 52   19E6
                                  3B  19B6
52 55 43 43 4F 20 52 4F 52 52 45 00'  19F2   1168 MSG20_WRITE::   .ASCIC  /ERROR OCCURED EXECUTING WRITE DATA COMMAND/
47 4E 49 54 55 43 45 58 45 20 44 45   19FE
20 41 54 41 44 20 45 54 49 52 57 20   1A0A
            44 4E 41 4D 4D 4F 43      1A16
                                  2A  19F2
45 4C 49 41 46 20 45 54 49 52 57 00'  1A1D   1169 MSG20_WRT_BND:: .ASCIC  /WRITE FAILED WHILE OVERLAPPING PAGE BOUNDRIES/
52 45 56 4F 20 45 4C 49 48 57 20 44   1A29
45 47 41 50 20 47 4E 49 50 50 41 4C   1A35
      53 45 49 52 44 4E 55 4F 42 20   1A41
                                  2D  1A1D
52 55 43 43 4F 20 52 4F 52 52 45 00'  1A4B   1170 MSG20A_WRITE::  .ASCIC  /ERROR OCCURED DURING A 2 SECTOR WRITE DATA COMMAND/
20 41 20 47 4E 49 52 55 44 20 44 45   1A57
49 52 57 20 52 4F 54 43 45 53 20 32   1A63
4D 4D 4F 43 20 41 54 41 44 20 45 54   1A6F
                        44 4E 41      1A7B
                                  32  1A4B
4E 49 52 55 44 20 52 4F 52 52 45 00'  1A7E   1171 MSG20A_FORMAT:: .ASCIC  /ERROR DURING WRITE FORMAT OPERATION/
4D 52 4F 46 20 45 54 49 52 57 20 47   1A8A
4E 4F 49 54 41 52 45 50 4F 20 54 41   1A96
                                  23  1A7E
52 55 43 43 4F 20 52 4F 52 52 45 00'  1AA2   1172 MSG20B_WRITE::  .ASCIC  /ERROR OCCURED DURING WRITE DATA WITH MAXIMUM WORD COUNT/
52 57 20 47 4E 49 52 55 44 20 44 45   1AAE
54 49 57 20 41 54 41 44 20 45 54 49   1ABA
4F 57 20 4D 55 4D 49 58 41 4D 20 48   1AC6
         54 4E 55 4F 43 20 44 52      1AD2
                                  37  1AA2
41 52 45 50 4F 20 45 54 49 52 57 00'  1ADA   1173 MSG20C_WRITE::  .ASCIC  /WRITE OPERATION FAILED FOLLOWING FORMATTING TRACK/
20 44 45 4C 49 41 46 20 4E 4F 49 54   1AE6
4F 46 20 47 4E 49 57 4F 4C 4C 4F 46   1AF2
41 52 54 20 47 4E 49 54 54 41 4D 52   1AFE
                              4B 43   1B0A
                                  31  1ADA
52 55 43 43 4F 20 52 4F 52 52 45 00'  1B0C   1174 MSG21_READ::    .ASCIC  /ERROR OCCURED EXECUTING READ DATA COMMAND/
47 4E 49 54 55 43 45 58 45 20 44 45   1B18
43 20 41 54 41 44 20 44 41 45 52 20   1B24
            44 4E 41 4D 4D 4F         1B30
                                  29  1B0C
52 55 43 43 4F 20 52 4F 52 52 45 00'  1B36   1175 MSG21A_READ::   .ASCIC  /ERROR OCCURED DURING A 2 SECTOR READ DATA COMMAND/
20 41 20 47 4E 49 52 55 44 20 44 45   1B42
41 45 52 20 52 4F 54 43 45 53 20 32   1B4E
41 4D 4D 4F 43 20 41 54 41 44 20 44   1B5A
                              44 4E   1B66
                                  31  1B36
49 44 41 45 52 20 52 4F 52 52 45 00'  1B68   1176 MSG21A_READH::  .ASCIC  /ERROR READING HEADERS AFTER FORMAT OPERATION/
41 20 53 52 45 44 41 45 48 20 47 4E   1B74
20 54 41 4D 52 4F 46 20 52 45 54 46   1B80
         4E 4F 49 54 41 52 45 50 4F   1B8C
                                  2C  1B68
52 55 43 43 4F 20 52 4F 52 52 45 00'  1B95   1177 MSG21B_READ::   .ASCIC  /ERROR OCCURED DURING READ DATA WITH MAXIMUM WORD COUNT/
```

```
45 52 20 47 4E 49 52 55 44 20 44 45   1BA1
48 54 49 57 20 41 54 41 44 20 44 41   1BAD
52 4F 57 20 4D 55 4D 49 58 41 4D 20   1BB9
            54 4E 55 4F 43 20 44   1BC5
                              36   1B95
52 55 43 43 4F 20 52 4F 52 52 45 00'  1BCC   1178 MSG22_READNHC:: .ASCIC  /ERROR OCCURED EXECUTING READ WITH NO HEADER COMPARE CMD/
47 4E 49 54 55 43 45 58 45 20 44 45   1BD8
4E 20 48 54 49 57 20 44 41 45 52 20   1BE4
4D 4F 43 20 52 45 44 41 45 48 20 4F   1BF0
            44 4D 43 20 45 52 41 50   1BFC
                              37   1BCC
44 4E 41 20 4C 4F 52 54 4E 4F 43 00'  1C04   1179 MSG23_CS_INC::  .ASCIC  /CONTROL AND STATUS REGISTER INCORRECT/
49 47 45 52 20 53 55 54 41 54 53 20   1C10
45 52 52 4F 43 4E 49 20 52 45 54 53   1C1C
                           54 43   1C28
                              25   1C04
49 47 45 52 20 52 41 42 43 44 49 00'  1C2A   1180 MSG24_BA_MOD::  .ASCIC  /IDCBAR REGISTER WAS MODIFIED/
44 4F 4D 20 53 41 57 20 52 45 54 53   1C36
                  44 45 49 46 49   1C42
                              1C   1C2A
49 47 45 52 20 54 4E 43 43 44 49 00'  1C47   1181 MSG24A_BC_MOD:: .ASCIC  /IDCCNT REGISTER WAS MODIFIED/
44 4F 4D 20 53 41 57 20 52 45 54 53   1C53
                  44 45 49 46 49   1C5F
                              1C   1C47
49 47 45 52 20 52 41 44 43 44 49 00'  1C64   1182 MSG25_DA_MOD::  .ASCIC  /IDCDAR REGISTER WAS MODIFIED/
44 4F 4D 20 53 41 57 20 52 45 54 53   1C70
                  44 45 49 46 49   1C7C
                              1C   1C64
49 47 45 52 20 52 50 4D 43 44 49 00'  1C81   1183 MSG26_MP_MOD::  .ASCIC  /IDCMPR REGISTER WAS MODIFIED/
44 4F 4D 20 53 41 57 20 52 45 54 53   1C8D
                  44 45 49 46 49   1C99
                              1C   1C81
4F 50 52 55 50 20 49 54 4C 55 4D 00'  1C9E   1184 MSG27_MP_INC::  .ASCIC  /MULTI PURPOSE REGISTER INCORRECT/
20 52 45 54 53 49 47 45 52 20 45 53   1CAA
         54 43 45 52 52 4F 43 4E 49   1CB6
                              20   1C9E
53 45 52 44 44 41 20 4B 53 49 44 00'  1CBF   1185 MSG28_DA_INC::  .ASCIC  /DISK ADDRESS REGISTER INCORRECT/
49 20 52 45 54 53 49 47 45 52 20 53   1CCB
         54 43 45 52 52 4F 43 4E   1CD7
                              1F   1CBF
53 53 45 52 44 44 41 20 53 55 42 00'  1CDF   1186 MSG29_BA_INC::  .ASCIC  /BUS ADDRESS REGISTER INCORRECT/
4E 49 20 52 45 54 53 49 47 45 52 20   1CEB
            54 43 45 52 52 4F 43   1CF7
                              1E   1CDF
20 52 45 4C 4C 4F 52 54 4E 4F 43 00'  1CFE   1187 MSG30_CRDY::    .ASCIC  /CONTROLLER READY FAILED TO SET/
44 45 4C 49 41 46 20 59 44 41 45 52   1D0A
            54 45 53 20 4F 54 20   1D16
                              1E   1CFE
47 4E 49 50 50 41 4C 52 45 56 4F 00'  1D1D   1188 MSG30A_CRDY::   .ASCIC  /OVERLAPPING SEEKS - CRDY FAILED TO SET FOR OTHER DRIVE/
44 52 43 20 2D 20 53 4B 45 45 53 20   1D29
20 4F 54 20 44 45 4C 49 41 46 20 59   1D35
45 48 54 4F 20 52 4F 46 20 54 45 53   1D41
            45 56 49 52 44 20 52   1D4D
                              36   1D1D
59 44 41 45 52 20 45 56 49 52 44 00'  1D54   1189 MSG31_DRDY::    .ASCIC  /DRIVE READY FAILED TO SET/
53 20 4F 54 20 44 45 4C 49 41 46 20   1D60
```

```
                              54 45  1D6C
                                 19  1D54
20 52 45 4C 4C 4F 52 54 4E 4F 43 00' 1D6E   1190 MSG32_CERR::     .ASCIC  /CONTROLLER ERROR DETECTED/
54 43 45 54 45 44 20 52 4F 52 52 45  1D7A
                              44 45  1D86
                                 19  1D6E
52 4F 52 52 45 20 45 56 49 52 44 00' 1D88   1191 MSG33_DERR::     .ASCIC  /DRIVE ERROR DETECTED/
         44 45 54 43 45 54 45 44 20  1D94
                                 14  1D88
                                     1D9D   1192 MSG33A_UXEATTN::
20 44 45 54 43 45 50 58 45 4E 55 00' 1D9D   1193                  .ASCIC  /UNEXPECTED DRIVE ATTENTION/
54 4E 45 54 54 41 20 45 56 49 52 44  1DA9
                           4E 4F 49  1DB5
                                 1A  1D9D
4E 45 54 54 41 20 45 56 49 52 44 00' 1DB8   1194 MSG33B_NOATTN:: .ASCIC  /DRIVE ATTENTION FAILED TO SET/
20 44 45 4C 49 41 46 20 4E 4F 49 54  1DC4
                  54 45 53 20 4F 54  1DD0
                                 1D  1DB8
45 47 20 44 45 54 55 43 45 58 45 00' 1DD6   1195 MSG34_GS_OPI::  .ASCIC  /EXECUTED GET STATUS WITHOUT GS BIT SET TO FORCE OPI ERROR/
54 49 57 20 53 55 54 41 54 53 20 54  1DE2
20 54 49 42 20 53 47 20 54 55 4F 48  1DEE
45 43 52 4F 46 20 4F 54 20 54 45 53  1DFA
      52 4F 52 52 45 20 49 50 4F 20  1E06
                                 39  1DD6
57 20 54 55 4F 20 44 45 4D 49 54 00' 1E10   1196 MSG35_OPI_TO::  .ASCIC  /TIMED OUT WAITING FOR OPI ERROR TO SET/
4F 20 52 4F 46 20 47 4E 49 54 49 41  1E1C
20 4F 54 20 52 4F 52 52 45 20 49 50  1E28
                           54 45 53  1E34
                                 26  1E10
53 20 52 4F 52 52 45 20 49 50 4F 00' 1E37   1197 MSG36_OPI_RLY:: .ASCIC  /OPI ERROR SET TO EARLY OR LESS THAN 200 MS./
20 59 4C 52 41 45 20 4F 54 20 54 45  1E43
4E 41 48 54 20 53 53 45 4C 20 52 4F  1E4F
            2E 53 4D 20 30 30 32 20  1E5B
                                 2B  1E37
53 20 52 4F 52 52 45 20 49 50 4F 00' 1E63   1198 MSG37_OPI_LAT:: .ASCIC  /OPI ERROR SET TO LATE OR GTR THAN 400 MS./
4F 20 45 54 41 4C 20 4F 54 20 54 45  1E6F
34 20 4E 41 48 54 20 52 54 47 20 52  1E7B
               2E 53 4D 20 30 30  1E87
                                 29  1E63
4F 4E 20 44 45 54 55 43 45 58 45 00' 1E8D   1199 MSG38_NOOP_IE:: .ASCIC  /EXECUTED NOOP CMD WITH INTERRUPTS ENABLED/
20 48 54 49 57 20 44 4D 43 20 50 4F  1E99
45 20 53 54 50 55 52 52 45 54 4E 49  1EA5
                  44 45 4C 42 41 4E  1EB1
                                 29  1E8D
44 20 54 50 55 52 52 45 54 4E 49 00' 1EB7   1200 MSG39_NO_INT::  .ASCIC  /INTERRUPT DID NOT OCCUR OR WAS LATE/
52 55 43 43 4F 20 54 4F 4E 20 44 49  1EC3
45 54 41 4C 20 53 41 57 20 52 4F 20  1ECF
                                 23  1EB7
20 44 45 54 43 45 50 58 45 4E 55 00' 1EDB   1201 MSG39A_INT::    .ASCIC  /UNEXPECTED ATTENTION INTERRUPT FROM ANOTHER DRIVE/
4E 49 20 4E 4F 49 54 4E 45 54 54 41  1EE7
4D 4F 52 46 20 54 50 55 52 52 45 54  1EF3
49 52 44 20 52 45 48 54 4F 4E 41 20  1EFF
                              45 56  1F0B
                                 31  1EDB
45 54 4E 49 20 45 43 49 56 45 44 00' 1F0D   1202 MSG40_BAD_BR::  .ASCIC  /DEVICE INTERRUPTED AT THE WRONG BR LEVEL/
54 20 54 41 20 44 45 54 50 55 52 52  1F19
```

```
20 52 42 20 47 4E 4F 52 57 20 45 48   1F25
               4C 45 56 45 4C   1F31
                           28   1F0D
45 54 4E 49 20 45 43 49 56 45 44 00'  1F36   1203 MSG41_BAD_VEC:: .ASCIC  /DEVICE INTERRUPTED TO THE WRONG VECTOR/
54 20 4F 54 20 44 45 54 50 55 52 52   1F42
43 45 56 20 47 4E 4F 52 57 20 45 48   1F4E
                     52 4F 54   1F5A
                           26   1F36
45 52 20 44 45 54 41 45 50 45 52 00'  1F5D   1204 MSG42_RPT_CYL:: .ASCIC  /REPEATED READ HDRS DID NOT YIELD SAME CYLINDER AND HEAD/
20 44 49 44 20 53 52 44 48 20 44 41   1F69
41 53 20 44 4C 45 49 59 20 54 4F 4E   1F75
20 52 45 44 4E 49 4C 59 43 20 45 4D   1F81
         44 41 45 48 20 44 4E 41   1F8D
                           37   1F5D
44 52 4F 57 20 52 45 44 41 45 48 00'  1F95   1205 MSG43_MP2_NZ::  .ASCIC  /HEADER WORD #2 NOT EQUAL TO ZERO/
41 55 51 45 20 54 4F 4E 20 32 23 20   1FA1
      4F 52 45 5A 20 4F 54 20 4C   1FAD
                           20   1F95
20 44 45 54 41 4C 55 43 4C 41 43 00'  1FB6   1206 MSG44_BAD_CRC:: .ASCIC  /CALCULATED CRC DOES NOT EQUAL ACTUAL CRC/
54 4F 4E 20 53 45 4F 44 20 43 52 43   1FC2
41 55 54 43 41 20 4C 41 55 51 45 20   1FCE
               43 52 43 20 4C   1FDA
                           28   1FB6
4E 49 20 45 52 41 57 54 46 4F 53 00'  1FDF   1207 MS44A_INC_CRC:: .ASCIC  /SOFTWARE INITIATED CHECK OF HEADER CRC FAILED/
43 45 48 43 20 44 45 54 41 49 54 49   1FEB
20 52 45 44 41 45 48 20 46 4F 20 4B   1FF7
   44 45 4C 49 41 46 20 43 52 43   2003
                           2D   1FDF
20 41 20 44 45 54 55 43 45 58 45 00'  200D   1208 MSG45_SK_ZERO:: .ASCIC  /EXECUTED A ZERO DIFFERENCE SEEK/
45 52 45 46 46 49 44 20 4F 52 45 5A   2019
         4B 45 45 53 20 45 43 4E   2025
                           1F   200D
                              202D   1209 MS45A_HD_PROX::
44 4E 41 20 30 20 53 44 41 45 48 00'  202D   1210            .ASCIC  /HEADS 0 AND 1 FOLLOWING DIFFERENT TRACKS - CHECK ALIGNMENT/
47 4E 49 57 4F 4C 4C 4F 46 20 31 20   2039
54 20 54 4E 45 52 45 46 46 49 44 20   2045
43 45 48 43 20 2D 20 53 4B 43 41 52   2051
   54 4E 45 4D 4E 47 49 4C 41 20 4B   205D
                           3A   202D
47 4E 49 4E 4F 49 54 49 53 4F 50 00'  2068   1211 MSG46_POS_ERR:: .ASCIC  /POSITIONING ERROR OCCURED/
52 55 43 43 4F 20 52 4F 52 52 45 20   2074
                        44 45   2080
                           19   2068
55 54 41 54 53 20 45 56 49 52 44 00'  2082   1212 MSG47_DS_INC::  .ASCIC  /DRIVE STATUS INCORRECT/
   54 43 45 52 52 4F 43 4E 49 20 53   208E
                           16   2082
20 41 20 44 45 54 55 43 45 58 45 00'  2099   1213 MSG48_SK_ONE::  .ASCIC  /EXECUTED A ONE TRACK SEEK/
45 53 20 4B 43 41 52 54 20 45 4E 4F   20A5
                        4B 45   20B1
                           19   2099
59 44 41 45 52 20 45 56 49 52 44 00'  20B3   1214 MSG49_DRDY_NC:: .ASCIC  /DRIVE READY DID NOT CLEAR/
45 4C 43 20 54 4F 4E 20 44 49 44 20   20BF
                        52 41   20CB
                           19   20B3
45 53 20 44 45 54 55 43 45 58 45 00'  20CD   1215 MSG50_SK_OUTR:: .ASCIC  /EXECUTED SEEK INTO OUTER GUARD BAND/
45 54 55 4F 20 4F 54 4E 49 20 4B 45   20D9
```

```
44 4E 41 42 20 44 52 41 55 47 20 52   20E5
                                 23   20CD
45 53 20 44 45 54 55 43 45 58 45 00'  20F1   1216 MSG51_SK_INNR:: .ASCIC  /EXECUTED SEEK INTO INNER GUARD BAND/
45 4E 4E 49 20 4F 54 4E 49 20 4B 45   20FD
44 4E 41 42 20 44 52 41 55 47 20 52   2109
                                 23   20F1
52 55 43 43 4F 20 52 4F 52 52 45 00'  2115   1217 MSG52_RD_HNF::  .ASCIC  /ERROR OCCURED FORCING HEADER NOT FOUND ERROR/
48 20 47 4E 49 43 52 4F 46 20 44 45   2121
4F 46 20 54 4F 4E 20 52 45 44 41 45   212D
         52 4F 52 52 45 20 44 4E 55   2139
                                 2C   2115
52 55 43 43 4F 20 52 4F 52 52 45 00'  2142   1218 MSG53_WRT_NXM:: .ASCIC  /ERROR OCCURRED FORCING NON-EXISTANT MEMORY ERROR/
20 47 4E 49 43 52 4F 46 20 44 45 52   214E
54 4E 41 54 53 49 58 45 2D 4E 4F 4E   215A
4F 52 52 45 20 59 52 4F 4D 45 4D 20   2166
                                 52   2172
                                 30   2142
53 45 52 44 44 41 20 4B 53 49 44 00'  2173   1219 MSG54_DA_UPD::  .ASCIC  /DISK ADDRESS FAILED TO UPDATE (INCREMENT) PROPERLY/
20 4F 54 20 44 45 4C 49 41 46 20 53   217F
52 43 4E 49 28 20 45 54 41 44 50 55   218B
45 50 4F 52 50 20 29 54 4E 45 4D 45   2197
                           59 4C 52   21A3
                                 32   2173
44 45 4C 49 41 46 20 41 54 41 44 00'  21A6   1220 MSG55_DAT_CP1:: .ASCIC  /DATA FAILED TO COMPARE FOLLOWING ONE SECTOR READ CMD/
20 45 52 41 50 4D 4F 43 20 4F 54 20   21B2
4E 4F 20 47 4E 49 57 4F 4C 4C 4F 46   21BE
41 45 52 20 52 4F 54 43 45 53 20 45   21CA
                  44 4D 43 20 44      21D6
                                 34   21A6
44 45 4C 49 41 46 20 41 54 41 44 00'  21DB   1221 MSG55_DAT_BND:: .ASCIC  /DATA FAILED TO COMPARE READING OOVER PAGE BOUNDRIES/
20 45 52 41 50 4D 4F 43 20 4F 54 20   21E7
45 56 4F 4F 20 47 4E 49 44 41 45 52   21F3
44 4E 55 4F 42 20 45 47 41 50 20 52   21FF
                        53 45 49 52   220B
                                 33   21DB
49 41 46 20 53 52 45 44 41 45 48 00'  220F   1222 MSG55_HDRCMP::  .ASCIC  /HEADERS FAILED TO COMPARE AFTER FORMAT OPERATION/
41 50 4D 4F 43 20 4F 54 20 44 45 4C   221B
52 4F 46 20 52 45 54 46 41 20 45 52   2227
4F 49 54 41 52 45 50 4F 20 54 41 4D   2233
                                 4E   223F
                                 30   220F
44 45 4C 49 41 46 20 41 54 41 44 00'  2240   1223 MSG56_DAT_UNQ:: .ASCIC  /DATA FAILED TO COMPARE AFTER WRITING SECTOR WITH 128/-
20 45 52 41 50 4D 4F 43 20 4F 54 20   224C
4E 49 54 49 52 57 20 52 45 54 46 41   2258
54 49 57 20 52 4F 54 43 45 53 20 47   2264
                  38 32 31 20 48      2270
41 54 41 44 20 45 55 51 49 4E 55 20   2275   1224                             / UNIQUE DATA WORDS/
               53 44 52 4F 57 20      2281
                                 46   2240
52 55 43 43 4F 20 52 4F 52 52 45 00'  2287   1225 MSG57_WCK_FRC:: .ASCIC  /ERROR OCCURED FORCING A WRITE CHECK ERROR/
41 20 47 4E 49 43 52 4F 46 20 44 45   2293
4B 43 45 48 43 20 45 54 49 52 57 20   229F
               52 4F 52 52 45 20      22AB
                                 29   2287
52 55 43 43 4F 20 52 4F 52 52 45 00'  22B1   1226 MSG58_PAR_WRT:: .ASCIC  /ERROR OCCURED DURING PARTIAL SECTOR WRITE DATA CMD/
41 50 20 47 4E 49 52 55 44 20 44 45   22BD
```

```
52 4F 54 43 45 53 20 4C 41 49 54 52   22C9
20 41 54 41 44 20 45 54 49 52 57 20   22D5
                           44 4D 43   22E1
                                 32   22B1
52 55 43 43 4F 20 52 4F 52 52 45 00'  22E4   1227 MSG59_PAR_RD::  .ASCIC  /ERROR OCCURED DURING PARTIAL SECTOR READ DATA CMD/
41 50 20 47 4E 49 52 55 44 20 44 45   22F0
52 4F 54 43 45 53 20 4C 41 49 54 52   22FC
43 20 41 54 41 44 20 44 41 45 52 20   2308
                              44 4D   2314
                                 31   22E4
52 55 43 43 4F 20 52 4F 52 52 45 00'  2316   1228 MSG60_WRT_PRD:: .ASCIC  /ERROR OCCURED DURING READ OF A PARTIALLY WRITTEN SECTOR/
45 52 20 47 4E 49 52 55 44 20 44 45   2322
54 52 41 50 20 41 20 46 4F 20 44 41   232E
45 54 54 49 52 57 20 59 4C 4C 41 49   233A
         52 4F 54 43 45 53 20 4E      2346
                                 37   2316
                                      234E   1229 MSG61_DAT_PAR::
44 45 4C 49 41 46 20 41 54 41 44 00'  234E   1230            .ASCIC  /DATA FAILED TO COMPARE AFTER READ OF A PARTIALLY WRITTEN SECTOR/
20 45 52 41 50 4D 4F 43 20 4F 54 20   235A
4F 20 44 41 45 52 20 52 45 54 46 41   2366
4C 4C 41 49 54 52 41 50 20 41 20 46   2372
45 53 20 4E 45 54 54 49 52 57 20 59   237E
                        52 4F 54 43   238A
                                 3F   234E
44 45 4C 49 41 46 20 41 54 41 44 00'  238E   1231 MSG62_DC_PAR::  .ASCIC  /DATA FAILED TO COMPARE AFTER PARTIAL SECTOR READ DATA CMD/
20 45 52 41 50 4D 4F 43 20 4F 54 20   239A
41 49 54 52 41 50 20 52 45 54 46 41   23A6
41 45 52 20 52 4F 54 43 45 53 20 4C   23B2
      44 4D 43 20 41 54 41 44 20 44   23BE
                                 39   238E
43 45 53 20 4C 41 49 54 52 41 50 00'  23C8   1232 MSG63_PRD_TMW:: .ASCIC  /PARTIAL SECTOR READ TRANSFERED TO MANY WORDS/
41 52 54 20 44 41 45 52 20 52 4F 54   23D4
4D 20 4F 54 20 44 45 52 45 46 53 4E   23E0
         53 44 52 4F 57 20 59 4E 41   23EC
                                 2C   23C8
52 55 43 43 4F 20 52 4F 52 52 45 00'  23F5   1233 MSG64_WCK_PAR:: .ASCIC  /ERROR OCCURED DURING WRITE CHECK OF PARTIAL SECTOR/
52 57 20 47 4E 49 52 55 44 20 44 45   2401
46 4F 20 4B 43 45 48 43 20 45 54 49   240D
43 45 53 20 4C 41 49 54 52 41 50 20   2419
                           52 4F 54   2425
                                 32   23F5
4F 4E 20 44 49 44 20 41 54 41 44 00'  2428   1234 MSG65_DAT_CP2:: .ASCIC  /DATA DID NOT COMPARE FOLLOWING A TWO SECTOR READ DATA CMD/
4F 46 20 45 52 41 50 4D 4F 43 20 54   2434
57 54 20 41 20 47 4E 49 57 4F 4C 4C   2440
41 45 52 20 52 4F 54 43 45 53 20 4F   244C
      44 4D 43 20 41 54 41 44 20 44   2458
                                 39   2428
52 55 43 43 4F 20 52 4F 52 52 45 00'  2462   1235 MSG66_HNF_LT::  .ASCIC  /ERROR OCCURED FORCING HEADER NOT FOUND AT LAST TRACK/
48 20 47 4E 49 43 52 4F 46 20 44 45   246E
4F 46 20 54 4F 4E 20 52 45 44 41 45   247A
20 54 53 41 4C 20 54 41 20 44 4E 55   2486
               4B 43 41 52 54         2492
                                 34   2462
                                      2497   1236 MSG67_DAT_ES::
44 45 4C 49 41 46 20 41 54 41 44 00'  2497   1237             .ASCIC  /DATA FAILED TO COMPARE AFTER WRITING EACH SECTOR WITH ITS SECTOR  -
cont> NUMBER/
20 45 52 41 50 4D 4F 43 20 4F 54 20   24A3
```

```
4E 49 54 49 52 57 20 52 45 54 46 41   24AF
4F 54 43 45 53 20 48 43 41 45 20 47   24BB
53 20 53 54 49 20 48 54 49 57 20 52   24C7
52 45 42 4D 55 4E 20 52 4F 54 43 45   24D3
                                  47   2497
                                       24DF   1238 MSG68_DAT_MAX::
44 45 4C 49 41 46 20 41 54 41 44 00'  24DF   1239          .ASCIC  /DATA FAILED TO COMPARE AFTER DATA TRANSFER WITH MAXIMUM WORD COUNT/
20 45 52 41 50 4D 4F 43 20 4F 54 20   24EB
54 20 41 54 41 44 20 52 45 54 46 41   24F7
48 54 49 57 20 52 45 46 53 4E 41 52   2503
52 4F 57 20 4D 55 4D 49 58 41 4D 20   250F
            54 4E 55 4F 43 20 44   251B
                                  42   24DF
                                       2522   1240 MSG69_DAT_NHC::
44 45 4C 49 41 46 20 41 54 41 44 00'  2522   1241          .ASCIC  /DATA FAILED TO COMPARE AFTER READ WITH NO HEADER COMPARE CMD/
20 45 52 41 50 4D 4F 43 20 4F 54 20   252E
57 20 44 41 45 52 20 52 45 54 46 41   253A
45 44 41 45 48 20 4F 4E 20 48 54 49   2546
4D 43 20 45 52 41 50 4D 4F 43 20 52   2552
                                  44   255E
                                  3C   2522
53 20 44 4E 49 46 20 54 4E 41 43 00'  255F   1242 MSG70_FND_S39:: .ASCIC  /CANT FIND SECTOR 39 WITHIN 60 TRIES/
54 49 57 20 39 33 20 52 4F 54 43 45   256B
53 45 49 52 54 20 30 36 20 4E 49 48   2577
                                  23   255F
                                       2583   1243 MSG70A_NFSEC31::
4C 20 4F 54 20 45 4C 42 41 4E 55 00'  2583   1244          .ASCIC  /UNABLE TO LOCATED SECTOR 31/
4F 54 43 45 53 20 44 45 54 41 43 4F   258F
                     31 33 20 52   259B
                                  1B   2583
                                       259F   1245 MSG71_RD_BKT::
20 52 4F 52 52 45 20 44 41 45 52 00'  259F   1246          .ASCIC  /READ ERROR OCCURED, APPARENTLY CAUSED BY ADJACENT CYLN INTERFEREN -
cont> CE/
50 50 41 20 2C 44 45 52 55 43 43 4F   25AB
53 55 41 43 20 59 4C 54 4E 45 52 41   25B7
45 43 41 4A 44 41 20 59 42 20 44 45   25C3
45 54 4E 49 20 4E 4C 59 43 20 54 4E   25CF
         45 43 4E 45 52 45 46 52   25DB
                                  43   259F
                                       25E3   1247 MSG72_DAT_BKT::
44 45 4C 49 41 46 20 41 54 41 44 00'  25E3   1248          .ASCIC  /DATA FAILED TO COMPARE TESTING FOR ADJACENT CYLN INTERFERENCE/
20 45 52 41 50 4D 4F 43 20 4F 54 20   25EF
20 52 4F 46 20 47 4E 49 54 53 45 54   25FB
4C 59 43 20 54 4E 45 43 41 4A 44 41   2607
4E 45 52 45 46 52 45 54 4E 49 20 4E   2613
                               45 43   261F
                                  3D   25E3
                                       2621   1249 MSG73_GS_UND::
55 54 41 54 53 20 45 56 49 52 44 00'  2621   1250          .ASCIC  /DRIVE STATUS INCORRECT AFTER HEADS UNLOADED/
20 54 43 45 52 52 4F 43 4E 49 20 53   262D
20 53 44 41 45 48 20 52 45 54 46 41   2639
         44 45 44 41 4F 4C 4E 55   2645
                                  2B   2621
                                       264D   1251 MSG74_DRDY_NC::
59 44 41 45 52 20 45 56 49 52 44 00'  264D   1252          .ASCIC  /DRIVE READY NOT CLEAR AFTER HEADS UNLOADED/
41 20 52 41 45 4C 43 20 54 4F 4E 20   2659
55 20 53 44 41 45 48 20 52 45 54 46   2665
```

```
                44 45 44 41 4F 4C 4E  2671
                                  2A  264D
                                      2678   1253 MSG75_NO_ST1::
57 20 54 55 4F 20 44 45 4D 49 54 00'  2678   1254           .ASCIC  /TIMED OUT WAITING FOR DRIVE STATE 1 TO OCCUR/
44 20 52 4F 46 20 47 4E 49 54 49 41   2684
31 20 45 54 41 54 53 20 45 56 49 52   2690
         52 55 43 43 4F 20 4F 54 20   269C
                                  2C  2678
                                      26A5   1255 MSG76_ST1_BS::
44 20 54 43 45 52 52 4F 43 4E 49 00'  26A5   1256           .ASCIC  /INCORRECT DRIVE STATUS DURING SPIN UP STATE/
20 53 55 54 41 54 53 20 45 56 49 52   26B1
20 4E 49 50 53 20 47 4E 49 52 55 44   26BD
            45 54 41 54 53 20 50 55   26C9
                                  2B  26A5
                                      26D1   1257 MSG77_NO_ST2::
57 20 54 55 4F 20 44 45 4D 49 54 00'  26D1   1258           .ASCIC  /TIMED OUT WAITING FOR DRIVE STATE 2 TO OCCUR/
44 20 52 4F 46 20 47 4E 49 54 49 41   26DD
32 20 45 54 41 54 53 20 45 56 49 52   26E9
         52 55 43 43 4F 20 4F 54 20   26F5
                                  2C  26D1
                                      26FE   1259 MSG77_NO_ST3::
57 20 54 55 4F 20 44 45 4D 49 54 00'  26FE   1260           .ASCIC  /TIMED OUT WAITING FOR DRIVE STATE 3 TO OCCUR/
44 20 52 4F 46 20 47 4E 49 54 49 41   270A
33 20 45 54 41 54 53 20 45 56 49 52   2716
         52 55 43 43 4F 20 4F 54 20   2722
                                  2C  26FE
                                      272B   1261 MSG78_ST2_BS::
44 20 54 43 45 52 52 4F 43 4E 49 00'  272B   1262           .ASCIC  /INCORRECT DRIVE STATUS DURING BRUSH CYCLE STATE/
20 53 55 54 41 54 53 20 45 56 49 52   2737
48 53 55 52 42 20 47 4E 49 52 55 44   2743
45 54 41 54 53 20 45 4C 43 59 43 20   274F
                                  2F  272B
                                      275B   1263 MSG79_BH_NC::
57 20 54 55 4F 20 44 45 4D 49 54 00'  275B   1264           .ASCIC  /TIMED OUT WAITING FOR BRUSH CYCLE TO START (BH NOT CLEAR)/
42 20 52 4F 46 20 47 4E 49 54 49 41   2767
54 20 45 4C 43 59 43 20 48 53 55 52   2773
20 48 42 28 20 54 52 41 54 53 20 4F   277F
      29 52 41 45 4C 43 20 54 4F 4E   278B
                                  39  275B
                                      2795   1265 MSG80_BH_NF::
57 20 54 55 4F 20 44 45 4D 49 54 00'  2795   1266           .ASCIC  /TIMED OUT WAITING FOR BRUSH CYCLE TO COMPLETE (BH NOT SET)/
42 20 52 4F 46 20 47 4E 49 54 49 41   27A1
54 20 45 4C 43 59 43 20 48 53 55 52   27AD
28 20 45 54 45 4C 50 4D 4F 43 20 4F   27B9
   29 54 45 53 20 54 4F 4E 20 48 42   27C5
                                  3A  2795
                                      27D0   1267 MSG81_ST3_BS::
44 20 54 43 45 52 52 4F 43 4E 49 00'  27D0   1268           .ASCIC  /INCORRECT DRIVE STATUS DURING LOAD HEADS STATE/
20 53 55 54 41 54 53 20 45 56 49 52   27DC
20 44 41 4F 4C 20 47 4E 49 52 55 44   27E8
   45 54 41 54 53 20 53 44 41 45 48   27F4
                                  2E  27D0
                                      27FF   1269 MSG82_NO_ST4::
57 20 54 55 4F 20 44 45 4D 49 54 00'  27FF   1270           .ASCIC  /TIMED OUT WAITING FOR DRIVE STATE 4 TO OCCUR/
44 20 52 4F 46 20 47 4E 49 54 49 41   280B
34 20 45 54 41 54 53 20 45 56 49 52   2817
```

```
      52 55 43 43 4F 20 4F 54 20  2823
                              2C  27FF
                                  282C  1271 MSG83_ST4_BS::
53 20 54 43 45 52 52 4F 43 4E 49 00'  282C  1272         .ASCIC  /INCORRECT STATUS DURING SEEK STATE OF HEAD LOAD OPERATION/
47 4E 49 52 55 44 20 53 55 54 41 54  2838
20 45 54 41 54 53 20 4B 45 45 53 20  2844
44 41 4F 4C 20 44 41 45 48 20 46 4F  2850
      4E 4F 49 54 41 52 45 50 4F 20  285C
                                 39  282C
                                  2866  1273 MS83A_ST4_DE::
52 4F 52 52 45 20 45 56 49 52 44 00'  2866  1274         .ASCIC  /DRIVE ERROR BIT IN IDCCSR SHOULD BE SET IN DRIVE STATE 4/
43 43 44 49 20 4E 49 20 54 49 42 20  2872
45 42 20 44 4C 55 4F 48 53 20 52 53  287E
56 49 52 44 20 4E 49 20 54 45 53 20  288A
      34 20 45 54 41 54 53 20 45  2896
                                 38  2866
                                  289F  1275 MSG84_NO_ST5::
57 20 54 55 4F 20 44 45 4D 49 54 00'  289F  1276         .ASCIC  /TIMED OUT WAITING FOR DRIVE STATE 5 TO OCCUR/
44 20 52 4F 46 20 47 4E 49 54 49 41  28AB
35 20 45 54 41 54 53 20 45 56 49 52  28B7
      52 55 43 43 4F 20 4F 54 20  28C3
                                 2C  289F
                                  28CC  1277 MSG85_ST5_BS::
44 20 54 43 45 52 52 4F 43 4E 49 00'  28CC  1278         .ASCIC  /INCORRECT DRIVE STATUS DURING LOCKED ON TRACK STATE/
20 53 55 54 41 54 53 20 45 56 49 52  28D8
45 4B 43 4F 4C 20 47 4E 49 52 55 44  28E4
53 20 4B 43 41 52 54 20 4E 4F 20 44  28F0
                     45 54 41 54  28FC
                                 33  28CC
                                  2900  1279 MSG86_DRDY_NS::
59 44 41 45 52 20 45 56 49 52 44 00'  2900  1280         .ASCIC  /DRIVE READY IN IDCCSR NOT SET AFTER HEAD LOAD OPERATION/
4E 20 52 53 43 43 44 49 20 4E 49 20  290C
52 45 54 46 41 20 54 45 53 20 54 4F  2918
4F 20 44 41 4F 4C 20 44 41 45 48 20  2924
         4E 4F 49 54 41 52 45 50  2930
                                 37  2900
                                  2938  1281 MSG87_VC_NC::
43 45 48 43 20 45 4D 55 4C 4F 56 00'  2938  1282         .ASCIC  /VOLUME CHECK FAILED TO CLEAR AFTER DRIVE RESET OPERATION/
20 4F 54 20 44 45 4C 49 41 46 20 4B  2944
20 52 45 54 46 41 20 52 41 45 4C 43  2950
20 54 45 53 45 52 20 45 56 49 52 44  295C
      4E 4F 49 54 41 52 45 50 4F  2968
                                 38  2938
                                  2971  1283 MSG88_NO_ST6::
57 20 54 55 4F 20 44 45 4D 49 54 00'  2971  1284         .ASCIC  /TIMED OUT WAITING FOR DRIVE STATE 6 TO OCCUR/
44 20 52 4F 46 20 47 4E 49 54 49 41  297D
36 20 45 54 41 54 53 20 45 56 49 52  2989
      52 55 43 43 4F 20 4F 54 20  2995
                                 2C  2971
                                  299E  1285 MSG89_ST6_BS::
44 20 54 43 45 52 52 4F 43 4E 49 00'  299E  1286         .ASCIC  /INCORRECT DRIVE STATUS DURING HEADS UNLOADING STATE/
20 53 55 54 41 54 53 20 45 56 49 52  29AA
53 44 41 45 48 20 47 4E 49 52 55 44  29B6
53 20 47 4E 49 44 41 4F 4C 4E 55 20  29C2
                     45 54 41 54  29CE
                                 33  299E
```

```
                                  29D2   1287 MSG90_NO_ST7::
57 20 54 55 4F 20 44 45 4D 49 54 00' 29D2   1288          .ASCIC  /TIMED OUT WAITING FOR DRIVE STATE 7 TO OCCUR/
44 20 52 4F 46 20 47 4E 49 54 49 41  29DE
37 20 45 54 41 54 53 20 45 56 49 52  29EA
         52 55 43 43 4F 20 4F 54 20  29F6
                                  2C  29D2
                                  29FF   1289 MSG91_ST7_BS::
44 20 54 43 45 52 52 4F 43 4E 49 00' 29FF   1290          .ASCIC  /INCORRECT DRIVE STATUS DURING SPIN DOWN STATE/
20 53 55 54 41 54 53 20 45 56 49 52  2A0B
20 4E 49 50 53 20 47 4E 49 52 55 44  2A17
      45 54 41 54 53 20 4E 57 4F 44  2A23
                                  2D  29FF
                                  2A2D   1291 MSG92_NO_ST0::
57 20 54 55 4F 20 44 45 4D 49 54 00' 2A2D   1292          .ASCIC  /TIMED OUT WAITING FOR DRIVE STATE 0 TO OCCUR/
44 20 52 4F 46 20 47 4E 49 54 49 41  2A39
30 20 45 54 41 54 53 20 45 56 49 52  2A45
         52 55 43 43 4F 20 4F 54 20  2A51
                                  2C  2A2D
                                  2A5A   1293 MSG93_RELV_UP::
20 44 45 54 43 45 50 58 45 4E 55 00' 2A5A   1294          .ASCIC  /UNEXPECTED ERROR OCCURED AFTER CHANGING UNIT PLUG NUMBER/
45 52 55 43 43 4F 20 52 4F 52 52 45  2A66
4E 41 48 43 20 52 45 54 46 41 20 44  2A72
4C 50 20 54 49 4E 55 20 47 4E 49 47  2A7E
         52 45 42 4D 55 4E 20 47 55  2A8A
                                  38  2A5A
                                  2A93   1295 MSG94_UXE_DSE::
20 44 45 54 43 45 50 58 45 4E 55 00' 2A93   1296          .ASCIC  /UNEXPECTED ERROR OCCURED TRYING TO FORCE DRIVE SELECT ERROR/
45 52 55 43 43 4F 20 52 4F 52 52 45  2A9F
20 4F 54 20 47 4E 49 59 52 54 20 44  2AAB
20 45 56 49 52 44 20 45 43 52 4F 46  2AB7
52 4F 52 52 45 20 54 43 45 4C 45 53  2AC3
                                  3B  2A93
                                  2ACF   1297 MS94A_NO_OTH::
52 55 43 43 4F 20 52 4F 52 52 45 00' 2ACF   1298          .ASCIC  /ERROR OCCURED TRYING TO ACCESS OTHER DRIVE/
4F 54 20 47 4E 49 59 52 54 20 44 45  2ADB
45 48 54 4F 20 53 53 45 43 43 41 20  2AE7
            45 56 49 52 44 20 52     2AF3
                                  2A  2ACF
                                  2AFA   1299 MSG95_DSE_NS::
43 45 4C 45 53 20 45 56 49 52 44 00' 2AFA   1300          .ASCIC  /DRIVE SELECT ERROR AND-OR DRIVE ERROR BITS FAILED TO SET/
2D 44 4E 41 20 52 4F 52 52 45 20 54  2B06
52 52 45 20 45 56 49 52 44 20 52 4F  2B12
4C 49 41 46 20 53 54 49 42 20 52 4F  2B1E
         54 45 53 20 4F 54 20 44 45  2B2A
                                  38  2AFA
                                  2B33   1301 MSG96_DSE_NC::
43 45 4C 45 53 20 45 56 49 52 44 00' 2B33   1302          .ASCIC  /DRIVE SELECT ERROR FAILED TO CLEAR WITH DRIVE RESET/
4C 49 41 46 20 52 4F 52 52 45 20 54  2B3F
20 52 41 45 4C 43 20 4F 54 20 44 45  2B4B
52 20 45 56 49 52 44 20 48 54 49 57  2B57
                     54 45 53 45     2B63
                                  33  2B33
                                  2B67   1303 MSG97_GS_NXD::
20 41 20 44 45 54 55 43 45 58 45 00' 2B67   1304          .ASCIC  /EXECUTED A GET STATUS CMD TO A NON-EXISTANT DRIVE/
43 20 53 55 54 41 54 53 20 54 45 47  2B73
2D 4E 4F 4E 20 41 20 4F 54 20 44 4D  2B7F
```

```
49 52 44 20 54 4E 41 54 53 49 58 45  2B8B
                              45 56  2B97
                                 31  2B67
                                     2B99   1305 MSG98_WRT_WL::
52 57 20 44 45 54 55 43 45 58 45 00' 2B99   1306          .ASCIC  /EXECUTED WRITE CMD WITH WRITE LOCK SWITCH SET/
48 54 49 57 20 44 4D 43 20 45 54 49  2BA5
20 4B 43 4F 4C 20 45 54 49 52 57 20  2BB1
      54 45 53 20 48 43 54 49 57 53  2BBD
                                 2D  2B99
                                     2BC7   1307 MSG99_RST_WGE::
20 44 45 54 43 45 50 58 45 4E 55 00' 2BC7   1308          .ASCIC  /UNEXPECTED ERROR OCCURED TRYING TO CLEAR WRITE PROTECT ERROR/
45 52 55 43 43 4F 20 52 4F 52 52 45  2BD3
20 4F 54 20 47 4E 49 59 52 54 20 44  2BDF
20 45 54 49 52 57 20 52 41 45 4C 43  2BEB
4F 52 52 45 20 54 43 45 54 4F 52 50  2BF7
                                 52  2C03
                                 3C  2BC7
                                     2C04   1309 MS100_DAT_DST::
49 44 20 53 41 57 20 41 54 41 44 00' 2C04   1310          .ASCIC  /DATA WAS DISTURBED WITH WRITE LOCK SWITCH SET/
48 54 49 57 20 44 45 42 52 55 54 53  2C10
20 4B 43 4F 4C 20 45 54 49 52 57 20  2C1C
      54 45 53 20 48 43 54 49 57 53  2C28
                                 2D  2C04
                                     2C32   1311 MS103_SK_ONE::
45 44 4E 49 4C 59 43 20 45 4E 4F 00' 2C32   1312          .ASCIC  /ONE CYLINDER SEEK TIMING EXCEEDS LIMITS/
4E 49 4D 49 54 20 4B 45 45 53 20 52  2C3E
49 4C 20 53 44 45 45 43 58 45 20 47  2C4A
                     53 54 49 4D     2C56
                                 27  2C32
                                     2C5A   1313 MS104_SK_MAX::
4C 59 43 20 4D 55 4D 49 58 41 4D 00' 2C5A   1314          .ASCIC  /MAXIMUM CYLINDER SEEK TIMING EXCEEDS LIMITS/
54 20 4B 45 45 53 20 52 45 44 4E 49  2C66
44 45 45 43 58 45 20 47 4E 49 4D 49  2C72
            53 54 49 4D 49 4C 20 53  2C7E
                                 2B  2C5A
                                     2C86   1315 MS105_SK_AVG::
4C 59 43 20 45 47 41 52 45 56 41 00' 2C86   1316          .ASCIC  /AVERAGE CYLINDER SEEK TIMING EXCEEDS LIMITS/
54 20 4B 45 45 53 20 52 45 44 4E 49  2C92
44 45 45 43 58 45 20 47 4E 49 4D 49  2C9E
            53 54 49 4D 49 4C 20 53  2CAA
                                 2B  2C86
                                     2CB2   1317 MS106_SP_TIM::
20 4C 41 4E 4F 49 54 41 54 4F 52 00' 2CB2   1318          .ASCIC  /ROTATIONAL TIME OUT OF RANGE/
20 46 4F 20 54 55 4F 20 45 4D 49 54  2CBE
                  45 47 4E 41 52     2CCA
                                 1C  2CB2
                                     2CCF   1319 MS107_RECAL::
52 55 43 43 4F 20 52 4F 52 52 45 00' 2CCF   1320          .ASCIC  /ERROR OCCURED EXECUTING A DRIVE RECAL FUNCTION/
47 4E 49 54 55 43 45 58 45 20 44 45  2CDB
43 45 52 20 45 56 49 52 44 20 41 20  2CE7
   4E 4F 49 54 43 4E 55 46 20 4C 41  2CF3
                                 2E  2CCF
                                     2CFE   1321 MS107A_SEEKERR::
4E 49 52 55 44 20 52 4F 52 52 45 00' 2CFE   1322          .ASCIC  /ERROR DURING SEEK TO INVALID CYLINDER/
4E 49 20 4F 54 20 4B 45 45 53 20 47  2D0A
44 4E 49 4C 59 43 20 44 49 4C 41 56  2D16
```

```
                                52 45   2D22
                                   25   2CFE
                                        2D24   1323 MS107B_RECAL::
4E 49 52 55 44 20 52 4F 52 52 45 00'    2D24   1324            .ASCIC  /ERROR DURING RECOVERY FROM SEEK ERROR/
46 20 59 52 45 56 4F 43 45 52 20 47     2D30
52 52 45 20 4B 45 45 53 20 4D 4F 52     2D3C
                                52 4F   2D48
                                   25   2D24
                                        2D4A   1325 MS108_SECTNZ::
4E 55 4F 43 20 52 4F 54 43 45 53 00'    2D4A   1326            .ASCIC  /SECTOR COUNTER IN STATUS WORD NEVER REACHED ZERO/
55 54 41 54 53 20 4E 49 20 52 45 54     2D56
52 45 56 45 4E 20 44 52 4F 57 20 53     2D62
52 45 5A 20 44 45 48 43 41 45 52 20     2D6E
                                   4F   2D7A
                                   30   2D4A
                                        2D7B   1327 MS109_BADCNT::
4E 55 4F 43 20 52 4F 54 43 45 53 00'    2D7B   1328            .ASCIC  /SECTOR COUNTER IN STATUS WORD NOT COUNTING PROPERLY/
55 54 41 54 53 20 4E 49 20 52 45 54     2D87
43 20 54 4F 4E 20 44 52 4F 57 20 53     2D93
50 4F 52 50 20 47 4E 49 54 4E 55 4F     2D9F
                          59 4C 52 45   2DAB
                                   33   2D7B
                                        2DAF   1329 MS110_UEHNF31::
20 44 45 54 43 45 50 58 45 4E 55 00'    2DAF   1330            .ASCIC  /UNEXPECTED ERR WHILE FORCING HNF BY ACCESSING SECTOR 31/
4F 46 20 45 4C 49 48 57 20 52 52 45     2DBB
59 42 20 46 4E 48 20 47 4E 49 43 52     2DC7
53 20 47 4E 49 53 53 45 43 43 41 20     2DD3
            31 33 20 52 4F 54 43 45     2DDF
                                   37   2DAF
                                        2DE7   1331 MS111_EHNF31::
45 4C 49 48 57 20 52 4F 52 52 45 00'    2DE7   1332            .ASCIC  /ERROR WHILE ACCESSING SECTOR 31 WITH SKIP SECTORS ENABLED/
53 20 47 4E 49 53 53 45 43 43 41 20     2DF3
54 49 57 20 31 33 20 52 4F 54 43 45     2DFF
4F 54 43 45 53 20 50 49 4B 53 20 48     2E0B
      44 45 4C 42 41 4E 45 20 53 52     2E17
                                   39   2DE7
                                        2E21   1333 MS112_ACC31::
45 4C 49 48 57 20 52 4F 52 52 45 00'    2E21   1334            .ASCIC  /ERROR WHILE ACCESSING SECTOR 31/
53 20 47 4E 49 53 53 45 43 43 41 20     2E2D
            31 33 20 52 4F 54 43 45     2E39
                                   1F   2E21
                                        2E41   1335 MS113_UWRTERR::
20 44 45 54 43 45 50 58 45 4E 55 00'    2E41   1336            .ASCIC  /UNEXPECTED ERROR WRITING WITH SKIP SECTORS DISABLED/
4E 49 54 49 52 57 20 52 4F 52 52 45     2E4D
20 50 49 4B 53 20 48 54 49 57 20 47     2E59
41 53 49 44 20 53 52 4F 54 43 45 53     2E65
                          44 45 4C 42   2E71
                                   33   2E41
                                        2E75   1337 MS114_URDERR::
20 44 45 54 43 45 50 58 45 4E 55 00'    2E75   1338            .ASCIC  /UNEXPECTED ERROR READING WITH AUTO SKIP DISABLED/
4E 49 44 41 45 52 20 52 4F 52 52 45     2E81
20 4F 54 55 41 20 48 54 49 57 20 47     2E8D
45 4C 42 41 53 49 44 20 50 49 4B 53     2E99
                                   44   2EA5
                                   30   2E75
                                        2EA6   1339 MS114A_UWCKERR::
```

```
20 44 45 54 43 45 50 58 45 4E 55 00' 2EA6  1340            .ASCIC  /UNEXPECTED ERROR WRITE CHECKING WITH AUTO SKIP DISABLED/
20 45 54 49 52 57 20 52 4F 52 52 45  2EB2
54 49 57 20 47 4E 49 4B 43 45 48 43  2EBE
20 50 49 4B 53 20 4F 54 55 41 20 48  2ECA
            44 45 4C 42 41 53 49 44  2ED6
                                 37  2EA6
                                     2EDE  1341 MS115_RDRST::
52 55 43 43 4F 20 52 4F 52 52 45 00' 2EDE  1342            .ASCIC  /ERROR OCCURED RESTARTING READ FUNCTION AFTER SKIP SECTOR ERROR/
4E 49 54 52 41 54 53 45 52 20 44 45  2EEA
54 43 4E 55 46 20 44 41 45 52 20 47  2EF6
4B 53 20 52 45 54 46 41 20 4E 4F 49  2F02
52 45 20 52 4F 54 43 45 53 20 50 49  2F0E
                           52 4F 52  2F1A
                                 3E  2EDE
                                     2F1D  1343 MS115A_WKRST::
52 55 43 43 4F 20 52 4F 52 52 45 00' 2F1D  1344            .ASCIC  /ERROR OCCURED RESTARTING WRITECHECK AFTER SKIP SECTOR ERROR/
4E 49 54 52 41 54 53 45 52 20 44 45  2F29
4B 43 45 48 43 45 54 49 52 57 20 47  2F35
20 50 49 4B 53 20 52 45 54 46 41 20  2F41
52 4F 52 52 45 20 52 4F 54 43 45 53  2F4D
                                 3B  2F1D
                                     2F59  1345 MS116_SKPCMP::
44 45 4C 49 41 46 20 41 54 41 44 00' 2F59  1346            .ASCIC  /DATA FAILED TO COMPARE AFTER SKIP SECTOR READ/
20 45 52 41 50 4D 4F 43 20 4F 54 20  2F65
53 20 50 49 4B 53 20 52 45 54 46 41  2F71
      44 41 45 52 20 52 4F 54 43 45  2F7D
                                 2D  2F59
                                     2F87  1347 MS116A_AUTOCMP::
44 45 4C 49 41 46 20 41 54 41 44 00' 2F87  1348            .ASCIC  /DATA FAILED TO COMPARE AFTER AUTO SKIP SECTOR READ/
20 45 52 41 50 4D 4F 43 20 4F 54 20  2F93
53 20 4F 54 55 41 20 52 45 54 46 41  2F9F
52 20 52 4F 54 43 45 53 20 50 49 4B  2FAB
                           44 41 45  2FB7
                                 32  2F87
                                     2FBA  1349 MS117_AUTOWRT::
4E 49 52 55 44 20 52 4F 52 52 45 00' 2FBA  1350            .ASCIC  /ERROR DURING WRITE TO SKIP SECTOR WITH AUTO SKIP ENABLED/
53 20 4F 54 20 45 54 49 52 57 20 47  2FC6
57 20 52 4F 54 43 45 53 20 50 49 4B  2FD2
49 4B 53 20 4F 54 55 41 20 48 54 49  2FDE
         44 45 4C 42 41 4E 45 20 50  2FEA
                                 38  2FBA
                                     2FF3  1351 MS117_AUTORD::
4E 49 52 55 44 20 52 4F 52 52 45 00' 2FF3  1352            .ASCIC  /ERROR DURING WRITE CHECK TO SKIP SECTOR WITH AUTO SKIP ENABLED/
43 45 48 43 20 45 54 49 52 57 20 47  2FFF
45 53 20 50 49 4B 53 20 4F 54 20 4B  300B
55 41 20 48 54 49 57 20 52 4F 54 43  3017
42 41 4E 45 20 50 49 4B 53 20 4F 54  3023
                           44 45 4C  302F
                                 3E  2FF3
                                     3032  1353 MS117_AUTOWCK::
4E 49 52 55 44 20 52 4F 52 52 45 00' 3032  1354            .ASCIC  /ERROR DURING READ TO SKIP SECTOR WITH AUTO SKIP ENABLED/
4B 53 20 4F 54 20 44 41 45 52 20 47  303E
49 57 20 52 4F 54 43 45 53 20 50 49  304A
50 49 4B 53 20 4F 54 55 41 20 48 54  3056
            44 45 4C 42 41 4E 45 20  3062
                                 37  3032
```

```
                                    306A  1355 MS118_SEEKIDC::
52 55 43 43 4F 20 52 4F 52 52 45 00' 306A  1356          .ASCIC  /ERROR OCCURED WITH IDC TRACKING ALL CURRENT SEEK POSITIONS/
20 43 44 49 20 48 54 49 57 20 44 45  3076
4C 4C 41 20 47 4E 49 4B 43 41 52 54  3082
45 45 53 20 54 4E 45 52 52 55 43 20  308E
   53 4E 4F 49 54 49 53 4F 50 20 4B  309A
                                 3A  306A
                                    30A5  1357 MS119_IDCPOS::
47 4E 49 4E 4F 49 54 49 53 4F 50 00' 30A5  1358          .ASCIC  /POSITIONING ERROR - AFTER 512 IDC CALCULATED SEEKS/
54 46 41 20 2D 20 52 4F 52 52 45 20  30B1
43 20 43 44 49 20 32 31 35 20 52 45  30BD
45 53 20 44 45 54 41 4C 55 43 4C 41  30C9
                           53 4B 45  30D5
                                 32  30A5
                                    30D8  1359 .NLIST MEB
```

```
                   30D8
                   30D8          .SBTTL  INITIALIZATION CODE.
                   30D8    ;++
                   30D8    ; FUNCTIONAL DESCRIPTION::
                   30D8    ;
                   30D8    ;       THIS ROUTINE WILL BE EXECUTED AT THE BEGINNING OF EACH PASS.
                   30D8    ;       THIS ROUTINE FIRST WILL SELECT EACH LOGICAL UNIT FROM THE
                   30D8    ;       P-TABLE AND EXECUTE A HOME BLOCK READ TO DETERMINE THE PACK
                   30D8    ;       NAME.  IF AN ERROR OCCURED READING THE HOME BLOCK OR THE PACK
                   30D8    ;       NAME IS NOT 'SCRATCH' THE OPERATOR IS PROMPTED FOR A DECISION
                   30D8    ;       AS TO CONTINUE TESTING OR NOT.
                   30D8    ;       ONCE THE PROTECTION CHECK HAS BEEN MADE ON EACH LOGICAL UNIT
                   30D8    ;       TESTING BEGINS ON EACH DEVICE PROVIDING IT WAS NOT ABORTED
                   30D8    ;       FROM THE TEST LIST (ABORT_FLAG).
                   30D8    ;
                   30D8    ;
                   30D8    ; CALLING SEQUENCE::
                   30D8    ;
                   30D8    ;       ** NONE **
                   30D8    ;
                   30D8    ; INPUT PARAMETERS::
                   30D8    ;
                   30D8    ;       ** NONE **
                   30D8    ;
                   30D8    ; IMPLICIT INPUTS::
                   30D8    ;
                   30D8    ;       ** NONE **
                   30D8    ;
                   30D8    ; OUTPUT PARAMETERS::
                   30D8    ;
                   30D8    ;       ** NONE **
                   30D8    ;
                   30D8    ; IMPLICIT OUTPUTS::
                   30D8    ;
                   30D8    ;       ** NONE **
                   30D8    ;
                   30D8    ; COMPLETION CODES::
                   30D8    ;
                   30D8    ;       ** NONE **
                   30D8    ;
                   30D8    ; SIDE EFFECTS::
                   30D8    ;
                   30D8    ;       ** NONE **
                   30D8    ;
                   30D8    ;--
                   30D8
                   30D8          $DS_BGNINIT
011C'CF     D4     0002          CLRL    CONFIG                          ; CLEAR ANY PROGRAM FLAGS
                   0006          $DS_BPASS0      10$                     ; IF PASS0 GOTO 10$
   01FC     31     000E          BRW     120$                            ; SKIP PROTECTION CHECK
                   0011    10$:  $READEF_S -                             ; READ EVENT FLAGS
                   0011            EFN=          #0,-                    ; GROUP NUMBER 0
                   0011            STATE=        EVENT_FLAGS             ; PUT FLAGS HERE
010C'CF     D4     001E          CLRL    PNT_FLAG                        ; CLEAR PRINT FLAGS
0054'CF     D4     0022          CLRL    ABORT_FLAG                      ; CLEAR ABORT FLAG
     53     D4     0026          CLRL    R3                              ; RESET LUN COUNTER
```

```
                             0028  1417 15$:    $DS_GPHARD_S -                          ; GET P-TABLE ADDRS
                             0028  1418           DEVNUM=       R3,-                    ; LOGICAL UNIT
                             0028  1419           ADRLOC=       HARD_TABLE              ; STORAGE FOR PTABLE ADDRESS
                             0035  1420         $DS_BNERROR     20$                     ; BRANCH IF VALID UNIT
              01B8     31    0045  1421         BRW     100$                            ; GET UNITS TO TEST
      52    0058'CF    D0    0048  1422 20$:    MOVL    HARD_TABLE,R2                   ; R2 <-- P-TABLE ADDRESS
        58    0B A2    9A    004D  1423         MOVZBL  HP$B_DRIVE(R2),R8               ; GET PHYSICAL DRIVE #
      55    58    10   C1    0051  1424         ADDL3   #16,R8,R5                       ; CALC ATTENTION BIT POSITON
  003C'CF   01    55   9C    0055  1425         ROTL    R5,#1,P$ATTN                    ; POSITION EXPECTED ATTENTION BIT
  0034'CF   58    08   9C    005B  1426         ROTL    #8,R8,P$UNIT                    ; POSITION DRIVE SELECT CORRECTLY
     0044'CF      62   7D    0061  1427         MOVQ    HP$Q_DEVICE(R2),DEV_NAME        ; GET DEVICE MNEMONIC POINTER
        55    26 A2    DE    0066  1428         MOVAL   HP$T_TYPE(R2),R5                ; GET ADDRESS OF HARDWARE DRIVE TYPE
        52    20 A2    D0    006A  1429         MOVL    HP$A_LINK(R2),R2                ; R2 <-- P-TABLE FOR IDC
   018C'CF    24 A2    3C    006E  1430         MOVZWL  HP$W_VECTOR(R2),E.VEC           ; GET DEVICE VECTOR ADDRESS
   0188'CF    36 A2    9A    0074  1431         MOVZBL  RB730$B_BR(R2),E.BR             ; GET DEVICE BR LEVEL
   0014'CF    18 A2    D0    007A  1432         MOVL    HP$A_DEVICE(R2),BASE_ADDR       ; GET BASE ADDRESS
                             0080  1433         $DS_PROBE_S -                           ; CHECK CSR REG IS READABLE
                             0080  1434           UNIT=         R3,-                    ; LOGICAL UNIT
                             0080  1435           ADDRESS=      @BASE_ADDR,-            ; ADDRESS OF REGISTER
                             0080  1436           LENGTH=       #4                      ; SPECIFY LONGWORD ACCESS
                             008F  1437         $DS_BNERROR     25$                     ; IF REGISTER READABLE GOTO 25$
                             009F  1438         $DS_ERRSYS_S ,-                         ; REPORT SYSTEM ERROR
                             009F  1439           UNIT=         R3,-                    ; CURRNET UNIT SELECTED
                             009F  1440           MSGADR=       IDC_ADRFAIL             ; REPORT PROBE ERROR
                             00A9       ;;; INITIALIZATION ERROR 1
                             00B0  1441         $DS_ABORT                               ; DIE
      59    04FC'CF    DE    00B7  1442 25$:    MOVAL   RL02_TBL,R9                     ; SET UP FOR RL TYPE DEVICE
0034'CF   00000080 8F  C9    00BC  1443         BISL3   #CSR$M_CRDY,P$UNIT,@BASE_ADDR   ; SELECT CURRENT UNIT
               0014'DF       00C5
0014'DF   04000000 8F  D3    00C8  1444         BITL    #CSR$M_R80,@BASE_ADDR           ; IF DRIVE IS AND RL02
                  05   13    00D1  1445         BEQL    27$                             ; THEN GOTO 27$
      59    0580'CF    DE    00D3  1446         MOVAL   R80_TBL,R9                      ; ELSE SETUP FOR R80 DISK TYPE
   5B    1C A9    01   C1    00D8  1447 27$:    ADDL3   #1,OFST_DEVMEM(R9),R11          ; CALC ADDRESS OF STRING - COUNT
                  0F   BB    00DD  1448         PUSHR   #^M<R0,R1,R2,R3>                ; SAVE REGISTERS
   6B    01 A5    02   29    00DF  1449         CMPC3   #2,1(R5),(R11)                  ; IF DRIVE ATTACHED IS CORRECT
                  2A   13    00E4  1450         BEQL    28$                             ; THEN GOTO 28$
                  0F   BA    00E6  1451         POPR    #^M<R0,R1,R2,R3>                ; RESTORE REGISTERS
                             00E8  1452         $DS_ERRSYS_S ,-                         ; REPORT SYSTEM ERROR
                             00E8  1453           UNIT=         R3,-                    ; UNIT UNDER TEST
                             00E8  1454           MSGADR=       MSG_OPERATTCH           ; OPERATOR ATTACH ERROR
                             00F2       ;;; INITIALIZATION ERROR 2
                             00F9  1455         $DS_PRINTF_S -                          ; SHOW OPERATOR THE PROBLEM
                             00F9  1456           FORMAT=       FMT_ATTACH,-            ; ATTACH PRPBLEM
                             00F9  1457           P0=           OFST_DEVMEM(R9),-       ; HARDWARE SAYS DEVICE IS
                             00F9  1458           P1=           R5                      ; OPER SAYS DEVICE IS
                             0109  1459         $DS_ABORT                               ; DIE
                  0F   BA    0110  1460 28$:    POPR    #^M<R0,R1,R2,R3>                ; RESTORE REGISTERS
                             0112  1461         $DS_PRINTF_S -                          ; PRINT DRIVE TYPE INFO
                             0112  1462           FORMAT=       FMT_DRVTYP,-            ; FMT MESSAGE
                             0112  1463           P0=           OFST_DEVMEM(R9),-       ; DEVICE MNEMONIC
                             0112  1464           P1=           #DEV_NAME               ; DEVICE NAME
                  5B   D4    0126  1465         CLRL    R11                             ; RESET INDEX COUNTER
      5A    02DC'CF43  D0    0128  1466         MOVL    LUN_TBL[R3],R10                 ; GET ADDRESS OF DEVICE TABLE
         6A4B   694B   D0    012E  1467 29$:    MOVL    (R9)[R11],(R10)[R11]            ; COPY TABLE
         F7 5B    21   F2    0133  1468         AOBLSS  #NUMENT,R11,29$                 ; DO ALL ENTRIES
         04 AA    58   D0    0137  1469         MOVL    R8,OFST_PHYU(R10)               ; SAVE PHYSICAL UNIT NUMBER
```

```
          002C'CF    53   D0  013B  1470        MOVL    R3,L$UNIT                        ; LOAD LOGICAL UNIT
       20 AA   0044'CF    7D  0140  1471        MOVQ    DEV_NAME,OFST_DEVNAM(R10)        ; SAVE DEVICE NAME DESCRIPTOR
          0000'CF    5A   D0  0146  1472        MOVL    R10,DRIVE_TBL                    ; SET UP DRIVE INFO TABLE
 011C'CF   00000040  8F   C8  014B  1473        BISL2   #NOCHK$M_FTL,CONFIG              ; DONT ABORT FOR FATAL DRIVE ERROR
                     0F   BB  0154  1474        PUSHR   #^M<R0,R1,R2,R3>                 ; SAVE REGISTERS
          1107'CF    00   FB  0156  1475        CALLS   #0,RD_HOME_BLK                   ; READ THE HOME BLOCK
 011C'CF   00000040  8F   CA  015B  1476        BICL2   #NOCHK$M_FTL,CONFIG              ; CLEAR PROGRAM FLAG
                     0F   BA  0164  1477        POPR    #^M<R0,R1,R2,R3>                 ; RESTORE REGISTERS
               007C'CF    D5  0166  1478        TSTL    HMB_ERR_FLAG                     ; ERROR READING HOME BLOCK ?
                     13   13  016A  1479        BEQL    30$                              ; BRANCH IF NOT
                              016C  1480        $DS_PRINTF_S -                           ; PRINT HOME BLOCK READ ERROR
                              016C  1481          FORMAT=         FMT_HMB_ERROR,-        ; ERROR MESSAGE
                              016C  1482          P0=             #DEV_NAME              ; FOR THIS DEVICE
                     5E   11  017D  1483        BRB     65$                              ; GO ASK HIM IF WE CONTINUE
          55   02C4'CF    D0  017F  1484 30$:   MOVL    READ_BUFFER,R5                   ; R5 <-- ADDRS OF BUFFER
               0101  8F   B1  0184  1485        CMPW    #HMB$C_LEVEL1,-                  ; IS PACK ODS 1
                  0C A5       0188  1486                HMB$W_STRUCLEV(R5)
                     05   12  018A  1487        BNEQ    40$                              ; BRANCH IF NOT
               55    0E   C0  018C  1488        ADDL2   #HMB$T_VOLNAME,R5                ; ADD IN OFFSET FOR ODS 1
                     07   11  018F  1489        BRB     45$                              ; AND BRANCH
       55   000001D8 8F   C0  0191  1490 40$:   ADDL2   #HMB$T_VOLNAME2,R5               ; ADD IN OFFSET FOR ODS 2
          004C'CF    0C   D0  0198  1491 45$:   MOVL    #12,ACT_NAME                     ; CREATE STRING DESCRIPTOR
          0050'CF    55   D0  019D  1492        MOVL    R5,ACT_NAME+4                    ; GET ADDRS OF STRING
          56   02B4'CF    9A  01A2  1493        MOVZBL  VOL_NAME,R6                      ; R6 <-- BYTE COUNT OF STRING
 57    01   000002B4'8F   C1  01A7  1494        ADDL3   #VOL_NAME,#1,R7                  ; R7 <-- ADDRS OF ASCII
                     58   D4  01AF  1495        CLRL    R8                               ; R8 <-- INDEX TO ASCII
            87     6548   91  01B1  1496 50$:   CMPB    (R5)[R8],(R7)+                   ; COMPARE STRINGS
                     0F   12  01B5  1497        BNEQ    60$                              ; BRANCH IF NOT NAME NOT SCRATCH
            F6 58    56   F2  01B7  1498        AOBLSS  R6,R8,50$                        ; CHECK STRING TO THE END
          50   01    53   78  01BB  1499        ASHL    R3,#1,R0                         ; GET LUN BIT POSITION
          0054'CF    50   C8  01BF  1500        BISL2   R0,ABORT_FLAG                    ; SET TEST FLAG
                     34   11  01C4  1501        BRB     70$                              ; AND GOTO 70$
                              01C6  1502 60$:   $DS_PRINTF_S -                           ; PRINT VOLUME PROTECTION STATUS
                              01C6  1503          FORMAT=         FMT_VOL,-              ; MESSAGE FORMAT
                              01C6  1504          P0=             #DEV_NAME,-            ; DEVICE NAME
                              01C6  1505          P1=             #ACT_NAME              ; ACTUAL VOLUME NAME
                              01DD  1506 65$:   $DS_ASKLGCL_S -                          ; PROMPT OPERATOR
                              01DD  1507          MSGADR=         MSG_ABORT,-            ; ASK OPERATOR FOR DECISION
                              01DD  1508          DATADR=         ABORT_FLAG,-           ; STORE ANSWER HERE
                              01DD  1509          POS=            R3,-                   ; ANSWER MASK POSITION
                              01DD  1510          YEXFER=         70$,-                  ; IF YES GOTO 70$
                              01DD  1511          NOXFER=         70$,-                  ; IF NO GOTO 70$
                              01DD  1512          DEFALT=         #PAR$_NO               ; DEFAULT ANSWER IS NO
     FE28 53     01      07   F1  01FA  1513 70$:   ACBL    #7,#1,R3,15$                     ; GET NEXT DEVICE INFO
          0054'CF         D5  0200  1514 100$:  TSTL    ABORT_FLAG                       ; ANY UNITS TO BE TESTED
                     0E   12  0204  1515        BNEQ    130$                             ; BRANCH IF YES
                              0206  1516        $DS_ABORT                                ; NO UNITS TO TEST
                              020D  1517 120$:  $DS_ENDPASS_S                            ; MUST BE END OF PASS THEN
 002C'CF   0054'CF   03   00   EA  0214  1518 130$:  FFS     #0,#3,ABORT_FLAG,L$UNIT          ; SET UP INITIAL LOGICAL UNIT
          07A5'CF    00   FB  021D  1519        CALLS   #0,DRIVE_INIT$A                  ; RESET ALL DRIVES
                              0222  1520        $DS_CHANNEL_S   L$UNIT,#CHC$_ENINT,-     ; SET UP INTERRUPT VECTOR
                              0222  1521                        IDC_ISR$A,IDC_STATUS
                              023B  1522        $DS_SETIPL_S -                           ; ALLOW INTERRUPTS
                              023B  1523          LEVEL=          #0                     ; IPL = 0
                              0244  1524        $DS_ENDINIT
```

```
                                30D8  1526           .SBTTL  CLEAN_UP CODE.
                                30D8  1527 ;++
                                30D8  1528 ; FUNCTIONAL DESCRIPTION::
                                30D8  1529 ;
                                30D8  1530 ;         THIS ROUTINE IS EXECUTED AT THE COMPLETION OF EACH PASS.
                                30D8  1531 ;
                                30D8  1532 ; CALLING SEQUENCE::
                                30D8  1533 ;
                                30D8  1534 ;         THE DIAGNOSTIC SUPERVISOR AUTOMATICALLY CALLS THIS ROUTINE WITH
                                30D8  1535 ;         A CALLG INSTRUCTION.
                                30D8  1536 ;
                                30D8  1537 ; INPUT PARAMETERS::
                                30D8  1538 ;
                                30D8  1539 ;         ** NONE **
                                30D8  1540 ;
                                30D8  1541 ; IMPLICIT INPUTS::
                                30D8  1542 ;
                                30D8  1543 ;         ** NONE **
                                30D8  1544 ;
                                30D8  1545 ; OUTPUT PARAMETERS::
                                30D8  1546 ;
                                30D8  1547 ;         ** NONE **
                                30D8  1548 ;
                                30D8  1549 ; IMPLICIT OUTPUTS::
                                30D8  1550 ;
                                30D8  1551 ;         ** NONE **
                                30D8  1552 ;
                                30D8  1553 ; COMPLETION CODES::
                                30D8  1554 ;
                                30D8  1555 ;         ** NONE **
                                30D8  1556 ;
                                30D8  1557 ; SIDE EFFECTS::
                                30D8  1558 ;
                                30D8  1559 ;         ** NONE **
                                30D8  1560 ;
                                30D8  1561 ;--
                                30D8  1562
                                30D8  1563           $DS_BGNCLEAN
          52    0014'CF    D0   0002  1564           MOVL    BASE_ADDR,R2                         ; GET IDC ADDRESS
      62    000F0080 8F    D0   0007  1565           MOVL    #^XF0080,IDCCSR(R2)                  ; RESET ATTENTIONS
      62    01000080 8F    D0   000E  1566           MOVL    #<CSR$M_IR!CSR$M_CRDY>,IDCCSR(R2)    ; REST ANY PENDING INTERRUPTS
                                0015  1567           $DS_CHANNEL_S -                              ; DISABLE INTR
                                0015  1568             UNIT=          L$UNIT,-                    ; UNDER UNDER TEST
                                0015  1569             FUNC=          #CHC$_DSINT                 ; FUNCTION = DISABLE INTERRUPTS
                     53    D4   002A  1570           CLRL    R3                                   ; RESET LUN COUNTER
                                002C  1571 10$:      GET_LUN -                                    ; GET LOGCICAL UNIT
                                002C  1572             UNIT=          R3                          ;
                     53    DD   002C                 PUSHL   R3
         126C'CF     01    FB   002E                 CALLS   #1,GET_LUN$A
                  02 50    E8   0033  1573           BLBS    R0,20$                               ; IF LUN FOUND GOTO 20$
                  16 11    11   0036  1574           BRB     30$                                  ; ELSE GOTO 30$
 011C'CF    00000040 8F    C8   0038  1575 20$:      BISL2   #NOCHK$M_FTL,CONFIG                  ; SET NO FATAL ERROR ALLOWED HERE
         0823'CF     00    FB   0041  1576           CALLS   #0,SUBSYS_INIT                       ; INIT DRIVE
             011C'CF       D4   0046  1577           CLRL    CONFIG                               ; CLEAR FLAGS
            DE 53    04    F2   004A  1578           AOBLSS  #4,R3,10$                            ; DO NEXT DEVICE
                                004E  1579 30$:      $CANTIM_S -                                  ; CANCEL TIMERS
```

```
004E  1580        REQIDT=      #0              ; CANCEL TIMER #0
0057  1581     $CANTIM_S -                     ; CANCEL BOTH TIMER REQUESTS
0057  1582        REQIDT=      #1              ; CANCEL TIMER #1
0062  1583     $DS_RELBUF_S -                  ; RELEASE ANY ALLOCATED BUFFERS
0062  1584        PAGCNT=      #40,-           ; PAGE COUNT = 40 PAGES
0062  1585        RETADR=      VIR_WRT_BUF     ; ADDRESS TO RELEASE
0071  1586     $DS_ENDCLEAN
```

```
       30D8
       30D8   ;++
       30D8   ;         SUMMARY CODE SECTION.  THIS SECTION IS NOT USED BY THIS
       30D8   ;         PROGRAM; THEREFORE, NO CODE IS PRESENT.
       30D8   ;--
       30D8             $DS_BGNSUMMARY
   01  0002             NOP
       0003             $DS_ENDSUMMARY
```

```
                            30D8          .SBTTL  PRINT REGISTER SUBROUTINES
                        00000000          .PSECT  SUBROUTINES,LONG
                            0000
                            0000   ;++
                            0000   ; FUNCTIONAL DESCRIPTION:
                            0000   ;
                            0000   ;        THIS ROUTINE WILL BE CALLED BY THE DIAGNOSTIC SUPERVISOR TO REPORT
                            0000   ;        REGISTER DATA COMPARE ERRORS.IT IS CALLED VIA THE "PRLINK" ARG
                            0000   ;        SPECIFIED IN THE VARIOUS ERROR CALLS. THE ROUTINE WILL ALSO CALL
                            0000   ;        THE DUMP_REG ROUTINE TO REPORT ADDITIONAL REGISTER DATA.
                            0000   ;
                            0000   ; CALLING SEQ:
                            0000   ;
                            0000   ;        CALLED BY THE SUPERVISIOR
                            0000   ;
                            0000   ;        ERR$_P1(AP) = ADDRESS OF ADDITIONAL (FAO) ERROR MESSAGE
                            0000   ;        ERR$_P2(AP) = EXPECTED REGISTER DATA
                            0000   ;        ERR$_P3(AP) = RECEIVED (ACTUAL) REGISTER DATA
                            0000   ;        ERR$_P4(AP) = ADDRESS OF REGISTER BIT MNEUMONICS
                            0000   ;
                            0000   ; INPUT PARAMETERS:
                            0000   ;
                            0000   ;        *NONE*
                            0000   ;
                            0000   ; IMPLICIT INPUTS:
                            0000   ;
                            0000   ;        *NONE*
                            0000   ;
                            0000   ; OUTPUT PARAMETERS:
                            0000   ;        *NONE*
                            0000   ;
                            0000   ; IMPLICIT OUTPUT:
                            0000   ;
                            0000   ;        *NONE*
                            0000   ;
                            0000   ;
                            0000   ; COMPLETION CODES:
                            0000   ;
                            0000   ;        *NONE*
                            0000   ;
                            0000   ;--
                            0000
                            0000   REG_ERRPT::
                            0000   $DS_BGNMESSAGE  REGMASK = <R2,R3,R4,R5>
      52     14 BC    DE    0002          MOVAL   @ERR$_P1(AP),R2             ; GET ADDITIONAL MESSAGE
                14    13    0006          BEQL    10$                         ; BRANCH IF NO MESSAGE
                            0008          $DS_PRINTB_S    FMT_CRLF1
                            0013          $DS_PRINTB_S    (R2)
                            001C
      53     18 AC    D0    001C   10$:   MOVL    ERR$_P2(AP),R3              ; GET EXPECTED DATA
      54     1C AC    D0    0020          MOVL    ERR$_P3(AP),R4              ; GET RECEIVED DATA
      55     20 BC    DE    0024          MOVAL   @ERR$_P4(AP),R5             ; GET ADDR OF BIT MNEUM
                05    12    0028          BNEQ    20$                         ; NONE SPECIFIED
   55    0039'CF      DE    002A          MOVAL   W^COM_BIT32,R5              ; USE COMMON DEFS
                            002F
```

```
            52    54    53    CD  002F   1653 20$:     XORL3    R3,R4,R2                          ; CAL XOR OF GOOD & BAD
                                  0033   1654          $DS_PRINTB_S     FMT_CRLF1
                                  003E   1655          $DS_CVTREG_S     #31,R3,(R5),W^CHAR_BUFFER,#BUFLEN
                                  0061   1656          $DS_PRINTB_S     FMT_GOOD,R3,#CHAR_BUFFER
                                  0074   1657          $DS_CVTREG_S     #31,R4,(R5),W^CHAR_BUFFER,#BUFLEN
                                  0097   1658          $DS_PRINTB_S     FMT_BAD,R4,#CHAR_BUFFER
                                  00AA   1659
                                  00AA   1660          $DS_PRINTB_S     FMT_XOR,R2
                                  00B7   1661          $DS_PRINTB_S     FMT_CRLF1
         03 00A8'CF    06     E0  00C2   1662          BBS      #PRT$V_NONE,RGLIST,30$            ; CHECK PROGRAM CONFIGURATION
                    005E      30  00C8   1663          BSBW     DUMP_REG                          ; PRINT DEFINED REG DATA
                                  00CB   1664 30$:     $DS_PRINTB_S     FMT_CRLF1
                                  00D6   1665
                                  00D6   1666 $DS_ENDMESSAGE
```

```
                        00D7          .SBTTL  PRINT REGISTER ROUTINE
                        00D7     ;++
                        00D7     ; FUNCTIONAL DESCRIPTION:
                        00D7     ;
                        00D7     ;        THIS ROUTINE WILL BE CALLED BY THE DIAGNOSTIC SUPERVISOR TO PRINT
                        00D7     ;        ANY (OPTIONAL) ADDITIONAL MESSAGES WHILE REPORTING ERRORS. IT WILL
                        00D7     ;        ALSO CALL "DUMP_REG" ROUTINE TO PRINT THE CONTENTS OF VARIOUS
                        00D7     ;        (IDC-RL02-R80) REGISTERS.IT IS CALLED VIA THE "PRLINK" ARG
                        00D7     ;        PROVIDED IN THE VARIOUS ERROR CALLS
                        00D7     ;
                        00D7     ; CALLING SEQ:
                        00D7     ;
                        00D7     ;        CALLED BY THE SUPERVISOR
                        00D7     ;
                        00D7     ;        ERR$_P1(AP) = ADDRESS OF ADDITONAL (FAO) MESSAGE
                        00D7     ;        ERR$_P2(AP) = ADDRESS OF ADD'L TEXT TO BE INSERTED
                        00D7     ;                      IN FAO MESSAGE (IF SPECIFIED)
                        00D7     ;
                        00D7     ; INPUT PARAMETERS:
                        00D7     ;
                        00D7     ;        *NONE*
                        00D7     ;
                        00D7     ; IMPLICIT INPUTS:
                        00D7     ;
                        00D7     ;        *NONE*
                        00D7     ;
                        00D7     ; OUTPUT PARAMETERS:
                        00D7     ;
                        00D7     ;        *NONE*
                        00D7     ;
                        00D7     ; IMPLICIT OUTPUTS:
                        00D7     ;
                        00D7     ;        *NONE*
                        00D7     ;
                        00D7     ; COMPLETION CODES:
                        00D7     ;
                        00D7     ;        *NONE*
                        00D7     ;
                        00D7     ;--
                        00D7
                        00D7     PRINT_REG::
                        00D7     $DS_BGNMESSAGE  REGMASK = <>
       05    6C    D1   00D9          CMPL    (AP),#5                          ; ANY ADD'L MSG
             31    19   00DC          BLSS    20$                              ; BRANCH IF NOT
             17    14   00DE          BGTR    10$                              ; BRANCH IF ADD'L TEXT
                        00E0          $DS_PRINTB_S     FMT_CRLF1
                        00EB          $DS_PRINTB_S     @ERR$_P1(AP)
             18    11   00F5          BRB     20$
                        00F7     10$:  $DS_PRINTB_S     FMT_CRLF1
                        0102          $DS_PRINTB_S     @ERR$_P1(AP),ERR$_P2(AP)
                        010F     20$:  $DS_PRINTB_S     FMT_CRLF1
           000C    30   011A          BSBW    DUMP_REG                                 ; PRINT DEFINED REG DATA
                        011D          $DS_PRINTB_S     FMT_CRLF1
                        0128          $DS_ENDMESSAGE
```

```
                        0129          .SBTTL  DUMP REGISTER ROUTINE
                        0129  ;++
                        0129  ; FUNCTIONAL DESCRIPTION:
                        0129  ;
                        0129  ;       THIS ROUTINE WILL BE CALLED TO PRINT (REPORT) THE CONTENTS
                        0129  ;       OF VARIOUS IDC AND DRIVE STATUS MESSAGES DEFINED PRIOR TO
                        0129  ;       CALLING THE ROUTINE.
                        0129  ;
                        0129  ; CALLING SEQ:
                        0129  ;       BSBW    DUMP_REG
                        0129  ;
                        0129  ; INPUT PARAMETERS:
                        0129  ;
                        0129  ;       *NONE*
                        0129  ;
                        0129  ; IMPLICIT INPUTS:
                        0129  ;
                        0129  ;       RGLIST - THIS GLOBAL LOCATION MUST BE SET UP PRIOR TO
                        0129  ;                CALLING THE ROUTINE TO DEFINE THE REGS TO PRINT
                        0129  ;
                        0129  ; OUTPUT PARAMETERS:
                        0129  ;
                        0129  ;       *NONE*
                        0129  ;
                        0129  ; IMPLICIT OUTPUTS:
                        0129  ;
                        0129  ;       *NONE*
                        0129  ;
                        0129  ; COMPLETION CODES:
                        0129  ;
                        0129  ;       *NONE*
                        0129  ;
                        0129  ;--
                        0129
                        0129  DUMP_REG::
                        0129  .ENABL  LSB
           10     BB    0129          PUSHR   #^M<R4>                       ; SAVE REGISTER
   01D1'CF 00     FB    012B          CALLS   #0,PRINT_REG_HDR              ; PRINT REGISTER HEADERS
   01DF'CF 00     FB    0130          CALLS   #0,PRINT_CS                   ; PRINT CS REGISTER
   022C'CF 00     FB    0135          CALLS   #0,PRINT_BA                   ; PRINT BA REGISTER
   0276'CF 00     FB    013A          CALLS   #0,PRINT_BC                   ; PRINT BC REGISTER
   54   0000'CF   D0    013F          MOVL    DRIVE_TBL,R4                  ; GET ADDRESS OF DRIVE TABLE
                        0144          CASE    TYPE=B,SRC=OFST_DRVTYP(R4),-  ; GET DEVICE TYPE AND -
                        0144                  LIMIT=#RL02$T,DISPLIST=<-     ; PRINT DRIVE SPECIFIC INFO
                        0144                  DUMP_RL02,-                   ; RL02 REGISTER INFO
                        0144                  DUMP_R80>                     ; R80 REGISTER INFO
                        014C  DUMP_RL02:
                        014C
   02C0'CF 00     FB    014C          CALLS   #0,PRINT_DA_02                ; PRINT DA FOR DATA XFR
   03D6'CF 00     FB    0151          CALLS   #0,PRINT_MP_02                ; PRINT MP REGISTER
   046E'CF 00     FB    0156          CALLS   #0,PRINT_POS                  ; PRINT ECC POSITION REG
   04B8'CF 00     FB    015B          CALLS   #0,PRINT_PAT                  ; PRINT ECC PATTERN REG
05 00A8'CF 00     E1    0160          BBC     #PRT$V_MPR_1,RGLIST,32$       ; BRANCH - DONT PRINT
   0502'CF 00     FB    0166          CALLS   #0,PRINT_MP1_02               ; PRINT MP1 HDR DATA
31 00A8'CF 01     E1    016B  32$:    BBC     #PRT$V_MPR_2,RGLIST,38$       ; BRANCH - DONT PRINT
   059A'CF 00     FB    0171          CALLS   #0,PRINT_MP2_02               ; PRINT MP2 HDR DATA
```

```
                2A   11  0176       BRB     38$                                     ; GOTO 38$
                         0178
                         0178  DUMP_R80:
                         0178
       034E'CF   00  FB  0178       CALLS   #0,PRINT_DA_80                          ; PRINT DA FOR DATA XFR
       0424'CF   00  FB  017D       CALLS   #0,PRINT_MP_80                          ; PRINT MP REGISTER
       046E'CF   00  FB  0182       CALLS   #0,PRINT_POS                            ; PRINT ECC POSITION REG
       04B8'CF   00  FB  0187       CALLS   #0,PRINT_PAT                            ; PRINT ECC PATTERN REG
    05 00A8'CF   00  E1  018C       BBC     #PRT$V_MPR_1,RGLIST,34$                 ; BRANCH - DONT PRINT
       054F'CF   00  FB  0192       CALLS   #0,PRINT_MP1_80                         ; PRINT MP1 HDR DATA
    05 00A8'CF   01  E1  0197  34$: BBC     #PRT$V_MPR_2,RGLIST,38$                 ; BRANCH - DONT PRINT
       05DA'CF   00  FB  019D       CALLS   #0,PRINT_MP2_80                         ; PRINT MP2 HDR DATA
                         01A2
    05 00A8'CF   02  E1  01A2  38$: BBC     #PRT$V_MPR_3,RGLIST,40$                 ; BRANCH - DONT PRINT
       061A'CF   00  FB  01A8       CALLS   #0,PRINT_MP3                            ; PRINT MP3 HDR DATA
    05 00A8'CF   03  E5  01AD  40$: BBCC    #PRT$V_EXT_1,RGLIST,50$                 ; BRANCH - DONT PRINT
       065A'CF   00  FB  01B3       CALLS   #0,PRINT_EXT_1                          ; PRINT EXTENDED INFO
    05 00A8'CF   04  E5  01B8  50$: BBCC    #PRT$V_EXT_2,RGLIST,60$                 ; BRANCH - DONT PRINT
       06A6'CF   00  FB  01BE       CALLS   #0,PRINT_EXT_2                          ; PRINT ADDITIONAL INFO
                         01C3  60$: $DS_PRINTB_S    FMT_CRLF
                10   BA  01CE       POPR    #^M<R4>                                 ; RESTORE REGISTER
                     05  01D0       RSB                                             ; RETURN TO CALLING ROUTINE
                         01D1  .DSABL   LSB
                         01D1  PRINT_REG_HDR::
                   0000  01D1       .WORD   ^M<>
                         01D3       $DS_PRINTB_S    FMT_HDR
                     04  01DE       RET
                         01DF
                         01DF  PRINT_CS::
                   000C  01DF       .WORD   ^M<R2,R3>
    52    0014'CF    D0  01E1       MOVL    BASE_ADDR,R2                            ; GET BASE ADDRESS
    53    0120'CF    D0  01E6       MOVL    T.CS,R3                                 ; R3 <-- ACTUAL DATA
                         01EB       $DS_CVTREG_S    #31,R3,IDCCSR_BITS,CHAR_BUFFER,#BUFLEN,-
                         01EB                       CVT_DRVTYP,CVT_ERR_TABLE,CVT_FUN_TABLE
                         0216       $DS_PRINTB_S    FMT_CS,R2,R3,#CHAR_BUFFER
                     04  022B       RET
                         022C
                         022C  PRINT_BA::
                   000C  022C       .WORD   ^M<R2,R3>
    52    0014'CF    D0  022E       MOVL    BASE_ADDR,R2                            ; GET BASE ADDRESS
          52   04    C0  0233       ADDL2   #IDCBAR,R2                              ; CALC REG ADDRESS
    53    0124'CF    D0  0236       MOVL    T.BA,R3                                 ; R3 <-- ACTUAL DATA
                         023B       $DS_CVTREG_S    #31,R3,IDCBAR_BITS,CHAR_BUFFER,#BUFLEN
                         0260       $DS_PRINTB_S    FMT_BA,R2,R3,#CHAR_BUFFER
                     04  0275       RET
                         0276
                         0276  PRINT_BC::
                   000C  0276       .WORD   ^M<R2,R3>
    52    0014'CF    D0  0278       MOVL    BASE_ADDR,R2                            ; GET BASE ADDRESS
          52   08    C0  027D       ADDL2   #IDCCNT,R2                              ; CALC REG ADDRESS
    53    0128'CF    D0  0280       MOVL    T.BC,R3                                 ; R3 <-- ACTUAL DATA
                         0285       $DS_CVTREG_S    #31,R3,IDCCNT_BITS,CHAR_BUFFER,#BUFLEN
                         02AA       $DS_PRINTB_S    FMT_BC,R2,R3,#CHAR_BUFFER
                     04  02BF       RET
                         02C0
                         02C0  PRINT_DA_02::
```

```
                                 02C0  .ENABL LSB
                          007C   02C0              .WORD    ^M<R2,R3,R4,R5,R6>
           52   0014'CF    D0    02C2              MOVL     BASE_ADDR,R2                              ; GET BASE ADDRESS
                 52   0C   C0    02C7              ADDL2    #IDCDAR,R2                                ; CALC REG ADDRESS
           53   012C'CF    D0    02CA              MOVL     T.DA,R3                                   ; R3 <-- ACTUAL DATA
                      55   7C    02CF              CLRQ     R5                                        ; CLEAR CONVERSION ADDRESSES
      54   00FC'CF  FF 8F  9C    02D1              ROTL     #-1,CURR_CMD,R4                           ; SET UP FOR CASE INST
                                 02D8              CASE     TYPE=B,SRC=R4,LIMIT=#0,DISPLIST=<-
                                 02D8                       USE02_COMMON,-
                                 02D8                       USE02_DMA,-
                                 02D8                       USE02_GS,-
                                 02D8                       USE02_SEEK,-
                                 02D8                       USE02_COMMON,-
                                 02D8                       USE02_DMA,-
                                 02D8                       USE02_DMA,-
                                 02D8                       USE02_DMA>
                                 02EC
                                 02EC
                                 02EC  USE02_COMMON:
           54   0039'CF    DE    02EC              MOVAL    COM_BIT32,R4                              ; USE COMMON BITS
                      22   11    02F1              BRB      10$                                       ; GOTO 10$
                                 02F3  USE02_GS:
           54   03AA'CF    DE    02F3              MOVAL    DAR_GS02_BITS,R4                          ; USE GET STATUS BITS
                      1B   11    02F8              BRB      10$                                       ; GOTO 10$
                                 02FA  USE02_SEEK:
           54   02F4'CF    DE    02FA              MOVAL    DAR_SK02_BITS,R4                          ; USE SEEK BITS
           55   069A'CF    DE    02FF              MOVAL    CVT_SUR_TABLE,R5                          ; SET HEAD SELECT TABLE
           56   06A6'CF    DE    0304              MOVAL    CVT_DIR_TABLE,R6                          ; SET DIRECTION SELECT TABLE
                      0A   11    0309              BRB      10$                                       ; GOTO 10$
                                 030B  USE02_DMA:
           54   0294'CF    DE    030B              MOVAL    DAR_DMA02_BITS,R4                         ; USE DATA TRANSFER BITS
           55   069A'CF    DE    0310              MOVAL    CVT_SUR_TABLE,R5                          ; SET HEAD SELECT TABLE
                                 0315
                                 0315  10$:        $DS_CVTREG_S     #31,R3,(R4),CHAR_BUFFER,#BUFLEN,(R5),(R6)
                                 0338              $DS_PRINTB_S     FMT_DA,R2,R3,#CHAR_BUFFER
                           04    034D              RET
                                 034E  .DSABL LSB
                                 034E
                                 034E  PRINT_DA_80::
                                 034E  .ENABL LSB
                          007C   034E              .WORD    ^M<R2,R3,R4,R5,R6>
           52   0014'CF    D0    0350              MOVL     BASE_ADDR,R2                              ; GET BASE ADDRESS
                 52   0C   C0    0355              ADDL2    #IDCDAR,R2                                ; CALC REG ADDRESS
           53   012C'CF    D0    0358              MOVL     T.DA,R3                                   ; R3 <-- ACTUAL DATA
                      55   7C    035D              CLRQ     R5                                        ; CLEAR CONVERSION ADDRESSES
      54   00FC'CF  FF 8F  9C    035F              ROTL     #-1,CURR_CMD,R4                           ; SET UP FOR CASE INST
                                 0366              CASE     TYPE=B,SRC=R4,LIMIT=#0,DISPLIST=<-
                                 0366                       USE80_COMMON,-
                                 0366                       USE80_DMA,-
                                 0366                       USE80_COMMON,-
                                 0366                       USE80_SEEK,-
                                 0366                       USE80_COMMON,-
                                 0366                       USE80_DMA,-
                                 0366                       USE80_DMA,-
                                 0366                       USE80_DMA>
                                 037A
```

```
                              037A
                              037A  USE80_COMMON:
           54    0039'CF  DE  037A          MOVAL   COM_BIT32,R4                  ; USE COMMON BITS
                      1C  11  037F          BRB     10$                           ; GOTO 10$
                              0381  USE80_SEEK:
           54    0337'CF  DE  0381          MOVAL   DAR_SK80_BITS,R4              ; USE SEEK BITS
  00008040 8F    012C'CF  D1  0386          CMPL    T.DA,#^X8040                  ; IF COMMAND WAS NOT A RECAL
                      0C  12  038F          BNEQ    10$                           ; THEN GOTO 10$
           54    0370'CF  DE  0391          MOVAL   DAR_RCL80_BITS,R4             ; ELSE USE RECAL BITS
                      05  11  0396          BRB     10$                           ; GOTO 10$
                              0398  USE80_DMA:
           54    02C5'CF  DE  0398          MOVAL   DAR_DMA80_BITS,R4             ; USE DATA TRANSFER BITS
                              039D
                              039D  10$:    $DS_CVTREG_S    #31,R3,(R4),CHAR_BUFFER,#BUFLEN,(R5),(R6)
                              03C0          $DS_PRINTB_S    FMT_DA,R2,R3,#CHAR_BUFFER
                          04  03D5          RET
                              03D6  .DSABL LSB
                              03D6
                              03D6  PRINT_MP_02::
                        000C  03D6          .WORD   ^M<R2,R3>
           52    0014'CF  D0  03D8          MOVL    BASE_ADDR,R2                  ; GET BASE ADDRESS
                 52   10  C0  03DD          ADDL2   #IDCMPR,R2                    ; CALC REG ADDRESS
           53    0130'CF  D0  03E0          MOVL    T.MP,R3                       ; R3 <-- ACTUAL DATA
                              03E5          $DS_CVTREG_S    #31,R3,IDCMPR02_BITS,CHAR_BUFFER,#BUFLEN,-
                              03E5                          CVT_SUR_TABLE,CVT_STA_TABLE
                              040E          $DS_PRINTB_S    FMT_MP,R2,R3,#CHAR_BUFFER
                          04  0423          RET
                              0424
                              0424  PRINT_MP_80::
                        000C  0424          .WORD   ^M<R2,R3>
           52    0014'CF  D0  0426          MOVL    BASE_ADDR,R2                  ; GET BASE ADDRESS
                 52   10  C0  042B          ADDL2   #IDCMPR,R2                    ; CALC REG ADDRESS
           53    0130'CF  D0  042E          MOVL    T.MP,R3                       ; R3 <-- ACTUAL DATA
                              0433          $DS_CVTREG_S    #31,R3,IDCMPR80_BITS,CHAR_BUFFER,#BUFLEN
                              0458          $DS_PRINTB_S    FMT_MP,R2,R3,#CHAR_BUFFER
                          04  046D          RET
                              046E
                              046E  PRINT_POS::
                        000C  046E          .WORD   ^M<R2,R3>
           52    0014'CF  D0  0470          MOVL    BASE_ADDR,R2                  ; GET BASE ADDRESS
                 52   14  C0  0475          ADDL2   #IDCPOS,R2                    ; CALC REG ADDRESS
           53    014C'CF  D0  0478          MOVL    T.POS,R3                      ; R3 <-- ACTUAL DATA
                              047D          $DS_CVTREG_S    #31,R3,IDCPOS_BITS,CHAR_BUFFER,#BUFLEN
                              04A2          $DS_PRINTB_S    FMT_POS,R2,R3,#CHAR_BUFFER
                          04  04B7          RET
                              04B8
                              04B8  PRINT_PAT::
                        000C  04B8          .WORD   ^M<R2,R3>
           52    0014'CF  D0  04BA          MOVL    BASE_ADDR,R2                  ; GET BASE ADDRESS
                 52   18  C0  04BF          ADDL2   #IDCPAT,R2                    ; CALC REG ADDRESS
           53    0148'CF  D0  04C2          MOVL    T.PAT,R3                      ; R3 <-- ACTUAL DATA
                              04C7          $DS_CVTREG_S    #31,R3,IDCPAT_BITS,CHAR_BUFFER,#BUFLEN
                              04EC          $DS_PRINTB_S    FMT_PAT,R2,R3,#CHAR_BUFFER
                          04  0501          RET
                              0502
                              0502  PRINT_MP1_02::
```

```
                           0008  0502              .WORD    ^M<R3>
              53  0134'CF    D0  0504              MOVL     T.MP1,R3                        ; R3 <-- ACTUAL DATA
                                 0509              $DS_CVTREG_S     #15,R3,HDR1_02_BITS,CHAR_BUFFER,#BUFLEN,-
                                 0509                               CVT_SUR_TABLE
                                 0530              $DS_PRINTB_S     FMT_CRLF1
                                 053B              $DS_PRINTB_S     FMT_MP1_02,R3,#CHAR_BUFFER
                             04  054E              RET
                                 054F
                                 054F      PRINT_MP1_80::
                           0008  054F              .WORD    ^M<R3>
              53  0134'CF    D0  0551              MOVL     T.MP1,R3                        ; R3 <-- ACTUAL DATA
                                 0556              $DS_CVTREG_S     #15,R3,HDR1_80_BITS,CHAR_BUFFER,#BUFLEN
                                 057B              $DS_PRINTB_S     FMT_CRLF1
                                 0586              $DS_PRINTB_S     FMT_MP1_80,R3,#CHAR_BUFFER
                             04  0599              RET
                                 059A
                                 059A      PRINT_MP2_02::
                           0008  059A              .WORD    ^M<R3>
              53  0138'CF    D0  059C              MOVL     T.MP2,R3                        ; R3 <-- ACTUAL DATA
                                 05A1              $DS_CVTREG_S     #15,R3,COM_BIT16,CHAR_BUFFER,#BUFLEN
                                 05C6              $DS_PRINTB_S     FMT_MP2_02,R3,#CHAR_BUFFER
                             04  05D9              RET
                                 05DA
                                 05DA      PRINT_MP2_80::
                           0008  05DA              .WORD    ^M<R3>
              53  0138'CF    D0  05DC              MOVL     T.MP2,R3                        ; R3 <-- ACTUAL DATA
                                 05E1              $DS_CVTREG_S     #15,R3,HDR2_80_BITS,CHAR_BUFFER,#BUFLEN
                                 0606              $DS_PRINTB_S     FMT_MP2_80,R3,#CHAR_BUFFER
                             04  0619              RET
                                 061A
                                 061A      PRINT_MP3::
                           0008  061A              .WORD    ^M<R3>
              53  013C'CF    D0  061C              MOVL     T.MP3,R3                        ; R3 <-- ACTUAL DATA
                                 0621              $DS_CVTREG_S     #15,R3,COM_BIT16,CHAR_BUFFER,#BUFLEN
                                 0646              $DS_PRINTB_S     FMT_MP3,R3,#CHAR_BUFFER
                             04  0659              RET
                                 065A
                                 065A      PRINT_EXT_1::
                           0008  065A              .WORD    ^M<R3>
              53  0100'CF    D0  065C              MOVL     DIFF_WORD,R3                    ; R3 <-- DIFFERENCE WORD
         00E0'CF  00E4'CF    D1  0661              CMPL     DEST_CYLN,PREV_CYLN             ; FIND DIRECTION OF SEEK
                       05    18  0668              BGEQ     5$                              ; BRANCH IF FWD SK OR ZERO SK
              53  0100'CF    CE  066A              MNEGL    DIFF_WORD,R3                    ; ELSE CHANGE SIGN OF SK TO REV
                                 066F      5$:     $DS_PRINTX_S     FMT_CRLF1
                                 067A              $DS_PRINTX_S     FMT_EXT_1A,PREV_CYLN
                                 0689              $DS_PRINTX_S     FMT_EXT_1B,R3
                                 0696              $DS_PRINTX_S     FMT_EXT_1C,DEST_CYLN
                             04  06A5              RET
                                 06A6
                                 06A6      PRINT_EXT_2::
                           0000  06A6              .WORD    ^M<>
                                 06A8              $DS_PRINTX_S     FMT_EXT_2,CURR_CYLN
                             04  06B7              RET
```

```
                               06B8       .SBTTL  DATA ERROR PRINT ROUTINE
                               06B8
                               06B8  ;++
                               06B8  ;
                               06B8  ; FUNCTIONAL DESCRIPTION:
                               06B8  ;
                               06B8  ; THIS IS A ROUTINE TO PRINT OUT DATA ERRORS THAT WERE DETECTED BY THE
                               06B8  ; BY THE CMPDATA ROUTINE.THE ROUTINE WILL PRINT THE FAILING MEMEORY
                               06B8  ; ADDRESS,GOOD DATA WORD (16 BIT WORD),BAD WORD,XOR DATA AND THE FAILING
                               06B8  ; WORD NUMBER IN THE TRANSFER .THIS SHOULD INDICATE WHERE IN THE SECTOR
                               06B8  ; THAT WAS READ , THE DATA ERROR WAS DETECTED.
                               06B8  ;
                               06B8  ; CALLING SEQ:
                               06B8  ;
                               06B8  ; CALLED BY THE SUPERVISOR BY VIA THE PRLINK ARG IN THE ERROR CALL
                               06B8  ;
                               06B8  ; INPUT PARAMETERS:
                               06B8  ;
                               06B8  ;       *NONE*
                               06B8  ;
                               06B8  ; IMPLICIT INPUTS:
                               06B8  ;
                               06B8  ;       GOOD_DATA = ADDRESS OF EXPECTED DATA
                               06B8  ;        BAD_DATA = ADDRESS OF RECEIVED DATA
                               06B8  ;            CYLN = TRANSFER CYLINDER
                               06B8  ;         SURFACE = TRANSFER HEAD
                               06B8  ;          SECTOR = TRANSFER SECTOR
                               06B8  ;         ERR_CNT = # OF WORDS IN ERROR
                               06B8  ;
                               06B8  ; OUTPUT PARAMETERS:
                               06B8  ;
                               06B8  ;       *NONE*
                               06B8  ;
                               06B8  ; IMPLICIT OUTPUTS:
                               06B8  ;
                               06B8  ;       *NONE*
                               06B8  ;
                               06B8  ; COMPLETION CODES:
                               06B8  ;
                               06B8  ;       *NONE*
                               06B8  ;
                               06B8  ;--
                               06B8
                               06B8
                               06B8  PRINT_DATA_ERR::
                               06B8  $DS_BGNMESSAGE  REGMASK = <R2,R3,R4,R5,R6,R7,R8>
                               06BA
                   52     D4   06BA        CLRL    R2                      ; USE R2 AS INDEX COUNT
          58  01B8'CF     D0   06BC        MOVL    ERR_CNT,R8              ; GET # OF ERRORS
              53   03     9A   06C1        MOVZBL  #3,R3                   ; SET PRINT LIMIT
             FA61  CF     16   06C4        JSB     DUMP_REG                ; DUMP IDC REGISTERS
                               06C8        $DS_PRINTB_S    FMT_HDR_A,CYLN,SURFACE,SECTOR
                               06DF        $DS_PRINTB_S    FMT_WDCNT,WORD_COUNT
                               06EE        $DS_PRINTB_S    FMT_HEADER1
    54     0190'CF42      D0   06F9  10$:  MOVL    W^GOOD_DATA[R2],R4      ; R4 = GOOD DATA ADDRESS
    55     01A4'CF42      D0   06FF        MOVL    W^BAD_DATA[R2],R5       ; R5 = BAD DATA ADDRESS
```

```
         56   65   64  CD  0705  2057         XORL3   (R4),(R5),R6              ; R6 = XOR
                           0709  2058
                           0709  2059         ;
                           0709  2060         ; CALCULATE FAILING WORD #
                           0709  2061         ;
                           0709  2062
         57   00A4'CF  D0  0709  2063         MOVL    BUFF_INUSE,R7             ; GET DATA BUFFER ADDRESS
         57   55   57  C3  070E  2064         SUBL3   R7,R5,R7                  ; GET OFFSET INTO BUFFER
    57   57   FF 8F    78  0712  2065         ASHL    #-1,R7,R7                 ; DIV BY 2 = WORD #
                           0717  2066
                           0717  2067         $DS_PRINTB_S    W^FMT_DATA,R5,(R4),(R5),R6,R7
         58   58   01  C3  072C  2068         SUBL3   #1,R8,R8                  ; DECR ERROR COUNT
                   04  13  0730  2069         BEQL    15$                       ; EXIT IF DONE
           C3 52   53  F2  0732  2070         AOBLSS  R3,R2,10$
                           0736  2071 15$:    $DS_PRINTB_S    FMT_ERRCOUNT,W^ERR_CNT
                       04  0745  2072         RET
```

```
                                0746          .SBTTL  DATA COMPARE ROUTINE
                                0746     ;++
                                0746     ;
                                0746     ; FUNCTIONAL DESCRIPTION:
                                0746     ;
                                0746     ;        THIS ROUTINE WILL DO A WORD BY WORD DATA COMPARISON ON THE DATA
                                0746     ;        BUFFERS SPECIFIED BY THE ARG LIST.
                                0746     ;
                                0746     ;
                                0746     ;        CALLING SEQUENCE:
                                0746     ;
                                0746     ;                PUSHAL  EXPECTED           ; ADDRESS OF EXPECTED DATA
                                0746     ;                PUSHAL  ACTUAL             ; ADDRESS OF ACTUAL DATA
                                0746     ;                PUSHL   WORD_COUNT         ; WORD COUNT
                                0746     ;                CALLS   #3,CMPDATA
                                0746     ;
                                0746     ; INPUT PARAMETERS:
                                0746     ;
                                0746     ;         12(AP) = ADDRESS OF EXPECTED DATA
                                0746     ;          8(AP) = ADDRESS OF RECEIVED DATA
                                0746     ;        4(AP) = # WORDS TO COMPARE
                                0746     ;
                                0746     ; IMPLICIT INPUTS:
                                0746     ;
                                0746     ;        *NONE*
                                0746     ;
                                0746     ; OUTPUT PARAMETERS:
                                0746     ;
                                0746     ;        *NONE*
                                0746     ;
                                0746     ; IMPLICIT OUTPUTS:
                                0746     ;
                                0746     ;        *NONE*
                                0746     ;
                                0746     ; COMPLETION CODES:
                                0746     ;
                                0746     ;        R0  =  1            => NO ERRORS
                                0746     ;        R0  =  2            => ERROR(S) DETECTED
                                0746     ;
                                0746     ;        IF R0 = 2 THEN, R1 = # OF ERRORS DETECTED
                                0746     ;
                                0746     ;--
                                0746
                                0746     CMPDATA::
                          007C  0746          .WORD   ^M<R2,R3,R4,R5,R6>
        52    04 AC       D0    0748          MOVL    4(AP),R2                   ; R2 = WORD COUNT
        53    08 AC       D0    074C          MOVL    8(AP),R3                   ; R3 = ADDRESS OF RECEIVEDDATA
   00A4'CF       53       D0    0750          MOVL    R3,BUFF_INUSE              ; SAVE ADRS OF BUFF FOR ERR REPORT
        54    0C AC       D0    0755          MOVL    12(AP),R4                  ; R4 =ADDRESS OF EXPECTED DATA
                 56       D4    0759          CLRL    R6                         ; CLEAR INDEX
      0190'CF46           D4    075B     15$: CLRL    W^GOOD_DATA[R6]            ; CLEAR GOOD BUFFER
      01A4'CF46           D4    0760          CLRL    W^BAD_DATA[R6]             ; CLEAR BAD BUFFER
     F2 56       03       F2    0765          AOBLSS  #3,R6,15$                  ; LOOP BACK
                 51       D4    0769          CLRL    R1                         ; CLEAR ERROR COUNT
                 55       D4    076B          CLRL    R5                         ; CLEAR INDEX
    6445       6345       B1    076D     10$: CMPW    (R3)[R5],(R4)[R5]          ; DO THE COMPARE
```

```
                 1B   13  0772  2130          BEQL    25$                     ; BRANCH IF OK
          03     51   D1  0774  2131          CMPL    R1,#3                   ; LIMIT EXCEEDED?
                 14   18  0777  2132          BGEQ    20$                     ; BRANCH IF LIMIT EXCEEDED
        50     6345   3E  0779  2133          MOVAW   (R3)[R5],R0             ; SAVE ADDRESS OF BAD DATA
    01A4'CF41    50   D0  077D  2134          MOVL    R0,W^BAD_DATA[R1]       ;
        50     6445   3E  0783  2135          MOVAW   (R4)[R5],R0             ; SAVE ADDRESS OF GOOD DATA
    0190'CF41    50   D0  0787  2136          MOVL    R0,W^GOOD_DATA[R1]      ;
                 51   D6  078D  2137 20$:     INCL    R1                      ; BUMP ERROR COUNT
       DA 55     52   F2  078F  2138 25$:     AOBLSS  R2,R5,10$               ; LOOP BACK
                 51   D5  0793  2139          TSTL    R1                      ; ANY ERRORS?
                 0A   13  0795  2140          BEQL    30$                     ; BRANCH IF NONE
      01B8'CF    51   D0  0797  2141          MOVL    R1,W^ERR_CNT            ; SAVE ERROR COUNT
          50     02   9A  079C  2142          MOVZBL  #2,R0                   ; SET ERROR FLAG
                 03   11  079F  2143          BRB     CMPDATAX                ; AND EXIT
          50     01   D0  07A1  2144 30$:     MOVL    #1,R0                   ; SET SUCCESS FLAG
                          07A4  2145
                          07A4  2146 CMPDATAX:
                      04  07A4  2147          RET
                          07A5  2148
```

```
                                  07A5          .SBTTL DRIVE INIT ROUTINE
                                  07A5    ;++
                                  07A5    ; FUNCTIONAL DESCRIPTION::
                                  07A5    ;
                                  07A5    ; THIS ROUTINE IS CALLED FROM THE INIT CODE TO CLEAR ALL DRIVE
                                  07A5    ; FAULTS EVEN ON DRIVES THAT ARE NOT SELECTED FOR TESTING. THIS
                                  07A5    ; IS NECESSARY TO PREVENT UNSOLICITED ATTENTION INTERRUPTS DURING
                                  07A5    ; INTERRUPT AND TIMING TESTS. DRIVE CLEAR FUNCTIONS ARE USED FOR
                                  07A5    ; RL02 DRIVES AND RECAL OPERATIONS ARE USED FOR THE R80.
                                  07A5    ;
                                  07A5    ; CALLING SEQUENCE::
                                  07A5    ;
                                  07A5    ;       CALLS   #0,DRIVE_INIT$A
                                  07A5    ;
                                  07A5    ; INPUT PARAMETERS::
                                  07A5    ;
                                  07A5    ;       ** NONE **
                                  07A5    ;
                                  07A5    ; IMPLICIT INPUTS::
                                  07A5    ;
                                  07A5    ;       ** NONE **
                                  07A5    ;
                                  07A5    ; OUTPUT PARAMETERS::
                                  07A5    ;
                                  07A5    ;       ** NONE **
                                  07A5    ;
                                  07A5    ; IMPLICIT OUTPUTS::
                                  07A5    ;
                                  07A5    ;       ** NONE **
                                  07A5    ;
                                  07A5    ; COMPLETION CODES::
                                  07A5    ;
                                  07A5    ;       R0 = 1 SUCCESS
                                  07A5    ;       R0 = 2 ERROR
                                  07A5    ;       R1 = ADDRESS OF AN APPROPRIATE ERROR MESSAGE
                                  07A5    ;
                                  07A5    ; SIDE EFFECTS::
                                  07A5    ;
                                  07A5    ; DRIVE INIT WILL BE DONE TO ALL RL02 DRIVES AND
                                  07A5    ; DRIVE RECAL WILL BE DONE TO ALL R80 DRIVES.
                                  07A5    ;
                                  07A5    ;--
                                  07A5
                                  07A5    DRIVE_INIT$A::
                            003C  07A5          .WORD   ^M<R2,R3,R4,R5>
                     53       D4  07A7          CLRL    R3                              ; RESET DRIVE SELECT NUMBER
                     55       D4  07A9    10$:  CLRL    R5                              ; RESET COUNTER
          52   0014'CF        D0  07AB          MOVL    BASE_ADDR,R2                    ; GET IDC BASE ADDRESS
      62   000F0080 8F        D0  07B0          MOVL    #^XF0080,IDCCSR(R2)             ; RESET ATTENTION BITS
      62   01000000 8F        C8  07B7          BISL2   #CSR$M_IR,IDCCSR(R2)            ; CLEAR OUTSTANDING INTERRUPTS
  62  53   00000080 8F        C9  07BE          BISL3   #CSR$M_CRDY,R3,IDCCSR(R2)       ; SELECT DRIVE
                54   62       D0  07C6          MOVL    IDCCSR(R2),R4                   ; GET CSR CONTENTS
             26 54   1A       E0  07C9          BBS     #CSR$V_R80,R4,20$               ; IF R80 TYPE GOTO 20$
  62  53   00000084 8F        C9  07CD          BISL3   #<CSR$M_CRDY!GETSTAT>,R3,IDCCSR(R2) ; LOAD FOR CMD DECODE
             10 A2   0B       D0  07D5          MOVL    #^XB,IDCMPR(R2)                 ; SET UP FOR DRIVE RESET
      62   00000080 8F        CA  07D9          BICL2   #CSR$M_CRDY,IDCCSR(R2)          ; START DRIVE RESET FUNCTION
```

```
        62    00000080 8F   D3  07E0   2206 15$:    BITL    #CSR$M_CRDY,IDCCSR(R2)              ; IF REST DONE
                       28   12  07E7   2207         BNEQ    30$                                 ; THEN GOTO 30$
     EF 55    0000FFFF 8F   F2  07E9   2208         AOBLSS  #^XFFFF,R5,15$                      ; TRY AGAIN
                       1E   11  07F1   2209         BRB     30$                                 ; GOTO 30$
   62    53   00000086 8F   C9  07F3   2210 20$:    BISL3   #<CSR$M_CRDY!RECAL>,R3,IDCCSR(R2)   ; LOAD FOR CMD DECODE
                 0C A2  00  D2  07FB   2211         MCOML   #0,IDCDAR(R2)                       ; SET DAR TO RECAL FUNCTION
        62    00000080 8F   CA  07FF   2212         BICL2   #CSR$M_CRDY,IDCCSR(R2)              ; START DRIVE RECAL
                                0806   2213         $DS_WAITMS_S    #20                         ; WAIT FOR RECAL TO COMPLETE
  FF8E 53     0100 8F   0300 8F   3D  0811   2214 30$:    ACBW    #<BIT9!BIT8>,#BIT8,R3,10$   ; SELECT NEXT DRIVE
        62    000F0080 8F   D0  081B   2215         MOVL    #^XF0080,IDCCSR(R2)                 ; RESET ATTENTION BITS
                            04  0822   2216         RET                                         ; RETURN TO CALLER
```

```
                              0823        .SBTTL SUBSYSTEM INITIALIZE ROUTINE
                              0823  ;++
                              0823  ; FUNCTIONAL DESCRIPTION::
                              0823  ;
                              0823  ;       THIS ROUTINE WILL ISSUE A NO OPERATION COMMAND TO CYCLE
                              0823  ;       THE IDC MICROCODE THEN A GET STATUS FUNCTION IS
                              0823  ;       EXECUTED WITH THE RST BIT SET IN THE IDCDAR TO CLEAR ANY DRIVE
                              0823  ;       ERRORS.  THE IDCCSR REGISTER IS CHECKED TO ASSURE CONTROLLER
                              0823  ;       READY AND DRIVE READY ARE SET AND THAT NO ERROR BITS ARE SET.
                              0823  ;       DRIVE STATUS IS CHECKED FOR ANY ERRORS AND THAT THE DRIVE
                              0823  ;       IS IN A READY TO TEST STATE (LOCKED ON TRACK).
                              0823  ;
                              0823  ; CALLING SEQUENCE::
                              0823  ;
                              0823  ;       CALLS   #0,SUBSYS_INIT
                              0823  ;
                              0823  ; INPUT PARAMETERS::
                              0823  ;
                              0823  ;       ** NONE **
                              0823  ;
                              0823  ; IMPLICIT INPUTS::
                              0823  ;
                              0823  ;       ** NONE **
                              0823  ;
                              0823  ; OUTPUT PARAMETERS::
                              0823  ;
                              0823  ;       ** NONE **
                              0823  ;
                              0823  ; IMPLICIT OUTPUTS::
                              0823  ;
                              0823  ;       ** NONE **
                              0823  ;
                              0823  ; COMPLETION CODES::
                              0823  ;
                              0823  ;       R0 = 1 SUCCESS
                              0823  ;       R0 = 2 ERROR
                              0823  ;       R1 = ADDRESS OF AN APPROPRIATE ERROR MESSAGE
                              0823  ;
                              0823  ; SIDE EFFECTS::
                              0823  ;
                              0823  ;       ** NONE **
                              0823  ;
                              0823  ;--
                              0823
                              0823  SUBSYS_INIT::
                        00F8  0823        .WORD   ^M<R3,R4,R5,R6,R7>             ; REGISTER SAVE MASK
         52   0014'CF    D0  0825        MOVL    BASE_ADDR,R2                   ; R2 <-- IDC BASE ADDRESS
         55   0000'CF    D0  082A        MOVL    DRIVE_TBL,R5                   ; GET ADDRESS OF DRIVE TABLE
           56    50 A5   D0  082F        MOVL    OFST_DRVRST(R5),R6             ; GET STATUS CMD BITS
           57    4C A5   D0  0833        MOVL    OFST_DRVERR(R5),R7             ; GET DRIVE ERROR BIT MASK
                    53   D4  0837        CLRL    R3                             ; CLEAR R3
         00A0'CF    00   D0  0839        MOVL    #NO_OP,HOLD_CS                 ; DO A NO OP TO CYCLE FIRMWARE
 00A0'CF    0034'CF      C8  083E        BISL2   P$UNIT,HOLD_CS                 ; OR IN THE UNIT NUMBER
         62   00A0'CF    D0  0845        MOVL    HOLD_CS,IDCCSR(R2)             ; EXECUTE NO OP CMD
                              084A        $DS_WAITUS_S    #10                    ; WAIT A BIT
                              0855  10$:  $SETIMR_S       #33,TIME_300MS,,#1     ; START TIMER
```

```
         10 A2   56   D0  0866  2274          MOVL    R6,IDCMPR(R2)                      ; IDCMPR = GET STAT AND DRV RESET
      00FC'CF    04   D0  086A  2275          MOVL    #GETSTAT,CURR_CMD                  ; SET CURRENT COMMAND
           53    04   D0  086F  2276          MOVL    #GETSTAT,R3                        ; DEFINE COMMAND
      53  0034'CF     C8  0872  2277          BISL2   P$UNIT,R3                          ; OR IN DRIVE NUMBER
           62    53   D0  0877  2278          MOVL    R3,IDCCSR(R2)                      ; EXECUTE GET STATUS CMD
         08EF'CF     16   087A  2279 20$:     JSB     RD_IDCREGS                         ; UPDATE IDC REGISTERS
   1D 0120'CF    07   E0  087E  2280          BBS     #CSR$V_CRDY,T.CS,30$               ; BRANCH IF CRDY SET
                          0884  2281          $READEF_S      #33,TIME_OUT                ; GET EVENT FLAGS
   E3 0080'CF    01   E1  0891  2282          BBC     #1,TIME_OUT,20$                    ; BRANCH IF NO TIME OUT
      51    17C2'CF   DE  0897  2283          MOVAL   MSG11_GS_CRDY,R1                   ; R1 <-- ERROR MSG
           50    02   D0  089C  2284          MOVL    #2,R0                              ; R0 <-- FAILURE FLAG
                 4D   11  089F  2285          BRB     SUBSYS_INIT_X                      ; EXIT ROUTINE
                          08A1  2286 30$:     $CANTIM_S       #1                         ; CANCEL TIMER
           50    01   D0  08AC  2287          MOVL    #1,R0                              ; INIT RETURN STATUS = OK
      011C'CF    02   E4  08AF  2288          BBSC    #NOCHK$V_DRDY,CONFIG,-             ; SKIP IF NOT CHECKING DRDY BIT
                 24       08B4  2289                  45$
                 00   DD  08B5  2290          PUSHL   #0                                 ; DUMMY DESTINATION
                 00   DD  08B7  2291          PUSHL   #0                                 ; DUMMY NEXT COMMAND
      0B60'CF    02   FB  08B9  2292          CALLS   #2,RESTORE_DRIVE                   ; RESTORE DRIVE IN NECESSARY
   0A 0120'CF    00   E0  08BE  2293          BBS     #CSR$V_DRDY,T.CS,40$               ; BR IF DRIVE RDY SET
      51    17F6'CF   DE  08C4  2294          MOVAL   MSG12_GS_DRDY,R1                   ; R1 <-- ERROR MSG
           50    02   D0  08C9  2295          MOVL    #2,R0                              ; R0 <-- FAILURE FLAG
                 20   11  08CC  2296          BRB     SUBSYS_INIT_X                      ; EXIT ROUTINE
0120'CF   0000FC00 8F D3  08CE  2297 40$:     BITL    #<CSR$M_ERR!CSR$M_DE!CSR$M_NXM!- ; ANY ERRORS STILL SET?
                          08D7  2298                  CSR$M_HNF!CSR$M_DCK!CSR$M_OPI>,T.CS ;
                 0D   12  08D7  2299          BNEQ    50$                                ; BRANCH IF YES
      011C'CF    03   E4  08D9  2300 45$:     BBSC    #NOCHK$V_DERR,CONFIG,-             ; SKIP IF NOT CHECKING DRV ERRORS
                 0F       08DE  2301                  SUBSYS_INIT_X
      0130'CF    57   D3  08DF  2302          BITL    R7,T.MP                            ; DO WE HAVE GOOD DRIVE STATUS
                 08   13  08E4  2303          BEQL    SUBSYS_INIT_X                      ; YES EXIT ROUTINE
      51    1810'CF   DE  08E6  2304 50$:     MOVAL   MSG13_INI_ERR,R1                   ; R1 <-- ERROR MSG
           50    02   D0  08EB  2305          MOVL    #2,R0                              ; R0 <-- FAILURE FLAG
                          08EE  2306 SUBSYS_INIT_X:
                 04       08EE  2307          RET
```

```
                              08EF          .SBTTL  READ IDC REGISTERS ROUTINE
                              08EF    ;++
                              08EF    ; FUNCTIONAL DESCRIPTION::
                              08EF    ;
                              08EF    ;       THIS ROUTINE WILL READ ALL THE INTERNAL IDC REGISTERS
                              08EF    ;       AND STORE THEM IN THE GLOBAL DATA SECTION OF MEMORY.  THE
                              08EF    ;       LOCATIONS ARE ASSIGNED TAGS WHICH CONTAIN A REGISTER
                              08EF    ;       MNEUMONIC PLUS A PREFIX.  EX: T.CS = IDCCSR RECEIVED DATA.
                              08EF    ;
                              08EF    ; CALLING SEQUENCE::
                              08EF    ;
                              08EF    ;       JSB     RD_IDCREGS
                              08EF    ;
                              08EF    ; INPUT PARAMETERS::
                              08EF    ;
                              08EF    ;       ** NONE **
                              08EF    ;
                              08EF    ; IMPLICIT INPUTS::
                              08EF    ;
                              08EF    ;       ** NONE **
                              08EF    ;
                              08EF    ; OUTPUT PARAMETERS::
                              08EF    ;
                              08EF    ;       ** NONE **
                              08EF    ;
                              08EF    ; IMPLICIT OUTPUTS::
                              08EF    ;
                              08EF    ;       ** NONE **
                              08EF    ;
                              08EF    ; COMPLETION CODES::
                              08EF    ;
                              08EF    ;       ** NONE **
                              08EF    ;
                              08EF    ; SIDE EFFECTS::
                              08EF    ;
                              08EF    ;       ** NONE **
                              08EF    ;
                              08EF    ;--
                              08EF
                              08EF    RD_IDCREGS::
                  18    BB    08EF            PUSHR   #^M<R3,R4>                      ; REGISTER SAVE MASK
        0120'CF   62    D0    08F1            MOVL    IDCCSR(R2),T.CS                 ; READ CONTROL AND STATUS REG
     0124'CF   04 A2    D0    08F6            MOVL    IDCBAR(R2),T.BA                 ; READ BUS ADDRESS REGISTER
     0128'CF   08 A2    D0    08FC            MOVL    IDCCNT(R2),T.BC                 ; READ BYTE COUNT REGISTER
     012C'CF   0C A2    D0    0902            MOVL    IDCDAR(R2),T.DA                 ; READ DISK ADDRESS REGISTER
     014C'CF   14 A2    D0    0908            MOVL    IDCPOS(R2),T.POS                ; READ ECC PATTERN REG
     0148'CF   18 A2    D0    090E            MOVL    IDCPAT(R2),T.PAT                ; READ ECC PATTERN REG
     49 0120'CF   07    E1    0914            BBC     #CSR$V_CRDY,T.CS,RD_IDCREGS_X   ; IF RDY NOT SET DONT READ MP
        00FC'CF   08    D1    091A            CMPL    #READH,CURR_CMD                 ; IS THIS A READ HDR CMD
                  3C    12    091F            BNEQ    10$                             ; BRANCH IF NOT
                              0921            ;++
                              0921            ; HERE IT IS NECESSARY TO CHECK WHETHER THE
                              0921            ; CURRENT DRIVE IS AND RL02 OR AN R80 SINCE
                              0921            ; FOR A READ HEADER COMMAND THE IDCMPR REGISTER
                              0921            ; IS READ TWICE FOR THE RL02, WHILE THE R80
                              0921            ; IS READ THREE TIMES.
```

```
                            0921  2365          ;--
          54   0000'CF  D0  0921  2366          MOVL    DRIVE_TBL,R4                    ; GET DRIVE INFO TABLE
               02   64  91  0926  2367          CMPB    OFST_DRVTYP(R4),#R80$T          ; IF DRIVE IS AN R80
                    0F  13  0929  2368          BEQL    5$                              ; THEN GOTO 5$
     0134'CF    10  A2  D0  092B  2369          MOVL    IDCMPR(R2),T.MP1                ; GET FIRST 2 RL HEADER WORDS
0138'CF    0136'CF      3C  0931  2370          MOVZWL  T.MP1+2,T.MP2                   ; LOAD PSEUDO SECOND WORD
                    0C  11  0938  2371          BRB     7$                              ; GOTO 7$
     0134'CF    10  A2  D0  093A  2372  5$:     MOVL    IDCMPR(R2),T.MP1                ; READ MULTI PURPOSE REG 1ST WORD
     0138'CF    10  A2  D0  0940  2373          MOVL    IDCMPR(R2),T.MP2                ; READ MULTI PURPOSE REG 2ND WORD
     013C'CF    10  A2  D0  0946  2374  7$:     MOVL    IDCMPR(R2),T.MP3                ; READ MULTI PURPOSE REG 3RD WORD
          53   0140'CF  DE  094C  2375          MOVAL   T.HDR,R3                        ; GET SAVE ADDRESS FOR HEADER
          83   0134'CF  B0  0951  2376          MOVW    T.MP1,(R3)+                     ; SAVE FIRST HEADER WORD
          63   0138'CF  B0  0956  2377          MOVW    T.MP2,(R3)                      ; SAVE SECOND HEADER WORD
                    06  11  095B  2378          BRB     RD_IDCREGS_X                    ; EXIT ROUTINE
     0130'CF    10  A2  D0  095D  2379  10$:    MOVL    IDCMPR(R2),T.MP                 ; GET DRIVE STATUS WORD
                            0963  2380  RD_IDCREGS_X:
                    18  BA  0963  2381          POPR    #^M<R3,R4>                              ; RESTORE REGISTERS
                        05  0965  2382          RSB
```

```
0966          .SBTTL  START COMMAND ROUTINE
0966    ;++
0966    ; FUNCTIONAL DESCRIPTION::
0966    ;
0966    ;       THIS PURPOSE OF THIS ROUTINE IS TO LOAD THE IDC REGISTERS
0966    ;       IN ORDER TO PERFORM A SPECIFIC FUNCTION.  THE ROUTINE IS ALSO
0966    ;       USED TO SET CERTAIN PROGRAM CONFIGURATION FLAGS BASED UPON
0966    ;       THE CURRENT COMMAND AND SAVE PERTENANT INFORMATION REGARDING
0966    ;       THE FUNCTION EXECUTION PARAMETERS FOR ERROR REPORTING
0966    ;       PURPOSES.
0966    ;
0966    ;       PROGRAMMERS NOTE: IT IS NECESSARY TO LOAD THE IDCCSR WITH THE FUNCTION
0966    ;                         FIRST, THEN LOAD THE IDCDAR, AND FINIALLY CLEAR
0966    ;                         THE CRDY BIT IN THE IDCCSR.  THIS IS A REQUIREMENT
0966    ;                         SINCE THE IDC MICROCODE MUST DECODE THE IDCDAR CONTENTS
0966    ;                         PRIOR TO STARTING THE FUNCTION.
0966    ;
0966    ; CALLING SEQUENCE::
0966    ;
0966    ;       PUSHL   IDCMPR DATA
0966    ;       PUSHL   IDCDAR DATA
0966    ;       PUSHL   IDCCNT DATA
0966    ;       PUSHL   IDCBAR DATA
0966    ;       PUSHL   COMMAND TO BE EXECUTED
0966    ;       CALLS   #5,START_CMD
0966    ;
0966    ; INPUT PARAMETERS::
0966    ;
0966    ;       REGISTER DATA AND COMMAND DATA
0966    ;
0966    ; IMPLICIT INPUTS::
0966    ;
0966    ;       ** NONE **
0966    ;
0966    ; OUTPUT PARAMETERS::
0966    ;
0966    ;       ** NONE **
0966    ;
0966    ; IMPLICIT OUTPUTS::
0966    ;
0966    ;       ** NONE **
0966    ;
0966    ; COMPLETION CODES::
0966    ;
0966    ;       ** NONE **
0966    ;
0966    ; SIDE EFFECTS::
0966    ;
0966    ;       1. CONTROLS PROGRAM CONFIGURATION BASED ON CURRENT COMMAND
0966    ;       2. PERFORMS RECOVERY FROM PREVIOUS DRIVE ERRORS IF NECESSARY
0966    ;          BEFORE STARTING NEXT FUNCTION.
0966    ;
0966    ;--
0966
0966    .ENABL LSB
0966    START_CMD::
```

```
                           0018  0966          .WORD   ^M<R3,R4>                    ; REGISTER SAVE MASK
          52    0014'CF    D0    0968          MOVL    BASE_ADDR,R2                 ; GET BASE ADDRESS IN R2
                00A8'CF    D4    096D          CLRL    RGLIST                       ; RESET REGISTER PRINT LIST
     0B 00B4'CF     00     E5    0971          BBCC    #0,DERR_FLAG,5$              ; BRANCH IF NO DRV ERR SET
                  10 AC    DD    0977          PUSHL   16(AP)                       ; PASS CURRENT DESTINATION
                  04 AC    DD    097A          PUSHL   4(AP)                        ; NEXT COMMAND
        0B60'CF     02     FB    097D          CALLS   #2,RESTORE_DRIVE             ; CLEAR DRIVE ERRORS
                00A8'CF    D4    0982  5$:     CLRL    RGLIST                       ; INIT REGISTERS TO PRINT
          53      04 AC    9A    0986          MOVZBL  4(AP),R3                     ; R3 <-- COMMAND
          53      40 8F    8A    098A          BICB2   #CSR$M_IE,R3                 ; CLR ALL BUT CMD BITS
        00FC'CF     53     D0    098E          MOVL    R3,CURR_CMD                  ; SAVE CURRENT DRIVE FUNCTION
     53   53      FF 8F    9C    0993          ROTL    #-1,R3,R3                    ; POSITION CMD FOR CASE INST
                                 0998          CASE    TYPE=B,SRC=R3,LIMIT=#0,-     ; CASE ON INSTRUCTION TYPE
                                 0998                  DISPLIST=<-
                                 0998          ST_NOOP,-                            ; NOOP FUNCTION
                                 0998          ST_WRTCHK,-                          ; WRITECHECK FUNCTION
                                 0998          ST_GETSTAT,-                         ; GET STATUS FUNCTION
                                 0998          ST_SEEK,-                            ; SEEK FUNCTION
                                 0998          ST_READHDR,-                         ; READ HEADER FUNCTION
                                 0998          ST_WRITE,-                           ; WRITE DATA FUNCTION
                                 0998          ST_READ,-                            ; READ DATA FUNCTION
                                 0998          ST_READ_NOHDR>                       ; READ WITH NO HEADER COMPARASIN
                                 09AC
                                 09AC  ST_NOOP:
                                 09AC  ST_WRTCHK:
                      0043 31    09AC          BRW     90$
                                 09AF  ST_GETSTAT:
                      0040 31    09AF          BRW     90$
                                 09B2  ST_SEEK:
        00FC'CF     06     D0    09B2          MOVL    #SEEK,CURR_CMD               ; SAVE COMMAND
   00E0'CF     00D4'CF     D0    09B7          MOVL    CURR_CYLN,PREV_CYLN          ; CURRENT CYLN BECOMES THE PREVIOUS
  53   0C AC    08   08    EF    09BE          EXTZV   #8,#8,12(AP),R3
                                 09C4  ;       $DS_PRINTF_S    MSG_TRACE,CURR_CYLN,R3
                        2C 11    09C4          BRB     90$
                                 09C6  ST_READHDR:
        00A8'CF     07     D0    09C6          MOVL    #PRT$M_ALL,RGLIST            ; DEFINE REGISTERS TO PRINT
                        25 11    09CB          BRB     90$
                                 09CD  ST_WRITE:
     01BC'CF    0C AC      CE    09CD          MNEGL   12(AP),WORD_COUNT            ; SAVE WORD_COUNT
        01BC'CF     02     C6    09D3          DIVL2   #2,WORD_COUNT                ; CHANGE BYTE TO WORD COUNT
 00C4'CF   10 AC    08  00 EF    09D8          EXTZV   #0,#8,16(AP),SECTOR          ; SAVE SECTOR
 00C8'CF   10 AC    08  08 EF    09E0          EXTZV   #8,#8,16(AP),SURFACE         ; SAVE XFR SURFACE
 00D0'CF   10 AC    10  10 EF    09E8          EXTZV   #16,#16,16(AP),CYLN          ; SAVE XFR CYLINDER
                        00 11    09F0          BRB     90$
                                 09F2  ST_READ:
                                 09F2  ST_READ_NOHDR:
          54      04 AC    D0    09F2  90$:    MOVL    4(AP),R4                     ; R4 <-- COMMAND
        54      0034'CF    C8    09F6          BISL2   P$UNIT,R4                    ; OR IN DRIVE NUMBER
   62    54    000F0080 8F C9    09FB          BISL3   #^XF0080,R4,IDCCSR(R2)       ; LOAD FOR FUNCTION DECODE
                                 0A03                                               ; AND CLEAR ATTENTIONS
       04 A2      08 AC    D0    0A03          MOVL    8(AP),IDCBAR(R2)             ; LOAD IDCBAR REGISTER
       08 A2      0C AC    D0    0A08          MOVL    12(AP),IDCCNT(R2)            ; LOAD IDCCNT REGISTER
       0C A2      10 AC    D0    0A0D          MOVL    16(AP),IDCDAR(R2)            ; LOAD IDCDAR REGISTER
        00FC'CF     04     D1    0A12          CMPL    #GETSTAT,CURR_CMD            ; IF COMMAND IS NOT GET STATUS
                    05     12    0A17          BNEQ    100$                         ; THEN DONT LOAD MPR
       10 A2      14 AC    D0    0A19          MOVL    20(AP),IDCMPR(R2)            ; LOAD MULTI PURP REGISTER
```

```
          62    54   D0  0A1E   2496 100$:    MOVL     R4,IDCCSR(R2)                       ; LOAD AND START COMMAND
                     04  0A21   2497          RET                                          ; RETURN TO TEST FLOW
                         0A22   2498 .DSABL LSB
```

```
0A22          .SBTTL  STATUS CHECK ROUTINE
0A22     ;++
0A22     ; FUNCTIONAL DESCRIPTION::
0A22     ;
0A22     ;       THE PURPOSE OF THIS ROUTINE IS TO CHECK THE SETTING OF
0A22     ;       CONTROLLER READY AND DRIVE READY.  ONCE THESE BITS HAVE
0A22     ;       CHECKED CONTROLLER AND DRIVE ERROR BITS ARE TESTED.
0A22     ;       PROPER ERROR MESSAGES WILL BE LOGGED FOR ERROR REPORTING
0A22     ;       AT THE TEST LEVEL OF THE PROGRAM.
0A22     ;       SOME PROGRAM FLAGS ARE USED TO CONTROL THE FLOW OF THIS SUBROUTINE.
0A22     ;       THEY ARE:
0A22     ;       NOCHK_CERR      DONT CHECK FOR CONTROLLER ERRORS
0A22     ;       NOCHK_DERR      DONT CHECK FOR DRIVE ERRORS
0A22     ;       NOCHK_CRDY      DONT CHECK CONTROLLER READY BIT
0A22     ;       NOCHK_DRDY      DONT CHECK DRIVE READY BIT
0A22     ;       NOCHK_DSTA      DELETE GET STATUS OPERATION
0A22     ;       NOCHK_ATTN      DONT CHECK FOR ATTENTION STATUS AT THIS TIME
0A22     ;
0A22     ; THESE ARE USED FOR EXAMPLE WHEN A PARTICULAR ERROR IS BEING FORCED
0A22     ; AND THE ERROR WILL BE CHECK IN THE TEST CODE AND NOT IN THE SUB-
0A22     ; ROUTINE.
0A22     ;
0A22     ; CALLING SEQUENCE::
0A22     ;
0A22     ;       CALLS   #0,STATUS_CHK
0A22     ;
0A22     ; INPUT PARAMETERS::
0A22     ;
0A22     ;       ** NONE **
0A22     ;
0A22     ; IMPLICIT INPUTS::
0A22     ;
0A22     ;       1.  CONFIG -    CONTAINS PROGRAM FLAGS TO CONTROL THE ACTION
0A22     ;                       OF THE SUBROUTINE.
0A22     ;       2.  CURR_CMD -  ALSO CONTROLS ACTION OF THE SUBROUTINE BASED
0A22     ;                       ON COMMAND CURRENTLY IN PROGRESS.
0A22     ;
0A22     ; OUTPUT PARAMETERS::
0A22     ;
0A22     ;       ** NONE **
0A22     ;
0A22     ; IMPLICIT OUTPUTS::
0A22     ;
0A22     ;       ** NONE **
0A22     ;
0A22     ; COMPLETION CODES::
0A22     ;
0A22     ;       R0 = 1 SUCCESS
0A22     ;       R0 = 2 FAILURE DETECTED
0A22     ;       R1 = ADDRESS OF AN APPROPRIATE ERROR MESSAGE
0A22     ;
0A22     ; SIDE EFFECTS::
0A22     ;
0A22     ;       ** NONE **
0A22     ;
0A22     ;--
```

```
                              0A22
                              0A22  STATUS_CHK::
                        0FF8  0A22        .WORD   ^M<R3,R4,R5,R6,R7,R8,R9,R10,R11>; REGISTER SAVE MASK
        57   0000'CF     D0   0A24        MOVL    DRIVE_TBL,R7               ; GET ADRS OF DRV INFO
          58   34 A7     D0   0A29        MOVL    OFST_GTSCMD(R7),R8         ; GET STATUS CMD BITS
          59   4C A7     D0   0A2D        MOVL    OFST_DRVERR(R7),R9         ; GET DRIVE ERROR MASK
             55   01     D0   0A31        MOVL    #1,R5                      ; INIT RETURN STATUS
   20 011C'CF     00     E4   0A34        BBSC    #NOCHK$V_CRDY,CONFIG,25$   ; CHECK PROGRAM FLAGS
                  5A     D4   0A3A        CLRL    R10                        ; RESET COUNTER
           FEAF CF       16   0A3C  10$:  JSB     RD_IDCREGS                 ; UPDATE REGISTERS
   13 0120'CF     07     E0   0A40        BBS     #CSR$V_CRDY,T.CS,20$       ; BRANCH IF CRDY SET
EE 5A   0000FFFF 8F      F2   0A46        AOBLSS  #^XFFFF,R10,10$            ; TRY AGAIN
        56   1CFE'CF     DE   0A4E        MOVAL   MSG30_CRDY,R6              ; R6 <-- ERROR MSG
             55   02     D0   0A53        MOVL    #2,R5                      ; R5 <-- FAILURE FLAG
                00FB     31   0A56        BRW     STATUS_CHK_X               ; AND EXIT
                         01   0A59  20$:  NOP
   41 011C'CF     02     E4   0A5A  25$:  BBSC    #NOCHK$V_DRDY,CONFIG,45$   ; CHECK PROGRAM FLAG
                              0A60        $SETIMR_S       #33,TIME_800MS,,#1 ; START TIMER
          62      01     D3   0A71  30$:  BITL    #CSR$M_DRDY,IDCCSR(R2)     ; TEST IF DRDY SET
                  20     12   0A74        BNEQ    40$                        ; BRANCH IF IT IS
                              0A76        $READEF_S       #33,TIME_OUT       ; GET EVENT FLAGS
   E8 0080'CF     01     E1   0A83        BBC     #1,TIME_OUT,30$            ; BRANCH IF NO TIME OUT
        56   1D54'CF     DE   0A89        MOVAL   MSG31_DRDY,R6              ; R6 <-- ERROR MSG
        00B4'CF   01     D0   0A8E        MOVL    #1,DERR_FLAG               ; SET POSSIBLE DRIVE ERROR FLAG
             55   02     D0   0A93        MOVL    #2,R5                      ; R5 <-- FAILURE FLAG
                              0A96  40$:  $CANTIM_S       #1                 ; STOP TIMER
        0120'CF   62     D0   0AA1  45$:  MOVL    IDCCSR(R2),T.CS            ; UPDATE IDCCSR REGISTER
        011C'CF   04     E5   0AA6        BBCC    #NOCHK$V_DSTA,CONFIG,-     ; SKIP IF GETTING DRV STATUS
                  03          0AAB                46$
                0080     31   0AAC        BRW     70$                        ; ELSE GOTO 70$
        00FC'CF   04     D1   0AAF  46$:  CMPL    #GETSTAT,CURR_CMD          ; IF THIS IS GET STATUS CMD -
                  28     13   0AB4        BEQL    50$                        ; WE DONT NEED TO GET IT AGAIN
          10 A2   58     D0   0AB6        MOVL    R8,IDCMPR(R2)              ; SET UP IDCMPR FOR GET STATUS CMD
             54   04     D0   0ABA        MOVL    #GETSTAT,R4                ; DEFINE COMMAND
        54   0034'CF     C8   0ABD        BISL2   P$UNIT,R4                  ; OR IN DRIVE NUMBER
             62   54     D0   0AC2        MOVL    R4,IDCCSR(R2)              ; EXECUTE GET STATUS CMD
                  54     D4   0AC5        CLRL    R4                         ; RESET WAIT COUNT
    62     00000080 8F   D3   0AC7  47$:  BITL    #CSR$M_CRDY,IDCCSR(R2)     ; IF READY SET
                  08     12   0ACE        BNEQ    49$                        ; THEN GOTO 49$
EF 54      00000064 8F   F2   0AD0        AOBLSS  #100,R4,47$                ; TRY FOR READY AGAIN
   0130'CF     10 A2     D0   0AD8  49$:  MOVL    IDCMPR(R2),T.MP            ; GET DRIVE STATUS
               73 55     E9   0ADE  50$:  BLBC    R5,STATUS_CHK_X            ; BR IF AN ERROR ALREADY LOGGED
   48 011C'CF     03     E0   0AE1        BBS     #NOCHK$V_DERR,CONFIG,70$   ; BR IF CHECK NO ERRORS HERE
   33 0120'CF     0E     E0   0AE7        BBS     #CSR$V_DE,T.CS,60$         ; BRANCH IF DRV ERROR SET
      0130'CF     59     D3   0AED        BITL    R9,T.MP                    ; TEST ERRORS IN STATUS WORD
                  2C     12   0AF2        BNEQ    60$                        ; BRANCH IF ERRORS FOUND
   35 011C'CF     07     E4   0AF4        BBSC    #NOCHK$V_ATTN,CONFIG,70$   ; SKIP IF NO ATTN CHECKING
      06     00FC'CF     D1   0AFA        CMPL    CURR_CMD,#SEEK             ; IF COMMAND IS A SEEK
                  0C     13   0AFF        BEQL    55$                        ; THEN GOTO 55$
                              0B01  ;     BITL    P$ATTN,T.CS                ; IF ATTENTION NOT SET
                              0B01  ;     BEQL    70$                        ; THEN GOTO 70$
                  2C     11   0B01        BRB     70$                        ; GOTO 70$
        56   1D9D'CF     DE   0B03        MOVAL   MSG33A_UXEATTN,R6          ; R6 <-- ERROR MSG
             55   02     D0   0B08        MOVL    #2,R5                      ; SET ERROR CODE
                  47     11   0B0B        BRB     STATUS_CHK_X               ; AND EXIT
0120'CF   003C'CF        D3   0B0D  55$:  BITL    P$ATTN,T.CS                ; IF ATTENTION NOT CLEAR
```

```
                    19    12  0B14  2612          BNEQ    70$                              ; THEN GOTO 70$
          56    1DB8'CF   DE  0B16  2613          MOVAL   MSG33B_NOATTN,R6                 ; R6 <-- ERROR MSG
                55    02  D0  0B1B  2614          MOVL    #2,R5                            ; SET ERROR CODE
                    34    11  0B1E  2615          BRB     STATUS_CHK_X                     ; AND EXIT
          56    1D88'CF   DE  0B20  2616 60$:     MOVAL   MSG33_DERR,R6                    ; R6 <-- ERROR MSG
          00B4'CF     01  D0  0B25  2617          MOVL    #1,DERR_FLAG                     ; SET DRIVE ERROR FLAG
                55    02  D0  0B2A  2618          MOVL    #2,R5                            ; R5 <-- FAILURE FLAG
                    25    11  0B2D  2619          BRB     STATUS_CHK_X                     ; AND EXIT
          011C'CF     01  E4  0B2F  2620 70$:     BBSC    #NOCHK$V_CERR,CONFIG,-           ; CHECK PROGRAM FLAG
                    1F        0B34  2621                  STATUS_CHK_X
   0120'CF   00003C00 8F  D3  0B35  2622          BITL    #<CSR$M_NXM!-                    ; CHECK IDC FOR ANY ERRORS
                              0B3E  2623                  CSR$M_HNF!CSR$M_DCK!CSR$M_OPI>,T.CS ;
                    0C    12  0B3E  2624          BNEQ    80$                              ; BRANCH IF ERRORS FOUND
       0E 011C'CF     03  E4  0B40  2625          BBSC    #NOCHK$V_DERR,CONFIG,STATUS_CHK_X ; BR IF DRIVE TESTING DELETED
       08 0120'CF     0F  E1  0B46  2626          BBC     #CSR$V_ERR,T.CS,STATUS_CHK_X     ; BR IF COMPOSITE ERROR RESET
          56    1D6E'CF   DE  0B4C  2627 80$:     MOVAL   MSG32_CERR,R6                    ; R6 <-- ERROR MSG
                55    02  D0  0B51  2628          MOVL    #2,R5                            ; R5 <-- FAILURE FLAG
                              0B54  2629 STATUS_CHK_X:
          011C'CF     08  CA  0B54  2630          BICL2   #NOCHK$M_DERR,CONFIG             ; MAKE SURE FLAG IS RESET
                50    55  D0  0B59  2631          MOVL    R5,R0                            ; PUT FLAG IN R0
                51    56  D0  0B5C  2632          MOVL    R6,R1                            ; PUT ERROR MSG IN R1
                          04  0B5F  2633          RET                                      ; EXIT SUBROUTINE
```

```
                          0B60          .SBTTL  RESTORE DRIVE ROUTINE
                          0B60  ;++
                          0B60  ; FUNCTIONAL DESCRIPTION::
                          0B60  ;
                          0B60  ; THIS ROUTINE IS USED TO CLEAR ANY DRIVE ERRORS THAT MAY
                          0B60  ; OCCUR BEFORE STARTING THE NEXT COMMAND OPERATION.
                          0B60  ; RL02 OPERATION:
                          0B60  ;       IF DRIVE ERROR = TRUE
                          0B60  ;         THEN RESET DRIVE
                          0B60  ; R80 OPERATION:
                          0B60  ;       IF DRIVE ERROR = TRUE
                          0B60  ;         THEN RECAL DRIVE
                          0B60  ;       IF NEXT OPERATION = DMA TYPE
                          0B60  ;         THEN REPOSITION TO DESIRED CYLINDER
                          0B60  ;
                          0B60  ; CALLING SEQUENCE::
                          0B60  ;
                          0B60  ;       PUSHL   DESTINATION
                          0B60  ;       PUSHL   COMMAND
                          0B60  ;       CALLS   #2,RESTORE_DRIVE
                          0B60  ;
                          0B60  ; INPUT PARAMETERS::
                          0B60  ;
                          0B60  ;       DESTINATION (IF ANY) FOR NEXT COMMAND OPERATION
                          0B60  ;       NEXT COMMAND TO BE ISSUED TO IDC
                          0B60  ;
                          0B60  ; IMPLICIT INPUTS::
                          0B60  ;
                          0B60  ;       DRIVE TABLE
                          0B60  ;       DRIVE STATUS FROM LAST OPERATION
                          0B60  ;
                          0B60  ; OUTPUT PARAMETERS::
                          0B60  ;
                          0B60  ;       ** NONE **
                          0B60  ;
                          0B60  ; IMPLICIT OUTPUTS::
                          0B60  ;
                          0B60  ;       DRIVE RESET AND/OR DRIVE REPOSITONED
                          0B60  ;
                          0B60  ; COMPLETION CODES::
                          0B60  ;
                          0B60  ;       **NONE**
                          0B60  ;
                          0B60  ; SIDE EFFECTS::
                          0B60  ;
                          0B60  ;       ** NONE **
                          0B60  ;
                          0B60  ;--
                          0B60
                          0B60  RESTORE_DRIVE::
                    0070  0B60          .WORD   ^M<R4,R5,R6>                    ; REGISTER SAVE MASK
                          0B62  .ENABL  LSB
     54   0000'CF    D0   0B62          MOVL    DRIVE_TBL,R4                    ; GET DRIVE INFOR TABLE
                          0B67          CASE    TYPE=B,SRC=OFST_DRVTYP(R4),-
                          0B67                  LIMIT=#RL02$T,DISPLIST=<-
                          0B67                  RESTORE_RL02,-
```

```
                                    0B67  2691                   RESTORE_R80>
                                    0B6F  2692
                                    0B6F  2693 RESTORE_RL02:
                 0C4C'CF      16    0B6F  2694          JSB      CHECK_FATAL                        ; CHECK FOR ANY FATAL ERRORS
      0130'CF    00001F00 8F  D3    0B73  2695          BITL     #<MPR_02$M_DSE!MPR_02$M_WGE!-      ; CHECK RL02 DRIVE ERRORS
                                    0B7C  2696                   MPR_02$M_SPE!MPR_02$M_SKTO!MPR_02$M_VC>,T.MP ; FROM LAST STATUS WO -
cont> RD OBTAINED
                      03      12    0B7C  2697          BNEQ     10$                                ; BRANCH IF ERRORS FOUND
                    00CA      31    0B7E  2698          BRW      EXIT_RESTORE                       ; ELSE EXIT
            10 A2    50 A4    D0    0B81  2699 10$:     MOVL     OFST_DRVRST(R4),IDCMPR(R2)         ; LOAD MPR FOR DRIVE RESET OEPRATION
                 55    04     D0    0B86  2700          MOVL     #GETSTAT,R5                        ; GET COMMAND
              00FC'CF  55     D0    0B89  2701          MOVL     R5,CURR_CMD                        ; SAVE CURRENT COMMAND
              55    0034'CF   C8    0B8E  2702          BISL2    P$UNIT,R5                          ; OR IN DRIVE SELECT BITS
                 62    55     D0    0B93  2703          MOVL     R5,IDCCSR(R2)                      ; EXECUTE DRIVE RESET
                                    0B96  2704          $DS_WAITUS_S     #10                        ; WAIT FOR 100 US
                    004A      31    0BA1  2705          BRW      25$                                ; CHECK NEXT FUNCTION
                                    0BA4  2706
                                    0BA4  2707 RESTORE_R80:
          40 0130'CF    0A    E1    0BA4  2708          BBC      #MPR_80$V_SKE,T.MP,20$             ; IF NO SEEK ERROR GOTO 20$
                 55    06     D0    0BAA  2709          MOVL     #RECAL,R5                          ; GET COMMAND
              55    0034'CF   C8    0BAD  2710          BISL2    P$UNIT,R5                          ; OR IN DRIVE SELECT BITS
      62    55   000F0080 8F  C9    0BB2  2711          BISL3    #^XF0080,R5,IDCCSR(R2)             ; LOAD FOR COMMAND DECODE
            0C A2       00    D2    0BBA  2712          MCOML    #0,IDCDAR(R2)                      ; LOAD -1 FOR RECAL OPERATION
                 62    55     D0    0BBE  2713          MOVL     R5,IDCCSR(R2)                      ; EXECUTE RECAL COMMAND
                                    0BC1  2714          $DS_WAITMS_S     #20                        ; WAIT 100 MS
                 55    04     90    0BCC  2715          MOVB     #GETSTAT,R5                        ; GET COMMAND
              00FC'CF  55     D0    0BCF  2716          MOVL     R5,CURR_CMD                        ; SAVE CURRENT COMMAND
                 FD17 CF      16    0BD4  2717          JSB      RD_IDCREGS                         ; UPDATED IDC REGISTERS
                 62    55     D0    0BD8  2718          MOVL     R5,IDCCSR(R2)                      ; GET NEW DIRVE STATUS
                                    0BDB  2719          $DS_WAITUS_S     #10                        ; WAIT 100 US
                 FD05 CF      16    0BE6  2720          JSB      RD_IDCREGS                         ; SNAPSHOT REGISTERS
                 0C4C'CF      16    0BEA  2721 20$:     JSB      CHECK_FATAL                        ; CHECK FOR FATAL ERRORS
      56    04 AC    03    01 EF    0BEE  2722 25$:     EXTZV    #1,#3,4(AP),R6                     ; GET NEXT COMMAND
                                    0BF4  2723          CASE     TYPE=B,SRC=R6,LIMIT=#0,DISPLIST=<-
                                    0BF4  2724                   EXIT_RESTORE,-
                                    0BF4  2725                   DMA_CMD,-
                                    0BF4  2726                   EXIT_RESTORE,-
                                    0BF4  2727                   EXIT_RESTORE,-
                                    0BF4  2728                   EXIT_RESTORE,-
                                    0BF4  2729                   DMA_CMD,-
                                    0BF4  2730                   DMA_CMD,-
                                    0BF4  2731                   DMA_CMD>
                                    0C08  2732
                                    0C08  2733 DMA_CMD:
                 01    64     91    0C08  2734          CMPB     OFST_DRVTYP(R4),#RL02$T            ; IF DEVICE IS NOT AN RL02
                       1D     12    0C0B  2735          BNEQ     30$                                ; THEN GOTO 30$
         55    0034'CF   08   C9    0C0D  2736          BISL3    #READH,P$UNIT,R5                   ; SET UP FOR A READ HEADER CMD
                 62    55     D0    0C13  2737          MOVL     R5,IDCCSR(R2)                      ; TO FIX POSITION IN IDC LOGIC
                                    0C16  2738          $DS_WAITUS_S     #100                       ; WAIT 1 MS FOR CMD COMPLETTION
            10 A2    10 A2    D1    0C25  2739          CMPL     IDCMPR(R2),IDCMPR(R2)              ; GET MICRO CODE TO SAVE CURRENT A -
cont> DDRESS
         55    0034'CF   06   C9    0C2A  2740 30$:     BISL3    #SEEK,P$UNIT,R5                    ; SET UP FOR SEEK COMMAND
      62    55   00000080 8F  C9    0C30  2741          BISL3    #CSR$M_CRDY,R5,IDCCSR(R2)          ; LOAD FOR COMMAND DECODE
            0C A2    08 AC    D0    0C38  2742          MOVL     8(AP),IDCDAR(R2)                   ; LOAD DESTINATION ADDRESS
                 62    55     D0    0C3D  2743          MOVL     R5,IDCCSR(R2)                      ; EXECUTE SEEK TO DESTINATION
                                    0C40  2744          $DS_WAITMS_S     #6                         ; WAIT 60 MS
                                    0C4B  2745
```

```
0C4B  2746 EXIT_RESTORE:
```

```
 04  0C4B  2747        RET                         ; RETURN TO CALLER
     0C4C  2748 .DSABL  LSB
```

```
                                0C4C          .SBTTL  CHECK DRIVE FATAL ERROR ROUTINE
                                0C4C    ;++
                                0C4C    ; FUNCTIONAL DESCRIPTION::
                                0C4C    ;
                                0C4C    ; THIS ROUTINE CHECKS THE DRIVE STATUS WORD FOR FATAL ERRORS.
                                0C4C    ;
                                0C4C    ; RL02 FATAL DRIVE ERRORS:
                                0C4C    ;         HCE, WDE, AND DRIVE SPINNING DOWN OR ALREADY SPUN DOWN.
                                0C4C    ; R80 FATAL DRIVE ERRORS:
                                0C4C    ;         FLT
                                0C4C    ;
                                0C4C    ; CALLING SEQUENCE::
                                0C4C    ;
                                0C4C    ;         JSB     CHECK_FATAL
                                0C4C    ;
                                0C4C    ; INPUT PARAMETERS::
                                0C4C    ;
                                0C4C    ;         DRIVE STATUS WORD
                                0C4C    ;
                                0C4C    ; IMPLICIT INPUTS::
                                0C4C    ;
                                0C4C    ;         DRIVE TABLE
                                0C4C    ;
                                0C4C    ; OUTPUT PARAMETERS::
                                0C4C    ;
                                0C4C    ;         ** NONE **
                                0C4C    ;
                                0C4C    ; IMPLICIT OUTPUTS::
                                0C4C    ;
                                0C4C    ;         ** NONE **
                                0C4C    ;
                                0C4C    ; COMPLETION CODES::
                                0C4C    ;
                                0C4C    ;         ** NONE **
                                0C4C    ;
                                0C4C    ; SIDE EFFECTS::
                                0C4C    ;
                                0C4C    ;         PROGRAM ABORTED ON DETECTION OF FATAL DRIVE ERROR
                                0C4C    ;
                                0C4C    ;--
                                0C4C
                                0C4C    CHECK_FATAL::
                                0C4C    .ENABL  LSB
                     30    BB   0C4C            PUSHR   #^M<R4,R5>                        ; REGISTER SAVE MASK
        5A 011C'CF   06    E4   0C4E            BBSC    #NOCHK$V_FTL,CONFIG,NON_FATAL     ; SKIP IF THIS CHECKING EXCLUDED
           54   0000'CF    D0   0C54            MOVL    DRIVE_TBL,R4                      ; GET DRIVE INFO TABLE
                                0C59            CASE    TYPE=B,SRC=OFST_DRVTYP(R4),-
                                0C59                    LIMIT=#RL02$T,DISPLIST=<-
                                0C59                    CHKFTL_RL02,-
                                0C59                    CHKFTL_R80>
                                0C61
                                0C61    CHKFTL_RL02:
  0130'CF    0000C000 8F   D3   0C61            BITL    #<MPR_02$M_HCE!MPR_02$M_WDE>,T.MP ; IF FATAL ERRORS
                     26    12   0C6A            BNEQ    DRV_FATAL                         ; THEN GOTO DRV_FATAL
 55    0130'CF    03   00  EF   0C6C            EXTZV   #0,#3,T.MP,R5                     ; GET DRIVE STATE BITS
                                0C73            CASE    TYPE=B,SRC=R5,LIMIT=#0,DISPLIST=<-
```

```
                                 0C73                    DRV_FATAL,-
                                 0C73                    NON_FATAL,-
                                 0C73                    NON_FATAL,-
                                 0C73                    NON_FATAL,-
                                 0C73                    NON_FATAL,-
                                 0C73                    NON_FATAL,-
                                 0C73                    DRV_FATAL,-
                                 0C73                    DRV_FATAL>
                                 0C87
                                 0C87      CHKFTL_R80:
       0130'CF   00000100 8F  D3 0C87              BITL   #<MPR_80$M_FLT!MPR_80$M_FLT>,T.MP  ; IF ERRORS CLEAR
                          1C  13 0C90              BEQL   NON_FATAL                          ; THEN EXIT
                                 0C92
                                 0C92      DRV_FATAL:
                                 0C92              $DS_ERRDEV_S    ,L$UNIT,MSG_DRVFATAL,-
                                 0C92                              PRINT_REG
                                 0CA0      ;;; GLOBAL ERROR 1
                                 0CA7              $DS_ABORT                                 ; DIE
                                 0CAE
                                 0CAE      NON_FATAL:
                          30  BA 0CAE              POPR   #^M<R4,R5>                         ; RESTORE REGISTER
                              05 0CB0              RSB                                       ; RETURN TO CALLER
                                 0CB1      .DSABL  LSB
```

```
                          0CB1         .SBTTL  STEP HEADS ROUTINE
                          0CB1   ;++
                          0CB1   ; FUNCTIONAL DESCRIPTION::
                          0CB1   ;
                          0CB1   ;        THIS ROUTINE IS USED TO POSITION THE DRIVE AT A SPECIFIC
                          0CB1   ;        CYLINDER USING MINUS ONE AND PLUS ONE SEEKS.
                          0CB1   ;
                          0CB1   ; CALLING SEQUENCE::
                          0CB1   ;
                          0CB1   ;        PUSHL   #0                 ; PUSH DESIRED HEAD
                          0CB1   ;        PUSHL   NEW_CYLN           ; PUSH DESTINATION CYLINDER
                          0CB1   ;        CALLS   #2,STEP_HDS        ; EXECUTE SEEK
                          0CB1   ;
                          0CB1   ; INPUT PARAMETERS::
                          0CB1   ;
                          0CB1   ;        ** NONE **
                          0CB1   ;
                          0CB1   ; IMPLICIT INPUTS::
                          0CB1   ;
                          0CB1   ;        ** NONE **
                          0CB1   ;
                          0CB1   ; OUTPUT PARAMETERS::
                          0CB1   ;
                          0CB1   ;        ** NONE **
                          0CB1   ;
                          0CB1   ; IMPLICIT OUTPUTS::
                          0CB1   ;
                          0CB1   ;        CURRENT CYLINDER
                          0CB1   ;        DESTINATION CYLINDER
                          0CB1   ;        DIFFERENCE WORD
                          0CB1   ;
                          0CB1   ; COMPLETION CODES::
                          0CB1   ;
                          0CB1   ;        R0 = 1  SUCCESS FLAG
                          0CB1   ;        R0 = 2  FAILURE FLAG
                          0CB1   ;        R1 = ADDRESS OF ERROR MESSAGE
                          0CB1   ;
                          0CB1   ; SIDE EFFECTS::
                          0CB1   ;
                          0CB1   ;        IF IN AN ERROR LOOP CONDITION SUBROUTINE CAUSES HEADS TO SEEK
                          0CB1   ;        BACK TO ORIGIONAL CYLINDER AND TRY MISSED SEEK AGAIN. ALSO IF
                          0CB1   ;        THE ERROR WAS CAUSED BY A READ HEADER FUNCTION THEN THE SUBROUTINE
                          0CB1   ;        WILL ONLY LOOP ON THAT FUNCTION.
                          0CB1   ;
                          0CB1   ;--
                          0CB1
                          0CB1   STEP_HDS::
                  00F8    0CB1         .WORD   ^M<R3,R4,R5,R6,R7>          ; REGISTER SAVE MASK
      56   0000'CF   D0   0CB3         MOVL    DRIVE_TBL,R6                ; GET DRIVE INFO TABLE
       57     48 A6  D0   0CB8         MOVL    OFST_HDRTMP(R6),R7          ; GET ADDRESS OF HDR TEMPLATE
       53     04 AC  D0   0CBC         MOVL    4(AP),R3                    ; R3 <-- DESTINATION
              00     DD   0CC0         PUSHL   #0                          ; DEFINE FLAGS
    0EB0'CF   01     FB   0CC2         CALLS   #1,GET_POSITION             ; GET CURRNET DRIVE POSITION
           0F 50     E8   0CC7         BLBS    R0,10$                      ; BRANCH IF NO ERRORS
    011C'CF   20     C8   0CCA         BISL2   #HDR$M_ERR,CONFIG           ; SET READ HDR ERR FLAG
 00B8'CF   198A'CF   DE   0CCF         MOVAL   MSG19_READH,ERR_MSG         ; SET UP ERROR MSG
```

```
                          00E4  31  0CD6        BRW     STEP_HDS_X                      ; AND EXIT
 00D4'CF   0140'CF   67     87  EF  0CD9  10$:  EXTZV   (R7)+,(R7),T.HDR,CURR_CYLN      ; GET CURRENT CYLINDER
                                    0CE2        $DS_INLOOP_S                            ; ARE WE IN AN ERROR LOOP
                         09 50  E9  0CE9        BLBC    R0,14$                          ; BRANCH IF NOT
               03 011C'CF   05  E1  0CEC        BBC     #HDR$V_ERR,CONFIG,14$           ; BRANCH IF NO READ HDR ERROR
                          00C5  31  0CF2        BRW     70$                             ; ELSE EXIT (LOOP ON READ HDR ONLY)
                         55 01  D0  0CF5  14$:  MOVL    #1,R5                           ; SET UP FOR +1 SEEK
                  53   00D4'CF  D1  0CF8        CMPL    CURR_CYLN,R3                    ; FIND RELATION TO DESTINATION
                            03  19  0CFD        BLSS    16$                             ; BRANCH IF FWD SEEK
                         55 55  CE  0CFF        MNEGL   R5,R5                           ; CHANGE DIRECTION TO REVERSE
       00E4'CF     00D4'CF  55  C1  0D02  16$:  ADDL3   R5,CURR_CYLN,DEST_CYLN          ; CALC DESTINATION CYLN
                  0100'CF   01  D0  0D0A        MOVL    #1,DIFF_WORD                    ; SAVE DIFFERENCE WORD
                            54  D4  0D0F        CLRL    R4                              ; CLEAR WORK LOCATION
        54    08     08  08 AC  F0  0D11        INSV    8(AP),#8,#8,R4                  ; OR IN DESIRED HEAD
     54    10     10   00E4'CF  F0  0D17        INSV    DEST_CYLN,#16,#16,R4            ; OR IN DESIRED CYLINDER
                            00  DD  0D1E        PUSHL   #0                              ; IDCMPR DATA
                            54  DD  0D20        PUSHL   R4                              ; IDCDAR DATA
                            00  DD  0D22        PUSHL   #0                              ; IDCCNT DATA
                            00  DD  0D24        PUSHL   #0                              ; IDCBAR DATA
                            06  DD  0D26        PUSHL   #SEEK                           ; DEFINE COMMAND
                  FC39 CF   05  FB  0D28        CALLS   #5,START_CMD                    ; EXECUTE SEEK CMD
                  FCF0 CF   00  FB  0D2D        CALLS   #0,STATUS_CHK                   ; CHECK FOR ANY ERRORS
                         1D 50  E8  0D32        BLBS    R0,30$                          ; BRANCH IF NONE
            00E8'CF    00E4'CF  D0  0D35        MOVL    DEST_CYLN,OLD_CYLN              ; SAVE DESTINATION OF SEEK
            00EC'CF    00E0'CF  D0  0D3C        MOVL    PREV_CYLN,NEW_CYLN              ; SAVE ORIGIN OF SEEK
            00B8'CF    2099'CF  DE  0D43        MOVAL   MSG48_SK_ONE,ERR_MSG            ; SET UP ERROR MSG
                  00A8'CF   08  C8  0D4A        BISL2   #PRT$M_EXT_1,RGLIST             ; PNT EXTENDED INFO
                          006B  31  0D4F        BRW     STEP_HDS_X                      ; AND EXIT
                            00  DD  0D52  30$:  PUSHL   #0                              ; DEFINE FLAGS
                  0EB0'CF   01  FB  0D54        CALLS   #1,GET_POSITION                 ; GET CURRNET DRIVE POSITION
                         0F 50  E8  0D59        BLBS    R0,50$                          ; BRANCH IF NONE
                  011C'CF   20  C8  0D5C        BISL2   #HDR$M_ERR,CONFIG               ; SET READ HDR ERR FLAG
            00B8'CF    198A'CF  DE  0D61        MOVAL   MSG19_READH,ERR_MSG             ; SET UP ERROR MSG
                          0052  31  0D68        BRW     STEP_HDS_X                      ; AND EXIT
            00E4'CF    00D4'CF  D1  0D6B  50$:  CMPL    CURR_CYLN,DEST_CYLN             ; DID SEEK POSITION PROPERLY
                            24  13  0D72        BEQL    60$                             ; BRANCH IF YES
            00E8'CF    00E4'CF  D0  0D74        MOVL    DEST_CYLN,OLD_CYLN              ; SAVE DESTINATION OF SEEK
            00EC'CF    00E0'CF  D0  0D7B        MOVL    PREV_CYLN,NEW_CYLN              ; SAVE ORIGIN OF SEEK
                            18  C8  0D82        BISL2   #<PRT$M_EXT_1!PRT$M_EXT_2>,-    ; PNT EXTENDED INFO
                       00A8'CF      0D84                RGLIST
            00B8'CF    2099'CF  DE  0D87        MOVAL   MSG48_SK_ONE,ERR_MSG            ; SET UP ERROR MSG
                  51   2068'CF  DE  0D8E        MOVAL   MSG46_POS_ERR,R1                ; SET UP ADDITIONAL MSG
                         50 02  D0  0D93        MOVL    #2,R0                           ; R0 <-- FAILURE FLAG
                            25  11  0D96        BRB     STEP_HDS_X                      ; EXIT
                  53   00D4'CF  D1  0D98  60$:  CMPL    CURR_CYLN,R3                    ; ARE ALL SEEKS DONE ?
                            03  13  0D9D        BEQL    65$                             ; BRANCH IF YES
                          FF53  31  0D9F        BRW     14$                             ; GO DO NEXT SEEK
                                    0DA2  65$:  $DS_INLOOP_S                            ; ARE WE IN A SEEK ERROR LOOP
                         0E 50  E9  0DA9        BLBC    R0,70$                          ; BRANCH IF NOT
            00EC'CF    00E8'CF  D0  0DAC        MOVL    OLD_CYLN,NEW_CYLN               ; SET UP TO OSCILLATE ON ERROR
            00E8'CF    00D4'CF  D0  0DB3        MOVL    CURR_CYLN,OLD_CYLN              ; THAT CAUSED THE ERROR
                         50 01  D0  0DBA  70$:  MOVL    #1,R0                           ; R0 <-- SUCCESS FLAG
                                    0DBD  STEP_HDS_X:
                                04  0DBD        RET                                     ; RETURN TO TEST
                                    0DBE
```

```
                              0DBE             .SBTTL  POSITION HEADS ROUTINE
                              0DBE      ;++
                              0DBE      ; FUNCTIONAL DESCRIPTION::
                              0DBE      ;
                              0DBE      ;       THIS ROUTINE IS USED TO POSITION THE DRIVE TO A SPECIFIC
                              0DBE      ;       CYLINDER.
                              0DBE      ;
                              0DBE      ; CALLING SEQUENCE::
                              0DBE      ;
                              0DBE      ;       PUSHL   #0                ; PUSH DESIRED CYLINDER
                              0DBE      ;       PUSHL   NEW_CYLN          ; PUSH DESTINATION
                              0DBE      ;       CALLS   #2,POSITION_HDS   ; EXECUTE SEEK
                              0DBE      ;
                              0DBE      ; INPUT PARAMETERS::
                              0DBE      ;
                              0DBE      ;       ** NONE **
                              0DBE      ;
                              0DBE      ; IMPLICIT INPUTS::
                              0DBE      ;
                              0DBE      ;       ** NONE **
                              0DBE      ;
                              0DBE      ; OUTPUT PARAMETERS::
                              0DBE      ;
                              0DBE      ;       ** NONE **
                              0DBE      ;
                              0DBE      ; IMPLICIT OUTPUTS::
                              0DBE      ;
                              0DBE      ;       CURRENT CYLINDER
                              0DBE      ;       DESTINATION CYLINDER
                              0DBE      ;       DIFFERENCE WORD
                              0DBE      ;
                              0DBE      ; COMPLETION CODES::
                              0DBE      ;
                              0DBE      ;       R0 = 1  SUCCESS
                              0DBE      ;       R0 = 2  FAILURE
                              0DBE      ;       R1 = ADDRESS OF AN ERROR MESSAGE
                              0DBE      ;
                              0DBE      ; SIDE EFFECTS::
                              0DBE      ;
                              0DBE      ;       IF SUBROUTINE CONTAINED WITHIN AN ERROR LOOP CAUSES HEADS TO
                              0DBE      ;       SEEK BACK TO THE ORIGIONAL CYLINDER AND TRY SEEK AGAIN.
                              0DBE      ;       ALSO IF ERROR WAS CAUSED BY A READ HEADER FUNCTION THEN THE
                              0DBE      ;       SUBROUTINE WILL LOOP ON THAT FUNCTION ONLY.
                              0DBE      ;
                              0DBE      ;--
                              0DBE
                              0DBE      POSITION_HDS::
                        0078  0DBE             .WORD   ^M<R3,R4,R5,R6>                 ; REGISTER SAVE MASK
           55   0000'CF   D0  0DC0             MOVL    DRIVE_TBL,R5                    ; GET DRIVE INFO TABLE
             56   48 A5   D0  0DC5             MOVL    OFST_HDRTMP(R5),R6              ; GET ADDRESS OF HDR TEMPLATE
        00E4'CF   04 AC   D0  0DC9             MOVL    4(AP),DEST_CYLN                 ; GET DESTINATION FOR SEEK
        03 0028'CF   0A   E1  0DCF             BBC     #DEL$V_READH,CURR_FLAGS,5$      ; IF READ HEADERS NOT WAIVERED GOT -
cont> O 5$
                 0052     31  0DD5             BRW     26$                             ; ELSE GOTO 26$
                   00     DD  0DD8      5$:    PUSHL   #0                              ; DEFINE FLAGS
          0EB0'CF  01     FB  0DDA             CALLS   #1,GET_POSITION                 ; GET CURRNET DRIVE POSITION
                0F 50     E8  0DDF             BLBS    R0,10$                          ; BRANCH IF NONE
```

```
          011C'CF   20  C8  0DE2  2997          BISL2   #HDR$M_ERR,CONFIG               ; SET READ HDR ERR FLAG
      00B8'CF   198A'CF  DE  0DE7  2998          MOVAL   MSG19_READH,ERR_MSG             ; SET UP ERROR MSG
                  00BE  31  0DEE  2999          BRW     POSITION_HDS_X                  ; EXIT
 00D4'CF  0140'CF  66  86  EF  0DF1  3000 10$:  EXTZV   (R6)+,(R6),T.HDR,CURR_CYLN      ; GET CURRENT CYLINDER
                            0DFA  3001          $DS_INLOOP_S                            ; ARE WE IN AN ERROR LOOP
                 1B 50  E9  0E01  3002          BLBC    R0,20$                          ; BRANCH IF NOT
       03 011C'CF   05  E1  0E04  3003          BBC     #HDR$V_ERR,CONFIG,12$           ; BRANCH IF NO READ HEADER ERROR
                  009F  31  0E0A  3004          BRW     50$                             ; EXIT
      00E4'CF   00D4'CF  D1  0E0D  3005 12$:    CMPL    CURR_CYLN,DEST_CYLN             ; DID WE REACH THE DESTINATION ?
                    10  12  0E14  3006          BNEQ    25$                             ; BR IF WE DIDNT AND KEEP TRYING
      00E4'CF   00E8'CF  D0  0E16  3007          MOVL    OLD_CYLN,DEST_CYLN              ; ELSE USE REFERENCE CYLINDER
                    07  11  0E1D  3008          BRB     25$                             ; SKIP OVER SETTING OLD CYLINDER
      00E8'CF   00D4'CF  D0  0E1F  3009 20$:    MOVL    CURR_CYLN,OLD_CYLN              ; SAVE STARTING CYLINDER
                0FF8'CF  16  0E26  3010 25$:    JSB     CALC_DIFF                       ; CALCULATE DIFFERENCE WORD
                    54  D4  0E2A  3011 26$:     CLRL    R4                              ; CLEAR WORK LOCATION
      54   08   08  08 AC  F0  0E2C  3012          INSV    8(AP),#8,#8,R4                  ; OR IN DESIRED TRACK
      54   10   10  04 AC  F0  0E32  3013          INSV    4(AP),#16,#16,R4                ; OR IN DESIRED CYLINDER
       03 00CC'CF   00  E5  0E38  3014          BBCC    #RECAL$V_INPROG,POS_TYPE,27$    ; IF SEEK OPERATION GOTO 27$
                 54 00  D2  0E3E  3015          MCOML   #0,R4                           ; ELSE DAR= -1 FOR RECAL
                    00  DD  0E41  3016 27$:     PUSHL   #0                              ; IDCMPR DATA
                    54  DD  0E43  3017          PUSHL   R4                              ; IDCDAR DATA
                    00  DD  0E45  3018          PUSHL   #0                              ; IDCCNT DATA
                    00  DD  0E47  3019          PUSHL   #0                              ; IDCBAR DATA
                    06  DD  0E49  3020          PUSHL   #SEEK                           ; DEFINE COMMAND
            FB16 CF 05  FB  0E4B  3021          CALLS   #5,START_CMD                    ; EXECUTE SEEK CMD
            FBCD CF 00  FB  0E50  3022          CALLS   #0,STATUS_CHK                   ; CHECK FOR ANY ERRORS
                 0F 50  E8  0E55  3023          BLBS    R0,30$                          ; BRANCH IF NONE
           00A8'CF  08  C8  0E58  3024          BISL2   #PRT$M_EXT_1,RGLIST             ; PNT EXTENDED INFO
      00B8'CF   1943'CF  DE  0E5D  3025          MOVAL   MSG18_SEEK,ERR_MSG              ; SET UP ERROR MSG
                  0048  31  0E64  3026          BRW     POSITION_HDS_X                  ; EXIT
       3F 0028'CF   0A  E0  0E67  3027 30$:     BBS     #DEL$V_READH,CURR_FLAGS,50$     ; IF READ HEADERS WAIVERED GOTO 50$
                    00  DD  0E6D  3028          PUSHL   #0                              ; DEFINE FLAGS
           0EB0'CF  01  FB  0E6F  3029          CALLS   #1,GET_POSITION                 ; GET CURRNET DRIVE POSITION
                 0E 50  E8  0E74  3030          BLBS    R0,40$                          ; BRANCH IF NONE
      00B8'CF   198A'CF  DE  0E77  3031          MOVAL   MSG19_READH,ERR_MSG             ; SET UP ERROR MSG
          011C'CF   20  C8  0E7E  3032          BISL2   #HDR$M_ERR,CONFIG               ; SET READ HDR ERR FLAG
                    2A  11  0E83  3033          BRB     POSITION_HDS_X                  ; EXIT
      00E4'CF   00D4'CF  D1  0E85  3034 40$:    CMPL    CURR_CYLN,DEST_CYLN             ; DID SEEK POSITION PROPERLY
                    08  12  0E8C  3035          BNEQ    45$                             ; BRANCH IF IT DID NOT
        08 AC   00D8'CF  D1  0E8E  3036          CMPL    CURR_HD,8(AP)                   ; WAS PROPER HEAD SELECTED
                    16  13  0E94  3037          BEQL    50$                             ; IF SO THEN GOTO 50$
      00B8'CF   1943'CF  DE  0E96  3038 45$:    MOVAL   MSG18_SEEK,ERR_MSG              ; SET UP ERROR MSG
           51   2068'CF  DE  0E9D  3039          MOVAL   MSG46_POS_ERR,R1                ; SET UPP ADDITIONAL MSG
                    18  C8  0EA2  3040          BISL2   #<PRT$M_EXT_1!PRT$M_EXT_2>,-    ; PNT EXTENDED INFO
               00A8'CF      0EA4  3041                  RGLIST
                 50 02  D0  0EA7  3042          MOVL    #2,R0                           ; R0 <-- FAILURE FLAG
                    03  11  0EAA  3043          BRB     POSITION_HDS_X                  ; EXIT
                 50 01  D0  0EAC  3044 50$:     MOVL    #1,R0                           ; R0 <-- SUCCESS FLAG
                            0EAF  3045 POSITION_HDS_X:
                        04  0EAF  3046          RET                                     ; RETURN TO TEST
```

```
                         0EB0         .SBTTL  GET POSITION ROUTINE
                         0EB0   ;++
                         0EB0   ; FUNCTIONAL DESCRIPTION::
                         0EB0   ;
                         0EB0   ;       THIS SUBROUTINE EXECUTES A READ HEADER COMMAND AND RETURNS
                         0EB0   ;       WITH THE CURRENT DRIVE POSITION IN CURR_CYLN.
                         0EB0   ;       IT IS ALSO RESPONSIBLE FOR SAVING CURRENT POSITIONING
                         0EB0   ;       INFORMATION AS WELL AS PERFORMING A SOFTWARE CHECK
                         0EB0   ;       OF THE HEADER CRC WORD.
                         0EB0   ;
                         0EB0   ; CALLING SEQUENCE::
                         0EB0   ;
                         0EB0   ;       PUSHL   FLAGS
                         0EB0   ;       JSB     GET_POSITION
                         0EB0   ;
                         0EB0   ; INPUT PARAMETERS::
                         0EB0   ;
                         0EB0   ;       FLAGS FOR DEFINING SKIP SECTOR OPERATION IN THE IDC
                         0EB0   ;
                         0EB0   ; IMPLICIT INPUTS::
                         0EB0   ;
                         0EB0   ;       ** NONE **
                         0EB0   ;
                         0EB0   ; OUTPUT PARAMETERS::
                         0EB0   ;
                         0EB0   ;       ** NONE **
                         0EB0   ;
                         0EB0   ; IMPLICIT OUTPUTS::
                         0EB0   ;
                         0EB0   ;       CURRENT CYLINDER
                         0EB0   ;
                         0EB0   ; COMPLETION CODES::
                         0EB0   ;
                         0EB0   ;       R0 = 1  SUCCESS
                         0EB0   ;       R0 = 2  FAILURE
                         0EB0   ;       R1 = ADDRESS OF AN ERROR MESSAGE
                         0EB0   ;
                         0EB0   ; SIDE EFFECTS::
                         0EB0   ;
                         0EB0   ;       ** NONE **
                         0EB0   ;
                         0EB0   ;--
                         0EB0
                         0EB0   GET_POSITION::
                   0078  0EB0         .WORD   ^M<R3,R4,R5,R6>          ; REGISTER SAVE MASK
               56    D4  0EB2         CLRL    R6                       ; RESET COUNTER
         53    08    D0  0EB4   5$:   MOVL    #READH,R3                ; LOAD COMMAND
     53  04    AC    C8  0EB7         BISL2   4(AP),R3                 ; OR IN FLAGS
               00    DD  0EBB         PUSHL   #0                       ; IDCMPR DATA
               00    DD  0EBD         PUSHL   #0                       ; IDCDAR DATA
               00    DD  0EBF         PUSHL   #0                       ; IDCBCR DATA
               00    DD  0EC1         PUSHL   #0                       ; IDCBAR DATA
               53    DD  0EC3         PUSHL   R3                       ; DEFINE COMMAND
   FA9C CF     05    FB  0EC5         CALLS   #5,START_CMD             ; EXECUTE READ HEADER
   FB53 CF     00    FB  0ECA         CALLS   #0,STATUS_CHK            ; CHECK FOR ANY ERRORS
         0F    50    E8  0ECF         BLBS    R0,10$                   ; BRANCH IF NONE
```

```
      03  0120'CF      02    0A  ED  0ED2             CMPZV   #10,#2,T.CS,#3                  ; IF ERROR IS HCRC
                             62  13  0ED9             BEQL    20$                             ; THEN GOTO 20$
                     50      02  D0  0EDB             MOVL    #2,R0                           ; R0 <-- FAILURE FLAG
                           006A  31  0EDE             BRW     GET_POSITION_X                  ; EXIT
                53         0000'CF D0 0EE1     10$:   MOVL    DRIVE_TBL,R3                    ; GET DEVICE INFO TABLE
                   53     48 A3  D0  0EE6             MOVL    OFST_HDRTMP(R3),R3              ; GET ADDR OF HDR TEMPLATE
                54      0140'CF  DE  0EEA             MOVAL   T.HDR,R4                        ; GET ADDRESS OF SAVED HEADER
   00D4'CF      64     83    83  EF  0EEF             EXTZV   (R3)+,(R3)+,(R4),CURR_CYLN      ; GET CURRENT CYLINDER
   00D8'CF      64     83    83  EF  0EF6             EXTZV   (R3)+,(R3)+,(R4),CURR_HD        ; GET CURRENT HEAD SELECTED
   00F0'CF      64     63    83  EF  0EFD             EXTZV   (R3)+,(R3),(R4),CURR_SECTOR     ; GET CURRENT SECTOR
00DC'CF     00D8'CF    00D4'CF   C1  0F04             ADDL3   CURR_CYLN,CURR_HD,CURR_CYL_HD   ; CCOMBINE CURRENT HEAD AND CYLN
            0114'CF    0134'CF   B0  0F0E             MOVW    T.MP1,WORK                      ; GET 1ST HEADER WORD
            0116'CF    0138'CF   B0  0F15             MOVW    T.MP2,WORK+2                    ; GET SECOND HEADER WORD
                  011C'CF    05  E4  0F1C             BBSC    #NOCHK$V_CRC,CONFIG,-           ; SKIP IF NO CRC CHECKING
                             29      0F21                     GET_POSITION_X
                             3F  BB  0F22             PUSHR   #^M<R0,R1,R2,R3,R4,R5>          ; SAVE REGISTERS
0114'CF     04      00  0228'CF  0B  0F24             CRC     CRC_TABLE,#0,#4,WORK            ; CALC CRC
                0178'CF      50  3C  0F2D             MOVZWL  R0,E.MP3                        ; GET EXPECTED CRC
                             3F  BA  0F32             POPR    #^M<R0,R1,R2,R3,R4,R5>          ; RESTORE REGISTERS
            013C'CF    0178'CF   D1  0F34             CMPL    E.MP3,T.MP3                     ; IS CRC OK ?
                             0E  13  0F3B             BEQL    GET_POSITION_X                  ; BRANCH IF IT IS
            FF71 56     01   02  F1  0F3D      20$:   ACBL    #2,#1,R6,5$                     ; TRY AGAIN
                        50   02  D0  0F43             MOVL    #2,R0                           ; R0 <-- FAILURE FLAG
                51     1FDF'CF   DE  0F46             MOVAL   MS44A_INC_CRC,R1                ; R1 <-- ERROR MESSAGE
                                     0F4B      GET_POSITION_X:
                                 04  0F4B             RET                                     ; RETURN TO CALLER
```

```
                                 0F4C              .SBTTL  BUILD HEADERS ROUTINE
                                 0F4C
                                 0F4C       ;++
                                 0F4C       ; FUNCTIONAL DESCRIPTION::
                                 0F4C       ;
                                 0F4C       ; THIS ROUTINE IS RESPONSIBLE FOR BUILDING ALL HEADERS
                                 0F4C       ; FOR THE R80 DEVICE.
                                 0F4C       ;
                                 0F4C       ; CALLING SEQUENCE::
                                 0F4C       ;
                                 0F4C       ;       PUSHL   FLAGS
                                 0F4C       ;       PUSHL   TRACK
                                 0F4C       ;       PUSHL   CYLINDER
                                 0F4C       ;       CALLS   #3,BUILD_HDRS
                                 0F4C       ;
                                 0F4C       ; INPUT PARAMETERS::
                                 0F4C       ;
                                 0F4C       ;       DESIRED TRACK
                                 0F4C       ;       DESIRED CYLINDER
                                 0F4C       ;       32 BIT FLAG WORD SPECIFING WHICH SECTOR
                                 0F4C       ;         IS TO BE DEFINED AS A SKIP SECTOR
                                 0F4C       ;
                                 0F4C       ; IMPLICIT INPUTS::
                                 0F4C       ;
                                 0F4C       ;       ** NONE **
                                 0F4C       ;
                                 0F4C       ; OUTPUT PARAMETERS::
                                 0F4C       ;
                                 0F4C       ;       ** NONE **
                                 0F4C       ;
                                 0F4C       ; IMPLICIT OUTPUTS::
                                 0F4C       ;
                                 0F4C       ;       WRITE BUFFER LOAD WITH HEADER WORD DATA
                                 0F4C       ;
                                 0F4C       ; COMPLETION CODES::
                                 0F4C       ;
                                 0F4C       ;       ** NONE **
                                 0F4C       ;
                                 0F4C       ; SIDE EFFECTS::
                                 0F4C       ;
                                 0F4C       ;       ** NONE **
                                 0F4C       ;
                                 0F4C       ;--
                                 0F4C
                                 0F4C       BUILD_HDRS::
                           0FFC  0F4C              .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
                       53    7C  0F4E              CLRQ    R3                              ; INIT COUNTER/POINTER
              55    04 AC    D0  0F50              MOVL    4(AP),R5                        ; GET DESIRED CYLINDER
              56    08 AC    D0  0F54              MOVL    8(AP),R6                        ; GET DESIRED TRACK
              57    0C AC    D0  0F58              MOVL    12(AP),R7                       ; GET SKIP SECTOR FLAG WORD
       54    57    20    00  EA  0F5C              FFS     #0,#32,R7,R4                    ; FIND POSITION OF SKIP SECTOR
                         0C  13  0F61              BEQL    10$                             ; BRANCH IF NONE
                   5A    00  D2  0F63              MCOML   #0,R10                          ; MAKE A WORD OF ALL 1'S
             59    20    54  C3  0F66              SUBL3   R4,#32,R9                       ; CALC NUMBER OF SECTORS TO MARK A -
cont> S SKIP
       57    59    54    5A  F0  0F6A              INSV    R10,R4,R9,R7                    ; AND SET THEIR FLAGS
             54    0000'CF   D0  0F6F       10$:   MOVL    DRIVE_TBL,R4                    ; GET DRIVE INFO TABLE
```

```
          59    02BC'CF   D0  0F74  3187          MOVL    WRITE_BUFFER,R9                     ; GET ADDRESS OF WRITE BUFFER
             58     48 A4   D0  0F79  3188 20$:     MOVL    OFST_HDRTMP(R4),R8                  ; GET HEADER CONTROL FIELD
                       69   D4  0F7D  3189          CLRL    (R9)                                ; START WITH ZERO WORD
       69    88    88  55   F0  0F7F  3190          INSV    R5,(R8)+,(R8)+,(R9)                 ; LOAD CYLINDER
       69    01    0C  01   F0  0F84  3191          INSV    #1,#HDR_80$V_FMT,#HDR_80$S_FMT,(R9) ; LOAD FORMAT 16 BIT
       69    01    0D  57   F0  0F89  3192          INSV    R7,#HDR_80$V_SSF,#HDR_80$S_SSF,(R9) ; LOAD SKIP SECTOR FLAG
       69    88    88  56   F0  0F8E  3193          INSV    R6,(R8)+,(R8)+,(R9)                 ; LOAD TRACK
       69    68    88  53   F0  0F93  3194          INSV    R3,(R8)+,(R8),(R9)                  ; LOAD SECTOR
        57    57    FF 8F   9C  0F98  3195          ROTL    #-1,R7,R7                           ; GET NEXT SKIP SECTOR FLAG
                 59    04   C0  0F9D  3196          ADDL2   #4,R9                               ; GET NEXT HEADER ADDRESS
              D5 53    20   F2  0FA0  3197          AOBLSS  #32,R3,20$                          ; LOAD NEXT HEADER
                            04  0FA4  3198          RET                                         ; RETURN TO CALLER
```

```
                                0FA5         .SBTTL  READ TRACK HEADERS
                                0FA5    ;++
                                0FA5    ; FUNCTIONAL DESCRIPTION::
                                0FA5    ;
                                0FA5    ; THIS ROUTINE READS ALL THE HEADERS ON ONE TRACK SEQUENTIALLY
                                0FA5    ;
                                0FA5    ; CALLING SEQUENCE::
                                0FA5    ;
                                0FA5    ;        CALLS   #0,READ_HEADERS
                                0FA5    ;
                                0FA5    ; INPUT PARAMETERS::
                                0FA5    ;
                                0FA5    ;        ** NONE **
                                0FA5    ;
                                0FA5    ; IMPLICIT INPUTS::
                                0FA5    ;
                                0FA5    ;        ** NONE **
                                0FA5    ;
                                0FA5    ; OUTPUT PARAMETERS::
                                0FA5    ;
                                0FA5    ;        ** NONE **
                                0FA5    ;
                                0FA5    ; IMPLICIT OUTPUTS::
                                0FA5    ;
                                0FA5    ;        ** NONE **
                                0FA5    ;
                                0FA5    ; COMPLETION CODES::
                                0FA5    ;
                                0FA5    ;        R0 = 1  SUCCESS
                                0FA5    ;        R0 = 2  FAILURE
                                0FA5    ;        R1 = ADDRESS OF AN ERROR MESSAGE
                                0FA5    ;
                                0FA5    ; SIDE EFFECTS::
                                0FA5    ;
                                0FA5    ;        ** NONE **
                                0FA5    ;
                                0FA5    ;--
                                0FA5
                                0FA5    READ_HEADERS::
                      0038      0FA5            .WORD   ^M<R3,R4,R5>                ; REGISTER SAVE MASK
                        53  D4  0FA7            CLRL    R3                          ; RESET COUNTER
         55   02C4'CF       D0  0FA9            MOVL    READ_BUFFER,R5              ; GET ADDRESS OF READ BUFFER
          08400000 8F       DD  0FAE    10$:    PUSHL   #<CSR$M_ASSI!CSR$M_SSEI>    ; DEFINE FLAGS
         FEF7 CF        01  FB  0FB4            CALLS   #1,GET_POSITION             ; GO READ HEADER
                     03 50  E8  0FB9            BLBS    R0,20$                      ; BRANCH IF NO ERRORS
                      0038  31  0FBC            BRW     EXIT_READHDR                ; ELSE EXIT
         1F   00F0'CF       91  0FBF    20$:    CMPB    CURR_SECTOR,#31             ; IF LAST SECTOR FOUND
                        13  13  0FC4            BEQL    30$                         ; THEN GOTO 30$
   E0 53  00000040 8F       F2  0FC6            AOBLSS  #64,R3,10$                  ; READ NEXT HEADER
         51   2583'CF       DE  0FCE            MOVAL   MSG70A_NFSEC31,R1           ; SET ERROR MESSAGE
              50        02  D0  0FD3            MOVL    #2,R0                       ; SET ERROR CODE
                      001E  31  0FD6            BRW     EXIT_READHDR                ; AND EXIT
                        53  D4  0FD9    30$:    CLRL    R3                          ; RESET COUNTER
          08400000 8F       DD  0FDB    35$:    PUSHL   #<CSR$M_ASSI!CSR$M_SSEI>    ; DEFINE FLAGS
         FECA CF        01  FB  0FE1            CALLS   #1,GET_POSITION             ; READ SEQUENTIAL HEADER
                     02 50  E8  0FE6            BLBS    R0,40$                      ; BRANCH IF NO ERRORS
```

```
              0C   11  0FE9  3256           BRB     EXIT_READHDR          ; ELSE EXIT WITH ERROR
      85   0140'CF D0  0FEB  3257 40$:      MOVL    T.HDR,(R5)+           ; SAVE HEADER IN BUFFER
         E7 53  20 F2  0FF0  3258           AOBLSS  #32,R3,35$            ; READ NEXT HEADER
            50  01 D0  0FF4  3259           MOVL    #1,R0                 ; SET SUCCESS FLAG
                       0FF7  3260 EXIT_READHDR:
                   04  0FF7  3261           RET                           ; RETURN TO CALLER
```

```
                                   0FF8        .SBTTL  CALCULATE DIFFERENCE ROUTINE
                                   0FF8  ;++
                                   0FF8  ; FUNCTIONAL DESCRIPTION::
                                   0FF8  ;
                                   0FF8  ;       THIS ROUTINE WILL CALCULATE THE DIFFERENCE BETWEEN THE CURRENT
                                   0FF8  ;       CYLINDER AND THE DESTINATION CYLINDER. THE DIRECTION BIT IS ALSO
                                   0FF8  ;       CALCULATED HERE.
                                   0FF8  ;
                                   0FF8  ; CALLING SEQUENCE::
                                   0FF8  ;
                                   0FF8  ;       JSB     CALC_DIFF
                                   0FF8  ;
                                   0FF8  ; INPUT PARAMETERS::
                                   0FF8  ;
                                   0FF8  ;       ** NONE **
                                   0FF8  ;
                                   0FF8  ; IMPLICIT INPUTS::
                                   0FF8  ;
                                   0FF8  ;       CURR_CYLN
                                   0FF8  ;       DEST_CYLN
                                   0FF8  ;
                                   0FF8  ; OUTPUT PARAMETERS::
                                   0FF8  ;
                                   0FF8  ;       ** NONE **
                                   0FF8  ;
                                   0FF8  ; IMPLICIT OUTPUTS::
                                   0FF8  ;
                                   0FF8  ;       HOLD_DA
                                   0FF8  ;       DIFF_WORD
                                   0FF8  ;
                                   0FF8  ; COMPLETION CODES::
                                   0FF8  ;
                                   0FF8  ;       ** NONE **
                                   0FF8  ;
                                   0FF8  ; SIDE EFFECTS::
                                   0FF8  ;
                                   0FF8  ;       ** NONE **
                                   0FF8  ;
                                   0FF8  ;--
                                   0FF8
                                   0FF8  CALC_DIFF::
                     38      BB    0FF8          PUSHR   #^M<R3,R4,R5>                   ; REGISTER SAVE MASK
         53     00D4'CF      D0    0FFA          MOVL    CURR_CYLN,R3                    ; R3 <-- CURRENT CYLINDER
         54     00E4'CF      D0    0FFF          MOVL    DEST_CYLN,R4                    ; R4 <-- DESTINATION CYLINDER
         FFFF 8F     54      B1    1004          CMPW    R4,#-1                          ; IF RECAL OPERATION HAS NOT BEEN -
cont> REQUESTED
                     0B      12    1009          BNEQ    5$                              ; THEN GOTO 5$
                     54      D4    100B          CLRL    R4                              ; ELSE SET DESTINATION TO 0
         00CC'CF     01      C8    100D          BISL2   #RECAL$M_INPROG,POS_TYPE        ; SET RECAL FLAG
             00E4'CF         D4    1012          CLRL    DEST_CYLN                       ; CLEAR SAVED DESTINATION CYLINDER
         55     54   53      C3    1016  5$:     SUBL3   R3,R4,R5                        ; SUBT CURR FROM DESTINATION
                     03      18    101A          BGEQ    10$                             ; BRANCH IF FORWARD SEEK
                55   55      CE    101C          MNEGL   R5,R5                           ; MAKE DIFFERENCE POSITIVE
         0100'CF     55      D0    101F  10$:    MOVL    R5,DIFF_WORD                    ; SAVE DIFFERENCE WORD
                     38      BA    1024          POPR    #^M<R3,R4,R5>                   ; RESTORE REGISTER
                     05            1026          RSB
```

```
                         1027  3319          .SBTTL  ADAPTER MAPPING ROUTINE
                         1027  3320 ;++
                         1027  3321 ; FUNCTIONAL DESCRIPTION::
                         1027  3322 ;
                         1027  3323 ;       THIS PURPOSE OF THIS ROUTINE TO PERFORM THE REQUESTED
                         1027  3324 ;       MAPPING OF THE UNIBUS ADAPTER.
                         1027  3325 ;
                         1027  3326 ; CALLING SEQUENCE::
                         1027  3327 ;
                         1027  3328 ;       PUSHL   256                     ; PUSH BYTE COUNT
                         1027  3329 ;       PUSHAQ  WRITE_BUFFER            ; PUSH ADDRESS OF BUFFER
                         1027  3330 ;       CALLS   #2,$GS_SETMAP           ; MAP UBA DATA PATH #1
                         1027  3331 ;
                         1027  3332 ; INPUT PARAMETERS::
                         1027  3333 ;
                         1027  3334 ;       ** NONE **
                         1027  3335 ;
                         1027  3336 ; IMPLICIT INPUTS::
                         1027  3337 ;
                         1027  3338 ;       ** NONE **
                         1027  3339 ;
                         1027  3340 ; OUTPUT PARAMETERS::
                         1027  3341 ;
                         1027  3342 ;       ** NONE **
                         1027  3343 ;
                         1027  3344 ; IMPLICIT OUTPUTS::
                         1027  3345 ;
                         1027  3346 ;       ** NONE **
                         1027  3347 ;
                         1027  3348 ; COMPLETION CODES::
                         1027  3349 ;
                         1027  3350 ;       ** NONE **
                         1027  3351 ;
                         1027  3352 ; SIDE EFFECTS::
                         1027  3353 ;
                         1027  3354 ;       ** NONE **
                         1027  3355 ;
                         1027  3356 ;--
                         1027  3357
                         1027  3358 $GS_SETMAP::
                   0018  1027  3359         .WORD   ^M<R3,R4>                               ; REGISTER SAVE MASK
   54    04 AC       D0  1029  3360         MOVL    4(AP),R4                                ; GET ADDRESS OF BUFFER
   53    08 AC       D0  102D  3361         MOVL    8(AP),R3                                ; GET BYTE COUNT
                         1031  3362 1$:     $DS_CHANNEL_S   L$UNIT,#CHC$_CLEAR
                         1046  3363         $DS_BNERROR     10$
                         1056  3364         $DS_ERRSYS_S    ,L$UNIT,UBACLEAR_ERR
                         1062       ;;; GLOBAL ERROR 2
                         1069  3365         $DS_CKLOOP      1$
                         1071  3366         $DS_ABORT
                         1078  3367 10$:    $DS_SETMAP_S    L$UNIT,#CHM$_MFWDV,-    ; MAP UBA DATA PATH #1
                         1078  3368                         (R4),#0,R3,#1
                         108D  3369         $DS_BNERROR     20$
                         109D  3370         $DS_ERRSYS_S    ,L$UNIT,UBAMAP_ERR
                         10A9       ;;; GLOBAL ERROR 3
                         10B0  3371         $DS_CKLOOP      10$
                         10B8  3372         $DS_ABORT
```

```
          10BF  3373 20$:     $DS_CHANNEL_S    L$UNIT,#CHC$_PURGE        ; PURGE THE UBA DATA PATH
          10D4  3374          $DS_BNERROR      $GS_SETMAP_X
          10E4  3375          $DS_ERRSYS_S     ,L$UNIT,UBAPURGE_ERR
          10F0       ;;; GLOBAL ERROR 4
          10F7  3376          $DS_CKLOOP       20$
          10FF  3377          $DS_ABORT
          1106  3378 $GS_SETMAP_X:
      04  1106  3379          RET
```

```
                              1107          .SBTTL  READ HOME BLOCK ROUTINE
                              1107
                              1107  ;++
                              1107  ; FUNCTIONAL DESCRIPTION::
                              1107  ;
                              1107  ;       THIS ROUTINE IS CALLED BY THE INITIALIZATION CODE TO
                              1107  ;       READ THE HOME BLOCK. FIRST THE SUBSYSTEM (RL02 OR R80)
                              1107  ;       IS INITIALIZED, THEN THE HEADS ARE POSITIONED OVER CYLINDER
                              1107  ;       ZERO.  THE HOME BLOCK IS THEN READ AND CHECKED BY THE INIT
                              1107  ;       CODE FOR VALIDITY.
                              1107  ;
                              1107  ;
                              1107  ; CALLING SEQUENCE::
                              1107  ;
                              1107  ;       CALLS   #0,RD_HOME_BLK
                              1107  ;
                              1107  ; INPUT PARAMETERS::
                              1107  ;
                              1107  ;       ** NONE **
                              1107  ;
                              1107  ; IMPLICIT INPUTS::
                              1107  ;
                              1107  ;       ** NONE **
                              1107  ;
                              1107  ; OUTPUT PARAMETERS::
                              1107  ;
                              1107  ;       ** NONE **
                              1107  ;
                              1107  ; IMPLICIT OUTPUTS::
                              1107  ;
                              1107  ;       ** NONE **
                              1107  ;
                              1107  ; COMPLETION CODES::
                              1107  ;
                              1107  ;       HMB_ERR_FLAG     = 1 IF ANY ERROR WAS DETECTED
                              1107  ;                        = 0 IF NO ERROR WERE DETECTED
                              1107  ;
                              1107  ; SIDE EFFECTS::
                              1107  ;
                              1107  ;       ** NONE **
                              1107  ;
                              1107  ;--
                              1107
                              1107  RD_HOME_BLK::
                        0000  1107          .WORD   ^M<>                   ; REGISTER SAVE MASK
      F715 CF      00    FB   1109          CALLS   #0,SUBSYS_INIT         ; INIT SUBSYSTEM
         42 50     E9         110E          BLBC    R0,10$                 ; BRANCH IF ANY ERRORS
            00     DD         1111          PUSHL   #0                     ; DEFINE HEAD FOR SEEK
            00     DD         1113          PUSHL   #0                     ; DEFINE DESTINATION CYLN
      FCA4 CF      02    FB   1115          CALLS   #2,POSITION_HDS        ; DO SEEK AND VERIFY
         36 50     E9         111A          BLBC    R0,10$                 ; BRANCH IF ANY ERRORS
   00000200 8F     DD         111D          PUSHL   #512                   ; GET BYTE COUNT
       02C4'CF     7F         1123          PUSHAQ  READ_BUFFER            ; GET BUFFER ADDRESS
      FEFB CF      02    FB   1127          CALLS   #2,$GS_SETMAP          ; MAP DATA PATH #1
   53   0000'CF    D0         112C          MOVL    DRIVE_TBL,R3           ; GET DRIVE INFOR TABLE
            00     DD         1131          PUSHL   #0                     ; IDCMPR DATA
```

```
          5C A3   DD  1133  3437           PUSHL   OFST_HMBSEC(R3)          ; IDCDAR = HOME BLOCK ADDRESS
    FFFFFE00 8F   DD  1136  3438           PUSHL   #-512                    ; IDCCNT = BYTE COUNT
             00   DD  113C  3439           PUSHL   #0                       ; INIT BUS ADDRESS REG
             0C   DD  113E  3440           PUSHL   #READ                    ; DEFINE COMMAND
    F821 CF  05   FB  1140  3441           CALLS   #5,START_CMD             ; EXECUTE READ DATA CMD
    F8D8 CF  00   FB  1145  3442           CALLS   #0,STATUS_CHK            ; CHECK FOR ANY ERRORS
          06 50   E9  114A  3443           BLBC    R0,10$                   ; BRANCH IF ANY ERRORS
        007C'CF   D4  114D  3444           CLRL    HMB_ERR_FLAG             ; FLAG THAT NO ERRORS OCCURED
             05   11  1151  3445           BRB     RD_HOME_BLK_X            ; AND EXIT
   007C'CF   01   D0  1153  3446 10$:      MOVL    #1,HMB_ERR_FLAG          ; SET ERROR FLAG
                      1158  3447 RD_HOME_BLK_X:
                  04  1158  3448           RET
```

```
                                1159       .SBTTL  RANDOM NUMBER GENERATOR
                                1159
                                1159  ;++
                                1159  ; FUNCTIONAL DESCRIPTION::
                                1159  ;
                                1159  ;        THIS ROUTINE GENERATES RANDOM NUMBERS USED IN
                                1159  ;        VARIOUS TEST SEQUENCES.
                                1159  ;
                                1159  ; CALLING SEQUENCE::
                                1159  ;
                                1159  ;        JSB     GET_RANDOM
                                1159  ;
                                1159  ; INPUT PARAMETERS::
                                1159  ;
                                1159  ;        ** NONE **
                                1159  ;
                                1159  ; IMPLICIT INPUTS::
                                1159  ;
                                1159  ;        ** NONE **
                                1159  ;
                                1159  ; OUTPUT PARAMETERS::
                                1159  ;
                                1159  ;        ** NONE **
                                1159  ;
                                1159  ; IMPLICIT OUTPUTS::
                                1159  ;
                                1159  ;        R1 = RANDOM NUMBER
                                1159  ;
                                1159  ; COMPLETION CODES::
                                1159  ;
                                1159  ;        ** NONE **
                                1159  ;
                                1159  ; SIDE EFFECTS::
                                1159  ;
                                1159  ;        ** NONE **
                                1159  ;
                                1159  ;--
                                1159
                                1159  GET_RANDOM::
                      0C   BB   1159          PUSHR   #^M<R2,R3>               ; REGISTER SAVE MASK
                 001C'CF   D5   115B          TSTL    SYS_TIME                 ; IF GENERATOR ALREADY STARTED
                      12   12   115F          BNEQ    10$                      ; THEN GOTO 10$
                                1161          $GETTIM_S -                      ; GET SYSTEM TIME TO START GENERATOR
                                1161            TIMADR=       SYS_TIME         ; PUT 64 BIT TIME HERE
        00F4'CF  001C'CF   3C   116C          MOVZWL  SYS_TIME,SEED            ; AND STORE IN SEED
   12B9B0A1 8F   00F4'CF   7A   1173  10$:    EMUL    SEED,#314159265,-        ; CALC RANDOM NUMBER
        52    1B0CB17F 8F       117C                  #453816703,R2            ; PUT IN R0,R1
   53   52    7FFFFFFF 8F  7B   1182          EDIV    #2147483647,R2,-         ; PUT MODULUS BACK IN SEEK
                 00F4'CF        118A                  R3,SEED                  ;
            51   00F4'CF   D0   118D          MOVL    SEED,R1                  ; LOAD OUTPUT REGISTER
            001C'CF   01   D0   1192          MOVL    #1,SYS_TIME              ; MAKE SURE SYSTIME REMAINS <> 0
                                1197  GET_RANDOM_X:
                      0C   BA   1197          POPR    #^M<R2,R3>               ; RESTORE REGISTERS
                      05        1199          RSB                              ; RETURN TO CALLER
```

```
                                119A          .SBTTL  CHECK ERROR CORRECTION STATUS
                                119A
                                119A    ;++
                                119A    ; FUNCTIONAL DESCRIPTION::
                                119A    ;
                                119A    ;         THIS ROUTIINE WILL PERFORM ECC ERROR CORRECTION
                                119A    ;         AND CHECK THE RESEULT OF THE ERROR CORRECTION
                                119A    ;         WHEN COMPLETE. PREREQUESITES ARE THE DRIVE UNDER
                                119A    ;         TEST IS AN R80, THE CURRENT ERROR IS A DATA CHECK,
                                119A    ;         AND ECC STATUS INDICATES ERROR IS CORRECTABLE.
                                119A    ;
                                119A    ; CALLING SEQUENCE::
                                119A    ;
                                119A    ;         CALLS   #0,CHECK_ECC
                                119A    ;
                                119A    ; INPUT PARAMETERS::
                                119A    ;
                                119A    ;         ** NONE **
                                119A    ;
                                119A    ; IMPLICIT INPUTS::
                                119A    ;
                                119A    ;         READ_BUFFER
                                119A    ;
                                119A    ; OUTPUT PARAMETERS::
                                119A    ;
                                119A    ;         ** NONE **
                                119A    ;
                                119A    ; IMPLICIT OUTPUTS::
                                119A    ;
                                119A    ;         ** NONE **
                                119A    ;
                                119A    ; COMPLETION CODES::
                                119A    ;
                                119A    ;         ** NONE **
                                119A    ;
                                119A    ; SIDE EFFECTS::
                                119A    ;
                                119A    ;         $DS_ERRDEV MESSAGE IF ECC STATUS BITS INDICATE
                                119A    ;         ERROR WAS CORRECTABLE AND DATA COMPARE FOLLOWING
                                119A    ;         ERROR CORRECTION FAILED.
                                119A    ;
                                119A    ;--
                                119A
                                119A    CHECK_ECC::
                          03FC  119A          .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9>
          52   0000'CF    D0    119C          MOVL    DRIVE_TBL,R2                    ; GET DRIVE INFO TABLE
               02    62   91    11A1          CMPB    OFST_DRVTYP(R2),#R80$T          ; IF DRIVE NOT AN R80
                     71   12    11A4          BNEQ    EXIT_ECC                        ; THEN EXIT
                     54   D4    11A6          CLRL    R4                              ; RESET DIVIDEND
  53   0120'CF    02    0A   EF    11A8          EXTZV   #CSR$V_OPI,#2,T.CS,R3           ; GET IDC ERROR BITS
               02    53   91    11AF          CMPB    R3,#2                           ; IF NO A DATA CHECK ERROR
                     63   12    11B2          BNEQ    EXIT_ECC                        ; THEN EXIT
  53   0120'CF    02    14   EF    11B4          EXTZV   #CSR$V_ECS0,#2,T.CS,R3          ; GET ECC STATUS
               03    53   91    11BB          CMPB    R3,#3                           ; IF NOT CORRECTABLE ERROR
                     57   12    11BE          BNEQ    EXIT_ECC                        ; THEN EXIT
          53   014C'CF    D0    11C0          MOVL    T.POS,R3                        ; GET POSITION OF ECC ERROR
```

```
  55   54   53   20   7B  11C5  3561          EDIV    #32,R3,R4,R5                  ; R4 = LWORDS INTO BUFFER
                          11CA  3562                                                ; R5 = BITS LEFT OVER
       56   02C4'CF   D0  11CA  3563          MOVL    READ_BUFFER,R6                ; GET ADDRESS OF BUFFER
          57   6644   DE  11CF  3564          MOVAL   (R6)[R4],R7                   ; GET WORD IN ERROR
  58   67   0B   55   EF  11D3  3565          EXTZV   R5,#PAT$S_BITS,(R7),R8        ; EXTRACT THE 11 BIT BURST
  59   58   0148'CF   CD  11D8  3566          XORL3   T.PAT,R8,R9                   ; CALC THE CORRECTED DATA
  67   0B   55   59   F0  11DE  3567          INSV    R9,R5,#PAT$S_BITS,(R7)        ; LOAD WORD WITH GOOD DATA NOW
            02BC'CF   DD  11E3  3568          PUSHL   WRITE_BUFFER                  ; GET ADDRESS OF GOOD DATA
            02C4'CF   DD  11E7  3569          PUSHL   READ_BUFFER                   ; GET ADDRESS OF UNKNOWN DATA
            01BC'CF   DD  11EB  3570          PUSHL   WORD_COUNT                    ; GET NUMBER OF WORDS TO  CHECK
       F552 CF   03   FB  11EF  3571          CALLS   #3,CMPDATA                    ; CHECK THE DATA
              15 50   E8  11F4  3572          BLBS    R0,10$                        ; EXIT IF DATA COMPARED OK
                          11F7  3573          $DS_ERRDEV_S      ,L$UNIT,MSG_ECCFAIL
                          1203          ;;; GLOBAL ERROR 5
                0B   11   120A  3574          BRB     EXIT_ECC                      ; AND EXIT
                          120C  3575 10$:     $DS_PRINTF_S    MSG_ECCPASS           ; PRINT ECC WORKED OK
                          1217  3576 EXIT_ECC:
                     04   1217  3577          RET                                   ; RETURN TO CALLER
```

```
                         1218          .SBTTL  CONVERT DISK ADDRESS REGISTER
                         1218   ;++
                         1218   ; FUNCTIONAL DESCRIPTION::
                         1218   ;
                         1218   ; THE INTEGRATED DISK CONTROLLER MICROCODE PERFORMS DATA
                         1218   ; MANIPULATION ON THE DATA PRESENTED TO THE DISK ADDRESS
                         1218   ; REGISTER.  THEREFORE DATA LOAD IN A SPECIFIC PATTERN FOR
                         1218   ; CYLINDER, TRACK, AND SECTOR WILL BE REARRANGED TO MEET
                         1218   ; THE NEEDS OF THE CURRENT DISK SELECTED.  THIS ROUTINE
                         1218   ; WILL REBUILD THE DATA LOADED INTO THE IDCDAR SO THAT EXPECTED
                         1218   ; VALUES FOR THE DAR MAY BE TESTED AT A LATER TIME.
                         1218   ;
                         1218   ; CALLING SEQUENCE::
                         1218   ;
                         1218   ;       PUSHL   DAR LOAD DATA
                         1218   ;       CALLS   #1,CHANGE_DAR$A
                         1218   ;
                         1218   ; INPUT PARAMETERS::
                         1218   ;
                         1218   ;       DISK ADDRESS REGISTER IN THE FOLLOWING FORM
                         1218   ;         1. SECTOR ADDRESS    (BITS 0 - 7)
                         1218   ;         2. TRACK ADDRESS     (BITS 8 -15)
                         1218   ;         3. CYLINDER ADDRESS (BITS 16 - 31)
                         1218   ;
                         1218   ; IMPLICIT INPUTS::
                         1218   ;
                         1218   ;       DRIVE TYPE TABLE
                         1218   ;
                         1218   ; OUTPUT PARAMETERS::
                         1218   ;
                         1218   ;       R1 CONTAINS REFORMED DISK ADDRESS REGISTER IN A FORM
                         1218   ;       RESEMBLING THE MICROCODE CONVERSION APPLICABLE TO THE
                         1218   ;       SPECIFIC DISK CURRENTLY UNDER TEST.
                         1218   ;
                         1218   ; IMPLICIT OUTPUTS::
                         1218   ;
                         1218   ;       LOAD PROPER BIT MNEMONICS INTO DAR_MNEM FOR
                         1218   ;       ERROR REPORTING
                         1218   ;
                         1218   ; COMPLETION CODES::
                         1218   ;
                         1218   ;       ** NONE **
                         1218   ;
                         1218   ; SIDE EFFECTS::
                         1218   ;
                         1218   ;       ** NONE **
                         1218   ;
                         1218   ;--
                         1218
                         1218   .ENABL LSB
                         1218   CHANGE_DAR$A::
                   007C  1218          .WORD   ^M<R2,R3,R4,R5,R6>
   52  0000'CF      D0   121A          MOVL    DRIVE_TBL,R2                  ; GET DRIVE INFOROMATION
   53  0114'CF      DE   121F          MOVAL   WORK,R3                       ; GET ADDRESS OF WORK LOCATION
    63    04 AC     D0   1224          MOVL    4(AP),(R3)                    ; GET INPUTTED DAR DATA
       54    83     9A   1228          MOVZBL  (R3)+,R4                      ; GET SECTOR NUMBER
```

```
                55   83   9A  122B             MOVZBL  (R3)+,R5                      ; GET TRACK NUMBER
                56   63   3C  122E             MOVZWL  (R3),R6                       ; GET CYLINDER ADDRESS
                              1231             CASE    TYPE=B,SRC=OFST_DRVTYP(R2),-  ; CASE ON DEVICE TYPE DATA
                              1231                     LIMIT=#RL02$T,DISPLIST=<-
                              1231                     FIX02_DAR,-
                              1231                     FIX80_DAR>
                              1239
                              1239        FIX02_DAR:
                     51   D4  1239             CLRL    R1                            ; SETUP OUTPUT
  00BC'CF    0294'CF      DE  123B             MOVAL   DAR_DMA02_BITS,DAR_MNEM       ; SETUP REGISTER BIT MNEMONICS
  51   06    00      54   F0  1242             INSV    R4,#0,#6,R1                   ; LOAD SECTOR
  51   01    06      55   F0  1247             INSV    R5,#6,#1,R1                   ; LOAD TRACK
  51   09    07      56   F0  124C             INSV    R6,#7,#9,R1                   ; LOAD CYLINDER
                     18   11  1251             BRB     EXIT_CHANGE                   ; AND EXIT
                              1253
                              1253        FIX80_DAR:
                     51   D4  1253             CLRL    R1                            ; SETUP OUTPUT
  00BC'CF    02C5'CF      DE  1255             MOVAL   DAR_DMA80_BITS,DAR_MNEM       ; SETUP REGISTER BIT MNEMONICS
  51   05    00      54   F0  125C             INSV    R4,#0,#5,R1                   ; LOAD SECTOR
  51   04    05      55   F0  1261             INSV    R5,#5,#4,R1                   ; LOAD TRACK
  51   0A    09      56   F0  1266             INSV    R6,#9,#10,R1                  ; LOAD CYLINDER
                              126B
                              126B        EXIT_CHANGE:
                          04  126B             RET                                   ; RETURN TO CALLER
                              126C        .DSABL LSB
```

```
                              126C        .SBTTL  GET LOGICAL UNIT ROUTINE
                              126C
                              126C  ;++
                              126C  ; FUNCTIONAL DESCRIPTION::
                              126C  ;
                              126C  ; THIS ROUTINE PROVIDES PROGRAM SETUP FOR THE NEXT LOGCIAL
                              126C  ; DEVICE TO TEST IN THE TEST SEQUENCE.
                              126C  ;
                              126C  ; CALLING SEQUENCE::
                              126C  ;
                              126C  ;       PUSHL   UNIT
                              126C  ;       CALLS   #1,GET_LUN$A
                              126C  ;
                              126C  ; INPUT PARAMETERS::
                              126C  ;
                              126C  ;       LOGICAL UNIT NUMBER
                              126C  ;
                              126C  ; IMPLICIT INPUTS::
                              126C  ;
                              126C  ;       ** NONE **
                              126C  ;
                              126C  ; OUTPUT PARAMETERS::
                              126C  ;
                              126C  ;       ** NONE **
                              126C  ;
                              126C  ; IMPLICIT OUTPUTS::
                              126C  ;
                              126C  ;       ** NONE **
                              126C  ;
                              126C  ; COMPLETION CODES::
                              126C  ;
                              126C  ;       STANDARD COMPLETION CODES
                              126C  ;
                              126C  ; SIDE EFFECTS::
                              126C  ;
                              126C  ;       ** NONE **
                              126C  ;
                              126C  ;--
                              126C
                              126C  GET_LUN$A::
                        001C  126C        .WORD   ^M<R2,R3,R4>
                     50   D4  126E        CLRL    R0                            ; INIT RETURN STATUS
      53   0000FE0C'EF   01   C3  1270        SUBL3   #1,DSA$GL_UNITS,R3            ; GET HIGHEST LOGICAL UNIT NUMBER
             52    04 AC   D0  1278        MOVL    4(AP),R2                      ; PICKUP REQUESTED LUN
                  53   52   D1  127C        CMPL    R2,R3                         ; IF DEVICE IS SELECTED
                       02   15  127F        BLEQ    10$                           ; THEN GOTO 10$
                       33   11  1281        BRB     EXIT_GETLUN                   ; AND EXIT
         2D 0054'CF   52   E1  1283  10$:  BBC     R2,ABORT_FLAG,EXIT_GETLUN     ; EXIT IF OPERATOR SAID NO TEST TH -
cont> IS ONE
      0000'CF    02DC'CF42   D0  1289        MOVL    LUN_TBL[R2],DRIVE_TBL         ; LOAD DRIVE TABLE
            002C'CF   52   D0  1291        MOVL    R2,L$UNIT                     ; SET LOGICAL UNIT VALUE
          53    0000'CF   D0  1296        MOVL    DRIVE_TBL,R3                  ; GET DRIVE INFO TABLE
      0044'CF    20 A3   7D  129B        MOVQ    OFST_DEVNAM(R3),DEV_NAME      ; GET DEVICE NAME DESCRIPTOR
        54    04 A3   10   C1  12A1        ADDL3   #16,OFST_PHYU(R3),R4          ; GET BIT POSITION FOR ATTENTION BIT
        003C'CF   01   54   9C  12A6        ROTL    R4,#1,P$ATTN                  ; POSITION EXPECTED ATTENTION BIT
      0034'CF    04 A3   08   9C  12AC        ROTL    #8,OFST_PHYU(R3),P$UNIT       ; POSITION UNIT SELECT BITS
                  50   01   D0  12B3        MOVL    #1,R0                         ; SET DEVICE SETUP DONE FLAG
```

```
       12B6  3717 EXIT_GETLUN:
   04  12B6  3718          RET                                      ; RETURN TO CALLER
```

```
                                12B7         .SBTTL  GET SECONDARY UNIT ROUTINE
                                12B7
                                12B7   ;++
                                12B7   ; FUNCTIONAL DESCRIPTION::
                                12B7   ;
                                12B7   ; THIS ROUTINE PROVIDES PROGRAM SETUP FOR A SECONDARY LOGCIAL
                                12B7   ; UNIT TO USE OTHER THAN THE CURRENT UNIT UNDER TEST.
                                12B7   ; THIS IS PRIMARLY USED IN THE OVERLAP SEEK TEST
                                12B7   ;
                                12B7   ; CALLING SEQUENCE::
                                12B7   ;
                                12B7   ;        PUSHL   CURRENT UNIT
                                12B7   ;        CALLS   #1,GET_SECLUN$A
                                12B7   ;
                                12B7   ; INPUT PARAMETERS::
                                12B7   ;
                                12B7   ;        LOGICAL UNIT NUMBER
                                12B7   ;
                                12B7   ; IMPLICIT INPUTS::
                                12B7   ;
                                12B7   ;        ** NONE **
                                12B7   ;
                                12B7   ; OUTPUT PARAMETERS::
                                12B7   ;
                                12B7   ;        ** NONE **
                                12B7   ;
                                12B7   ; IMPLICIT OUTPUTS::
                                12B7   ;
                                12B7   ;        ** NONE **
                                12B7   ;
                                12B7   ; COMPLETION CODES::
                                12B7   ;
                                12B7   ;        STANDARD COMPLETION CODES
                                12B7   ;
                                12B7   ; SIDE EFFECTS::
                                12B7   ;
                                12B7   ;        ** NONE **
                                12B7   ;
                                12B7   ;--
                                12B7
                                12B7   GET_SECLUN$A::
                          001C  12B7         .WORD   ^M<R2,R3,R4>
                     50     D4  12B9         CLRL    R0                         ; INIT RETURN STATUS
 52    0000FE0C'EF   01     C3  12BB         SUBL3   #1,DSA$GL_UNITS,R2         ; GET HIGHEST LOGICAL UNIT NUMBER
                     2F     13  12C3         BEQL    EXIT_GETSEC                ; BR IF ONLY ONE DRIVE SELECTED
           04 AC     52     D1  12C5         CMPL    R2,4(AP)                   ; IF WE HAVE A DIFFERENT LUN
                     02     12  12C9         BNEQ    10$                        ; THEN GOTO 10$
                     52     D7  12CB         DECL    R2                         ; ELSE CHOOSE ANOTHER LUN
  0004'CF    02DC'CF42      D0  12CD   10$:  MOVL    LUN_TBL[R2],S_DRIVE_TBL    ; GET DRIVE TABLE
           0030'CF   52     D0  12D5         MOVL    R2,S_L$UNIT                ; SAVE IS LOGICAL UNIT NUMBER
           53    0004'CF    D0  12DA         MOVL    S_DRIVE_TBL,R3             ; GET ADDRESS OF DEVICE INFO
      54   04 A3     10     C1  12DF         ADDL3   #16,OFST_PHYU(R3),R4       ; GET BIT POSITION FOR ATTENTION BIT
    0040'CF  01      54     9C  12E4         ROTL    R4,#1,S_P$ATTN             ; POSITION EXPECTED ATTENTION BIT
  0038'CF    04 A3   08     9C  12EA         ROTL    #8,OFST_PHYU(R3),S_P$UNIT  ; POSITION UNIT SELECT BITS
           50        01     D0  12F1         MOVL    #1,R0                      ; SET DEVICE SETUP DONE FLAG
                                12F4   EXIT_GETSEC:
```

```
       04  12F4  3776          RET                                  ; RETURN TO CALLER
```

```
                              12F5  3778          .SBTTL  IDC INTERRUPT SERVICE ROUTINE
                              12F5  3779
                              12F5  3780 ;++
                              12F5  3781 ; FUNCTIONAL DESCRIPTION::
                              12F5  3782 ;
                              12F5  3783 ; THIS IS THE INTERRUPT SERVICE ROUTINE FOR THE IDC CONTROLLER.
                              12F5  3784 ; IT IS NORMALLY USED DURING INTERRUPT, SEEK AND ROTATIONAL TIMING
                              12F5  3785 ; TESTS.
                              12F5  3786 ;
                              12F5  3787 ; CALLING SEQUENCE::
                              12F5  3788 ;
                              12F5  3789 ;         **NONE** (HARDWARE INITIATED)
                              12F5  3790 ;
                              12F5  3791 ; INPUT PARAMETERS::
                              12F5  3792 ;
                              12F5  3793 ;         ** NONE **
                              12F5  3794 ;
                              12F5  3795 ; IMPLICIT INPUTS::
                              12F5  3796 ;
                              12F5  3797 ;         ** NONE **
                              12F5  3798 ;
                              12F5  3799 ; OUTPUT PARAMETERS::
                              12F5  3800 ;
                              12F5  3801 ;         ** NONE **
                              12F5  3802 ;
                              12F5  3803 ; IMPLICIT OUTPUTS::
                              12F5  3804 ;
                              12F5  3805 ;         ** NONE **
                              12F5  3806 ;
                              12F5  3807 ; COMPLETION CODES::
                              12F5  3808 ;
                              12F5  3809 ;         ** NONE **
                              12F5  3810 ;
                              12F5  3811 ; SIDE EFFECTS::
                              12F5  3812 ;
                              12F5  3813 ; PROVIDES CANCEL TIMER SERVICES FOR SEEK AND
                              12F5  3814 ; ROTATIONAL TIMING.
                              12F5  3815 ;
                              12F5  3816 ;--
                              12F5  3817          .ALIGN LONG
                              12F8  3818 IDC_ISR$A::
                     03   BB  12F8  3819          PUSHR   #^M<R0,R1>                  ; SAVE STANDARD REGISTERS
                     04   BB  12FA  3820          PUSHR   #^M<R2>                     ; SAVE R2
             52  0014'CF  D0  12FC  3821          MOVL    BASE_ADDR,R2                ; GET IDC ADDRESS
            0114'CF   62  D0  1301  3822          MOVL    IDCCSR(R2),WORK             ; GET CSR REGISTER
       0114'CF  003C'CF   CA  1306  3823          BICL2   P$ATTN,WORK                 ; RESET ATTENTION FOR THIS DRIV
 0114'CF  0114'CF   04    10  EF  130D  3824      EXTZV   #16,#4,WORK,WORK            ; GET ATTENTION BITS
                     27   13  1316  3825          BEQL    10$                         ; IF NO ATTENTION LEFT GOTO 10$
         62  00000040 8F  CA  1318  3826          BICL2   #CSR$M_IE,IDCCSR(R2)        ; TURN OFF INTERRUPTS
                F5CC CF   16  131F  3827          JSB     RD_IDCREGS                  ; SNAPSHOT REGISTERS
                              1323  3828          $DS_ERRSYS_S    ,L$UNIT,MSG39A_INT,-
                              1323  3829                          PRINT_REG
                              1331       ;;; GLOBAL ERROR 6
                              1338  3830          $DS_ABORT                           ; DIE DUE TO UNEXPECTED DRIVE ATTN
         62  000F0000 8F  C8  133F  3831 10$:     BISL2   #^XF0000,IDCCSR(R2)         ; RESET ATTENTION BITS
         62  01000000 8F  C8  1346  3832          BISL2   #CSR$M_IR,IDCCSR(R2)        ; RESET PENDING INTERRUPTS
```

```
04   BA  134D  3833        POPR    #^M<R2>                 ; RESTORE R2
         134F  3834        $DS_CANWAIT_S                   ; STOP TIMER
         1356  3835        $DS_ENDSERV                     ; RESTORE R0,R1 AND RETURN (REI)
```

```
                              1359         .SBTTL  READ BAD SECTOR FILE ROUTINE
                              1359
                              1359  ;++
                              1359  ; FUNCTIONAL DESCRIPTION::
                              1359  ;
                              1359  ; THIS ROUTINE WILL LOG THE DEVICE BAD SECTOR FILE INTO
                              1359  ; MEMORY FOR FUTURE USE
                              1359  ;
                              1359  ; CALLING SEQUENCE::
                              1359  ;
                              1359  ;      CALLS   #0,RD_BAD_BLK
                              1359  ;
                              1359  ; INPUT PARAMETERS::
                              1359  ;
                              1359  ;      DRIVE_TABLE
                              1359  ;
                              1359  ; IMPLICIT INPUTS::
                              1359  ;
                              1359  ;      ** NONE **
                              1359  ;
                              1359  ; OUTPUT PARAMETERS::
                              1359  ;
                              1359  ;      ** NONE **
                              1359  ;
                              1359  ; IMPLICIT OUTPUTS::
                              1359  ;
                              1359  ;      ** NONE **
                              1359  ;
                              1359  ; COMPLETION CODES::
                              1359  ;
                              1359  ;      R0 = 0 UNABLE TO READ FILE OR FILE IS CORRUPTED
                              1359  ;      R0 = 1 FILE HAS BEEN READ AND VALIDATED
                              1359  ;
                              1359  ; SIDE EFFECTS::
                              1359  ;
                              1359  ;      ** NONE **
                              1359  ;
                              1359  ;--
                              1359
                              1359  RD_BAD_BLK::
                       0FFC   1359         .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
         54   0000'CF    D0   135B         MOVL    DRIVE_TBL,R4                ; GET ADDRESS OF DRIVE TABLE
              56    02   D0   1360         MOVL    #2,R6                       ; SET INCREMENT VALUE
              58    12   D0   1363         MOVL    #18,R8                      ; AND UPPER SECTOR LIMIT
         53    08   A4   D0   1366         MOVL    OFST_LSTCYL(R4),R3          ; GET LAST CYLINDER
              01    64   91   136A         CMPB    OFST_DRVTYP(R4),#RL02$T     ; IF DRIVE IS AN RL02
                    09   13   136D         BEQL    20$                         ; THEN GOTO 20$
              53    02   C2   136F         SUBL2   #2,R3                       ; POINT TO BADBLK CYLINDER FOR R80
              56    01   D0   1372         MOVL    #1,R6                       ; SET INCREMENT VAL FOR 80
              58    09   D0   1375         MOVL    #9,R8                       ; SET UPPER LIMIT FOR R80
                    55   D4   1378  20$:   CLRL    R5                          ; RESET REGISTER
     55    10    10    53    F0   137A         INSV    R3,#16,#16,R5               ; LOAD CYLINDER ADDRESS
 55    08    08    0C A4    F0   137F         INSV    OFST_LSTTRK(R4),#8,#8,R5;   LOAD TRACK ADDRESS
                 0C A4    DD   1385         PUSHL   OFST_LSTTRK(R4)             ; TRACK FOR SEEK
                    53    DD   1388         PUSHL   R3                          ; CYLINDER FOR SEEK
       FA2F CF      02    FB   138A         CALLS   #2,POSITION_HDS             ; DO THE SEEK
```

```
             51 50   E9  138F  3893         BLBC    R0,EXIT_READBAD         ; IF ERROR THEN GOTO EXIT_READBAD
        00000200 8F  DD  1392  3894         PUSHL   #512                    ; MAP DATA PATH FOR THIS MANY BYTES
          02D4'CF    7F  1398  3895         PUSHAQ  MBSF_BUFFER             ; AND THIS BUFFER ADDRESS
      FC86 CF   02   FB  139C  3896         CALLS   #2,$GS_SETMAP           ; DO IT
                00   DD  13A1  3897 30$:    PUSHL   #0                      ; IDCMPR DATA
                55   DD  13A3  3898         PUSHL   R5                      ; IDC DAR DATA
        FFFFFE00 8F  DD  13A5  3899         PUSHL   #-512                   ; NUMBER OF BYTES TO READ
                00   DD  13AB  3900         PUSHL   #0                      ; STARTING ADDRESS
                0C   DD  13AD  3901         PUSHL   #READ                   ; FUNCTION IS READ DATA
      F5B2 CF   05   FB  13AF  3902         CALLS   #5,START_CMD            ; START READ FUNCTION
      F669 CF   00   FB  13B4  3903         CALLS   #0,STATUS_CHK           ; CHECK FOR ANY ERRORS
             1F 50   E9  13B9  3904         BLBC    R0,40$                  ; IF ERROR THEN GOTO 40$
      5A    0800'CF  DE  13BC  3905         MOVAL   MBSF,R10                ; GET ADDRESS OF THE FILE
             04 AA   D5  13C1  3906         TSTL    4(R10)                  ; IF THESE 2 WORDS <> 0
                15   12  13C4  3907         BNEQ    40$                     ; THEN GOTO 40$ FILE IS CORRUPT
                59   D4  13C6  3908         CLRL    R9                      ; RESET INDEX REGISTER
           08 AA49   D5  13C8  3909 35$:    TSTL    8(R10)[R9]              ; IF ANY ENTRY IS = 0
                0D   13  13CC  3910         BEQL    40$                     ; THEN GOTO 40$ FILE IS CORRUPT
  F2 59  0000007E 8F F2  13CE  3911         AOBLSS  #126,R9,35$             ; CHECK NEXT ENTRY
           50   01   D0  13D6  3912         MOVL    #1,R0                   ; SET SUCCESS CODE
                08   11  13D9  3913         BRB     EXIT_READBAD            ; AND EXIT
   FFC0 55   56 58   9D  13DB  3914 40$:    ACBB    R8,R6,R5,30$            ; TRY NEXT FILE
                50   D4  13E1  3915         CLRL    R0                      ; SET FAILURE CODE
                         13E3  3916 EXIT_READBAD:
                     04  13E3  3917         RET                             ; RETURN TO CALLER
                         13E4  3918
```

```
                                      13E4          .SBTTL  CHECK BAD SECTOR FILE ROUTINE
                                      13E4
                                      13E4     ;++
                                      13E4     ; FUNCTIONAL DESCRIPTION::
                                      13E4     ;
                                      13E4     ; THIS ROUTINE CHECKS THE BAD BLOCK FILE FOR A PARTICULAR
                                      13E4     ; CYLINDER TRACK AND SECTOR COMBINATION
                                      13E4     ;
                                      13E4     ; CALLING SEQUENCE::
                                      13E4     ;
                                      13E4     ;       PUSHL   SECTOR
                                      13E4     ;       PUSHL   TRACK
                                      13E4     ;       PUSHL   CYLINDER
                                      13E4     ;       CALLS   #3,CHECK_BADBLK
                                      13E4     ;
                                      13E4     ; INPUT PARAMETERS::
                                      13E4     ;
                                      13E4     ;       CYLINDER,TRACK, SECTOR AND BAD BLOCK FILE
                                      13E4     ;
                                      13E4     ; IMPLICIT INPUTS::
                                      13E4     ;
                                      13E4     ;       ** NONE **
                                      13E4     ;
                                      13E4     ; OUTPUT PARAMETERS::
                                      13E4     ;
                                      13E4     ;       ** NONE **
                                      13E4     ;
                                      13E4     ; IMPLICIT OUTPUTS::
                                      13E4     ;
                                      13E4     ;       ** NONE **
                                      13E4     ;
                                      13E4     ; COMPLETION CODES::
                                      13E4     ;
                                      13E4     ;       R0 = 0 SECTOR IS NOT IS BAD BLOCK FILE
                                      13E4     ;       R0 = 1 SECTOR IS IN BAD BLOCK FILE
                                      13E4     ;
                                      13E4     ; SIDE EFFECTS::
                                      13E4     ;
                                      13E4     ;       ** NONE **
                                      13E4     ;
                                      13E4     ;--
                                      13E4
                                      13E4     CHECK_BADBLK::
                                0FFC  13E4          .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
              54    0800'CF     DE    13E6          MOVAL   MBSF,R4                 ; GET ADDRESS OF BAD BLK FILE
                            55  D4    13EB          CLRL    R5                      ; RESET REGISTER
     55   10    00    04 AC     F0    13ED          INSV    4(AP),#0,#16,R5         ; SET CYLINDER
     55   08    10    08 AC     F0    13F3          INSV    8(AP),#16,#8,R5         ; SET TRACK
     55   08    18    0C AC     F0    13F9          INSV    12(AP),#24,#8,R5        ; SET SECTOR
                            56  D4    13FF          CLRL    R6                      ; RESET INDEX REGISTER
                      50    01  D0    1401          MOVL    #1,R0                   ; ASSUME SECTOR WILL BE THERE
              08 A446       55  D1    1404     10$: CMPL    R5,8(R4)[R6]            ; IF SECTOR IS FILED
                            15  13    1409          BEQL    EXIT_CHECK              ; THEN EXIT
     08 A446     FFFFFFFF   8F  D1    140B          CMPL    #-1,8(R4)[R6]           ; IF END OF FILE
                            08  13    1414          BEQL    20$                     ; THEN GOTO 20$
       E6 56     0000007E   8F  F2    1416          AOBLSS  #126,R6,10$             ; ELSE CHECK NEXT ENTRY
```

```
            50   D4  141E  3976 20$:     CLRL    R0                          ; SECTOR IS NOT FILED
                     1420  3977 EXIT_CHECK:
                 04  1420  3978          RET                                 ; RETURN TO CALLER
                     1421  3979          $DS_ENDMOD
                     1421  3980          .END
```

```
$$ARGS                 = 0000000A
$$E                    = 00000001
$$M                    = 00000004
$$N                    = 00000000
$$S                    = FFFFFFFF
$$T1                   = 00000000
$ENV                   = 00000003  G
$ER                    = 00000007
$GS_SETMAP               00001027 RG      0E
$GS_SETMAP_X             00001106 R       0E
$MO                    = 00000001  G
$ST                    = 00000000
$TN                    = 00000000
ABORT_FLAG               00000054 RG      08
ACT_NAME                 0000004C RG      08
ALL                    = 00000002  G
AL_DEVTYP                0000002D R       0A
A_HEADEND                00000058 R       03
BAD_DATA                 000001A4 RG      08
BASE_ADDR                00000014 RG      08
BDDAT                    00000106 RG      08
BIT...                 = 00000028
BIT0                   = 00000001
BIT1                   = 00000002
BIT10                  = 00000400
BIT11                  = 00000800
BIT12                  = 00001000
BIT13                  = 00002000
BIT14                  = 00004000
BIT15                  = 00008000
BIT16                  = 00010000
BIT17                  = 00020000
BIT18                  = 00040000
BIT19                  = 00080000
BIT2                   = 00000004
BIT20                  = 00100000
BIT21                  = 00200000
BIT22                  = 00400000
BIT23                  = 00800000
BIT24                  = 01000000
BIT25                  = 02000000
BIT26                  = 04000000
BIT27                  = 08000000
BIT28                  = 10000000
BIT29                  = 20000000
BIT3                   = 00000008
BIT30                  = 40000000
BIT31                  = 80000000
BIT4                   = 00000010
BIT5                   = 00000020
BIT6                   = 00000040
BIT7                   = 00000080
BIT8                   = 00000100
BIT9                   = 00000200
BUFFER1                  00000000 RG      09
BUFFER2                  00000400 RG      09
BUFF_INUSE               000000A4 RG      08
BUFLEN                 = 00000064
BUILD_HDRS               00000F4C RG      0E
CALC_DIFF                00000FF8 RG      0E
CEP_FUNCTIONAL         = 00000000
CEP_REPAIR             = 00000001
CHANGE_DAR$A             00001218 RG      0E
CHAR_BUFFER              000001C0 RG      08
CHC$_ABORT             = 00000002
CHC$_CLEAR             = 00000006
CHC$_CLRDFT            = 00000009
CHC$_DSINT             = 00000005
CHC$_ENINT             = 00000004
CHC$_INITA             = 00000000
CHC$_INITB             = 00000001
CHC$_PURGE             = 00000003
CHC$_SELF_TEST         = 0000000B
CHC$_SETDFT            = 00000008
CHC$_STATUS            = 00000007
CHC$_STOP              = 0000000A
CHECK_BADBLK             000013E4 RG      0E
CHECK_ECC                0000119A RG      0E
CHECK_FATAL              00000C4C RG      0E
CHI$M_CHNINT           = 00000001
CHI$M_DEVINT           = 00000002
CHI$M_IPL              = 0000007C
CHI$S_IPL              = 00000005
CHI$V_CHNINT           = 00000000
CHI$V_DEVINT           = 00000001
CHI$V_IPL              = 00000002
CHKFTL_R80               00000C87 R       0E
CHKFTL_RL02              00000C61 R       0E
CHM$_FORWARD           = 00000000
CHM$_INVALIDATE        = 00000001
CHM$_MAP               = 00000002
CHM$_MFWDN             = 00000002
CHM$_MFWDNO            = 0000000A
CHM$_MFWDV             = 00000003
CHM$_MFWDVO            = 0000000B
CHM$_MREVN             = 00000006
CHM$_MREVNO            = 0000000E
CHM$_MREVV             = 00000007
CHM$_MREVVO            = 0000000F
CHM$_NFWDN             = 00000000
CHM$_NREVN             = 00000004
CHM$_OFFSET            = 00000008
CHM$_REVERSE           = 00000004
CLEAN_UP                 00000000 R       0C
CLEAN_UP_X               00000071 R       0C
CMPDATA                  00000746 RG      0E
CMPDATAX                 000007A4 R       0E
COM_BIT16                000000EF RG      0A
COM_BIT32                00000039 RG      0A
CONFIG                   0000011C RG      08
CRC_TABLE                00000228 RG      08
CSR$M_ASSI             = 08000000
```

```
CSR$M_CRDY          = 00000080                 CVT_UNLOAD          0000070E RG   0A
CSR$M_DCK           = 00000800                 CVT_UPPER           0000072C RG   0A
CSR$M_DE            = 00004000                 CVT_WRITE           0000061F RG   0A
CSR$M_DRDY          = 00000001                 CVT_WRTCHK          000005F7 RG   0A
CSR$M_ERR           = 00008000                 CYLN                000000D0 RG   08
CSR$M_HNF           = 00001000                 CYLN_LIST           00000268 RG   08
CSR$M_IE            = 00000040                 CYLN_LIST_X         000002B2 RG   08
CSR$M_IR            = 01000000                 DAR_DMA02_BITS      00000294 RG   0A
CSR$M_NXM           = 00002000                 DAR_DMA80_BITS      000002C5 RG   0A
CSR$M_OPI           = 00000400                 DAR_GS02_BITS       000003AA RG   0A
CSR$M_R80           = 04000000                 DAR_MNEM            000000BC RG   08
CSR$M_SSEI          = 00400000                 DAR_RCL80_BITS      00000370 RG   0A
CSR$V_CRDY          = 00000007                 DAR_SK02_BITS       000002F4 RG   0A
CSR$V_DE            = 0000000E                 DAR_SK80_BITS       00000337 RG   0A
CSR$V_DRDY          = 00000000                 DEF               = 00000080
CSR$V_ECS0          = 00000014                 DEFAULT           = 00000000  G
CSR$V_ERR           = 0000000F                 DEL$V_READH       = 0000000A
CSR$V_OPI           = 0000000A                 DELTA_TIME          00000084 RG   08
CSR$V_R80           = 0000001A                 DERR_FLAG           000000B4 RG   08
CSR_BITS              000001CA RG   0A         DEST_CYLN           000000E4 RG   08
CURR_CMD              000000FC RG   08         DEV                 00000008 RG   08
CURR_CYLN             000000D4 RG   08         DEV_NAME            00000044 RG   08
CURR_CYL_HD           000000DC RG   08         DEV_REG             000000C0 RG   08
CURR_FLAGS            00000028 RG   08         DIFF_WORD           00000100 RG   08
CURR_HD               000000D8 RG   08         DISK_SECTION        000000AC RG   08
CURR_SECTOR           000000F0 RG   08         DISPATCH            00000000 R    06
CVT_BRUSH             000006DC RG   0A         DMA_CMD             00000C08 R    0E
CVT_DIR_TABLE         000006A6 RG   08         DRIVE_INIT$A        000007A5 RG   0E
CVT_DRVTYP            00000622 RG   08         DRIVE_TBL           00000000 RG   08
CVT_ERROR0            00000646 RG   0A         DRV_FATAL           00000C92 R    0E
CVT_ERROR1            0000064B RG   0A         DS$ABORT            00010020  G
CVT_ERROR2            00000660 RG   0A         DS$ASKADR           00010090  G
CVT_ERROR3            00000674 RG   0A         DS$ASKDATA          00010080  G
CVT_ERROR4            0000067C RG   0A         DS$ASKLGCL          00010098  G
CVT_ERROR5            00000686 RG   0A         DS$ASKSTR           000100A0  G
CVT_ERROR6            00000694 RG   0A         DS$ASKVLD           00010088  G
CVT_ERROR7            000006A9 RG   0A         DS$ATTACH           000101A8  G
CVT_ERR_TABLE         00000676 RG   08         DS$BGNSUB           00010030  G
CVT_FORWARD           00000740 RG   0A         DS$BRANCH           000100A8  G
CVT_FUN_TABLE         0000062E RG   08         DS$BREAK            00010058  G
CVT_GETSTAT           00000603 RG   0A         DS$CANWAIT          00010070  G
CVT_LDHEADS           000006E8 RG   0A         DS$CHANNEL          00010180  G
CVT_LOAD              000006C1 RG   0A         DS$CKLOOP           00010040  G
CVT_LOCKED            000006FE RG   0A         DS$CLRVEC           00010168  G
CVT_LOWER             00000732 RG   0A         DS$CNTRLC           00010078  G
CVT_NOOP              000005F1 RG   0A         DS$CVTREG           000100B0  G
CVT_RDNHC             00000634 RG   0A         DS$ENDPASS          00010010  G
CVT_READ              0000062A RG   0A         DS$ENDSUB           00010038  G
CVT_READH             00000613 RG   0A         DS$ERRDEV           000100C8  G
CVT_REVERSE           00000738 RG   0A         DS$ERRHARD          000100D0  G
CVT_SEEK              0000060E RG   0A         DS$ERRPREP          00010108  G
CVT_SK                000006F6 RG   0A         DS$ERRSOFT          000100D8  G
CVT_SPINDN            0000071E RG   0A         DS$ERRSYS           000100C0  G
CVT_SPINUP            000006D0 RG   0A         DS$ESCAPE           00010050  G
CVT_STA_TABLE         00000652 RG   08         DS$FREEDBGSYM       000101B8  G
CVT_SUR_TABLE         0000069A RG   08         DS$GETBUF           00010120  G
```

```
DS$GETMEM                 00010130   G            DS$PRINTF                   000100F0  G
DS$GETTERM                000101C8   G            DS$PRINTS                   000100F8  G
DS$GPHARD                 00010018   G            DS$PRINTSIG                 00010100  G
DS$HELP                   000101B0   G            DS$PRINTX                   000100E8  G
DS$INITSCB                00010170   G            DS$PROBE                    000101A0  G
DS$INLOOP                 00010048   G            DS$RELBUF                   00010128  G
DS$K_BBC                = 0000001D   G            DS$RELMEM                   00010138  G
DS$K_BBCC               = 00000021   G            DS$SERVICE_DISPATCHER       000101D0  G
DS$K_BBCCI              = 00000023   G            DS$SETIPL                   00010178  G
DS$K_BBCS               = 0000001F   G            DS$SETMAP                   00010188  G
DS$K_BBS                = 0000001C   G            DS$SETPRIEXV                00010110  G
DS$K_BBSC               = 00000020   G            DS$SETVEC                   00010160  G
DS$K_BBSS               = 0000001E   G            DS$SHOCHAN                  00010190  G
DS$K_BBSSI              = 00000022   G            DS$SUMMARY                  00010028  G
DS$K_BCC_M              = 0000001B   G            DS$WAITMS                   00010060  G
DS$K_BCS_M              = 0000001A   G            DS$WAITUS                   00010068  G
DS$K_BEQLU_B            = 00000005   G            DSA$AL_APTMAIL              0000FE00
DS$K_BEQLU_M            = 00000011   G            DSA$AT_APTTXT               0000FA00
DS$K_BEQL_B             = 00000004   G            DSA$GL_APTCOM               0000FE04
DS$K_BEQL_M             = 00000010   G            DSA$GL_DEVLEN               0000FE58
DS$K_BERROR             = 00000026   G            DSA$GL_ERRNO                0000FE44
DS$K_BGEQU_B            = 00000007   G            DSA$GL_EVENT                0000FE48
DS$K_BGEQU_M            = 00000013   G            DSA$GL_FLAGS                0000FE00
DS$K_BGEQ_B             = 00000006   G            DSA$GL_MSGTYP               0000FE40
DS$K_BGEQ_M             = 00000012   G            DSA$GL_PASSES               0000FE08
DS$K_BGTRU_B            = 00000009   G            DSA$GL_PASSNO               0000FE54
DS$K_BGTRU_M            = 00000015   G            DSA$GL_SECTNO               0000FE10
DS$K_BGTR_B             = 00000008   G            DSA$GL_SID                  0000FE14
DS$K_BGTR_M             = 00000014   G            DSA$GL_SUBTNO               0000FE4C
DS$K_BLBC               = 00000025   G            DSA$GL_TESTNO               0000FE50
DS$K_BLBS               = 00000024   G            DSA$GL_UNITS                0000FE0C
DS$K_BLEQU_B            = 00000003   G            DSA$GQ_MSGPTR               0000FE68
DS$K_BLEQU_M            = 0000000F   G            DSA$GT_DEVNAM               0000FE5C
DS$K_BLEQ_B             = 00000002   G            DSA$V_PASS0               = 0000001D
DS$K_BLEQ_M             = 0000000E   G            DUMP_R80                    00000178  R     0E
DS$K_BLSSU_B            = 00000001   G            DUMP_REG                    00000129  RG    0E
DS$K_BLSSU_M            = 0000000D   G            DUMP_RL02                   0000014C  R     0E
DS$K_BLSS_B             = 00000000   G            E.BA                        00000160  RG    08
DS$K_BLSS_M             = 0000000C   G            E.BC                        00000164  RG    08
DS$K_BNEQU_B            = 0000000B   G            E.BR                        00000188  RG    08
DS$K_BNEQU_M            = 00000017   G            E.CS                        0000015C  RG    08
DS$K_BNEQ_B             = 0000000A   G            E.DA                        00000168  RG    08
DS$K_BNEQ_M             = 00000016   G            E.DAT                       00000184  RG    08
DS$K_BNERROR            = 00000027   G            E.HDR                       0000017C  RG    08
DS$K_BVC_M              = 00000019   G            E.MP                        0000016C  RG    08
DS$K_BVS_M              = 00000018   G            E.MP1                       00000170  RG    08
DS$LOAD                   00010198   G            E.MP2                       00000174  RG    08
DS$LOADPCS                000101C0   G            E.MP3                       00000178  RG    08
DS$MAPDBGBLOCK            00010118   G            E.VEC                       0000018C  RG    08
DS$MEMSIZE                000101D8   G            ENV$M_DOMAIN              = 00000002
DS$MMOFF                  00010158   G            ENV$M_LEVEL               = 00000001
DS$MMON                   00010150   G            ENV$M_SUPER               = 000003FC
DS$MOVPHY                 00010148   G            ENV$S_DOMAIN              = 00000001
DS$MOVVRT                 00010140   G            ENV$S_LEVEL               = 00000001
DS$PARSE                  000100B8   G            ENV$S_SUPER               = 00000008
DS$PRINTB                 000100E0   G            ENV$V_DOMAIN              = 00000001
```

```
ENV$V_LEVEL                = 00000000              FMT_MP1_80                      00000C01 RG     0A
ENV$V_SUPER                = 00000002              FMT_MP2_02                      00000C2C RG     0A
ENV$_CPU                   = 00000000              FMT_MP2_80                      00000C57 RG     0A
ENV$_FUNCTIONAL            = 00000000              FMT_MP3                         00000C83 RG     0A
ENV$_REPAIR                = 00000001              FMT_NYI                         000009CD RG     0A
ENV$_SUPER                 = 00000001              FMT_ONE_SEEK                    00000941 RG     0A
ENV$_SYSTEM                = 00000001              FMT_PAT                         00000BB3 RG     0A
ERR$_MSGADR                = 0000000C              FMT_POS                         00000B90 RG     0A
ERR$_NARGS                 = 0000000A              FMT_SEEK1                       00000802 RG     0A
ERR$_NUM                   = 00000004              FMT_SEEK1_SP                    00000888 RG     0A
ERR$_P1                    = 00000014              FMT_SEEK2                       0000082C RG     0A
ERR$_P2                    = 00000018              FMT_SEEK2_SP                    000008C3 RG     0A
ERR$_P3                    = 0000001C              FMT_SEEK3                       0000085A RG     0A
ERR$_P4                    = 00000020              FMT_SEEK3_SP                    00000902 RG     0A
ERR$_P5                    = 00000024              FMT_SPINDLE                     00000E12 RG     0A
ERR$_P6                    = 00000028              FMT_SPINDLE_SP                  00000E30 RG     0A
ERR$_POINTER               = 00000010              FMT_SP_TIME                     00000E73 RG     0A
ERR$_UNIT                  = 00000008              FMT_VOL                         000009F4 RG     0A
ERR_CNT                      000001B8 RG     08    FMT_WDCNT                       00000D8B RG     0A
ERR_MSG                      000000B8 RG     08    FMT_XOR                         0000077D RG     0A
EVENT_FLAGS                  00000024 RG     08    GDDAT                           00000104 RG     08
EXIT_CHANGE                  0000126B R      0E    GETSTAT                       = 00000004
EXIT_CHECK                   00001420 R      0E    GET_LUN$A                       0000126C RG     0E
EXIT_ECC                     00001217 R      0E    GET_POSITION                    00000EB0 RG     0E
EXIT_GETLUN                  000012B6 R      0E    GET_POSITION_X                  00000F4B R      0E
EXIT_GETSEC                  000012F4 R      0E    GET_RANDOM                      00001159 RG     0E
EXIT_READBAD                 000013E3 R      0E    GET_RANDOM_X                    00001197 R      0E
EXIT_READHDR                 00000FF7 R      0E    GET_SECLUN$A                    000012B7 RG     0E
EXIT_RESTORE                 00000C4B R      0E    GOOD_DATA                       00000190 RG     08
FIX02_DAR                    00001239 R      0E    HARD_TABLE                      00000058 RG     08
FIX80_DAR                    00001253 R      0E    HDR$M_ERR                     = 00000020
FMT_ATTACH                   000007B6 RG     0A    HDR$V_ERR                     = 00000005
FMT_AVG_SEEK                 0000099D RG     0A    HDR02_BITS                      00000590 RG     0A
FMT_BA                       00000B04 RG     0A    HDR1_02_BITS                    00000534 RG     0A
FMT_BAD                      00000767 RG     0A    HDR1_80_BITS                    00000555 RG     0A
FMT_BADBLK                   00000E9C RG     0A    HDR2_80_BITS                    00000574 RG     0A
FMT_BC                       00000B27 RG     0A    HDR80_BITS                      000005C1 RG     0A
FMT_CRLF                     00000748 RG     0A    HDR_02$S_CYL                  = 00000009
FMT_CRLF1                    0000074D RG     0A    HDR_02$S_SEC                  = 00000006
FMT_CS                       00000AE1 RG     0A    HDR_02$S_TRK                  = 00000001
FMT_DA                       00000B4A RG     0A    HDR_02$V_CYL                  = 00000007
FMT_DATA                     00000DD0 RG     0A    HDR_02$V_SEC                  = 00000000
FMT_DRVTYP                   0000078C RG     0A    HDR_02$V_TRK                  = 00000006
FMT_ERRCOUNT                 00000DEB RG     0A    HDR_80$M_FMT                  = 00001000
FMT_EXT_1A                   00000CAD RG     0A    HDR_80$M_MF                   = 00008000
FMT_EXT_1B                   00000CD0 RG     0A    HDR_80$M_SSF                  = 00002000
FMT_EXT_1C                   00000CF0 RG     0A    HDR_80$M_UF                   = 00004000
FMT_EXT_2                    00000D12 RG     0A    HDR_80$S_CYL                  = 0000000A
FMT_GOOD                     00000750 RG     0A    HDR_80$S_FMT                  = 00000001
FMT_HDR                      00000A70 RG     0A    HDR_80$S_SEC                  = 00000005
FMT_HDR_A                    00000D33 RG     0A    HDR_80$S_SSF                  = 00000001
FMT_HEADER1                  00000DA6 RG     0A    HDR_80$S_TRK                  = 00000004
FMT_HMB_ERROR                00000A4F RG     0A    HDR_80$V_CYL                  = 00000000
FMT_MAX_SEEK                 0000096D RG     0A    HDR_80$V_FMT                  = 0000000C
FMT_MP                       00000B6D RG     0A    HDR_80$V_SEC                  = 00000010
FMT_MP1_02                   00000BD6 RG     0A    HDR_80$V_SSF                  = 0000000D
```

HDR_80$V_TRK	= 00000018		
HMB$B_EXTEND	0000002D		
HMB$B_LRU_LIM	0000002E		
HMB$B_WINDOW	0000002C		
HMB$C_LEVEL1	= 00000101		
HMB$L_IBMAPLBN	00000002		
HMB$T_CREDATE	0000003C		
HMB$T_FORMAT	000001F0		
HMB$T_OWNERNAME	000001E4		
HMB$T_VOLNAME	0000000E		
HMB$T_VOLNAME2	000001D8		
HMB$W_CHECKSUM1	0000003A		
HMB$W_CHECKSUM2	000001FE		
HMB$W_CLUSTER	00000008		
HMB$W_DEVTYPE	0000000A		
HMB$W_FILEPROT	00000024		
HMB$W_IBMAPSIZE	00000000		
HMB$W_MAXFILES	00000006		
HMB$W_PROTECT	00000020		
HMB$W_STRUCLEV	0000000C		
HMB$W_VOLCHAR	00000022		
HMB$W_VOLOWNER	0000001E		
HMB_ERR_FLAG	0000007C	RG	08
HOLD_CS	000000A0	RG	08
HOLD_DA	000000F8	RG	08
HP$A_DEPENDENT	00000032	G	
HP$A_DEVICE	00000018	G	
HP$A_DVA	0000001C	G	
HP$A_LINK	00000020	G	
HP$B_DRIVE	0000000B	G	
HP$B_FLAGS	0000000A	G	
HP$B_RB730_BR	00000036		
HP$K_RB730_LEN	00000037		
HP$L_RB730_CSR	00000032		
HP$M_ALLOC	= 00000001	G	
HP$M_NAME	= 00000004	G	
HP$M_WASALL	= 00000002	G	
HP$Q_DEVICE	00000000	G	
HP$T_DEVICE	0000000C	G	
HP$T_TYPE	00000026	G	
HP$V_ALLOC	= 00000000	G	
HP$V_NAME	= 00000002	G	
HP$V_WASALL	= 00000001	G	
HP$W_SIZE	00000008	G	
HP$W_VECTOR	00000024	G	
IDCBAR	= 00000004		
IDCBAR_BITS	00000253	RG	0A
IDCCNT	= 00000008		
IDCCNT_BITS	00000274	RG	0A
IDCCSR	= 00000000		
IDCCSR_BITS	00000145	RG	0A
IDCDAR	= 0000000C		
IDCDSINT_ERR	0000146F	RG	0A
IDCENINT_ERR	0000144D	RG	0A
IDCGETBUF	00001492	RG	0A
IDCMPR	= 00000010		
IDCMPR02_BITS	000003D0	RG	0A
IDCMPR80_BITS	0000046B	RG	0A
IDCPAT	= 00000018		
IDCPAT_BITS	00000512	RG	0A
IDCPOS	= 00000014		
IDCPOS_BITS	000004F1	RG	0A
IDC_ADRFAIL	0000139A	RG	0A
IDC_ISR$A	000012F8	RG	0E
IDC_SADRFAIL	000013C5	RG	0A
IDC_STAT$L_1	0000000C	RG	08
IDC_STAT$W_2	00000010	RG	08
IDC_STAT$W_VR	00000012	RG	08
IDC_STATUS	0000000C	RG	08
INH_PNT	00000108	RG	08
INITIALIZE	00000000	R	0B
INITIALIZE_X	00000244	R	0B
ITERAT	00000110	RG	08
L$A_CCP	00000040	R	03
L$A_DEVP	0000001C	R	03
L$A_DREG	00000024	R	03
L$A_DTP	00000018	R	03
L$A_ICP	0000003C	R	03
L$A_LASTAD	00000014	R	03
L$A_NAME	00000008	R	03
L$A_REPP	00000044	R	03
L$A_SECNAM	00000050	R	03
L$A_STATAB	00000048	R	03
L$A_TSTCNT	00000054	R	03
L$L_ENVIRON	00000004	R	03
L$L_ERRTYP	0000004C	R	03
L$L_HEADLENGTH	00000000	R	03
L$L_REV	0000000C	R	03
L$L_UNIT	00000020	R	03
L$L_UPDATE	00000010	R	03
L$UNIT	0000002C	RG	08
LASTAD	00000000	R	04
LUN0_TBL	000002EC	RG	08
LUN1_TBL	00000370	RG	08
LUN2_TBL	000003F4	RG	08
LUN3_TBL	00000478	RG	08
LUN_TBL	000002DC	RG	08
L_TSTCNT	00000000	R	05
MANUAL	= 00000001	G	
MBSF	00000800	RG	09
MBSF_BUFFER	000002D4	RG	08
MPR02_BITS	00000421	RG	0A
MPR80_BITS	000004A9	RG	0A
MPR_02$M_DSE	= 00000100		
MPR_02$M_HCE	= 00004000		
MPR_02$M_SKTO	= 00001000		
MPR_02$M_SPE	= 00000800		
MPR_02$M_VC	= 00000200		
MPR_02$M_WDE	= 00008000		
MPR_02$M_WGE	= 00000400		
MPR_80$M_FLT	= 00000100		
MPR_80$M_SECT	= 0000001F		

```
MPR_80$M_SKE             = 00000400
MPR_80$V_SKE             = 0000000A
MPR_GS$M_GS              = 00000002
MPR_GS$M_MARK            = 00000001
MPR_GS$M_RST             = 00000008
MS100_DAT_DST              00002C04 RG      0A
MS103_SK_ONE               00002C32 RG      0A
MS104_SK_MAX               00002C5A RG      0A
MS105_SK_AVG               00002C86 RG      0A
MS106_SP_TIM               00002CB2 RG      0A
MS107A_SEEKERR             00002CFE RG      0A
MS107B_RECAL               00002D24 RG      0A
MS107_RECAL                00002CCF RG      0A
MS108_SECTNZ               00002D4A RG      0A
MS109_BADCNT               00002D7B RG      0A
MS110_UEHNF31              00002DAF RG      0A
MS111_EHNF31               00002DE7 RG      0A
MS112_ACC31                00002E21 RG      0A
MS113_UWRTERR              00002E41 RG      0A
MS114A_UWCKERR             00002EA6 RG      0A
MS114_URDERR               00002E75 RG      0A
MS115A_WKRST               00002F1D RG      0A
MS115_RDRST                00002EDE RG      0A
MS116A_AUTOCMP             00002F87 RG      0A
MS116_SKPCMP               00002F59 RG      0A
MS117_AUTORD               00002FF3 RG      0A
MS117_AUTOWCK              00003032 RG      0A
MS117_AUTOWRT              00002FBA RG      0A
MS118_SEEKIDC              0000306A RG      0A
MS119_IDCPOS               000030A5 RG      0A
MS44A_INC_CRC              00001FDF RG      0A
MS45A_HD_PROX              0000202D RG      0A
MS83A_ST4_DE               00002866 RG      0A
MS94A_NO_OTH               00002ACF RG      0A
MSG10_RG_INIT              00001797 RG      0A
MSG11_GS_CRDY              000017C2 RG      0A
MSG12_GS_DRDY              000017F6 RG      0A
MSG13_INI_ERR              00001810 RG      0A
MSG14_SYSINIT              0000183E RG      0A
MSG15_NOOP                 00001869 RG      0A
MSG16A_WRTCHK              000018BA RG      0A
MSG16B_WRTCHK              000018E3 RG      0A
MSG16_WRTCHK               0000188E RG      0A
MSG17_GETSTAT              00001918 RG      0A
MSG18B_SEEK                00001968 RG      0A
MSG18_SEEK                 00001943 RG      0A
MSG19A_READH               000019B6 RG      0A
MSG19_READH                0000198A RG      0A
MSG1_ADR_ERR               00001613 RG      0A
MSG20A_FORMAT              00001A7E RG      0A
MSG20A_WRITE               00001A4B RG      0A
MSG20B_WRITE               00001AA2 RG      0A
MSG20C_WRITE               00001ADA RG      0A
MSG20_WRITE                000019F2 RG      0A
MSG20_WRT_BND              00001A1D RG      0A
MSG21A_READ                00001B36 RG      0A
MSG21A_READH               00001B68 RG      0A
MSG21B_READ                00001B95 RG      0A
MSG21_READ                 00001B0C RG      0A
MSG22_READNHC              00001BCC RG      0A
MSG23_CS_INC               00001C04 RG      0A
MSG24A_BC_MOD              00001C47 RG      0A
MSG24_BA_MOD               00001C2A RG      0A
MSG25_DA_MOD               00001C64 RG      0A
MSG26_MP_MOD               00001C81 RG      0A
MSG27_MP_INC               00001C9E RG      0A
MSG28_DA_INC               00001CBF RG      0A
MSG29_BA_INC               00001CDF RG      0A
MSG2_RG_ZERO               00001635 RG      0A
MSG30A_CRDY                00001D1D RG      0A
MSG30_CRDY                 00001CFE RG      0A
MSG31_DRDY                 00001D54 RG      0A
MSG32_CERR                 00001D6E RG      0A
MSG33A_UXEATTN             00001D9D RG      0A
MSG33B_NOATTN              00001DB8 RG      0A
MSG33_DERR                 00001D88 RG      0A
MSG34_GS_OPI               00001DD6 RG      0A
MSG35_OPI_TO               00001E10 RG      0A
MSG36_OPI_RLY              00001E37 RG      0A
MSG37_OPI_LAT              00001E63 RG      0A
MSG38_NOOP_IE              00001E8D RG      0A
MSG39A_INT                 00001EDB RG      0A
MSG39_NO_INT               00001EB7 RG      0A
MSG3_RG_ONE                0000164F RG      0A
MSG40_BAD_BR               00001F0D RG      0A
MSG41_BAD_VEC              00001F36 RG      0A
MSG42_RPT_CYL              00001F5D RG      0A
MSG43_MP2_NZ               00001F95 RG      0A
MSG44_BAD_CRC              00001FB6 RG      0A
MSG45_SK_ZERO              0000200D RG      0A
MSG46_POS_ERR              00002068 RG      0A
MSG47_DS_INC               00002082 RG      0A
MSG48_SK_ONE               00002099 RG      0A
MSG49_DRDY_NC              000020B3 RG      0A
MSG4_RG_FLT1               00001669 RG      0A
MSG50_SK_OUTR              000020CD RG      0A
MSG51_SK_INNR              000020F1 RG      0A
MSG52_RD_HNF               00002115 RG      0A
MSG53_WRT_NXM              00002142 RG      0A
MSG54_DA_UPD               00002173 RG      0A
MSG55_DAT_BND              000021DB RG      0A
MSG55_DAT_CP1              000021A6 RG      0A
MSG55_HDRCMP               0000220F RG      0A
MSG56_DAT_UNQ              00002240 RG      0A
MSG57_WCK_FRC              00002287 RG      0A
MSG58_PAR_WRT              000022B1 RG      0A
MSG59_PAR_RD               000022E4 RG      0A
MSG5_RG_FLT0               0000168C RG      0A
MSG60_WRT_PRD              00002316 RG      0A
MSG61_DAT_PAR              0000234E RG      0A
MSG62_DC_PAR               0000238E RG      0A
MSG63_PRD_TMW              000023C8 RG      0A
```

```
MSG64_WCK_PAR                   000023F5 RG     0A      MSG_R80                                000014FE RG     0A
MSG65_DAT_CP2                   00002428 RG     0A      MSG_RL02                               000014F9 RG     0A
MSG66_HNF_LT                    00002462 RG     0A      MSG_TRACE                              000014D9 RG     0A
MSG67_DAT_ES                    00002497 RG     0A      NEW_CYLN                               000000EC RG     08
MSG68_DAT_MAX                   000024DF RG     0A      NOCHK$M_DERR                         = 00000008
MSG69_DAT_NHC                   00002522 RG     0A      NOCHK$M_FTL                          = 00000040
MSG6_CS_FAIL                    000016AF RG     0A      NOCHK$V_ATTN                         = 00000007
MSG70A_NFSEC31                  00002583 RG     0A      NOCHK$V_CERR                         = 00000001
MSG70_FND_S39                   0000255F RG     0A      NOCHK$V_CRC                          = 00000005
MSG71_RD_BKT                    0000259F RG     0A      NOCHK$V_CRDY                         = 00000000
MSG72_DAT_BKT                   000025E3 RG     0A      NOCHK$V_DERR                         = 00000003
MSG73_GS_UND                    00002621 RG     0A      NOCHK$V_DRDY                         = 00000002
MSG74_DRDY_NC                   0000264D RG     0A      NOCHK$V_DSTA                         = 00000004
MSG75_NO_ST1                    00002678 RG     0A      NOCHK$V_FTL                          = 00000006
MSG76_ST1_BS                    000026A5 RG     0A      NON_FATAL                              00000CAE R      0E
MSG77_NO_ST2                    000026D1 RG     0A      NO_OP                                = 00000000
MSG77_NO_ST3                    000026FE RG     0A      NUMENT                               = 00000021
MSG78_ST2_BS                    0000272B RG     0A      OFST_AVGTIM                          = 00000080
MSG79_BH_NC                     0000275B RG     0A      OFST_CSRTYP                          = 00000058
MSG7A_BC_FAIL                   000016F9 RG     0A      OFST_DEVMEM                          = 0000001C
MSG7_BA_FAIL                    000016D4 RG     0A      OFST_DEVNAM                          = 00000020
MSG80_BH_NF                     00002795 RG     0A      OFST_DRVERR                          = 0000004C
MSG81_ST3_BS                    000027D0 RG     0A      OFST_DRVRST                          = 00000050
MSG82_NO_ST4                    000027FF RG     0A      OFST_DRVTYP                          = 00000000
MSG83_ST4_BS                    0000282C RG     0A      OFST_FMTLEN                          = 00000030
MSG84_NO_ST5                    0000289F RG     0A      OFST_GTSCMD                          = 00000034
MSG85_ST5_BS                    000028CC RG     0A      OFST_HDBITS                          = 00000044
MSG86_DRDY_NS                   00002900 RG     0A      OFST_HDRDNC                          = 00000054
MSG87_VC_NC                     00002938 RG     0A      OFST_HDRTMP                          = 00000048
MSG88_NO_ST6                    00002971 RG     0A      OFST_HIGRPM                          = 00000074
MSG89_ST6_BS                    0000299E RG     0A      OFST_HMBSEC                          = 0000005C
MSG8A_MP_FAIL                   00001743 RG     0A      OFST_IGBCYL                          = 00000018
MSG8_DA_FAIL                    0000171E RG     0A      OFST_LOWRPM                          = 00000070
MSG90_NO_ST7                    000029D2 RG     0A      OFST_LSTCYL                          = 00000008
MSG91_ST7_BS                    000029FF RG     0A      OFST_LSTSEC                          = 00000010
MSG92_NO_ST0                    00002A2D RG     0A      OFST_LSTTRK                          = 0000000C
MSG93_RELV_UP                   00002A5A RG     0A      OFST_MAXTIM                          = 0000007C
MSG94_UXE_DSE                   00002A93 RG     0A      OFST_MPBITS                          = 00000040
MSG95_DSE_NS                    00002AFA RG     0A      OFST_NMLSTS                          = 00000038
MSG96_DSE_NC                    00002B33 RG     0A      OFST_ONETIM                          = 00000078
MSG97_GS_NXD                    00002B67 RG     0A      OFST_OP1STS                          = 00000060
MSG98_WRT_WL                    00002B99 RG     0A      OFST_PHYU                            = 00000004
MSG99_RST_WGE                   00002BC7 RG     0A      OFST_RANCYL                          = 0000006C
MSG9_RG_RMW                     00001768 RG     0A      OFST_SECLEN                          = 00000028
MSG_ABORT                       0000153A RG     0A      OFST_SEKSTS                          = 0000003C
MSG_AVG                         0000150E RG     0A      OFST_STSDNC                          = 00000064
MSG_BLANK                       00001531 RG     0A      OFST_TRKLEN                          = 0000002C
MSG_DRVFATAL                    0000158A RG     0A      OFST_TSTCYL                          = 00000014
MSG_ECCFAIL                     000015DB RG     0A      OFST_WTPSTS                          = 00000068
MSG_ECCPASS                     000015B6 RG     0A      OLD_CYLN                               000000E8 RG     08
MSG_INNER                       00001528 RG     0A      OPR10_INS_ID                           0000114A RG     0A
MSG_MAX                         00001506 RG     0A      OPR11_ORIGIN                           000011BE RG     0A
MSG_MIDDLE                      0000151F RG     0A      OPR12_WRTLOCK                          00001225 RG     0A
MSG_ONE                         00001502 RG     0A      OPR134_RST_FLT                         00001269 RG     0A
MSG_OPERATTCH                   0000155B RG     0A      OPR13_RST_WL                           00001246 RG     0A
MSG_OUTER                       00001516 RG     0A      OPR14_ROTATE                           00001289 RG     0A
```

```
OPR15_SWAP                         000012E4 RG      0A      PRINT_REG_HDR                   000001D1 RG      0E
OPR16_SWAP_BK                      0000130B RG      0A      PRT$M_ALL                     = 00000007
OPR1_UNLOAD                        00000EE7 RG      0A      PRT$M_EXT_1                   = 00000008
OPR2_CLOSE                         00000F75 RG      0A      PRT$M_EXT_2                   = 00000010
OPR3_START                         00000FB2 RG      0A      PRT$V_EXT_1                   = 00000003
OPR4_MKRDY                         00000FDF RG      0A      PRT$V_EXT_2                   = 00000004
OPR5_UNLOAD                        00001013 RG      0A      PRT$V_MPR_1                   = 00000000
OPR6_REMOVE                        00001042 RG      0A      PRT$V_MPR_2                   = 00000001
OPR7_INSERT                        0000106F RG      0A      PRT$V_MPR_3                   = 00000002
OPR8_SETUP                         00001090 RG      0A      PRT$V_NONE                    = 00000006
OPR9_DUPPLUG                       000010C0 RG      0A      R80                           = 00000002
OPR_AVAIL                          00001137 RG      0A      R80$T                         = 00000002
OPR_FAULT                          00001363 RG      0A      R80_HDR_TMP                     00000613 RG      08
OPR_RETURN                         00000ED1 RG      0A      R80_TBL                         00000580 RG      08
P$ATTN                             0000003C RG      08      RB730$B_BR                      00000036
P$UNIT                             00000034 RG      08      RB730$K_LEN                     00000037
PAR$M_ATDEF                      = 00000010                 RB730$L_CSR                     00000032
PAR$M_ATHI                       = 00000008                 RD_BAD_BLK                      00001359 RG      0E
PAR$M_ATLO                       = 00000004                 RD_HOME_BLK                     00001107 RG      0E
PAR$M_NODEF                      = 00000002                 RD_HOME_BLK_X                   00001158 R       0E
PAR$V_ATDEF                      = 00000004                 RD_IDCREGS                      000008EF RG      0E
PAR$V_ATHI                       = 00000003                 RD_IDCREGS_X                    00000963 R       0E
PAR$V_ATLO                       = 00000002                 READ                          = 0000000C
PAR$V_NODEF                      = 00000001                 READH                         = 00000008
PAR$_BIN                         = 00000002                 READ_BUFFER                     000002C4 RG      08
PAR$_DEC                         = 0000000A                 READ_HEADERS                    00000FA5 RG      0E
PAR$_HEX                         = 00000010                 RECAL                         = 00000006
PAR$_NO                          = 00000000                 RECAL$M_INPROG                = 00000001
PAR$_OCT                         = 00000008                 RECAL$V_INPROG                = 00000000
PAR$_YES                         = 00000001                 REG_ERRPT                       00000000 RG      0E
PAT$S_BITS                       = 0000000B                 RESTORE_DRIVE                   00000B60 RG      0E
PHY_RD_BUF                         0000006C RG      08      RESTORE_R80                     00000BA4 R       0E
PHY_WRT_BUF                        0000005C RG      08      RESTORE_RL02                    00000B6F R       0E
PNT_FLAG                           0000010C RG      08      RGLIST                          000000A8 RG      08
POSITION_HDS                       00000DBE RG      0E      RL02                          = 00000001
POSITION_HDS_X                     00000EAF R       0E      RL02$T                        = 00000001
POS_TYPE                           000000CC RG      08      RL02_HDR_TMP                    00000604 RG      08
PREV_CYLN                          000000E0 RG      08      RL02_TBL                        000004FC RG      08
PRINT_BA                           0000022C RG      0E      SECD_ADDR                       00000018 RG      08
PRINT_BC                           00000276 RG      0E      SECTION                         00000013 R       0A
PRINT_CS                           000001DF RG      0E      SECTOR                          000000C4 RG      08
PRINT_DATA_ERR                     000006B8 RG      0E      SEED                            000000F4 RG      08
PRINT_DA_02                        000002C0 RG      0E      SEEK                          = 00000006
PRINT_DA_80                        0000034E RG      0E      SEEK_TYPE                       000000B0 RG      08
PRINT_EXT_1                        0000065A RG      0E      SEP_FUNCTIONAL                = 00000002
PRINT_EXT_2                        000006A6 RG      0E      SEP_REPAIR                    = 00000003
PRINT_MP1_02                       00000502 RG      0E      SIZ...                        = 00000001
PRINT_MP1_80                       0000054F RG      0E      SKIP_BUFFER                     000002CC RG      08
PRINT_MP2_02                       0000059A RG      0E      START_CMD                       00000966 RG      0E
PRINT_MP2_80                       000005DA RG      0E      STATUS_CHK                      00000A22 RG      0E
PRINT_MP3                          0000061A RG      0E      STATUS_CHK_X                    00000B54 R       0E
PRINT_MP_02                        000003D6 RG      0E      STEP_HDS                        00000CB1 RG      0E
PRINT_MP_80                        00000424 RG      0E      STEP_HDS_X                      00000DBD R       0E
PRINT_PAT                          000004B8 RG      0E      ST_GETSTAT                      000009AF R       0E
PRINT_POS                          0000046E RG      0E      ST_NOOP                         000009AC R       0E
PRINT_REG                          000000D7 RG      0E      ST_READ                         000009F2 R       0E
```

```
ST_READHDR                 000009C6 R      0E
ST_READ_NOHDR              000009F2 R      0E
ST_SEEK                    000009B2 R      0E
ST_WRITE                   000009CD R      0E
ST_WRTCHK                  000009AC R      0E
SUBSYS_INIT                00000823 RG     0E
SUBSYS_INIT_X              000008EE R      0E
SUMMARY                    00000000 R      0D
SUMMARY_X                  00000003 R      0D
SURFACE                    000000C8 RG     08
SYS$ALLOC                  00010238  G
SYS$ASCTIM                 00010248  G
SYS$ASSIGN                 00010250  G
SYS$BINTIM                 00010258  G
SYS$CANCEL                 00010260  G
SYS$CANTIM                 00010268  G
SYS$CLOSE                  000105B8  G
SYS$CLREF                  00010298  G
SYS$CONNECT                000105C0  G
SYS$DALLOC                 000102D8  G
SYS$DASSGN                 000102E0  G
SYS$DISCONNECT             000105D0  G
SYS$FAO                    00010350  G
SYS$FAOL                   00010358  G
SYS$GET                    00010580  G
SYS$GETCHN                 000104C8  G
SYS$GETTIM                 00010378  G
SYS$LKWSET                 000103A0  G
SYS$NUMTIM                 000103B8  G
SYS$OPEN                   00010608  G
SYS$QIO                    000103C8  G
SYS$QIOW                   00010200  G
SYS$READ                   00010590  G
SYS$READEF                 000103D0  G
SYS$SETAST                 000103F8  G
SYS$SETEF                  00010400  G
SYS$SETIMR                 00010420  G
SYS$SETPRI                 00010428  G
SYS$SETPRT                 00010430  G
SYS$SETRWM                 00010438  G
SYS$SETSWM                 00010448  G
SYS$ULKPAG                 00010460  G
SYS$ULWSET                 00010468  G
SYS$UNWIND                 00010470  G
SYS$WAITFR                 00010478  G
SYS$WFLAND                 00010488  G
SYS$WFLOR                  00010490  G
SYS_TIME                   0000001C RG     08
S_ALL                      0000000F R      0A
S_DEFAULT                  00000000 R      0A
S_DRIVE_TBL                00000004 RG     08
S_L$UNIT                   00000030 RG     08
S_MANUAL                   00000008 R      0A
S_P$ATTN                   00000040 RG     08
S_P$UNIT                   00000038 RG     08
T.BA                       00000124 RG     08
T.BC                       00000128 RG     08
T.BR                       00000154 RG     08
T.CS                       00000120 RG     08
T.DA                       0000012C RG     08
T.DAT                      00000150 RG     08
T.HDR                      00000140 RG     08
T.MP                       00000130 RG     08
T.MP1                      00000134 RG     08
T.MP2                      00000138 RG     08
T.MP3                      0000013C RG     08
T.PAT                      00000148 RG     08
T.POS                      0000014C RG     08
T.VEC                      00000158 RG     08
TEMP                       00000118 RG     08
TIME_300MS                 00000094 RG     08
TIME_800MS                 0000008C RG     08
TIME_LEFT                  0000009C RG     08
TIME_OUT                   00000080 RG     08
T_DEVICE                   00000023 RG     0A
T_NAME                     00000058 R      03
T_R80                      00000028 R      0A
T_RL02                     00000023 R      0A
UBACLEAR_ERR               00001428 RG     0A
UBAINIT_ERR                00001438 RG     0A
UBAMAP_ERR                 00001406 RG     0A
UBAPURGE_ERR               00001418 RG     0A
UNBINIT_ERR                000013F3 RG     0A
USE02_COMMON               000002EC R      0E
USE02_DMA                  0000030B R      0E
USE02_GS                   000002F3 R      0E
USE02_SEEK                 000002FA R      0E
USE80_COMMON               0000037A R      0E
USE80_DMA                  00000398 R      0E
USE80_SEEK                 00000381 R      0E
VIR_RD_BUF                 00000074 RG     08
VIR_WRT_BUF                00000064 RG     08
VOL_NAME                   000002B4 RG     08
WORD_COUNT                 000001BC RG     08
WORK                       00000114 RG     08
WRITE_BUFFER               000002BC RG     08
```

```
                                        +----------------+
                                        ! Psect synopsis !
                                        +----------------+

PSECT name                      Allocation          PSECT No.  Attributes
----------                      ----------          ---------  ----------
.  ABS  .                       00000000  (     0.)  00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
HEADER                          00000000  (     0.)  01 (  1.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC PAGE
$ABS$                           00010610  (67088.)  02 (  2.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
$HEADER                         0000007F  (   127.)  03 (  3.)  NOPIC   USR   CON   REL   LCL NOSHR NOEXE   RD   NOWRT NOVEC PAGE
_LAST                           00000000  (     0.)  04 (  4.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC PAGE
$TSTCNT                         00000000  (     0.)  05 (  5.)  NOPIC   USR   OVR   REL   LCL NOSHR NOEXE   RD   NOWRT NOVEC LONG
DISPATCH                        00000000  (     0.)  06 (  6.)  NOPIC   USR   CON   REL   LCL NOSHR NOEXE   RD   NOWRT NOVEC LONG
DISPATCH_X                      00000018  (    24.)  07 (  7.)  NOPIC   USR   CON   REL   LCL NOSHR NOEXE   RD   NOWRT NOVEC LONG
GLOBAL_DATA                     000006B2  (  1714.)  08 (  8.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC LONG
BUFFERS                         00000A00  (  2560.)  09 (  9.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC PAGE
GLOBAL_TEXT                     000030D8  (12504.)  0A ( 10.)  NOPIC   USR   CON   REL   LCL NOSHR NOEXE   RD   NOWRT NOVEC LONG
INITIALIZE                      0000024C  (   588.)  0B ( 11.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC LONG
CLEANUP                         00000079  (   121.)  0C ( 12.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC LONG
SUMMARY                         0000000B  (    11.)  0D ( 13.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC LONG
SUBROUTINES                     00001421  (  5153.)  0E ( 14.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC LONG
```

```
                                 +----------------------------+
                                 ! Symbol Cross Reference !
                                 +----------------------------+

SYMBOL                  VALUE       DEFINITION      REFERENCES...
------                  -----       ----------      -------------
$$ARGS                 =0000000A    76     (2)       76     (2)
$$E                    =00000001    1563   (16)      1524   (15)     1586   (16)
$$M                    =00000004    3829   (44)      1440   (15)     1454   (15)     2821   (30)     3364   (37)
                                                     3370   (37)     3375   (37)     3573   (40)     3829   (44)
$$N                    =00000000    3829   (44)      1440   (15)     1454   (15)     1458   (15)     1464   (15)
                                                     1482   (15)     1505   (15)   #-1644   (19)   #-1645   (19)
                                                   #-1654   (19)     1656   (19)     1658   (19)     1660   (19)
                                                   #-1661   (19)   #-1664   (19)   #-1713   (20)   #-1714   (20)
                                                   #-1716   (20)     1717   (20)   #-1718   (20)   #-1720   (20)
                                                   #-1798   (21)   #-1804   (21)     1813   (21)     1822   (21)
                                                     1831   (21)     1869   (21)     1905   (21)     1916   (21)
                                                     1925   (21)     1934   (21)     1943   (21)   #-1951   (21)
                                                     1952   (21)   #-1959   (21)     1960   (21)     1967   (21)
                                                     1974   (21)     1981   (21)   #-1990   (21)     1991   (21)
                                                     1992   (21)     1993   (21)     1998   (21)     2052   (22)
                                                     2053   (22)   #-2054   (22)     2067   (22)     2071   (22)
                                                     2821   (30)     3364   (37)     3370   (37)     3375   (37)
                                                     3573   (40)   #-3575   (40)     3829   (44)     797    (12)
                                                     805    (12)
$$S                    =FFFFFFFF    66     (2)
$$T1                   =00000000    2581   (28)      1580   (16)     1582   (16)     2273   (25)     2286   (25)
                                                     2573   (28)     2581   (28)     76     (2)
$ENV                   =00000003    66     (2)       97     (3)
$ER                    =00000007    3829   (44)      1407   (15)   #-1420   (15)   #-1437   (15)   #-1440   (15)
                                                   #-1454   (15)     1563   (16)   #-2821   (30)   #-3363   (37)
                                                   #-3364   (37)   #-3369   (37)   #-3370   (37)   #-3374   (37)
                                                   #-3375   (37)   #-3573   (40)   #-3829   (44)
$GS_SETMAP              00001027-R  3358   (37)      3434   (38)     3896   (45)
$GS_SETMAP_X            00001106-R  3378   (37)    #-3374   (37)
$MO                    =00000001    3979   (46)      3979   (46)
$ST                    =00000000    1586   (16)      1440   (15)     1454   (15)     1524   (15)     1586   (16)
                                                     2821   (30)     2887   (31)     2932   (31)     3001   (32)
                                                     3364   (37)     3365   (37)     3370   (37)     3371   (37)
                                                     3375   (37)     3376   (37)     3573   (40)     3829   (44)
$TN                    =00000000    66     (2)       1407   (15)     1563   (16)     1593   (18)     3979   (46)
ABORT_FLAG              00000054-R  446    (8)     #-1415   (15)   #-1500   (15)     1512   (15)   #-1514   (15)
                                                     1518   (15)     3708   (42)
ACT_NAME                0000004C-R  445    (8)     #-1491   (15)   #-1492   (15)   #-1505   (15)
ALL                    =00000002    797    (12)
AL_DEVTYP               0000002D-R  805    (12)      97     (3)
A_HEADEND               00000058-R  97     (3)       97     (3)
BAD_DATA                000001A4-R  518    (8)     #-2056   (22)   #-2125   (23)   #-2134   (23)
BASE_ADDR               00000014-R  433    (8)     #-1432   (15)     1436   (15)   #-1443   (15)   #-1444   (15)
                                                   #-1564   (16)   #-1809   (21)   #-1818   (21)   #-1827   (21)
                                                   #-1837   (21)   #-1876   (21)   #-1911   (21)   #-1921   (21)
                                                   #-1930   (21)   #-1939   (21)   #-2197   (24)   #-2264   (25)
                                                   #-2441   (27)   #-3821   (44)
BDDAT                   00000106-R  484    (8)
BIT...                 =00000028    3374   (37)      1420   (15)     1437   (15)     3363   (37)     3369   (37)
```

```
                                                        3374   (37)    66     (2)     69     (2)     73     (2)
                                                        74     (2)     81     (2)     82     (2)
BIT0                        =00000001    71     (2)
BIT1                        =00000002    71     (2)
BIT10                       =00000400    71     (2)
BIT11                       =00000800    71     (2)
BIT12                       =00001000    71     (2)
BIT13                       =00002000    71     (2)
BIT14                       =00004000    71     (2)     694    (9)
BIT15                       =00008000    71     (2)     694    (9)
BIT16                       =00010000    71     (2)
BIT17                       =00020000    71     (2)
BIT18                       =00040000    71     (2)
BIT19                       =00080000    71     (2)
BIT2                        =00000004    71     (2)
BIT20                       =00100000    71     (2)
BIT21                       =00200000    71     (2)
BIT22                       =00400000    71     (2)
BIT23                       =00800000    71     (2)
BIT24                       =01000000    71     (2)
BIT25                       =02000000    71     (2)
BIT26                       =04000000    71     (2)
BIT27                       =08000000    71     (2)
BIT28                       =10000000    71     (2)
BIT29                       =20000000    71     (2)
BIT3                        =00000008    71     (2)
BIT30                       =40000000    71     (2)
BIT31                       =80000000    71     (2)
BIT4                        =00000010    71     (2)
BIT5                        =00000020    71     (2)     694    (9)
BIT6                        =00000040    71     (2)     694    (9)
BIT7                        =00000080    71     (2)     694    (9)
BIT8                        =00000100    71     (2)    #-2214  (24)
BIT9                        =00000200    71     (2)    #-2214  (24)
BUFFER1                      00000000-R  602    (8)     588    (8)     589    (8)     594    (8)     595    (8)
BUFFER2                      00000400-R  603    (8)     591    (8)     592    (8)
BUFF_INUSE                   000000A4-R  459    (8)    #-2063  (22)   #-2121  (23)
BUFLEN                      =00000064                  #-1655  (19)   #-1657  (19)   #-1812  (21)   #-1821  (21)
                                                       #-1830  (21)   #-1868  (21)   #-1904  (21)   #-1915  (21)
                                                       #-1924  (21)   #-1933  (21)   #-1942  (21)   #-1950  (21)
                                                       #-1958  (21)   #-1966  (21)   #-1973  (21)   #-1980  (21)
                                                        526    (8)
BUILD_HDRS                   00000F4C-R  3175   (34)
CALC_DIFF                    00000FF8-R  3303   (36)    3010   (32)
CEP_FUNCTIONAL              =00000000    66     (2)
CEP_REPAIR                  =00000001    66     (2)
CHANGE_DAR$A                 00001218-R  3629   (41)
CHAR_BUFFER                  000001C0-R  526    (8)     1655   (19)   #-1656  (19)    1657  (19)   #-1658  (19)
                                                        1812   (21)   #-1813  (21)    1821  (21)   #-1822  (21)
                                                        1830   (21)   #-1831  (21)    1868  (21)   #-1869  (21)
                                                        1904   (21)   #-1905  (21)    1915  (21)   #-1916  (21)
                                                        1924   (21)   #-1925  (21)    1933  (21)   #-1934  (21)
                                                        1942   (21)   #-1943  (21)    1950  (21)   #-1952  (21)
                                                        1958   (21)   #-1960  (21)    1966  (21)   #-1967  (21)
                                                        1973   (21)   #-1974  (21)    1980  (21)   #-1981  (21)
CHC$_ABORT                  =00000002    73     (2)
```

```
CHC$_CLEAR                       =00000006     73      (2)     #-3362    (37)
CHC$_CLRDFT                      =00000009     73      (2)
CHC$_DSINT                       =00000005     73      (2)     #-1569    (16)
CHC$_ENINT                       =00000004     73      (2)     #-1521    (15)
CHC$_INITA                       =00000000     73      (2)
CHC$_INITB                       =00000001     73      (2)
CHC$_PURGE                       =00000003     73      (2)     #-3373    (37)
CHC$_SELF_TEST                   =0000000B     73      (2)
CHC$_SETDFT                      =00000008     73      (2)
CHC$_STATUS                      =00000007     73      (2)
CHC$_STOP                        =0000000A     73      (2)
CHECK_BADBLK                      000013E4-R   3962    (46)
CHECK_ECC                         0000119A-R   3548    (40)
CHECK_FATAL                       00000C4C-R   2791    (30)      2694    (29)      2721    (29)
CHI$M_CHNINT                     =00000001     74      (2)
CHI$M_DEVINT                     =00000002     74      (2)
CHI$M_IPL                        =0000007C     74      (2)
CHI$S_IPL                        =00000005     74      (2)
CHI$V_CHNINT                     =00000000     74      (2)
CHI$V_DEVINT                     =00000001     74      (2)
CHI$V_IPL                        =00000002     74      (2)
CHKFTL_R80                        00000C87-R   2815    (30)      2799    (30)
CHKFTL_RL02                       00000C61-R   2801    (30)      2799    (30)
CHM$_FORWARD                     =00000000     75      (2)       75      (2)
CHM$_INVALIDATE                  =00000001     75      (2)       75      (2)
CHM$_MAP                         =00000002     75      (2)       75      (2)
CHM$_MFWDN                       =00000002     75      (2)
CHM$_MFWDNO                      =0000000A     75      (2)
CHM$_MFWDV                       =00000003     75      (2)     #-3368    (37)
CHM$_MFWDVO                      =0000000B     75      (2)
CHM$_MREVN                       =00000006     75      (2)
CHM$_MREVNO                      =0000000E     75      (2)
CHM$_MREVV                       =00000007     75      (2)
CHM$_MREVVO                      =0000000F     75      (2)
CHM$_NFWDN                       =00000000     75      (2)
CHM$_NREVN                       =00000004     75      (2)
CHM$_OFFSET                      =00000008     75      (2)       75      (2)
CHM$_REVERSE                     =00000004     75      (2)       75      (2)
CLEAN_UP                          00000000-R   1563    (16)      97      (3)
CLEAN_UP_X                        00000071-R   1586    (16)
CMPDATA                           00000746-R   2117    (23)      3571    (40)
CMPDATAX                          000007A4-R   2146    (23)    #-2143    (23)
COM_BIT16                         000000EF-R   815     (13)      1966    (21)      1980    (21)
COM_BIT32                         00000039-R   810     (13)      1651    (19)      1854    (21)      1893    (21)
CONFIG                            0000011C-R   490     (8)     #-1408    (15)    #-1473    (15)    #-1476    (15)    #-1575    (16)
                                                               #-1577    (16)      2288    (25)      2300    (25)      2563    (28)
                                                                 2572    (28)      2583    (28)      2598    (28)      2602    (28)
                                                                 2620    (28)      2625    (28)    #-2630    (28)      2794    (30)
                                                               #-2883    (31)      2889    (31)    #-2916    (31)    #-2997    (32)
                                                                 3003    (32)    #-3032    (32)      3117    (33)
CRC_TABLE                         00000228-R   544     (8)       3120    (33)
CSR$M_ASSI                       =08000000                     #-3242    (35)    #-3253    (35)
CSR$M_CRDY                       =00000080                     #-1443    (15)    #-1566    (16)    #-2200    (24)    #-2203    (24)
                                                               #-2205    (24)    #-2206    (24)    #-2210    (24)    #-2212    (24)
                                                               #-2593    (28)    #-2741    (29)
CSR$M_DCK                        =00000800                     #-2298    (25)    #-2623    (28)
```

```
CSR$M_DE                         =00004000                         #-2297     (25)
CSR$M_DRDY                       =00000001                         #-2574     (28)
CSR$M_ERR                        =00008000                         #-2297     (25)
CSR$M_HNF                        =00001000                         #-2298     (25)     #-2623     (28)
CSR$M_IE                         =00000040                         #-2449     (27)     #-3826     (44)
CSR$M_IR                         =01000000                         #-1566     (16)     #-2199     (24)     #-3832     (44)
CSR$M_NXM                        =00002000                         #-2297     (25)     #-2622     (28)
CSR$M_OPI                        =00000400                         #-2298     (25)     #-2623     (28)
CSR$M_R80                        =04000000                         #-1444     (15)       690      (9)
CSR$M_SSEI                       =00400000                         #-3242     (35)     #-3253     (35)
CSR$V_CRDY                       =00000007                         #-2280     (25)     #-2356     (26)     #-2566     (28)
CSR$V_DE                         =0000000E                         #-2599     (28)
CSR$V_DRDY                       =00000000                         #-2293     (25)
CSR$V_ECS0                       =00000014                         #-3557     (40)
CSR$V_ERR                        =0000000F                         #-2626     (28)
CSR$V_OPI                        =0000000A                         #-3554     (40)
CSR$V_R80                        =0000001A                         #-2202     (24)
CSR_BITS                          000001CA-R      823     (13)
CURR_CMD                          000000FC-R      481     (8)      #-1841     (21)     #-1880     (21)     #-2275     (25)     #-2357     (26)
                                                                   #-2450     (27)     #-2469     (27)     #-2493     (27)     #-2586     (28)
                                                                   #-2603     (28)     #-2701     (29)     #-2716     (29)
CURR_CYLN                         000000D4-R      471     (8)      #-1998     (21)     #-2470     (27)     #-2886     (31)     #-2892     (31)
                                                                   #-2895     (31)     #-2919     (31)     #-2929     (31)     #-2935     (31)
                                                                   #-3000     (32)     #-3005     (32)     #-3009     (32)     #-3034     (32)
                                                                   #-3111     (33)     #-3114     (33)     #-3305     (36)
CURR_CYL_HD                       000000DC-R      473     (8)      #-3114     (33)
CURR_FLAGS                        00000028-R      437     (8)        2992     (32)       3027     (32)
CURR_HD                           000000D8-R      472     (8)      #-3036     (32)     #-3112     (33)     #-3114     (33)
CURR_SECTOR                       000000F0-R      478     (8)      #-3113     (33)     #-3246     (35)
CVT_BRUSH                         000006DC-R      926     (13)        746     (10)
CVT_DIR_TABLE                     000006A6-R      769     (10)       1862     (21)
CVT_DRVTYP                        00000622-R      726     (10)       1812     (21)
CVT_ERROR0                        00000646-R      900     (13)        755     (10)
CVT_ERROR1                        0000064B-R      902     (13)        756     (10)
CVT_ERROR2                        00000660-R      904     (13)        757     (10)
CVT_ERROR3                        00000674-R      906     (13)        758     (10)
CVT_ERROR4                        0000067C-R      908     (13)        759     (10)
CVT_ERROR5                        00000686-R      910     (13)        760     (10)
CVT_ERROR6                        00000694-R      912     (13)        761     (10)
CVT_ERROR7                        000006A9-R      914     (13)        762     (10)
CVT_ERR_TABLE                     00000676-R      753     (10)       1812     (21)
CVT_FORWARD                       00000740-R      956     (13)        772     (10)
CVT_FUN_TABLE                     0000062E-R      731     (10)       1812     (21)
CVT_GETSTAT                       00000603-R      882     (13)        735     (10)
CVT_LDHEADS                       000006E8-R      928     (13)        747     (10)
CVT_LOAD                          000006C1-R      922     (13)        744     (10)
CVT_LOCKED                        000006FE-R      932     (13)        749     (10)
CVT_LOWER                         00000732-R      946     (13)        767     (10)
CVT_NOOP                          000005F1-R      878     (13)        733     (10)
CVT_RDNHC                         00000634-R      892     (13)        740     (10)
CVT_READ                          0000062A-R      890     (13)        739     (10)
CVT_READH                         00000613-R      886     (13)        737     (10)
CVT_REVERSE                       00000738-R      954     (13)        771     (10)
CVT_SEEK                          0000060E-R      884     (13)        736     (10)
CVT_SK                            000006F6-R      930     (13)        748     (10)
CVT_SPINDN                        0000071E-R      936     (13)        751     (10)
```

```
CVT_SPINUP                      000006D0-R    924      (13)      745      (10)
CVT_STA_TABLE                   00000652-R    742      (10)     1915      (21)
CVT_SUR_TABLE                   0000069A-R    764      (10)     1861      (21)      1866    (21)      1915    (21)      1950    (21)
CVT_UNLOAD                      0000070E-R    934      (13)      750      (10)
CVT_UPPER                       0000072C-R    944      (13)      766      (10)
CVT_WRITE                       0000061F-R    888      (13)      738      (10)
CVT_WRTCHK                      000005F7-R    880      (13)      734      (10)
CYLN                            000000D0-R    470      (8)     #-2052     (22)    #-2482    (27)
CYLN_LIST                       00000268-R    566      (8)
CYLN_LIST_X                     000002B2-R    576      (8)
DAR_DMA02_BITS                  00000294-R    832      (13)     1865      (21)      3644    (41)
DAR_DMA80_BITS                  000002C5-R    834      (13)     1902      (21)      3652    (41)
DAR_GS02_BITS                   000003AA-R    843      (13)     1857      (21)
DAR_MNEM                        000000BC-R    465      (8)     #-3644     (41)    #-3652    (41)
DAR_RCL80_BITS                  00000370-R    841      (13)     1899      (21)
DAR_SK02_BITS                   000002F4-R    836      (13)     1860      (21)
DAR_SK80_BITS                   00000337-R    839      (13)     1896      (21)
DEF                            =00000080      779      (11)
DEFAULT                        =00000000      797      (12)
DEL$V_READH                    =0000000A                       #-2992     (32)    #-3027    (32)
DELTA_TIME                      00000084-R    454      (8)
DERR_FLAG                       000000B4-R    463      (8)       2443     (27)    #-2579    (28)    #-2617    (28)
DEST_CYLN                       000000E4-R    475      (8)     #-1987     (21)    #-1993    (21)    #-2895    (31)    #-2899    (31)
                                                               #-2908     (31)    #-2919    (31)    #-2921    (31)    #-2991    (32)
                                                               #-3005     (32)    #-3007    (32)    #-3034    (32)    #-3306    (36)
                                                               #-3311     (36)
DEV                             00000008-R    428      (8)
DEV_NAME                        00000044-R    444      (8)     #-1427     (15)    #-1464    (15)    #-1471    (15)    #-1482    (15)
                                                               #-1505     (15)    #-3712    (42)
DEV_REG                         000000C0-R    466      (8)        97      (3)
DIFF_WORD                       00000100-R    482      (8)     #-1986     (21)    #-1989    (21)    #-2896    (31)    #-3315    (36)
DISK_SECTION                    000000AC-R    461      (8)
DISPATCH                        00000000-R    110      (4)        97      (3)
DMA_CMD                         00000C08-R    2733     (29)     2731      (29)
DRIVE_INIT$A                    000007A5-R    2193     (24)     1519      (15)
DRIVE_TBL                       00000000-R    426      (8)     #-1472     (15)    #-1764    (21)    #-2265    (25)    #-2366    (26)
                                                               #-2559     (28)    #-2687    (29)    #-2795    (30)    #-2877    (31)
                                                               #-2989     (32)    #-3108    (33)    #-3186    (34)    #-3435    (38)
                                                               #-3550     (40)    #-3631    (41)    #-3709    (42)    #-3711    (42)
                                                               #-3878     (45)
DRV_FATAL                       00000C92-R    2819     (30)    #-2803     (30)      2813    (30)
DS$ABORT                        00010020                         1441     (15)      1459    (15)      1516    (15)      2822    (30)
                                                                 3366     (37)      3372    (37)      3377    (37)      3830    (44)
DS$ASKLGCL                      00010098                         1512     (15)
DS$BRANCH                       000100A8                         1420     (15)      1437    (15)      3363    (37)      3369    (37)
                                                                 3374     (37)
DS$BREAK                        00010058                         1524     (15)      1586    (16)      1595    (18)
DS$CANWAIT                      00010070                         3834     (44)
DS$CHANNEL                      00010180                         1521     (15)      1569    (16)      3362    (37)      3373    (37)
DS$CKLOOP                       00010040                         3365     (37)      3371    (37)      3376    (37)
DS$CVTREG                       000100B0                         1655     (19)      1657    (19)      1812    (21)      1821    (21)
                                                                 1830     (21)      1868    (21)      1904    (21)      1915    (21)
                                                                 1924     (21)      1933    (21)      1942    (21)      1950    (21)
                                                                 1958     (21)      1966    (21)      1973    (21)      1980    (21)
DS$ENDPASS                      00010010                         1517     (15)
DS$ERRDEV                       000100C8                         2821     (30)      3573    (40)
```

```
DS$ERRSYS                000100C0                   1440   (15)    1454   (15)    3364   (37)    3370   (37)
                                                    3375   (37)    3829   (44)
DS$GPHARD                00010018                   1419   (15)
DS$INLOOP                00010048                   2887   (31)    2932   (31)    3001   (32)
DS$K_BBC                =0000001D   3374   (37)
DS$K_BBCC               =00000021   3374   (37)
DS$K_BBCCI              =00000023   3374   (37)
DS$K_BBCS               =0000001F   3374   (37)
DS$K_BBS                =0000001C   3374   (37)
DS$K_BBSC               =00000020   3374   (37)
DS$K_BBSS               =0000001E   3374   (37)
DS$K_BBSSI              =00000022   3374   (37)
DS$K_BCC_M              =0000001B   3374   (37)
DS$K_BCS_M              =0000001A   3374   (37)
DS$K_BEQLU_B            =00000005   3374   (37)
DS$K_BEQLU_M            =00000011   3374   (37)
DS$K_BEQL_B             =00000004   3374   (37)
DS$K_BEQL_M             =00000010   3374   (37)
DS$K_BERROR             =00000026   3374   (37)
DS$K_BGEQU_B            =00000007   3374   (37)
DS$K_BGEQU_M            =00000013   3374   (37)
DS$K_BGEQ_B             =00000006   3374   (37)
DS$K_BGEQ_M             =00000012   3374   (37)
DS$K_BGTRU_B            =00000009   3374   (37)
DS$K_BGTRU_M            =00000015   3374   (37)
DS$K_BGTR_B             =00000008   3374   (37)
DS$K_BGTR_M             =00000014   3374   (37)
DS$K_BLBC               =00000025   3374   (37)
DS$K_BLBS               =00000024   3374   (37)
DS$K_BLEQU_B            =00000003   3374   (37)
DS$K_BLEQU_M            =0000000F   3374   (37)
DS$K_BLEQ_B             =00000002   3374   (37)
DS$K_BLEQ_M             =0000000E   3374   (37)
DS$K_BLSSU_B            =00000001   3374   (37)
DS$K_BLSSU_M            =0000000D   3374   (37)
DS$K_BLSS_B             =00000000   3374   (37)
DS$K_BLSS_M             =0000000C   3374   (37)
DS$K_BNEQU_B            =0000000B   3374   (37)
DS$K_BNEQU_M            =00000017   3374   (37)
DS$K_BNEQ_B             =0000000A   3374   (37)
DS$K_BNEQ_M             =00000016   3374   (37)
DS$K_BNERROR            =00000027   3374   (37)  #-1420   (15)  #-1437   (15)  #-3363   (37)  #-3369   (37)
                                                  #-3374   (37)
DS$K_BVC_M              =00000019   3374   (37)
DS$K_BVS_M              =00000018   3374   (37)
DS$PRINTB                000100E0                   1644   (19)    1645   (19)    1654   (19)    1656   (19)
                                                    1658   (19)    1660   (19)    1661   (19)    1664   (19)
                                                    1713   (20)    1714   (20)    1716   (20)    1717   (20)
                                                    1718   (20)    1720   (20)    1798   (21)    1804   (21)
                                                    1813   (21)    1822   (21)    1831   (21)    1869   (21)
                                                    1905   (21)    1916   (21)    1925   (21)    1934   (21)
                                                    1943   (21)    1951   (21)    1952   (21)    1959   (21)
                                                    1960   (21)    1967   (21)    1974   (21)    1981   (21)
                                                    2052   (22)    2053   (22)    2054   (22)    2067   (22)
                                                    2071   (22)
DS$PRINTF                000100F0                   1458   (15)    1464   (15)    1482   (15)    1505   (15)
```

```
                                                    3575    (40)
DS$PRINTX                    000100E8               1990    (21)     1991   (21)     1992   (21)     1993   (21)
                                                    1998    (21)
DS$PROBE                     000101A0               1436    (15)
DS$RELBUF                    00010128               1585    (16)
DS$SETIPL                    00010178               1523    (15)
DS$SETMAP                    00010188               3368    (37)
DS$WAITMS                    00010060               2213    (24)     2714   (29)     2744   (29)
DS$WAITUS                    00010068               2272    (25)     2704   (29)     2719   (29)     2738   (29)
DSA$GL_FLAGS                 0000FE00               1409    (15)
DSA$GL_UNITS                 0000FE0C              #-3703   (42)    #-3763  (43)
DSA$V_PASS0                 =0000001D              #-1409   (15)
DUMP_R80                     00000178-R   1781  (21)    1768    (21)
DUMP_REG                     00000129-R   1757  (21)   #-1663   (19)    #-1719  (20)     2051   (22)
DUMP_RL02                    0000014C-R   1769  (21)    1768    (21)
E.BA                         00000160-R   506   (8)
E.BC                         00000164-R   507   (8)
E.BR                         00000188-R   515   (8)  #-1431    (15)
E.CS                         0000015C-R   505   (8)
E.DA                         00000168-R   508   (8)
E.DAT                        00000184-R   514   (8)
E.HDR                        0000017C-R   513   (8)
E.MP                         0000016C-R   509   (8)
E.MP1                        00000170-R   510   (8)
E.MP2                        00000174-R   511   (8)
E.MP3                        00000178-R   512   (8)  #-3121    (33)    #-3123  (33)
E.VEC                        0000018C-R   516   (8)  #-1430    (15)
ENV$M_DOMAIN                =00000002     66    (2)
ENV$M_LEVEL                 =00000001     66    (2)
ENV$M_SUPER                 =000003FC     66    (2)
ENV$S_DOMAIN                =00000001     66    (2)
ENV$S_LEVEL                 =00000001     66    (2)
ENV$S_SUPER                 =00000008     66    (2)
ENV$V_DOMAIN                =00000001     66    (2)   66       (2)
ENV$V_LEVEL                 =00000000     66    (2)   66       (2)
ENV$V_SUPER                 =00000002     66    (2)   97       (3)
ENV$_CPU                    =00000000     66    (2)   66       (2)
ENV$_FUNCTIONAL             =00000000     66    (2)   66       (2)
ENV$_REPAIR                 =00000001     66    (2)   66       (2)
ENV$_SUPER                  =00000001     66    (2)   97       (3)
ENV$_SYSTEM                 =00000001     66    (2)   66       (2)
ERR$_MSGADR                 =0000000C     76    (2)
ERR$_NARGS                  =0000000A     76    (2)
ERR$_NUM                    =00000004     76    (2)
ERR$_P1                     =00000014     76    (2)   1642     (19)     1714   (20)     1717   (20)
ERR$_P2                     =00000018     76    (2)  #-1647    (19)    #-1717  (20)
ERR$_P3                     =0000001C     76    (2)  #-1648    (19)
ERR$_P4                     =00000020     76    (2)   1649     (19)
ERR$_P5                     =00000024     76    (2)
ERR$_P6                     =00000028     76    (2)
ERR$_POINTER                =00000010     76    (2)
ERR$_UNIT                   =00000008     76    (2)
ERR_CNT                      000001B8-R   519   (8)  #-2049    (22)    #-2071  (22)    #-2141  (23)
ERR_MSG                      000000B8-R   464   (8)  #-2884    (31)    #-2910  (31)    #-2917  (31)    #-2925  (31)
                                                     #-2998    (32)    #-3025  (32)    #-3031  (32)    #-3038  (32)
EVENT_FLAGS                  00000024-R   436   (8)   1413     (15)
```

```
EXIT_CHANGE      0000126B-R   3657   (41)   #-3648   (41)
EXIT_CHECK       00001420-R   3977   (46)   #-3972   (46)
EXIT_ECC         00001217-R   3576   (40)   #-3552   (40)   #-3556   (40)   #-3559   (40)   #-3574   (40)
EXIT_GETLUN      000012B6-R   3717   (42)   #-3707   (42)   #-3708   (42)
EXIT_GETSEC      000012F4-R   3775   (43)   #-3764   (43)
EXIT_READBAD     000013E3-R   3916   (45)   #-3893   (45)   #-3913   (45)
EXIT_READHDR     00000FF7-R   3260   (35)   #-3245   (35)   #-3251   (35)   #-3256   (35)
EXIT_RESTORE     00000C4B-R   2746   (29)   #-2698   (29)    2731    (29)
FIX02_DAR        00001239-R   3642   (41)    3640    (41)
FIX80_DAR        00001253-R   3650   (41)    3640    (41)
FMT_ATTACH       000007B6-R   969    (13)    1458    (15)
FMT_AVG_SEEK     0000099D-R   987    (13)
FMT_BA           00000B04-R   1000   (13)    1822    (21)
FMT_BAD          00000767-R   966    (13)    1658    (19)
FMT_BADBLK       00000E9C-R   1047   (13)
FMT_BC           00000B27-R   1002   (13)    1831    (21)
FMT_CRLF         00000748-R   963    (13)    1798    (21)
FMT_CRLF1        0000074D-R   964    (13)    1644    (19)    1654    (19)    1661    (19)    1664    (19)
                                             1713    (20)    1716    (20)    1718    (20)    1720    (20)
                                             1951    (21)    1959    (21)    1990    (21)
FMT_CS           00000AE1-R   998    (13)    1813    (21)
FMT_DA           00000B4A-R   1004   (13)    1869    (21)    1905    (21)
FMT_DATA         00000DD0-R   1037   (13)    2067    (22)
FMT_DRVTYP       0000078C-R   968    (13)    1464    (15)
FMT_ERRCOUNT     00000DEB-R   1039   (13)    2071    (22)
FMT_EXT_1A       00000CAD-R   1022   (13)    1991    (21)
FMT_EXT_1B       00000CD0-R   1024   (13)    1992    (21)
FMT_EXT_1C       00000CF0-R   1026   (13)    1993    (21)
FMT_EXT_2        00000D12-R   1028   (13)    1998    (21)
FMT_GOOD         00000750-R   965    (13)    1656    (19)
FMT_HDR          00000A70-R   995    (13)    1804    (21)
FMT_HDR_A        00000D33-R   1030   (13)    2052    (22)
FMT_HEADER1      00000DA6-R   1035   (13)    2054    (22)
FMT_HMB_ERROR    00000A4F-R   993    (13)    1482    (15)
FMT_MAX_SEEK     0000096D-R   985    (13)
FMT_MP           00000B6D-R   1006   (13)    1916    (21)    1925    (21)
FMT_MP1_02       00000BD6-R   1012   (13)    1952    (21)
FMT_MP1_80       00000C01-R   1014   (13)    1960    (21)
FMT_MP2_02       00000C2C-R   1016   (13)    1967    (21)
FMT_MP2_80       00000C57-R   1018   (13)    1974    (21)
FMT_MP3          00000C83-R   1020   (13)    1981    (21)
FMT_NYI          000009CD-R   989    (13)
FMT_ONE_SEEK     00000941-R   983    (13)
FMT_PAT          00000BB3-R   1010   (13)    1943    (21)
FMT_POS          00000B90-R   1008   (13)    1934    (21)
FMT_SEEK1        00000802-R   971    (13)
FMT_SEEK1_SP     00000888-R   977    (13)
FMT_SEEK2        0000082C-R   973    (13)
FMT_SEEK2_SP     000008C3-R   979    (13)
FMT_SEEK3        0000085A-R   975    (13)
FMT_SEEK3_SP     00000902-R   981    (13)
FMT_SPINDLE      00000E12-R   1041   (13)
FMT_SPINDLE_SP   00000E30-R   1043   (13)
FMT_SP_TIME      00000E73-R   1045   (13)
FMT_VOL          000009F4-R   990    (13)    1505    (15)
FMT_WDCNT        00000D8B-R   1033   (13)    2053    (22)
```

```
FMT_XOR                         0000077D-R   967     (13)      1660    (19)
GDDAT                           00000104-R   483     (8)
GETSTAT                        =00000004                     #-2203    (24)    #-2275    (25)    #-2276    (25)    #-2493    (27)
                                                             #-2586    (28)    #-2589    (28)    #-2700    (29)    #-2715    (29)
GET_LUN$A                       0000126C-R   3700    (42)      1572    (16)
GET_POSITION                    00000EB0-R   3091    (33)      2881    (31)      2914    (31)      2995    (32)      3029    (32)
                                                               3243    (35)      3254    (35)
GET_POSITION_X                  00000F4B-R   3128    (33)    #-3107    (33)    #-3118    (33)    #-3124    (33)
GET_RANDOM                      00001159-R   3488    (39)
GET_RANDOM_X                    00001197-R   3501    (39)
GET_SECLUN$A                    000012B7-R   3760    (43)
GOOD_DATA                       00000190-R   517     (8)     #-2055    (22)    #-2124    (23)    #-2136    (23)
HARD_TABLE                      00000058-R   447     (8)       1419    (15)    #-1422    (15)
HDR$M_ERR                      =00000020                     #-2883    (31)    #-2916    (31)    #-2997    (32)    #-3032    (32)
HDR$V_ERR                      =00000005                     #-2889    (31)    #-3003    (32)
HDR02_BITS                      00000590-R   868     (13)       646    (9)
HDR1_02_BITS                    00000534-R   862     (13)      1950    (21)
HDR1_80_BITS                    00000555-R   864     (13)      1958    (21)
HDR2_80_BITS                    00000574-R   866     (13)      1973    (21)
HDR80_BITS                      000005C1-R   870     (13)       682    (9)
HDR_02$S_CYL                   =00000009                        710    (9)
HDR_02$S_SEC                   =00000006                        714    (9)
HDR_02$S_TRK                   =00000001                        712    (9)
HDR_02$V_CYL                   =00000007                        709    (9)
HDR_02$V_SEC                   =00000000                        713    (9)
HDR_02$V_TRK                   =00000006                        711    (9)
HDR_80$M_FMT                   =00001000                        686    (9)
HDR_80$M_MF                    =00008000                        687    (9)
HDR_80$M_SSF                   =00002000                        689    (9)
HDR_80$M_UF                    =00004000                        688    (9)
HDR_80$S_CYL                   =0000000A                        718    (9)
HDR_80$S_FMT                   =00000001                     #-3191    (34)
HDR_80$S_SEC                   =00000005                        722    (9)
HDR_80$S_SSF                   =00000001                     #-3192    (34)
HDR_80$S_TRK                   =00000004                        720    (9)
HDR_80$V_CYL                   =00000000                        717    (9)
HDR_80$V_FMT                   =0000000C                     #-3191    (34)
HDR_80$V_SEC                   =00000010                        721    (9)
HDR_80$V_SSF                   =0000000D                     #-3192    (34)
HDR_80$V_TRK                   =00000018                        719    (9)
HMB$C_LEVEL1                   =00000101                     #-1485    (15)
HMB$T_VOLNAME                   0000000E                     #-1488    (15)
HMB$T_VOLNAME2                  000001D8                     #-1490    (15)
HMB$W_STRUCLEV                  0000000C                     #-1486    (15)
HMB_ERR_FLAG                    0000007C-R   452     (8)     #-1478    (15)    #-3444    (38)    #-3446    (38)
HOLD_CS                         000000A0-R   458     (8)     #-2269    (25)    #-2270    (25)    #-2271    (25)
HOLD_DA                         000000F8-R   480     (8)
HP$A_DEVICE                     00000018                     #-1432    (15)
HP$A_LINK                       00000020                     #-1429    (15)
HP$B_DRIVE                      0000000B                     #-1423    (15)
HP$Q_DEVICE                     00000000                     #-1427    (15)
HP$T_TYPE                       00000026                       1428    (15)
HP$W_VECTOR                     00000024                     #-1430    (15)
IDCBAR                         =00000004                     #-1819    (21)    #-2351    (26)    #-2490    (27)
IDCBAR_BITS                     00000253-R   828     (13)      1821    (21)
IDCCNT                         =00000008                     #-1828    (21)    #-2352    (26)    #-2491    (27)
```

```
IDCCNT_BITS                     00000274-R   830    (13)     1830    (21)
IDCCSR                         =00000000                    #-1565   (16)   #-1566   (16)   #-2198   (24)   #-2199   (24)
                                                            #-2200   (24)   #-2201   (24)   #-2203   (24)   #-2205   (24)
                                                            #-2206   (24)   #-2210   (24)   #-2212   (24)   #-2215   (24)
                                                            #-2271   (25)   #-2278   (25)   #-2350   (26)   #-2488   (27)
                                                            #-2496   (27)   #-2574   (28)   #-2582   (28)   #-2591   (28)
                                                            #-2593   (28)   #-2703   (29)   #-2711   (29)   #-2713   (29)
                                                            #-2718   (29)   #-2737   (29)   #-2741   (29)   #-2743   (29)
                                                            #-3822   (44)   #-3826   (44)   #-3831   (44)   #-3832   (44)
IDCCSR_BITS                     00000145-R   818    (13)     1812    (21)
IDCDAR                         =0000000C                    #-1838   (21)   #-1877   (21)   #-2211   (24)   #-2353   (26)
                                                            #-2492   (27)   #-2712   (29)   #-2742   (29)
IDCDSINT_ERR                    0000146F-R   1118   (13)
IDCENINT_ERR                    0000144D-R   1117   (13)
IDCGETBUF                       00001492-R   1119   (13)
IDCMPR                         =00000010                    #-1912   (21)   #-1922   (21)   #-2204   (24)   #-2274   (25)
                                                            #-2369   (26)   #-2372   (26)   #-2373   (26)   #-2374   (26)
                                                            #-2379   (26)   #-2495   (27)   #-2588   (28)   #-2596   (28)
                                                            #-2699   (29)   #-2739   (29)
IDCMPR02_BITS                   000003D0-R   845    (13)     1915    (21)
IDCMPR80_BITS                   0000046B-R   853    (13)     1924    (21)
IDCPAT                         =00000018                    #-1940   (21)   #-2355   (26)
IDCPAT_BITS                     00000512-R   860    (13)     1942    (21)
IDCPOS                         =00000014                    #-1931   (21)   #-2354   (26)
IDCPOS_BITS                     000004F1-R   858    (13)     1933    (21)
IDC_ADRFAIL                     0000139A-R   1110   (13)     1440    (15)
IDC_ISR$A                       000012F8-R   3818   (44)     1521    (15)
IDC_SADRFAIL                    000013C5-R   1111   (13)
IDC_STAT$L_1                    0000000C-R   430    (8)
IDC_STAT$W_2                    00000010-R   431    (8)
IDC_STAT$W_VR                   00000012-R   432    (8)
IDC_STATUS                      0000000C-R   429    (8)      1521    (15)
INH_PNT                         00000108-R   485    (8)
INITIALIZE                      00000000-R   1407   (15)     97      (3)
INITIALIZE_X                    00000244-R   1524   (15)
ITERAT                          00000110-R   487    (8)
L$A_CCP                         00000040-R   97     (3)
L$A_DEVP                        0000001C-R   97     (3)
L$A_DREG                        00000024-R   97     (3)
L$A_DTP                         00000018-R   97     (3)
L$A_ICP                         0000003C-R   97     (3)
L$A_LASTAD                      00000014-R   97     (3)
L$A_NAME                        00000008-R   97     (3)
L$A_REPP                        00000044-R   97     (3)
L$A_SECNAM                      00000050-R   97     (3)
L$A_STATAB                      00000048-R   97     (3)
L$A_TSTCNT                      00000054-R   97     (3)
L$L_ENVIRON                     00000004-R   97     (3)
L$L_ERRTYP                      0000004C-R   97     (3)
L$L_HEADLENGTH                  00000000-R   97     (3)
L$L_REV                         0000000C-R   97     (3)
L$L_UNIT                        00000020-R   97     (3)
L$L_UPDATE                      00000010-R   97     (3)
L$UNIT                          0000002C-R   438    (8)     #-1470   (15)   #-1518   (15)   #-1521   (15)   #-1569   (16)
                                                            #-2821   (30)   #-3362   (37)   #-3364   (37)   #-3368   (37)
                                                            #-3370   (37)   #-3373   (37)   #-3375   (37)   #-3573   (49)
```

```
                                                         #-3710    (42)    #-3829    (44)
LASTAD                           00000000-R   97      (3)      97     (3)
LUN0_TBL                         000002EC-R   616     (9)      611    (9)
LUN1_TBL                         00000370-R   619     (9)      612    (9)
LUN2_TBL                         000003F4-R   622     (9)      613    (9)
LUN3_TBL                         00000478-R   625     (9)      614    (9)
LUN_TBL                          000002DC-R   610     (9)    #-1466   (15)    #-3709    (42)    #-3768    (43)
L_TSTCNT                         00000000-R   97      (3)      97     (3)
MANUAL                          =00000001     797     (12)
MBSF                             00000800-R   604     (8)      3905   (45)     3964     (46)      597     (8)      598     (8)
MBSF_BUFFER                      000002D4-R   597     (8)      3895   (45)
MPR02_BITS                       00000421-R   849     (13)     645    (9)
MPR80_BITS                       000004A9-R   855     (13)     681    (9)
MPR_02$M_DSE                    =00000100                    #-2695   (29)
MPR_02$M_HCE                    =00004000                    #-2802   (30)
MPR_02$M_SKTO                   =00001000                    #-2696   (29)
MPR_02$M_SPE                    =00000800                    #-2696   (29)
MPR_02$M_VC                     =00000200                    #-2696   (29)
MPR_02$M_WDE                    =00008000                    #-2802   (30)
MPR_02$M_WGE                    =00000400                    #-2695   (29)
MPR_80$M_FLT                    =00000100                    #-2816   (30)      684     (9)
MPR_80$M_SECT                   =0000001F                      693    (9)
MPR_80$M_SKE                    =00000400                      684    (9)
MPR_80$V_SKE                    =0000000A                    #-2708   (29)
MPR_GS$M_GS                     =00000002                      642    (9)       650     (9)
MPR_GS$M_MARK                   =00000001                      641    (9)       649     (9)
MPR_GS$M_RST                    =00000008                      651    (9)
MS100_DAT_DST                    00002C04-R   1309    (13)
MS103_SK_ONE                     00002C32-R   1311    (13)
MS104_SK_MAX                     00002C5A-R   1313    (13)
MS105_SK_AVG                     00002C86-R   1315    (13)
MS106_SP_TIM                     00002CB2-R   1317    (13)
MS107A_SEEKERR                   00002CFE-R   1321    (13)
MS107B_RECAL                     00002D24-R   1323    (13)
MS107_RECAL                      00002CCF-R   1319    (13)
MS108_SECTNZ                     00002D4A-R   1325    (13)
MS109_BADCNT                     00002D7B-R   1327    (13)
MS110_UEHNF31                    00002DAF-R   1329    (13)
MS111_EHNF31                     00002DE7-R   1331    (13)
MS112_ACC31                      00002E21-R   1333    (13)
MS113_UWRTERR                    00002E41-R   1335    (13)
MS114A_UWCKERR                   00002EA6-R   1339    (13)
MS114_URDERR                     00002E75-R   1337    (13)
MS115A_WKRST                     00002F1D-R   1343    (13)
MS115_RDRST                      00002EDE-R   1341    (13)
MS116A_AUTOCMP                   00002F87-R   1347    (13)
MS116_SKPCMP                     00002F59-R   1345    (13)
MS117_AUTORD                     00002FF3-R   1351    (13)
MS117_AUTOWCK                    00003032-R   1353    (13)
MS117_AUTOWRT                    00002FBA-R   1349    (13)
MS118_SEEKIDC                    0000306A-R   1355    (13)
MS119_IDCPOS                     000030A5-R   1357    (13)
MS44A_INC_CRC                    00001FDF-R   1207    (13)     3127   (33)
MS45A_HD_PROX                    0000202D-R   1209    (13)
MS83A_ST4_DE                     00002866-R   1273    (13)
MS94A_NO_OTH                     00002ACF-R   1297    (13)
```

```
MSG10_RG_INIT                   00001797-R    1154    (13)
MSG11_GS_CRDY                   000017C2-R    1155    (13)    2283    (25)
MSG12_GS_DRDY                   000017F6-R    1156    (13)    2294    (25)
MSG13_INI_ERR                   00001810-R    1157    (13)    2304    (25)
MSG14_SYSINIT                   0000183E-R    1158    (13)
MSG15_NOOP                      00001869-R    1159    (13)
MSG16A_WRTCHK                   000018BA-R    1161    (13)
MSG16B_WRTCHK                   000018E3-R    1162    (13)
MSG16_WRTCHK                    0000188E-R    1160    (13)
MSG17_GETSTAT                   0000191B-R    1163    (13)
MSG18B_SEEK                     00001968-R    1165    (13)
MSG18_SEEK                      00001943-R    1164    (13)    3025    (32)    3038    (32)
MSG19A_READH                    000019B6-R    1167    (13)
MSG19_READH                     0000198A-R    1166    (13)    2884    (31)    2917    (31)    2998    (32)    3031    (32)
MSG1_ADR_ERR                    00001613-R    1143    (13)
MSG20A_FORMAT                   00001A7E-R    1171    (13)
MSG20A_WRITE                    00001A4B-R    1170    (13)
MSG20B_WRITE                    00001AA2-R    1172    (13)
MSG20C_WRITE                    00001ADA-R    1173    (13)
MSG20_WRITE                     000019F2-R    1168    (13)
MSG20_WRT_BND                   00001A1D-R    1169    (13)
MSG21A_READ                     00001B36-R    1175    (13)
MSG21A_READH                    00001B68-R    1176    (13)
MSG21B_READ                     00001B95-R    1177    (13)
MSG21_READ                      00001B0C-R    1174    (13)
MSG22_READNHC                   00001BCC-R    1178    (13)
MSG23_CS_INC                    00001C04-R    1179    (13)
MSG24A_BC_MOD                   00001C47-R    1181    (13)
MSG24_BA_MOD                    00001C2A-R    1180    (13)
MSG25_DA_MOD                    00001C64-R    1182    (13)
MSG26_MP_MOD                    00001C81-R    1183    (13)
MSG27_MP_INC                    00001C9E-R    1184    (13)
MSG28_DA_INC                    00001CBF-R    1185    (13)
MSG29_BA_INC                    00001CDF-R    1186    (13)
MSG2_RG_ZERO                    00001635-R    1144    (13)
MSG30A_CRDY                     00001D1D-R    1188    (13)
MSG30_CRDY                      00001CFE-R    1187    (13)    2568    (28)
MSG31_DRDY                      00001D54-R    1189    (13)    2578    (28)
MSG32_CERR                      00001D6E-R    1190    (13)    2627    (28)
MSG33A_UXEATTN                  00001D9D-R    1192    (13)    2608    (28)
MSG33B_NOATTN                   00001DB8-R    1194    (13)    2613    (28)
MSG33_DERR                      00001D88-R    1191    (13)    2616    (28)
MSG34_GS_OPI                    00001DD6-R    1195    (13)
MSG35_OPI_TO                    00001E10-R    1196    (13)
MSG36_OPI_RLY                   00001E37-R    1197    (13)
MSG37_OPI_LAT                   00001E63-R    1198    (13)
MSG38_NOOP_IE                   00001E8D-R    1199    (13)
MSG39A_INT                      00001EDB-R    1201    (13)    3829    (44)
MSG39_NO_INT                    00001EB7-R    1200    (13)
MSG3_RG_ONE                     0000164F-R    1145    (13)
MSG40_BAD_BR                    00001F0D-R    1202    (13)
MSG41_BAD_VEC                   00001F36-R    1203    (13)
MSG42_RPT_CYL                   00001F5D-R    1204    (13)
MSG43_MP2_NZ                    00001F95-R    1205    (13)
MSG44_BAD_CRC                   00001FB6-R    1206    (13)
MSG45_SK_ZERO                   0000200D-R    1208    (13)
```

```
MSG46_POS_ERR                       00002068-R    1211     (13)     2926    (31)     3039    (32)
MSG47_DS_INC                        00002082-R    1212     (13)
MSG48_SK_ONE                        00002099-R    1213     (13)     2910    (31)     2925    (31)
MSG49_DRDY_NC                       000020B3-R    1214     (13)
MSG4_RG_FLT1                        00001669-R    1146     (13)
MSG50_SK_OUTR                       000020CD-R    1215     (13)
MSG51_SK_INNR                       000020F1-R    1216     (13)
MSG52_RD_HNF                        00002115-R    1217     (13)
MSG53_WRT_NXM                       00002142-R    1218     (13)
MSG54_DA_UPD                        00002173-R    1219     (13)
MSG55_DAT_BND                       000021DB-R    1221     (13)
MSG55_DAT_CP1                       000021A6-R    1220     (13)
MSG55_HDRCMP                        0000220F-R    1222     (13)
MSG56_DAT_UNQ                       00002240-R    1223     (13)
MSG57_WCK_FRC                       00002287-R    1225     (13)
MSG58_PAR_WRT                       000022B1-R    1226     (13)
MSG59_PAR_RD                        000022E4-R    1227     (13)
MSG5_RG_FLT0                        0000168C-R    1147     (13)
MSG60_WRT_PRD                       00002316-R    1228     (13)
MSG61_DAT_PAR                       0000234E-R    1229     (13)
MSG62_DC_PAR                        0000238E-R    1231     (13)
MSG63_PRD_TMW                       000023C8-R    1232     (13)
MSG64_WCK_PAR                       000023F5-R    1233     (13)
MSG65_DAT_CP2                       00002428-R    1234     (13)
MSG66_HNF_LT                        00002462-R    1235     (13)
MSG67_DAT_ES                        00002497-R    1236     (13)
MSG68_DAT_MAX                       000024DF-R    1238     (13)
MSG69_DAT_NHC                       00002522-R    1240     (13)
MSG6_CS_FAIL                        000016AF-R    1148     (13)
MSG70A_NFSEC31                      00002583-R    1243     (13)     3249    (35)
MSG70_FND_S39                       0000255F-R    1242     (13)
MSG71_RD_BKT                        0000259F-R    1245     (13)
MSG72_DAT_BKT                       000025E3-R    1247     (13)
MSG73_GS_UND                        00002621-R    1249     (13)
MSG74_DRDY_NC                       0000264D-R    1251     (13)
MSG75_NO_ST1                        00002678-R    1253     (13)
MSG76_ST1_BS                        000026A5-R    1255     (13)
MSG77_NO_ST2                        000026D1-R    1257     (13)
MSG77_NO_ST3                        000026FE-R    1259     (13)
MSG78_ST2_BS                        0000272B-R    1261     (13)
MSG79_BH_NC                         0000275B-R    1263     (13)
MSG7A_BC_FAIL                       000016F9-R    1150     (13)
MSG7_BA_FAIL                        000016D4-R    1149     (13)
MSG80_BH_NF                         00002795-R    1265     (13)
MSG81_ST3_BS                        000027D0-R    1267     (13)
MSG82_NO_ST4                        000027FF-R    1269     (13)
MSG83_ST4_BS                        0000282C-R    1271     (13)
MSG84_NO_ST5                        0000289F-R    1275     (13)
MSG85_ST5_BS                        000028CC-R    1277     (13)
MSG86_DRDY_NS                       00002900-R    1279     (13)
MSG87_VC_NC                         00002938-R    1281     (13)
MSG88_NO_ST6                        00002971-R    1283     (13)
MSG89_ST6_BS                        0000299E-R    1285     (13)
MSG8A_MP_FAIL                       00001743-R    1152     (13)
MSG8_DA_FAIL                        0000171E-R    1151     (13)
MSG90_NO_ST7                        000029D2-R    1287     (13)
```

```
MSG91_ST7_BS                     000029FF-R   1289    (13)
MSG92_NO_ST0                     00002A2D-R   1291    (13)
MSG93_RELV_UP                    00002A5A-R   1293    (13)
MSG94_UXE_DSE                    00002A93-R   1295    (13)
MSG95_DSE_NS                     00002AFA-R   1299    (13)
MSG96_DSE_NC                     00002B33-R   1301    (13)
MSG97_GS_NXD                     00002B67-R   1303    (13)
MSG98_WRT_WL                     00002B99-R   1305    (13)
MSG99_RST_WGE                    00002BC7-R   1307    (13)
MSG9_RG_RMW                      00001768-R   1153    (13)
MSG_ABORT                        0000153A-R   1136    (13)      1512    (15)
MSG_AVG                          0000150E-R   1131    (13)
MSG_BLANK                        00001531-R   1135    (13)
MSG_DRVFATAL                     0000158A-R   1140    (13)      2821    (30)
MSG_ECCFAIL                      000015DB-R   1142    (13)      3573    (40)
MSG_ECCPASS                      000015B6-R   1141    (13)      3575    (40)
MSG_INNER                        00001528-R   1134    (13)
MSG_MAX                          00001506-R   1130    (13)
MSG_MIDDLE                       0000151F-R   1133    (13)
MSG_ONE                          00001502-R   1129    (13)
MSG_OPERATTCH                    0000155B-R   1138    (13)      1454    (15)
MSG_OUTER                        00001516-R   1132    (13)
MSG_R80                          000014FE-R   1128    (13)      673     (9)      729     (10)
MSG_RL02                         000014F9-R   1127    (13)      636     (9)      728     (10)
MSG_TRACE                        000014D9-R   1126    (13)
NEW_CYLN                         000000EC-R   477     (8)     #-2909    (31)   #-2922    (31)   #-2934    (31)
NOCHK$M_DERR                    =00000008                     #-2630    (28)
NOCHK$M_FTL                     =00000040                     #-1473    (15)   #-1476    (15)   #-1575    (16)
NOCHK$V_ATTN                    =00000007                     #-2602    (28)
NOCHK$V_CERR                    =00000001                     #-2620    (28)
NOCHK$V_CRC                     =00000005                     #-3117    (33)
NOCHK$V_CRDY                    =00000000                     #-2563    (28)
NOCHK$V_DERR                    =00000003                     #-2300    (25)   #-2598    (28)   #-2625    (28)
NOCHK$V_DRDY                    =00000002                     #-2288    (25)   #-2572    (28)
NOCHK$V_DSTA                    =00000004                     #-2583    (28)
NOCHK$V_FTL                     =00000006                     #-2794    (30)
NON_FATAL                        00000CAE-R   2824    (30)    #-2794    (30)     2813    (30)   #-2817    (30)
NO_OP                           =00000000                     #-2269    (25)
NUMENT                          =00000021                     #-1468    (15)     617     (9)      620     (9)      623     (9)
                                                                626     (9)
OFST_AVGTIM                     =00000080     82      (2)       779     (11)
OFST_CSRTYP                     =00000058     82      (2)       779     (11)
OFST_DEVMEM                     =0000001C     82      (2)     #-1447    (15)   #-1458    (15)   #-1464    (15)     779     (11)
OFST_DEVNAM                     =00000020     82      (2)     #-1471    (15)   #-3712    (42)     779     (11)
OFST_DRVERR                     =0000004C     82      (2)     #-2267    (25)   #-2561    (28)     779     (11)
OFST_DRVRST                     =00000050     82      (2)     #-2266    (25)   #-2699    (29)     779     (11)
OFST_DRVTYP                     =00000000     82      (2)     #-1768    (21)   #-2367    (26)   #-2691    (29)   #-2734    (29)
                                                              #-2799    (30)   #-3551    (40)   #-3640    (41)   #-3882    (45)
                                                                779     (11)
OFST_FMTLEN                     =00000030     82      (2)       779     (11)
OFST_GTSCMD                     =00000034     82      (2)     #-2560    (28)     779     (11)
OFST_HDBITS                     =00000044     82      (2)       779     (11)
OFST_HDRDNC                     =00000054     82      (2)       779     (11)
OFST_HDRTMP                     =00000048     82      (2)     #-2878    (31)   #-2990    (32)   #-3109    (33)   #-3188    (34)
                                                                779     (11)
OFST_HIGRPM                     =00000074     82      (2)       779     (11)
```

```
OFST_HMBSEC          =0000005C     82    (2)    #-3437   (38)     779   (11)
OFST_IGBCYL          =00000018     82    (2)      779    (11)
OFST_LOWRPM          =00000070     82    (2)      779    (11)
OFST_LSTCYL          =00000008     82    (2)    #-3881   (45)     779   (11)
OFST_LSTSEC          =00000010     82    (2)      779    (11)
OFST_LSTTRK          =0000000C     82    (2)    #-3889   (45)   #-3890  (45)     779   (11)
OFST_MAXTIM          =0000007C     82    (2)      779    (11)
OFST_MPBITS          =00000040     82    (2)      779    (11)
OFST_NMLSTS          =00000038     82    (2)      779    (11)
OFST_ONETIM          =00000078     82    (2)      779    (11)
OFST_OP1STS          =00000060     82    (2)      779    (11)
OFST_PHYU            =00000004     82    (2)    #-1469   (15)   #-3713  (42)   #-3715  (42)   #-3771  (43)
                                                #-3773   (43)     779   (11)
OFST_RANCYL          =0000006C     82    (2)      779    (11)
OFST_SECLEN          =00000028     82    (2)      779    (11)
OFST_SEKSTS          =0000003C     82    (2)      779    (11)
OFST_STSDNC          =00000064     82    (2)      779    (11)
OFST_TRKLEN          =0000002C     82    (2)      779    (11)
OFST_TSTCYL          =00000014     82    (2)      779    (11)
OFST_WTPSTS          =00000068     82    (2)      779    (11)
OLD_CYLN              000000E8-R   476   (8)    #-2908   (31)   #-2921  (31)   #-2934  (31)   #-2935  (31)
                                                #-3007   (32)   #-3009  (32)
OPR10_INS_ID          0000114A-R   1081  (13)
OPR11_ORIGIN          000011BE-R   1085  (13)
OPR12_WRTLOCK         00001225-R   1088  (13)
OPR13A_RST_FLT        00001269-R   1092  (13)
OPR13_RST_WL          00001246-R   1090  (13)
OPR14_ROTATE          00001289-R   1094  (13)
OPR15_SWAP            000012E4-R   1097  (13)
OPR16_SWAP_BK         0000130B-R   1099  (13)
OPR1_UNLOAD           00000EE7-R   1056  (13)
OPR2_CLOSE            00000F75-R   1061  (13)
OPR3_START            00000FB2-R   1063  (13)
OPR4_MKRDY            00000FDF-R   1065  (13)
OPR5_UNLOAD           00001013-R   1067  (13)
OPR6_REMOVE           00001042-R   1069  (13)
OPR7_INSERT           0000106F-R   1071  (13)
OPR8_SETUP            00001090-R   1073  (13)
OPR9_DUPPLUG          000010C0-R   1075  (13)
OPR_AVAIL             00001137-R   1079  (13)
OPR_FAULT             00001363-R   1102  (13)
OPR_RETURN            00000ED1-R   1054  (13)
P$ATTN                0000003C-R   442   (8)    #-1425   (15)   #-2611  (28)   #-3714  (42)   #-3823  (44)
P$UNIT                00000034-R   440   (8)    #-1426   (15)   #-1443  (15)   #-2270  (25)   #-2277  (25)
                                                #-2487   (27)   #-2590  (28)   #-2702  (29)   #-2710  (29)
                                                #-2736   (29)   #-2740  (29)   #-3715  (42)
PAR$M_ATDEF          =00000010     81    (2)
PAR$M_ATHI           =00000008     81    (2)
PAR$M_ATLO           =00000004     81    (2)
PAR$M_NODEF          =00000002     81    (2)
PAR$V_ATDEF          =00000004     81    (2)
PAR$V_ATHI           =00000003     81    (2)
PAR$V_ATLO           =00000002     81    (2)
PAR$V_NODEF          =00000001     81    (2)
PAR$_BIN             =00000002     81    (2)
PAR$_DEC             =0000000A     81    (2)
```

```
PAR$_HEX            =00000010      81     (2)
PAR$_NO             =00000000      81     (2)     #-1512  (15)
PAR$_OCT            =00000008      81     (2)
PAR$_YES            =00000001      81     (2)
PAT$S_BITS          =0000000B                     #-3565  (40)   #-3567  (40)
PHY_RD_BUF           0000006C-R    450    (8)
PHY_WRT_BUF          0000005C-R    448    (8)
PNT_FLAG             0000010C-R    486    (8)     #-1414  (15)
POSITION_HDS         00000DBE-R    2987   (32)     3430   (38)     3892  (45)
POSITION_HDS_X       00000EAF-R    3045   (32)    #-2999  (32)   #-3026  (32)   #-3033  (32)   #-3043  (32)
POS_TYPE             000000CC-R    469    (8)      3014   (32)   #-3310  (36)
PREV_CYLN            000000E0-R    474    (8)     #-1987  (21)   #-1991  (21)   #-2470  (27)   #-2909  (31)
                                                  #-2922  (31)
PRINT_BA             0000022C-R    1816   (21)     1762   (21)
PRINT_BC             00000276-R    1825   (21)     1763   (21)
PRINT_CS             000001DF-R    1807   (21)     1761   (21)
PRINT_DATA_ERR       000006B8-R    2045   (22)
PRINT_DA_02          000002C0-R    1834   (21)     1771   (21)
PRINT_DA_80          0000034E-R    1873   (21)     1783   (21)
PRINT_EXT_1          0000065A-R    1984   (21)     1795   (21)
PRINT_EXT_2          000006A6-R    1996   (21)     1797   (21)
PRINT_MP1_02         00000502-R    1946   (21)     1776   (21)
PRINT_MP1_80         0000054F-R    1955   (21)     1788   (21)
PRINT_MP2_02         0000059A-R    1963   (21)     1778   (21)
PRINT_MP2_80         000005DA-R    1970   (21)     1790   (21)
PRINT_MP3            0000061A-R    1977   (21)     1793   (21)
PRINT_MP_02          000003D6-R    1909   (21)     1772   (21)
PRINT_MP_80          00000424-R    1919   (21)     1784   (21)
PRINT_PAT            000004B8-R    1937   (21)     1774   (21)     1786  (21)
PRINT_POS            0000046E-R    1928   (21)     1773   (21)     1785  (21)
PRINT_REG            000000D7-R    1708   (20)     2821   (30)     3829  (44)
PRINT_REG_HDR        000001D1-R    1802   (21)     1760   (21)
PRT$M_ALL           =00000007                     #-2475  (27)
PRT$M_EXT_1         =00000008                     #-2911  (31)   #-2923  (31)   #-3024  (32)   #-3040  (32)
PRT$M_EXT_2         =00000010                     #-2923  (31)   #-3040  (32)
PRT$V_EXT_1         =00000003                     #-1794  (21)
PRT$V_EXT_2         =00000004                     #-1796  (21)
PRT$V_MPR_1         =00000000                     #-1775  (21)   #-1787  (21)
PRT$V_MPR_2         =00000001                     #-1777  (21)   #-1789  (21)
PRT$V_MPR_3         =00000002                     #-1792  (21)
PRT$V_NONE          =00000006                     #-1662  (19)
R80                 =00000002      805    (12)
R80$T               =00000002                     #-2367  (26)   #-3551  (40)     666  (9)
R80_HDR_TMP          00000613-R    716    (9)      683    (9)
R80_TBL              00000580-R    665    (9)      1446   (15)
RB730$B_BR           00000036                     #-1431  (15)
RD_BAD_BLK           00001359-R    3876   (45)
RD_HOME_BLK          00001107-R    3424   (38)     1475   (15)
RD_HOME_BLK_X        00001158-R    3447   (38)    #-3445  (38)
RD_IDCREGS           000008EF-R    2348   (26)     2279   (25)     2565  (28)     2717  (29)     2720  (29)
                                                   3827   (44)
RD_IDCREGS_X         00000963-R    2380   (26)    #-2356  (26)   #-2378  (26)
READ                =0000000C                     #-3440  (38)   #-3901  (45)
READH               =00000008                     #-2357  (26)   #-2736  (29)   #-3094  (33)
READ_BUFFER          000002C4-R    591    (8)     #-1484  (15)   #-3241  (35)     3433  (38)   #-3563  (40)
                                                  #-3569  (40)
```

```
READ_HEADERS                        00000FA5-R   3238     (35)
RECAL                              =00000006                       #-2210    (24)    #-2709    (29)
RECAL$M_INPROG                     =00000001                       #-3310    (36)
RECAL$V_INPROG                     =00000000                       #-3014    (32)
REG_ERRPT                           00000000-R   1640     (19)
RESTORE_DRIVE                       00000B60-R   2684     (29)      2292    (25)      2446    (27)
RESTORE_R80                         00000BA4-R   2707     (29)      2691    (29)
RESTORE_RL02                        00000B6F-R   2693     (29)      2691    (29)
RGLIST                              000000A8-R   460      (8)       1662    (19)      1775    (21)      1777    (21)      1787    (21)
                                                                    1789    (21)      1792    (21)      1794    (21)      1796    (21)
                                                                   #-2442    (27)    #-2447    (27)    #-2475    (27)    #-2911    (31)
                                                                   #-2924    (31)    #-3024    (32)    #-3041    (32)
RL02                               =00000001     805      (12)
RL02$T                             =00000001                       #-1768    (21)    #-2691    (29)    #-2734    (29)    #-2799    (30)
                                                                   #-3640    (41)    #-3882    (45)      629     (9)
RL02_HDR_TMP                        00000604-R   708      (9)        647    (9)
RL02_TBL                            000004FC-R   628      (9)       1442    (15)
SECD_ADDR                           00000018-R   434      (8)
SECTION                             00000013-R   797      (12)       97     (3)
SECTOR                              000000C4-R   467      (8)      #-2052    (22)    #-2480    (27)
SEED                                000000F4-R   479      (8)      #-3494    (39)    #-3495    (39)    #-3498    (39)    #-3499    (39)
SEEK                               =00000006                       #-2469    (27)    #-2603    (28)    #-2740    (29)    #-2904    (31)
                                                                   #-3020    (32)
SEEK_TYPE                           000000B0-R   462      (8)
SEP_FUNCTIONAL                     =00000002     66       (2)
SEP_REPAIR                         =00000003     66       (2)        66     (2)
SIZ...                             =00000001     81       (2)        66     (2)        74     (2)        81     (2)
SKIP_BUFFER                         000002CC-R   594      (8)
START_CMD                           00000966-R   2439     (27)      2905    (31)      3021    (32)      3101    (33)      3441    (38)
                                                                    3902    (45)
STATUS_CHK                          00000A22-R   2557     (28)      2906    (31)      3022    (32)      3102    (33)      3442    (38)
                                                                    3903    (45)
STATUS_CHK_X                        00000B54-R   2629     (28)     #-2570    (28)    #-2597    (28)    #-2610    (28)    #-2615    (28)
                                                                   #-2619    (28)    #-2621    (28)    #-2625    (28)    #-2626    (28)
STEP_HDS                            00000CB1-R   2875     (31)
STEP_HDS_X                          00000DBD-R   2937     (31)     #-2885    (31)    #-2912    (31)    #-2918    (31)    #-2928    (31)
ST_GETSTAT                          000009AF-R   2466     (27)      2461    (27)
ST_NOOP                             000009AC-R   2463     (27)      2461    (27)
ST_READ                             000009F2-R   2484     (27)      2461    (27)
ST_READHDR                          000009C6-R   2474     (27)      2461    (27)
ST_READ_NOHDR                       000009F2-R   2485     (27)      2461    (27)
ST_SEEK                             000009B2-R   2468     (27)      2461    (27)
ST_WRITE                            000009CD-R   2477     (27)      2461    (27)
ST_WRTCHK                           000009AC-R   2464     (27)      2461    (27)
SUBSYS_INIT                         00000823-R   2262     (25)      1576    (16)      3426    (38)
SUBSYS_INIT_X                       000008EE-R   2306     (25)     #-2285    (25)    #-2296    (25)    #-2301    (25)    #-2303    (25)
SUMMARY                             00000000-R   1593     (18)        97    (3)
SUMMARY_X                           00000003-R   1595     (18)
SURFACE                             000000C8-R   468      (8)      #-2052    (22)    #-2481    (27)
SYS$CANTIM                          00010268                        1580    (16)      1582    (16)      2286    (25)      2581    (28)
SYS$GETTIM                          00010378                        3493    (39)
SYS$READEF                          000103D0                        1413    (15)      2281    (25)      2576    (28)
SYS$SETIMR                          00010420                        2273    (25)      2573    (28)
SYS_TIME                            0000001C-R   435      (8)      #-3490    (39)      3493    (39)    #-3494    (39)    #-3500    (39)
S_ALL                               0000000F-R   797      (12)       797    (12)
S_DEFAULT                           00000000-R   797      (12)       797    (12)
```

```
S_DRIVE_TBL      00000004-R   427   (8)    #-3768  (43)  #-3770  (43)
S_L$UNIT         00000030-R   439   (8)    #-3769  (43)
S_MANUAL         00000008-R   797   (12)     797   (12)
S_P$ATTN         00000040-R   443   (8)    #-3772  (43)
S_P$UNIT         00000038-R   441   (8)    #-3773  (43)
T.BA             00000124-R   492   (8)    #-1820  (21)  #-2351  (26)
T.BC             00000128-R   493   (8)    #-1829  (21)  #-2352  (26)
T.BR             00000154-R   503   (8)
T.CS             00000120-R   491   (8)    #-1810  (21)    2280  (25)    2293  (25)  #-2298  (25)
                                           #-2350  (26)    2356  (26)    2566  (28)  #-2582  (28)
                                             2599  (28)  #-2611  (28)  #-2623  (28)    2626  (28)
                                             3104  (33)    3554  (40)    3557  (40)
T.DA             0000012C-R   494   (8)    #-1839  (21)  #-1878  (21)  #-1897  (21)  #-2353  (26)
T.DAT            00000150-R   502   (8)
T.HDR            00000140-R   499   (8)      2375  (26)    2886  (31)    3000  (32)    3110  (33)
                                           #-3257  (35)
T.MP             00000130-R   495   (8)    #-1913  (21)  #-1923  (21)  #-2302  (25)  #-2379  (26)
                                           #-2596  (28)  #-2600  (28)  #-2696  (29)    2708  (29)
                                           #-2802  (30)    2804  (30)  #-2816  (30)
T.MP1            00000134-R   496   (8)    #-1948  (21)  #-1957  (21)  #-2369  (26)  #-2370  (26)
                                           #-2372  (26)  #-2376  (26)  #-3115  (33)
T.MP2            00000138-R   497   (8)    #-1965  (21)  #-1972  (21)  #-2370  (26)  #-2373  (26)
                                           #-2377  (26)  #-3116  (33)
T.MP3            0000013C-R   498   (8)    #-1979  (21)  #-2374  (26)  #-3123  (33)
T.PAT            00000148-R   500   (8)    #-1941  (21)  #-2355  (26)  #-3566  (40)
T.POS            0000014C-R   501   (8)    #-1932  (21)  #-2354  (26)  #-3560  (40)
T.VEC            00000158-R   504   (8)
TEMP             00000118-R   489   (8)
TIME_300MS       00000094-R   456   (8)      2273  (25)
TIME_800MS       0000008C-R   455   (8)      2573  (28)
TIME_LEFT        0000009C-R   457   (8)
TIME_OUT         00000080-R   453   (8)      2281  (25)    2282  (25)    2576  (28)    2577  (28)
T_DEVICE         00000023-R   804   (12)
T_NAME           00000058-R   97    (3)      97    (3)
T_R80            00000028-R   805   (12)     805   (12)
T_RL02           00000023-R   805   (12)     805   (12)
UBACLEAR_ERR     00001428-R   1115  (13)     3364  (37)
UBAINIT_ERR      00001438-R   1116  (13)
UBAMAP_ERR       00001406-R   1113  (13)     3370  (37)
UBAPURGE_ERR     00001418-R   1114  (13)     3375  (37)
UNBINIT_ERR      000013F3-R   1112  (13)
USE02_COMMON     000002EC-R   1853  (21)     1850  (21)
USE02_DMA        0000030B-R   1864  (21)     1850  (21)
USE02_GS         000002F3-R   1856  (21)     1850  (21)
USE02_SEEK       000002FA-R   1859  (21)     1850  (21)
USE80_COMMON     0000037A-R   1892  (21)     1889  (21)
USE80_DMA        00000398-R   1901  (21)     1889  (21)
USE80_SEEK       00000381-R   1895  (21)     1889  (21)
VIR_RD_BUF       00000074-R   451   (8)
VIR_WRT_BUF      00000064-R   449   (8)      1585  (16)
VOL_NAME         000002B4-R   582   (8)    #-1493  (15)  #-1494  (15)
WORD_COUNT       000001BC-R   520   (8)    #-2053  (22)  #-2478  (27)  #-2479  (27)  #-3570  (40)
WORK             00000114-R   488   (8)    #-3115  (33)  #-3116  (33)    3120  (33)    3632  (41)
                                           #-3822  (44)  #-3823  (44)    3824  (44)
WRITE_BUFFER     000002BC-R   588   (8)    #-3187  (34)  #-3568  (40)
```

```
                                          +-------------------------+
                                          ! Macros Cross Reference !
                                          +-------------------------+

MACRO              SIZE         DEFINITION          REFERENCES...
-----              ----         ----------          -------------
$ASNPUSH           1            2273     (25)       2273     (25)       2573     (28)
$CANTIM_S          1            1580     (16)       1580     (16)       1582     (16)       2286     (25)       2581     (28)
$D1_ABPROGRAM      1            1441     (15)       1441     (15)       1459     (15)       1516     (15)       2822     (30)       3366     (37)
                                                    3372     (37)       3377     (37)       3830     (44)
$D1_ERR            1            1440     (15)       1440     (15)       1454     (15)       2821     (30)       3364     (37)       3370     (37)
                                                    3375     (37)       3573     (40)       3829     (44)
$D1_ERR_S          1            1440     (15)       1440     (15)       1454     (15)       2821     (30)       3364     (37)       3370     (37)
                                                    3375     (37)       3573     (40)       3829     (44)
$D1_PRINT_S        1            1458     (15)       1458     (15)       1464     (15)       1482     (15)       1505     (15)       1644     (19)
                                                    1645     (19)       1654     (19)       1656     (19)       1658     (19)       1660     (19)
                                                    1661     (19)       1664     (19)       1713     (20)       1714     (20)       1716     (20)
                                                    1717     (20)       1718     (20)       1720     (20)       1798     (21)       1804     (21)
                                                    1813     (21)       1822     (21)       1831     (21)       1869     (21)       1905     (21)
                                                    1916     (21)       1925     (21)       1934     (21)       1943     (21)       1951     (21)
                                                    1952     (21)       1959     (21)       1960     (21)       1967     (21)       1974     (21)
                                                    1981     (21)       1990     (21)       1991     (21)       1992     (21)       1993     (21)
                                                    1998     (21)       2052     (22)       2053     (22)       2054     (22)       2067     (22)
                                                    2071     (22)       3575     (40)
$D1_SEC            1            797      (12)       797      (12)
$DEF               1            3374     (37)
$DEFINI            1            78       (2)        118      (5)        281      (6)        314      (7)        70       (2)        72       (2)
                                                    78       (2)        79       (2)        80       (2)
$DS_ABORT          1            1441     (15)       1441     (15)       1459     (15)       1516     (15)       2822     (30)       3366     (37)
                                                    3372     (37)       3377     (37)       3830     (44)
$DS_ASKLGCL_S      1            1507     (15)       1507     (15)
$DS_BGNCLEAN       1            1563     (16)       1563     (16)
$DS_BGNINIT        1            1407     (15)       1407     (15)
$DS_BGNMESSAGE     1            1641     (19)       1641     (19)       1709     (20)       2046     (22)
$DS_BGNMOD         1            66       (2)        66       (2)
$DS_BGNSUMMARY     1            1593     (18)       1593     (18)
$DS_BITDEF         2            71       (2)        71       (2)
$DS_BNERROR        1            1420     (15)       1420     (15)       1437     (15)       3363     (37)       3369     (37)       3374     (37)
$DS_BPASS0         1            1409     (15)       1409     (15)
$DS_BREAK          1            1524     (15)       1524     (15)       1586     (16)       1595     (18)
$DS_CANWAIT_S      1            3834     (44)       3834     (44)
$DS_CHANNEL_S      1            1520     (15)       1520     (15)       1568     (16)       3362     (37)       3373     (37)
$DS_CHCDEF         1            73       (2)        73       (2)
$DS_CHIDEF         1            74       (2)        74       (2)
$DS_CHMDEF         2            75       (2)        75       (2)
$DS_CKLOOP         1            3365     (37)       3365     (37)       3371     (37)       3376     (37)
$DS_CVTREG_S       1            1655     (19)       1655     (19)       1657     (19)       1811     (21)       1821     (21)       1830     (21)
                                                    1868     (21)       1904     (21)       1914     (21)       1924     (21)       1933     (21)
                                                    1942     (21)       1949     (21)       1958     (21)       1966     (21)       1973     (21)
                                                    1980     (21)
$DS_DEFDEL         1            77       (2)        77       (2)
$DS_DEVTYP         1            805      (12)       805      (12)
$DS_DISPATCH       1            110      (4)        110      (4)
$DS_DSADEF         5            72       (2)        72       (2)
$DS_DSSDEF         8            70       (2)        70       (2)
```

```
$DS_ENDCLEAN       1          1586    (16)     1586    (16)
$DS_ENDINIT        1          1524    (15)     1524    (15)
$DS_ENDMESSAGE     1          1666    (19)     1666    (19)     1721    (20)
$DS_ENDMOD         1          3979    (46)     3979    (46)
$DS_ENDPASS_S      1          1517    (15)     1517    (15)
$DS_ENDSERV        1          3835    (44)     3835    (44)
$DS_ENDSUMMARY     1          1595    (18)     1595    (18)
$DS_ENVDEF         2          66      (2)      66      (2)
$DS_ERRDEF         1          76      (2)      76      (2)
$DS_ERRDEV_S       1          2820    (30)     2820    (30)     3573    (40)
$DS_ERRSYS_S       1          1438    (15)     1438    (15)     1452    (15)     3364    (37)     3370    (37)     3375    (37)
                                               3828    (44)
$DS_GPHARD_S       1          1418    (15)     1418    (15)
$DS_HEADER         4          94      (3)      94      (3)
$DS_HPODEF         2          78      (2)      78      (2)
$DS_INLOOP_S       1          2887    (31)     2887    (31)     2932    (31)     3001    (32)
$DS_PARDEF         1          81      (2)      81      (2)
$DS_PRINTB_S       1          1644    (19)     1644    (19)     1645    (19)     1654    (19)     1656    (19)     1658    (19)
                                               1660    (19)     1661    (19)     1664    (19)     1713    (20)     1714    (20)
                                               1716    (20)     1717    (20)     1718    (20)     1720    (20)     1798    (21)
                                               1804    (21)     1813    (21)     1822    (21)     1831    (21)     1869    (21)
                                               1905    (21)     1916    (21)     1925    (21)     1934    (21)     1943    (21)
                                               1951    (21)     1952    (21)     1959    (21)     1960    (21)     1967    (21)
                                               1974    (21)     1981    (21)     2052    (22)     2053    (22)     2054    (22)
                                               2067    (22)     2071    (22)
$DS_PRINTF_S       1          1456    (15)     1456    (15)     1462    (15)     1481    (15)     1503    (15)     3575    (40)
$DS_PRINTX_S       1          1990    (21)     1990    (21)     1991    (21)     1992    (21)     1993    (21)     1998    (21)
$DS_PROBE_S        1          1434    (15)     1434    (15)
$DS_QADEF          2          69      (2)      1420    (15)     1437    (15)     3363    (37)     3369    (37)     3374    (37)
                                               69      (2)
$DS_RB730_DEF      1          79      (2)      79      (2)
$DS_RELBUF_S       1          1584    (16)     1584    (16)
$DS_SECTION        2          797     (12)     797     (12)
$DS_SETIPL_S       1          1523    (15)     1523    (15)
$DS_SETMAP_S       1          3367    (37)     3367    (37)
$DS_WAITMS_S       1          2213    (24)     2213    (24)     2714    (29)     2744    (29)
$DS_WAITUS_S       1          2272    (25)     2272    (25)     2704    (29)     2719    (29)     2738    (29)
$EQU               1          3374    (37)     1420    (15)     1437    (15)     3363    (37)     3369    (37)     3374    (37)
                                               66      (2)      69      (2)      71      (2)      73      (2)      75      (2)
                                               81      (2)      82      (2)
$EQULS1            1          3374    (37)     1420    (15)     1437    (15)     3363    (37)     3369    (37)     3374    (37)
                                               66      (2)      69      (2)      73      (2)      82      (2)
$EQULST            1          82      (2)      1420    (15)     1437    (15)     3363    (37)     3369    (37)     3374    (37)
                                               66      (2)      69      (2)      73      (2)      82      (2)
$GBLINI            2          66      (2)      1420    (15)     1437    (15)     3363    (37)     3369    (37)     3374    (37)
                                               66      (2)      69      (2)      71      (2)      73      (2)      74      (2)
                                               75      (2)      81      (2)      82      (2)
$GETTIM_S          1          3493    (39)     3493    (39)
$HMBDEF            3          80      (2)      80      (2)
$OFFDEF            1          76      (2)      76      (2)
$PUSHADR           1          1413    (15)     1413    (15)     1440    (15)     1454    (15)     1512    (15)     1521    (15)
                                               1569    (16)     1585    (16)     1655    (19)     1657    (19)     1812    (21)
                                               1821    (21)     1830    (21)     1868    (21)     1904    (21)     1915    (21)
                                               1924    (21)     1933    (21)     1942    (21)     1950    (21)     1958    (21)
                                               1966    (21)     1973    (21)     1980    (21)     2213    (24)     2272    (25)
                                               2273    (25)     2281    (25)     2573    (28)     2576    (28)     2704    (29)
```

```
                                        2714   (29)      2719   (29)      2738   (29)      2744   (29)      2821   (30)
                                        3362   (37)      3364   (37)      3370   (37)      3373   (37)      3375   (37)
                                        3493   (39)      3573   (40)      3829   (44)
$PUSHTWO          1     1580   (16)     1580   (16)      1582   (16)      2286   (25)      2581   (28)
$QIOPUSH          1     2273   (25)     2273   (25)      2573   (28)
$READEF_S         1     1412   (15)     1412   (15)      2281   (25)      2576   (28)
$SETIMR_S         1     2273   (25)     2273   (25)      2573   (28)
$VIELD            1     66     (2)      66     (2)       74     (2)       81     (2)
$VIELD1           1     3374   (37)     66     (2)       74     (2)       81     (2)
CASE              1     1765   (21)     1765   (21)      1842   (21)      1881   (21)      2452   (27)      2688   (29)
                                        2723   (29)      2796   (30)      2805   (30)      3637   (41)
DRV_TBL           3     82     (2)      82     (2)
DRV_TBL_DEF       4     779    (11)     779    (11)
ENRKADEF          1     68     (2)      68     (2)
GET_LUN           1     1572   (16)     1572   (16)
```

```
                                    +--------------------------+
                                    ! Performance indicators !
                                    +--------------------------+

Phase                     Page faults    CPU Time       Elapsed Time
-----                     -----------    --------       ------------
Initialization                     40    00:00:00.24    00:00:00.64
Command processing                919    00:00:01.03    00:00:02.10
Pass 1                           4282    00:00:44.40    00:01:04.98
Symbol table sort                  41    00:00:02.02    00:00:02.25
Pass 2                           1259    00:00:15.89    00:00:32.02
Symbol table output               124    00:00:00.80    00:00:01.17
Psect synopsis output               6    00:00:00.08    00:00:00.11
Cross-reference output            388    00:00:05.57    00:00:10.19
Assembler run totals             7064    00:01:10.07    00:01:53.65

The working set limit was 962 pages.
414993 bytes (811 pages) of virtual memory were used to buffer the intermediate code.
There were 70 pages of symbol table space allocated to hold 1212 non-local and 139 local symbols.
3980 source lines were read in Pass 1, producing 119 object records in Pass 2.
183 pages of virtual memory were used to define 65 macros.

                                   +----------------------------+
                                   ! Macro library statistics !
                                   +----------------------------+

Macro library name                          Macros defined
------------------                          --------------
$1$DUA9:[BOWSER.TAPE]ENRKA.MLB;16                   5
SYS$COMMON:[SYSLIB]LIB.MLB;3                        1
$1$DUA9:[DIAGLIB.LIBRARY.BLISS]DIAG.MLB;1          56
SYS$COMMON:[SYSLIB]STARLET.MLB;3                   19
TOTALS (all libraries)                             81

1177 GETS were required to define 81 macros.

There were no errors, warnings or information messages.

MACRO/LIST/CROSS ENRKA0
```

Table of contents

```
    0000          .TITLE  ENRKA1  IDC DISK FUNCTIONAL DIAGNOSTIC - SECTION 1
    0000          .IDENT  /1.1/
    0000          .LIST   MEB
    0000          .NLIST  CND
00000000          .PSECT  HEADER, PAGE, NOWRT
    0000          .DEFAULT DISPLACEMENT, WORD
    0000
    0000    ;++
    0000    ; COPYRIGHT (C) 1982, 1986
    0000    ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS 01754
    0000    ;
    0000    ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON  A
    0000    ; SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLU-
    0000    ; SION OF THE ABOVE COPYRIGHT NOTICE.  THIS  SOFTWARE,  OR  ANY
    0000    ; OTHER  COPIES  THEREOF, MAY NOT BE PROVIDED OR OTHERWISE MADE
    0000    ; AVAILABLE TO ANY OTHER PERSON EXCEPT FOR USE ON  SUCH  SYSTEM
    0000    ; AND  TO  ONE WHO AGREES TO THESE LICENSE TERMS.  TITLE TO AND
    0000    ; OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN DEC.
    0000    ;
    0000    ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT
    0000    ; NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL
    0000    ; EQUIPMENT CORPORATION.
    0000    ;
    0000    ; DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF
    0000    ; ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
    0000    ;
    0000    ;++
    0000    ; FACILITY: VAX-11 DIAGNOSTIC LIBRARY
    0000    ;
    0000    ; ABSTRACT: THIS SECTION CONTAINS IDC SUBSYSTEM TESTS 1 THRU 15
    0000    ;
    0000    ; ENVIRONMENT: VAX-11 DIAGNOSTIC SUPERVISOR
    0000    ;
    0000    ;--
    0000    ;++
    0000    ; AUTHOR: JOHN L. WALSH
    0000    ;
    0000    ; CREATION DATE:  MARCH 1982
    0000    ;
    0000    ; MODIFIED BY:
    0000    ;       R.L. BOWSER     FEBURARY 1986   V1.1
    0000    ;
    0000    ;--
```

```
    0000            .SBTTL  DECLARATIONS
    0000    ;++
    0000    ; INCLUDE FILES
    0000    ;--
    0000            .LIBRARY /disk$diag:[DIAGLIB.LIBRARY.BLISS]DIAG.MLB/
    0000            .LIBRARY /SYS$LIBRARY:LIB.MLB/
    0000            .LIBRARY /disk$diag:[BOWSER.TAPE]ENRKA.MLB/
    0000
    0000    ;++
    0000    ; MACROS
    0000    ;--
    0000
    0000    ;++
    0000    ; EQUATED SYMBOLS
    0000    ;--
    0000
    0000    ;
    0000    ; CREATED SYMBOLS FOR THE PROGRAM/SUPERVISOR INTERFACE.
    0000    ;
    0000
    0000            $DS_BGNMOD      <SEP_REPAIR>,1
    0000    ;;;     Macro-32 Diagnostic macro library Version 8.2 "DIAG.MLB (233)"
    0000            $DS_BITDEF
    0000            $DS_SECDEF      MANUAL,ALL
    0000            $DS_CHDEF
    0000            ENRKADEF
    0000    ;;;     ENRKA MACRO LIBRARY VERSION     V1.1
    0000            DRV_TBL
    0000            $IDC_REGDEF     LOCAL
    0000            .SAVE   LOCAL_BLOCK
00000000            .RESTORE
    0000            $HDR_LAYOUTDEF  LOCAL
    0000            .SAVE   LOCAL_BLOCK
00000000            .RESTORE
    0000            $ENRKA_SYMDEF   LOCAL
    0000            .SAVE   LOCAL_BLOCK
00000000            .RESTORE
    0000            .NLIST  MEB
    0000            $DS_SBTTL <IDC PRIMARY ADDRESS TEST>,ALIGN=LONG
    0000            .SBTTL  TEST 1:  IDC PRIMARY ADDRESS TEST
    0000            $D2_SBTTL       001,<IDC PRIMARY ADDRESS TEST>,<LONG>
00000000                    .PSECT  TEST_001, LONG, NOWRT
    001C
```

```
                          001C     ;++
                          001C     ;
                          001C     ; TEST DESCRIPTION:
                          001C     ;
                          001C     ; THE PURPOSE OF THIS TEST IS TO INSURE PROPER COMMUNICATION
                          001C     ; BETWEEN THE CPU AND THE IDC CONTROLLER.  THIS IS ACCOMPLISHED
                          001C     ; BY READING EACH OF THE IDC REGISTERS.
                          001C     ;
                          001C     ; TEST STEPS:
                          001C     ;
                          001C     ; 1. PROBE IDC REGISTER
                          001C     ; 2. RETURN TO STEP 1. UNTIL ALL IDC REGISTERS HAVE BEEN ACCESSED
                          001C     ;
                          001C     ; DEBUG:
                          001C     ;
                          001C     ; IF THIS TEST FAILS PROBLEM IS LOCATED IN IDC ADDRESS DECODE LOGIC
                          001C     ;
                          001C     ;--
                          001C             $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
        52  0000'CF   D0  0022             MOVL    BASE_ADDR,R2                       ; GET IDC ADDRESS
                  53  D4  0027             CLRL    R3                                 ; CLEAR INDEX
                          0029     10$:    $DS_PROBE_S -                              ; PROBE DEVICE REGISTER
                          0029               UNIT=           L$UNIT,-                 ; UNIT UNDER TEST
                          0029               ADDRESS=        (R2)[R3],-               ; ADDRESS TO TEST
                          0029               LENGTH=         #4                       ; LONG WORD ACCESS
                          0039             $DS_BNERROR     20$
                          0049             $DS_ERRSYS_S    ,L$UNIT,IDC_ADRFAIL
                          0055     ;;; TEST 1, SUBTEST 0, ERROR 1
                          005C             $DS_CKLOOP      10$
                          0064             $DS_ABORT
FFB8 53   04      18  F1  006B     20$:    ACBL    #<6*4>,#4,R3,10$                   ; TEST ALL REGISTERS
                          0071             $DS_ENDTEST
                          007C             $DS_SBTTL <IDC SECONDARY ADDRESS TEST>,ALIGN=LONG
                          007C             .SBTTL  TEST 2:  IDC SECONDARY ADDRESS TEST
                          007C             $D2_SBTTL       002,<IDC SECONDARY ADDRESS TEST>,<LONG>
                      00000000                     .PSECT  TEST_002, LONG, NOWRT
                          0020
```

```
                             0020  ;++
                             0020  ;
                             0020  ; TEST DESCRIPTION:
                             0020  ;
                             0020  ; THE PURPOSE OF THIS TEST IS TO INSURE PROPER COMMUNICATION
                             0020  ; BETWEEN THE CPU AND THE IDC VIA ITS UNIBUS OR SECONDARY ADDRESS.
                             0020  ; THE ONLY REGISTER AVAILABLE ON THE UNIBUS IS THE LOWER 16 BITS
                             0020  ; OF THE IDCCSR.
                             0020  ;
                             0020  ; TEST STEPS:
                             0020  ;
                             0020  ; 1. ISSUE ADAPTER INIT
                             0020  ; 2. CLEAR UBA STATUS
                             0020  ; 3. PROBE IDC REGISTER
                             0020  ; 4. RETURN TO STEP 2. UNTIL ALL IDC REGISTERS HAVE BEEN ACCESSED
                             0020  ;
                             0020  ; DEBUG:
                             0020  ;
                             0020  ; IF THIS TEST FAILS PROBLEM IS LOCATED IN IDC ADDRESS DECODE LOGIC
                             0020  ; OR WITH UBA ADAPTER LOGIC.
                             0020  ;
                             0020  ;--
                             0020           $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
      52   0000'CF   D0      0026           MOVL    SECD_ADDR,R2                    ; GET SECONDARY ADDRESS
                             002B  5$:      $DS_CHANNEL_S   L$UNIT,#CHC$_INITA      ; INIT UBA
                             0040           $DS_BNERROR     10$
                             0050           $DS_ERRSYS_S    ,L$UNIT,UBAINIT_ERR
                             005C  ;;; TEST 2, SUBTEST 0, ERROR 1
                             0063           $DS_CKLOOP      5$
                             006B           $DS_ABORT
                             0072  10$:     $DS_CHANNEL_S   L$UNIT,#CHC$_CLEAR      ; CLEAR UBA STATUS REG
                             0087           $DS_BNERROR     20$
                             0097           $DS_ERRSYS_S    ,L$UNIT,UBACLEAR_ERR
                             00A3  ;;; TEST 2, SUBTEST 0, ERROR 2
                             00AA           $DS_CKLOOP      10$
                             00B2           $DS_ABORT
                             00B9           $DS_PROBE_S -                           ; PROBE DEVICE REGISTER
                             00B9             UNIT=         L$UNIT,-                ; UNIT UNDER TEST
                             00B9             ADDRESS=      (R2),-                  ; ADDRESS TO TEST
                             00B9             LENGTH=       #2                      ; WORD ACCESS
                             00C8           $DS_BNERROR     20$
                             00D8           $DS_ERRSYS_S    ,L$UNIT,IDC_SADRFAIL
                             00E4  ;;; TEST 2, SUBTEST 0, ERROR 3
                             00EB           $DS_CKLOOP      5$
                             00F4           $DS_ABORT
                             00FB  20$:     $DS_ENDTEST
                             0106           $DS_SBTTL <IDCCSR REG DATA TEST>,ALIGN=LONG
                             0106           .SBTTL  TEST 3:  IDCCSR REG DATA TEST
                             0106           $D2_SBTTL       003,<IDCCSR REG DATA TEST>,<LONG>
                         00000000                   .PSECT  TEST_003, LONG, NOWRT
                             0018
```

```
                              0018      ;++
                              0018      ;
                              0018      ; TEST DESCRIPTION:
                              0018      ;
                              0018      ; THIS TEST WILL CHECK THE DATA INTEGRITY OF THE CONTROL AND STATUS
                              0018      ; REGISTER ON A BIT BY BIT BASIS.  REGISTER WILL BE TESTED FOR HOLDING
                              0018      ; ALL ONES AND ALL ZEROES.  THE IDCCSR IS ALSO TESTED WITH A FLOATING ONES
                              0018      ; AND A FLOATING ZEROES PATTERN.  ONCE THE DATA INTEGRITY HAS BEEN
                              0018      ; VERIFIED THE TEST CHECKS THAT THE IDCCSR CAN BE LOADED USING A READ
                              0018      ; MODIFY WRITE INSTRUCTION.
                              0018      ;
                              0018      ; TEST STEPS:
                              0018      ;
                              0018      ; A. SUBTEST 1
                              0018      ;       1. WRITE IDCCSR WITH ONES
                              0018      ;       2. READ IDCCSR
                              0018      ;       3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
                              0018      ; B. SUBTEST 2
                              0018      ;       1. WRITE !DCCSR WITH ZEROES
                              0018      ;       2. READ IDCCSR
                              0018      ;       3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
                              0018      ; C. SUBTEST 3
                              0018      ;       1. WRITE IDCCSR USING FLOATING ONES PATTERN
                              0018      ;       2. READ IDCCSR
                              0018      ;       3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
                              0018      ;       4. RETURN TO STEP 1. UNTIL ALL BITS HAVE BEEN TESTED
                              0018      ; D. SUBTEST 4
                              0018      ;       1. WRITE IDCCSR WITH FLOATING ZERO PATTERN
                              0018      ;       2. READ IDCCSR
                              0018      ;       3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
                              0018      ;       4. RETURN TO STEP 1. UNTIL ALL BITS HAVE BEEN TESTED
                              0018      ; E. SUBTEST 5
                              0018      ;       1. LOAD IDCCSR WITH ONES USING A BIT SET INSTRUCTION
                              0018      ;       2. READ IDCCSR
                              0018      ;       3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
                              0018      ;
                              0018      ; DEBUG:
                              0018      ;
                              0018      ; IF THIS TEST FAILS THE IDCCSR REGISTER LOGIC SHOULD BE CHECKED
                              0018      ; AND ALSO THE REGISTER DECODE LOGIC.
                              0018      ;
                              0018      ;--
                              0018              $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
 0000'CF   00000040 8F    D0  001E              MOVL    #PRT$M_NONE,RGLIST              ; ONLY DUMP REG THAT FAILS
                              0027              $DS_BGNSUB
                              0030              STARS
                              0030      ;;*****************************************************************
                              0030      ;++
                              0030      ; CHECK THE IDCCSR REGISTER FOR ANY BITS STUCK AT A ZERO
                              0030      ;--
        52   0000'CF      D0  0030              MOVL    BASE_ADDR,R2                    ; R2 <-- GETS IDC BASE ADDRESS
    62   1A4F03CE 8F      D0  0035      10$:    MOVL    #^X1A4F03CE,IDCCSR(R2)          ; WRITE IDCCSR WITH 1'S
 0000'CF   1A4003CE 8F    D0  003C              MOVL    #^X1A4003CE,E.CS                ; SET EXPECTED IDCCSR
      0000'CF    62       D0  0045              MOVL    IDCCSR(R2),T.CS                 ; READ IDCCSR REGISTER
 0000'CF   C4B0FC31 8F    CA  004A              BICL2   #^XC4B0FC31,T.CS                ; CLEAR ERROR BITS-RB
   0000'CF   0000'CF      D1  0053              CMPL    E.CS,T.CS                       ; TEST THAT IDCCSR = EXPECTED VALUE
```

```
                     29   13  005A       BEQL    20$                                  ; BRANCH IF IT DOES
                              005C       $DS_ERRHARD_S    ,L$UNIT,MSG3_RG_ONE,-
                              005C                        REG_ERRPT,#MSG6_CS_FAIL,-
                              005C                        E.CS,T.CS,#CSR_BITS
                              007E  ;;; TEST 3, SUBTEST 1, ERROR 1
                              0085  20$:  $DS_CKLOOP      10$
                              008D       $DS_ENDSUB
                              0096
                              0096       $DS_BGNSUB
                              009F       STARS
                              009F  ;;****************************************************************
                              009F  ;++
                              009F  ;TEST FOR ANY IDCCSR BITS STUCK AT ONE
                              009F  ;--
                              009F
           62    80 8F   9A   009F  10$:  MOVZBL  #^X80,IDCCSR(R2)                   ; CLEAR ALL EXCEPT CRDY
      0000'CF    80 8F   9A   00A3       MOVZBL  #CSR$M_CRDY,E.CS                     ; SET EXPECTED IDCCSR REG
        0000'CF     62   D0   00A9       MOVL    IDCCSR(R2),T.CS                      ; READ IDCCSR REG BACK
0000'CF   C4B0FC31 8F   CA   00AE       BICL2   #^XC4B0FC31,T.CS                      ; CLEAR ERROR BITS-RB
   0000'CF    0000'CF   D1   00B7       CMPL    E.CS,T.CS                             ; EXPECTED = ACTUAL ?
                 29     13   00BE       BEQL    20$                                   ; BRANCH IF YES
                              00C0       $DS_ERRHARD_S    ,L$UNIT,MSG2_RG_ZERO,-
                              00C0                        REG_ERRPT,#MSG6_CS_FAIL,-
                              00C0                        E.CS,T.CS,#CSR_BITS
                              00E2  ;;; TEST 3, SUBTEST 2, ERROR 1
                              00E9  20$:  $DS_CKLOOP      10$
                              00F1       $DS_ENDSUB
                              00FA
                              00FA       $DS_BGNSUB
                              0103       STARS
                              0103  ;;****************************************************************
                              0103  ;++
                              0103  ; TEST IDCCSR WITH FLOATING ONE'S PATTERN
                              0103  ;--
                              0103
              53    02   D0   0103       MOVL    #BIT1,R3                             ; SET INITIAL PATTERN
              54    53   D0   0106  10$:  MOVL    R3,R4                                ; COPY R3 INTO WORK LOCATION
           54    80 8F   88   0109       BISB2   #CSR$M_CRDY,R4                       ; MAKE SURE WE WRITE CRDY BIT
     54   05BFFC31 8F   CA   010D       BICL2   #^X5BFFC31,R4                         ; CLEAR UNWRITEABLE BITS
              62    54   D0   0114       MOVL    R4,IDCCSR(R2)                        ; WRITE IDCCSR REGISTER
         0000'CF    54   D0   0117       MOVL    R4,E.CS                              ; SET IDCCSR EXPECTED VALUE
         0000'CF    62   D0   011C       MOVL    IDCCSR(R2),T.CS                      ; READ IDCCSR REGISTER
0000'CF   C4BFFC31 8F   CA   0121       BICL2   #^XC4BFFC31,T.CS                      ; CLEAR ERROR BITS-RB
   0000'CF    0000'CF   D1   012A       CMPL    E.CS,T.CS                             ; ACTUAL = EXPECTED ?
                 29     13   0131       BEQL    20$                                   ; BRANCH IF YES
                              0133       $DS_ERRHARD_S    ,L$UNIT,MSG4_RG_FLT1,-
                              0133                        REG_ERRPT,#MSG6_CS_FAIL,-
                              0133                        E.CS,T.CS,#CSR_BITS
                              0155  ;;; TEST 3, SUBTEST 3, ERROR 1
                              015C  20$:  $DS_CKLOOP      10$
        53    53    01   9C   0164       ROTL    #1,R3,R3                             ; ROTATE TO TEST NEXT BIT
           9A 53    1E   E1   0168       BBC     #CSR$V_FMT+1,R3,10$                  ; IF ALL BITS NOT DONE GOTO 10$
                              016C  30$:  $DS_ENDSUB
                              0175
                              0175       $DS_BGNSUB
                              017E       STARS
```

```
                               017E        ;;*************************************************************
                               017E        ;++
                               017E        ; TEST IDCCSR WITH A FLOATING 0'S PATTERN
                               017E        ;--
                               017E
             53     02    D0   017E                MOVL    #BIT1,R3                        ; R3 <-- STARTING TEST PATTERN
   54   1FFFFFFF 8F    53    CD   0181        10$:    XORL3   R3,#^X1FFFFFFF,R4               ; MAKE R4 HAVING 1 ZERO BIT
        54   C5BFFC31 8F    CA   0189                BICL2   #^XC5BFFC31,R4                  ; CLEAR ANY UNWRITABLE BITS
        54   00000080 8F    C8   0190                BISL2   #CSR$M_CRDY,R4                  ; MAKE SURE CRDY ISN'T A ZERO
             62     54    D0   0197                MOVL    R4,IDCCSR(R2)                   ; WRITE IDCCSR REGISTER
        0000'CF     54    D0   019A                MOVL    R4,E.CS                         ; SET EXPECTED IDCCSR REGISTER
        0000'CF     62    D0   019F                MOVL    IDCCSR(R2),T.CS                 ; READ IDCCSR REGISTER
0000'CF  C4B0FC31 8F    CA   01A4                BICL2   #^XC4B0FC31,T.CS                ; CLEAR ERROR BITS-RB
   0000'CF    0000'CF    D1   01AD                CMPL    E.CS,T.CS                       ; ACTUAL = EXPECTED ?
                   29    13   01B4                BEQL    20$                             ; BRANCH IF YES
                               01B6                $DS_ERRHARD_S    ,L$UNIT,MSG5_RG_FLT0,-
                               01B6                                 REG_ERRPT,#MSG6_CS_FAIL,-
                               01B6                                 E.CS,T.CS,#CSR_BITS
                               01D8        ;;; TEST 3, SUBTEST 4, ERROR 1
                               01DF        20$:    $DS_CKLOOP       10$
        53     53     01    9C   01E7                ROTL    #1,R3,R3                        ; GET NEXT BIT POSITION TO TEST
           92 53      1E    E1   01EB                BBC     #CSR$V_FMT+1,R3,10$             ; IF ALL BITS NOT DONE GOTO 10$
                               01EF        30$:    $DS_ENDSUB
                               01F8
                               01F8                $DS_BGNSUB
                               0201                STARS
                               0201        ;;*************************************************************
                               0201        ;++
                               0201        ; TEST IDCCSR WITH A READ MODIFY WRITE INSTRUCTION.
                               0201        ;--
                               0201
          62     80 8F    9A   0201        10$:    MOVZBL  #CSR$M_CRDY,IDCCSR(R2)          ; CLEAR IDCCSR RECISTER
     62   1A4F03CE 8F    C8   0205                BISL2   #^X1A4F03CE,IDCCSR(R2)          ; WRITE ALL POSSIBLE IDCCSR BITS
0000'CF   1A4003CE 8F    D0   020C                MOVL    #^X1A4003CE,E.CS                ; SET EXPECTED IDCCSR VALUE
        0000'CF     62    D0   0215                MOVL    IDCCSR(R2),T.CS                 ; READ IDCCSR REGISTER
0000'CF   C4B0FC31 8F    CA   021A                BICL2   #^XC4B0FC31,T.CS                ; CLEAR ERROR BITS-RB
   0000'CF    0000'CF    D1   0223                CMPL    E.CS,T.CS                       ; ACTUAL = EXPECTED ?
                   29    13   022A                BEQL    20$                             ; BRANCH IF YES
                               022C                $DS_ERRHARD_S    ,L$UNIT,MSG9_RG_RMW,-
                               022C                                 REG_ERRPT,#MSG6_CS_FAIL,-
                               022C                                 E.CS,T.CS,#CSR_BITS
                               024E        ;;; TEST 3, SUBTEST 5, ERROR 1
     62   000F0080 8F    D0   0255        20$:    MOVL    #^XF0080,IDCCSR(R2)             ; RESET ALL BITS IN CSR TO -
                               025C                                                        ; PREVENT SPURIOUS INTERUPTS
                               025C                $DS_CKLOOP       10$
                               0264                $DS_ENDSUB
                               026D                $DS_ENDTEST
                               0278                $DS_SBTTL <IDCBAR REG DATA TEST>,ALIGN=LONG
                               0278                .SBTTL  TEST 4:  IDCBAR REG DATA TEST
                               0278                $D2_SBTTL        004,<IDCBAR REG DATA TEST>,<LONG>
                           00000000                        .PSECT  TEST_004, LONG, NOWRT
                               0018
```

```
                              0018   306 ;++
                              0018   307 ;
                              0018   308 ; TEST DESCRIPTION:
                              0018   309 ;
                              0018   310 ; THIS TEST WILL CHECK THE DATA INTEGRITY OF THE BUS ADDRESS
                              0018   311 ; REGISTER ON A BIT BY BIT BASIS.  REGISTER WILL BE TESTED FOR HOLDING
                              0018   312 ; ALL ONES AND ALL ZEROES.  THE IDCBAR IS ALSO TESTED WITH A FLOATING ONES
                              0018   313 ; AND A FLOATING ZEROES PATTERN.  ONCE THE DATA INTEGRITY HAS BEEN
                              0018   314 ; VERIFIED THE TEST CHECKS THAT THE IDCBAR CAN BE LOADED USING A READ
                              0018   315 ; MODIFY WRITE INSTRUCTION.
                              0018   316 ;
                              0018   317 ; TEST STEPS:
                              0018   318 ;
                              0018   319 ; A. SUBTEST 1
                              0018   320 ;        1. WRITE IDCBAR WITH ONES
                              0018   321 ;        2. READ IDCBAR
                              0018   322 ;        3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
                              0018   323 ; B. SUBTEST 2
                              0018   324 ;        1. WRITE IDCBAR WITH ZEROES
                              0018   325 ;        2. READ IDCBAR
                              0018   326 ;        3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
                              0018   327 ; C. SUBTEST 3
                              0018   328 ;        1. WRITE IDCBAR USING FLOATING ONES PATTERN
                              0018   329 ;        2. READ IDCBAR
                              0018   330 ;        3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
                              0018   331 ;        4. RETURN TO STEP 1. UNTIL ALL BITS HAVE BEEN TESTED
                              0018   332 ; D. SUBTEST 4
                              0018   333 ;        1. WRITE IDCBAR WITH FLOATING ZERO PATTERN
                              0018   334 ;        2. READ IDCBAR
                              0018   335 ;        3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
                              0018   336 ;        4. RETURN TO STEP 1. UNTIL ALL BITS HAVE BEEN TESTED
                              0018   337 ; E. SUBTEST 5
                              0018   338 ;        1. LOAD IDCBAR WITH ONES USING A BIT SET INSTRUCTION
                              0018   339 ;        2. READ IDCBAR
                              0018   340 ;        3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
                              0018   341 ;
                              0018   342 ; DEBUG:
                              0018   343 ;
                              0018   344 ; IF THIS TEST FAILS THE IDCBAR REGISTER LOGIC SHOULD BE CHECKED
                              0018   345 ; AND ALSO THE REGISTER DECODE LOGIC.
                              0018   346 ;
                              0018   347 ;--
                              0018   348          $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
      0000'CF   00000040 8F   D0  001E   349          MOVL    #PRT$M_NONE,RGLIST              ; ONLY DUMP REG THAT FAILS
                              0027   350          $DS_BGNSUB
                              0030   351          STARS
                              0030       ;;**************************************************************
                              0030   352 ;++
                              0030   353 ; CHECK THE IDCBAR REGISTER FOR ANY BITS STUCK AT A ZERO
                              0030   354 ;--
            52   0000'CF   D0  0030   355          MOVL    BASE_ADDR,R2                    ; R2 <-- GETS IDC BASE ADDRESS
       04 A2   0003FFFF 8F   D0  0035   356 10$:     MOVL    #^X3FFFF,IDCBAR(R2)             ; WRITE IDCBAR WITH 1'S
     0000'CF   0003FFFF 8F   D0  003D   357          MOVL    #^X3FFFF,E.BA                   ; SET EXPECTED IDCBAR
          0000'CF   04 A2   D0  0046   358          MOVL    IDCBAR(R2),T.BA                 ; READ IDCBAR REGISTER
       0000'CF   0000'CF   D1  004C   359          CMPL    E.BA,T.BA                       ; TEST THAT IDCBAR = EXPECTED VALUE
                      23   13  0053   360          BEQL    20$                             ; BRANCH IF IT DOES
```

```
                                0055        $DS_ERRHARD_S   ,L$UNIT,MSG3_RG_ONE,-
                                0055                        REG_ERRPT,#MSG7_BA_FAIL,-
                                0055                        E.BA,T.BA
                                0071   ;;; TEST 4, SUBTEST 1, ERROR 1
                                0078   20$:    $DS_CKLOOP      10$
                                0080           $DS_ENDSUB
                                0089
                                0089           $DS_BGNSUB
                                0092           STARS
                                0092   ;;*****************************************************************
                                0092   ;++
                                0092   ;TEST FOR ANY IDCBAR BITS STUCK AT ONE
                                0092   ;--
                                0092
                04 A2     D4    0092   10$:    CLRL    IDCBAR(R2)                      ; CLEAR ALL IDCBAR BITS
             0000'CF      D4    0095           CLRL    E.BA                            ; SET EXPECTED IDCBAR REG
    0000'CF     04 A2     D0    0099           MOVL    IDCBAR(R2),T.BA                 ; READ IDCBAR REG BACK
  0000'CF    0000'CF      D1    009F           CMPL    E.BA,T.BA                       ; EXPECTED = ACTUAL ?
                   23     13    00A6           BEQL    20$                             ; BRANCH IF YES
                                00A8           $DS_ERRHARD_S   ,L$UNIT,MSG2_RG_ZERO,-
                                00A8                           REG_ERRPT,#MSG7_BA_FAIL,-
                                00A8                           E.BA,T.BA
                                00C4   ;;; TEST 4, SUBTEST 2, ERROR 1
                                00CB   20$:    $DS_CKLOOP      10$
                                00D3           $DS_ENDSUB
                                00DC
                                00DC           $DS_BGNSUB
                                00E5           STARS
                                00E5   ;;*****************************************************************
                                00E5   ;++
                                00E5   ; TEST IDCBAR WITH FLOATING ONE'S PATTERN
                                00E5   ;--
                53      01    D0    00E5           MOVL    #BIT0,R3                        ; SET INITIAL PATTERN
             04 A2      53    D0    00E8   10$:    MOVL    R3,IDCBAR(R2)                   ; WRITE IDCBAR REGISTER
           0000'CF      53    D0    00EC           MOVL    R3,E.BA                         ; SET IDCBAR EXPECTED VALUE
    0000'CF     04 A2     D0    00F1           MOVL    IDCBAR(R2),T.BA                 ; READ IDCBAR REGISTER
  0000'CF    0000'CF      D1    00F7           CMPL    E.BA,T.BA                       ; ACTUAL = EXPECTED ?
                   23     13    00FE           BEQL    20$                             ; BRANCH IF YES
                                0100           $DS_ERRHARD_S   ,L$UNIT,MSG4_RG_FLT1,-
                                0100                           REG_ERRPT,#MSG7_BA_FAIL,-
                                0100                           E.BA,T.BA
                                011C   ;;; TEST 4, SUBTEST 3, ERROR 1
                                0123   20$:    $DS_CKLOOP      10$
          53    53      01    9C    012B           ROTL    #1,R3,R3                        ; ROTATE TO TEST NEXT BIT
             B5 53      12    E1    012F           BBC     #18,R3,10$                      ; CONTINUE UNTIL ALL BITS TESTED
                                0133           $DS_ENDSUB
                                013C
                                013C           $DS_BGNSUB
                                0145           STARS
                                0145   ;;*****************************************************************
                                0145   ;++
                                0145   ; TEST IDCBAR WITH A FLOATING 0'S PATTERN
                                0145   ;--
                                0145
                53      01    D0    0145           MOVL    #BIT0,R3                        ; R3 <-- STARTING TEST PATTERN
  54    53    0003FFFF 8F     CD    0148   10$:    XORL3   #^X3FFFF,R3,R4                  ; MAKE R4 HAVING 1 ZERO BIT
```

```
         04 A2    54   D0  0150   411           MOVL    R4,IDCBAR(R2)                    ; WRITE IDCBAR REGISTER
       0000'CF    54   D0  0154   412           MOVL    R4,E.BA                          ; SET EXPECTED IDCBAR REGISTER
  0000'CF    04 A2     D0  0159   413           MOVL    IDCBAR(R2),T.BA                  ; READ IDCBAR REAIGTER
  0000'CF  0000'CF     D1  015F   414           CMPL    E.BA,T.BA                        ; ACTUAL = EXPECTED ?
                  23   13  0166   415           BEQL    20$                              ; BRANCH IF YES
                           0168   416           $DS_ERRHARD_S     ,L$UNIT,MSG5_RG_FLT0,-
                           0168   417                             REG_ERRPT,#MSG7_BA_FAIL,-
                           0168   418                             E.BA,T.BA
                           0184         ;;; TEST 4, SUBTEST 4, ERROR 1
                           018B   419 20$:      $DS_CKLOOP        10$
       53    53   01   9C  0193   420           ROTL    #1,R3,R3                         ; GET NEXT BIT POSITION TO TEST
          AD 53   12   E1  0197   421           BBC     #18,R3,10$                       ; CONTINUE UNTIL ALL BITS TESTED
                           019B   422           $DS_ENDSUB
                           01A4   423
                           01A4   424           $DS_BGNSUB
                           01AD   425           STARS
                           01AD         ;;*******************************************************************
                           01AD   426   ;++
                           01AD   427   ; TEST IDCBAR WITH A READ MODIFY WRITE INSTRUCTION.
                           01AD   428   ;--
                           01AD   429
                   04 A2   D4  01AD   430 10$:  CLRL    IDCBAR(R2)                       ; CLEAR IDCBAR REGISTER
   04 A2    0003FFFF 8F    C8  01B0   431       BISL2   #^X3FFFF,IDCBAR(R2)              ; WRITE ALL POSSIBLE IDCBAR BITS
 0000'CF    0003FFFF 8F    D0  01B8   432       MOVL    #^X3FFFF,E.BA                    ; SET EXPECTED IDCBAR VALUE
      0000'CF    04 A2     D0  01C1   433       MOVL    IDCBAR(R2),T.BA                  ; READ IDCBAR REGISTER
   0000'CF    0000'CF      D1  01C7   434       CMPL    E.BA,T.BA                        ; ACTUAL = EXPECTED ?
                      23   13  01CE   435       BEQL    20$                              ; BRANCH IF YES
                               01D0   436       $DS_ERRHARD_S     ,L$UNIT,MSG9_RG_RMW,-
                               01D0   437                         REG_ERRPT,#MSG7_BA_FAIL,-
                               01D0   438                         E.BA,T.BA
                               01EC         ;;; TEST 4, SUBTEST 5, ERROR 1
                               01F3   439 20$:  $DS_CKLOOP        10$
                               01FB   440       $DS_ENDSUB
                               0204   441       $DS_ENDTEST
                               020F   442       $DS_SBTTL <IDCCNT REG DATA TEST>,ALIGN=LONG
                               020F             .SBTTL  TEST 5:  IDCCNT REG DATA TEST
                               020F             $D2_SBTTL         005,<IDCCNT REG DATA TEST>,<LONG>
                           00000000                     .PSECT  TEST_005, LONG, NOWRT
                               0018
```

```
                                  0018  ;++
                                  0018  ;
                                  0018  ; TEST DESCRIPTION:
                                  0018  ;
                                  0018  ; THIS TEST WILL CHECK THE DATA INTEGRITY OF THE BYTE COUNT
                                  0018  ; REGISTER ON A BIT BY BIT BASIS.  REGISTER WILL BE TESTED FOR HOLDING
                                  0018  ; ALL ONES AND ALL ZEROES.  THE IDCCNT IS ALSO TESTED WITH A FLOATING ONES
                                  0018  ; AND A FLOATING ZEROES PATTERN.  ONCE THE DATA INTEGRITY HAS BEEN
                                  0018  ; VERIFIED THE TEST CHECKS THAT THE IDCCNT CAN BE LOADED USING A READ
                                  0018  ; MODIFY WRITE INSTRUCTION.
                                  0018  ;
                                  0018  ; TEST STEPS:
                                  0018  ;
                                  0018  ; A. SUBTEST 1
                                  0018  ;        1. WRITE IDCCNT WITH ONES
                                  0018  ;        2. READ IDCCNT
                                  0018  ;        3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
                                  0018  ; B. SUBTEST 2
                                  0018  ;        1. WRITE IDCCNT WITH ZEROES
                                  0018  ;        2. READ IDCCNT
                                  0018  ;        3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
                                  0018  ; C. SUBTEST 3
                                  0018  ;        1. WRITE IDCCNT USING FLOATING ONES PATTERN
                                  0018  ;        2. READ IDCCNT
                                  0018  ;        3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
                                  0018  ;        4. RETURN TO STEP 1. UNTIL ALL BITS HAVE BEEN TESTED
                                  0018  ; D. SUBTEST 4
                                  0018  ;        1. WRITE IDCCNT WITH FLOATING ZERO PATTERN
                                  0018  ;        2. READ IDCCNT
                                  0018  ;        3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
                                  0018  ;        4. RETURN TO STEP 1. UNTIL ALL BITS HAVE BEEN TESTED
                                  0018  ; E. SUBTEST 5
                                  0018  ;        1. LOAD IDCCNT WITH ONES USING A BIT SET INSTRUCTION
                                  0018  ;        2. READ IDCCNT
                                  0018  ;        3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
                                  0018  ;
                                  0018  ; DEBUG:
                                  0018  ;
                                  0018  ; IF THIS TEST FAILS THE IDCCNT REGISTER LOGIC SHOULD BE CHECKED
                                  0018  ; AND ALSO THE REGISTER DECODE LOGIC.
                                  0018  ;
                                  0018  ;--
                                  0018          $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
 0000'CF    00000040 8F     D0    001E          MOVL    #PRT$M_NONE,RGLIST              ; ONLY DUMP REG THAT FAILS
                                  0027          $DS_BGNSUB
                                  0030          STARS
                                  0030  ;;*************************************************************************
                                  0030  ;++
                                  0030  ; CHECK THE IDCCNT REGISTER FOR ANY BITS STUCK AT A ZERO
                                  0030  ;--
          52    0000'CF     D0    0030          MOVL    BASE_ADDR,R2                    ; R2 <-- GETS IDC BASE ADDRESS
      08 A2    0000FFFF 8F  D0    0035  10$:    MOVL    #^XFFFF,IDCCNT(R2)              ; WRITE IDCCNT WITH 1'S
     0000'CF       FFFF 8F  3C    003D          MOVZWL  #^XFFFF,E.BC                    ; SET EXPECTED IDCCNT
       0000'CF     08 A2    D0    0044          MOVL    IDCCNT(R2),T.BC                 ; READ IDCCNT REGISTER
     0000'CF     0000'CF    D1    004A          CMPL    E.BC,T.BC                       ; TEST THAT IDCCNT = EXPECTED VALUE
                      23    13    0051          BEQL    20$                             ; BRANCH IF IT DOES
```

```
                                0053   499            $DS_ERRHARD_S    ,L$UNIT,MSG3_RG_ONE,-
                                0053   500                             REG_ERRPT,#MSG7A_BC_FAIL,-
                                0053   501                             E.BC,T.BC
                                006F          ;;; TEST 5, SUBTEST 1, ERROR 1
                                0076   502 20$:    $DS_CKLOOP      10$
                                007E   503            $DS_ENDSUB
                                0087   504
                                0087   505            $DS_BGNSUB
                                0090   506            STARS
                                0090          ;;*********************************************************************
                                0090   507 ;++
                                0090   508 ;TEST FOR ANY IDCCNT BITS STUCK AT ONE
                                0090   509 ;--
                                0090   510
                08 A2     D4    0090   511 10$:    CLRL    IDCCNT(R2)                     ; CLEAR ALL IDCCNT BITS
             0000'CF      D4    0093   512            CLRL    E.BC                        ; SET EXPECTED IDCCNT REG
      0000'CF    08 A2    D0    0097   513            MOVL    IDCCNT(R2),T.BC             ; READ IDCCNT REG BACK
   0000'CF    0000'CF     D1    009D   514            CMPL    E.BC,T.BC                   ; EXPECTED = ACTUAL ?
                     23   13    00A4   515            BEQL    20$                         ; BRANCH IF YES
                                00A6   516            $DS_ERRHARD_S    ,L$UNIT,MSG2_RG_ZERO,-
                                00A6   517                             REG_ERRPT,#MSG7A_BC_FAIL,-
                                00A6   518                             E.BC,T.BC
                                00C2          ;;; TEST 5, SUBTEST 2, ERROR 1
                                00C9   519 20$:    $DS_CKLOOP      10$
                                00D1   520            $DS_ENDSUB
                                00DA   521
                                00DA   522            $DS_BGNSUB
                                00E3   523            STARS
                                00E3          ;;*********************************************************************
                                00E3   524 ;++
                                00E3   525 ; TEST IDCCNT WITH FLOATING ONE'S PATTERN
                                00E3   526 ;--
                53   02   D0    00E3   527            MOVL    #BIT1,R3                    ; SET INITIAL PATTERN
             08 A2   53   D0    00E6   528 10$:    MOVL    R3,IDCCNT(R2)                  ; WRITE IDCCNT REGISTER
          0000'CF    53   D0    00EA   529            MOVL    R3,E.BC                     ; SET IDCCNT EXPECTED VALUE
      0000'CF    08 A2    D0    00EF   530            MOVL    IDCCNT(R2),T.BC             ; READ IDCCNT REGISTER
   0000'CF    0000'CF     D1    00F5   531            CMPL    E.BC,T.BC                   ; ACTUAL = EXPECTED ?
                     23   13    00FC   532            BEQL    20$                         ; BRANCH IF YES
                                00FE   533            $DS_ERRHARD_S    ,L$UNIT,MSG4_RG_FLT1,-
                                00FE   534                             REG_ERRPT,#MSG7A_BC_FAIL,-
                                00FE   535                             E.BC,T.BC
                                011A          ;;; TEST 5, SUBTEST 3, ERROR 1
                                0121   536 20$:    $DS_CKLOOP      10$
             53   53   01   9C  0129   537            ROTL    #1,R3,R3                    ; ROTATE TO TEST NEXT BIT
             B5   53   10   E1  012D   538            BBC     #16,R3,10$                  ; CONTINUE UNTIL ALL BITS TESTED
                                0131   539            $DS_ENDSUB
                                013A   540
                                013A   541            $DS_BGNSUB
                                0143   542            STARS
                                0143          ;;*********************************************************************
                                0143   543 ;++
                                0143   544 ; TEST IDCCNT WITH A FLOATING 0'S PATTERN
                                0143   545 ;--
                                0143   546
                53   02   D0    0143   547            MOVL    #BIT1,R3                    ; R3 <-- STARTING TEST PATTERN
   54    53    0000FFFF 8F   CD 0146   548 10$:    XORL3   #^XFFFF,R3,R4               ; MAKE R4 HAVING 1 ZERO BIT
```

```
              08 A2    54    D0  014E    549            MOVL     R4,IDCCNT(R2)                          ; WRITE IDCCNT REGISTER
            0000'CF    54    D0  0152    550            MOVL     R4,E.BC                                ; SET EXPECTED IDCCNT REGISTER
      0000'CF   08 A2    D0  0157    551            MOVL     IDCCNT(R2),T.BC                        ; READ IDCCNT REAIGTER
    0000'CF   0000'CF    D1  015D    552            CMPL     E.BC,T.BC                              ; ACTUAL = EXPECTED ?
                    23    13  0164    553            BEQL     20$                                    ; BRANCH IF YES
                              0166    554            $DS_ERRHARD_S    ,L$UNIT,MSG5_RG_FLT0,-
                              0166    555                             REG_ERRPT,#MSG7A_BC_FAIL,-
                              0166    556                             E.BC,T.BC
                              0182        ;;; TEST 5, SUBTEST 4, ERROR 1
                              0189    557 20$:      $DS_CKLOOP       10$
           53   53    01    9C  0191    558            ROTL     #1,R3,R3                               ; GET NEXT BIT POSITION TO TEST
              AD 53    10    E1  0195    559            BBC      #16,R3,10$                             ; CONTINUE UNTIL ALL BITS TESTED
                              0199    560            $DS_ENDSUB
                              01A2    561
                              01A2    562            $DS_BGNSUB
                              01AB    563            STARS
                              01AB        ;;*******************************************************************
                              01AB    564 ;++
                              01AB    565 ; TEST IDCCNT WITH A READ MODIFY WRITE INSTRUCTION.
                              01AB    566 ;--
                              01AB    567
                 08 A2    D4  01AB    568 10$:      CLRL     IDCCNT(R2)                             ; CLEAR IDCCNT REGISTER
    08 A2    0000FFFF 8F    C8  01AE    569            BISL2    #^XFFFF,IDCCNT(R2)                     ; WRITE ALL POSSIBLE IDCCNT BITS
      0000'CF    FFFF 8F    3C  01B6    570            MOVZWL   #^XFFFF,E.BC                           ; SET EXPECTED IDCCNT VALUE
        0000'CF    08 A2    D0  01BD    571            MOVL     IDCCNT(R2),T.BC                        ; READ IDCCNT REGISTER
    0000'CF     0000'CF    D1  01C3    572            CMPL     E.BC,T.BC                              ; ACTUAL = EXPECTED ?
                    23    13  01CA    573            BEQL     20$                                    ; BRANCH IF YES
                              01CC    574            $DS_ERRHARD_S    ,L$UNIT,MSG9_RG_RMW,-
                              01CC    575                             REG_ERRPT,#MSG7A_BC_FAIL,-
                              01CC    576                             E.BC,T.BC
                              01E8        ;;; TEST 5, SUBTEST 5, ERROR 1
                              01EF    577 20$:      $DS_CKLOOP       10$
                              01F7    578            $DS_ENDSUB
                              0200    579            $DS_ENDTEST
                              020B    580            $DS_SBTTL <IDCDAR REG DATA TEST>,ALIGN=LONG
                              020B                   .SBTTL  TEST 6:  IDCDAR REG DATA TEST
                              020B                   $D2_SBTTL        006,<IDCDAR REG DATA TEST>,<LONG>
                          00000000                           .PSECT  TEST_006, LONG, NOWRT
                              0018
```

```
                                 0018  ;++
                                 0018  ;
                                 0018  ; TEST DESCRIPTION:
                                 0018  ;
                                 0018  ; THIS TEST WILL CHECK THE DATA INTEGRITY OF THE DISK ADDRESS
                                 0018  ; REGISTER ON A BIT BY BIT BASIS.  REGISTER WILL BE TESTED FOR HOLDING
                                 0018  ; ALL ONES AND ALL ZEROES.  THE IDCDAR IS ALSO TESTED WITH A FLOATING ONES
                                 0018  ; AND A FLOATING ZEROES PATTERN.  ONCE THE DATA INTEGRITY HAS BEEN
                                 0018  ; VERIFIED THE TEST CHECKS THAT THE IDCDAR CAN BE LOADED USING A READ
                                 0018  ; MODIFY WRITE INSTRUCTION. THE SUBTESTS THREE AND FOUR USE THE MULTI
                                 0018  ; PURPOSE REGISTER TO WRITE THE IDCDAR REGISTER SO THAT THE IDC
                                 0018  ; MICRO CODE WILL NOT PERFORM ANY MODIFICATIONS ON THE DATA
                                 0018  ;
                                 0018  ; TEST STEPS:
                                 0018  ;
                                 0018  ; A. SUBTEST 1
                                 0018  ;         1. WRITE IDCDAR WITH ONES
                                 0018  ;         2. READ IDCDAR
                                 0018  ;         3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
                                 0018  ; B. SUBTEST 2
                                 0018  ;         1. WRITE IDCDAR WITH ZEROES
                                 0018  ;         2. READ IDCDAR
                                 0018  ;         3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
                                 0018  ; C. SUBTEST 3
                                 0018  ;         1. WRITE IDCMPR USING FLOATING ONES PATTERN
                                 0018  ;         2. READ IDCDAR
                                 0018  ;         3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
                                 0018  ;         4. RETURN TO STEP 1. UNTIL ALL BITS HAVE BEEN TESTED
                                 0018  ; D. SUBTEST 4
                                 0018  ;         1. WRITE IDCMPR WITH FLOATING ZERO PATTERN
                                 0018  ;         2. READ IDCDAR
                                 0018  ;         3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
                                 0018  ;         4. RETURN TO STEP 1. UNTIL ALL BITS HAVE BEEN TESTED
                                 0018  ; E. SUBTEST 5
                                 0018  ;         1. LOAD IDCDAR WITH ONES USING A BIT SET INSTRUCTION
                                 0018  ;         2. READ IDCDAR
                                 0018  ;         3. IF NOT EQUAL TO EXPECTED VALUE - REPORT ERROR
                                 0018  ;
                                 0018  ; DEBUG:
                                 0018  ;
                                 0018  ; IF THIS TEST FAILS THE IDCDAR REGISTER LOGIC SHOULD BE CHECKED
                                 0018  ; AND ALSO THE REGISTER DECODE LOGIC.
                                 0018  ;
                                 0018  ;--
                                 0018          $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
 0000'CF   00000040 8F     D0    001E          MOVL    #PRT$M_NONE,RGLIST                  ; ONLY DUMP REG THAT FAILS
                                 0027          $DS_BGNSUB
                                 0030          STARS
                                 0030  ;;*****************************************************************
                                 0030  ;++
                                 0030  ; CHECK THE IDCDAR REGISTER FOR ANY BITS STUCK AT A ZERO
                                 0030  ;--
        52   0000'CF       D0    0030          MOVL    BASE_ADDR,R2                        ; R2 <-- GETS IDC BASE ADDRESS
     62      00000080 8F   D0    0035          MOVL    #CSR$M_CRDY,IDCCSR(R2)              ; SET UP SO THAT NO DAR
                                 003C                                                      ; MODIFICATIONS WILL BE DONE BY
                                 003C                                                      ; THE IDC MICRO CODE.
```

```
               04 A2    D4  003C   637          CLRL    IDCBAR(R2)                      ; RESET PREVIOUS REG
               08 A2    D4  003F   638          CLRL    IDCCNT(R2)                      ; RESET PREVIOUS REGISTER
   0C A2  FFFFFFFF 8F   D0  0042   639 10$:     MOVL    #-1,IDCDAR(R2)                  ; WRITE IDCDAR WITH 1'S
 0000'CF  000FFFFF 8F   D0  004A   640          MOVL    #^XFFFFF,E.DA                   ; SET EXPECTED IDCDAR
     0000'CF   0C A2    D0  0053   641          MOVL    IDCDAR(R2),T.DA                 ; READ IDCDAR REGISTER
  0000'CF    0000'CF    D1  0059   642          CMPL    E.DA,T.DA                       ; TEST THAT IDCDAR = EXPECTED VALUE
                  23    13  0060   643          BEQL    20$                             ; BRANCH IF IT DOES
                            0062   644          $DS_ERRHARD_S    ,L$UNIT,MSG3_RG_ONE,-
                            0062   645                           REG_ERRPT,#MSG8_DA_FAIL,-
                            0062   646                           E.DA,T.DA
                            007E       ;;; TEST 6, SUBTEST 1, ERROR 1
                            0085   647 20$:     $DS_CKLOOP       10$
                            008D   648          $DS_ENDSUB
                            0096   649
                            0096   650          $DS_BGNSUB
                            009F   651          STARS
                            009F       ;;***************************************************************
                            009F   652 ;++
                            009F   653 ;TEST FOR ANY IDCDAR BITS STUCK AT ONE
                            009F   654 ;--
                            009F   655
               0C A2    D4  009F   656 10$:     CLRL    IDCDAR(R2)                      ; CLEAR ALL IDCDAR BITS
             0000'CF    D4  00A2   657          CLRL    E.DA                            ; SET EXPECTED IDCDAR REG
     0000'CF   0C A2    D0  00A6   658          MOVL    IDCDAR(R2),T.DA                 ; READ IDCDAR REG BACK
  0000'CF    0000'CF    D1  00AC   659          CMPL    E.DA,T.DA                       ; EXPECTED = ACTUAL ?
                  23    13  00B3   660          BEQL    20$                             ; BRANCH IF YES
                            00B5   661          $DS_ERRHARD_S    ,L$UNIT,MSG2_RG_ZERO,-
                            00B5   662                           REG_ERRPT,#MSG8_DA_FAIL,-
                            00B5   663                           E.DA,T.DA
                            00D1       ;;; TEST 6, SUBTEST 2, ERROR 1
                            00D8   664 20$:     $DS_CKLOOP       10$
                            00E0   665          $DS_ENDSUB
                            00E9   666
                            00E9   667          $DS_BGNSUB
                            00F2   668          STARS
                            00F2       ;;***************************************************************
                            00F2   669 ;++
                            00F2   670 ; TEST IDCDAR WITH FLOATING ONE'S PATTERN
                            00F2   671 ;--
               53   01  D0  00F2   672          MOVL    #BIT0,R3                        ; SET INITIAL PATTERN
            10 A2   53  D0  00F5   673 10$:     MOVL    R3,IDCMPR(R2)                   ; WRITE IDCMPR REGISTER
          0000'CF   53  D0  00F9   674          MOVL    R3,E.DA                         ; SET IDCDAR EXPECTED VALUE
     0000'CF   0C A2    D0  00FE   675          MOVL    IDCDAR(R2),T.DA                 ; READ IDCDAR REGISTER
  0000'CF    0000'CF    D1  0104   676          CMPL    E.DA,T.DA                       ; ACTUAL = EXPECTED ?
                  23    13  010B   677          BEQL    20$                             ; BRANCH IF YES
                            010D   678          $DS_ERRHARD_S    ,L$UNIT,MSG4_RG_FLT1,-
                            010D   679                           REG_ERRPT,#MSG8_DA_FAIL,-
                            010D   680                           E.DA,T.DA
                            0129       ;;; TEST 6, SUBTEST 3, ERROR 1
                            0130   681 20$:     $DS_CKLOOP       10$
          53   53   01  9C  0138   682          ROTL    #1,R3,R3                        ; ROTATE TO TEST NEXT BIT
             B5 53  14  E1  013C   683          BBC     #20,R3,10$                      ; CONTINUE UNTIL ALL BITS TESTED
                            0140   684          $DS_ENDSUB
                            0149   685
                            0149   686          $DS_BGNSUB
                            0152   687          STARS
```

```
                                  0152       ;;*****************************************************************
                                  0152       ;++
                                  0152       ; TEST IDCDAR WITH A FLOATING 0'S PATTERN
                                  0152       ;--
                                  0152
                   53    01   D0  0152               MOVL    #BIT0,R3                          ; R3 <-- STARTING TEST PATTERN
      54   53  000FFFFF  8F   CD  0155       10$:    XORL3   #^XFFFFFF,R3,R4                   ; MAKE R4 HAVING 1 ZERO BIT
                10 A2    54   D0  015D               MOVL    R4,IDCMPR(R2)                     ; WRITE IDCMPR REGISTER
              0000'CF    54   D0  0161               MOVL    R4,E.DA                           ; SET EXPECTED IDCDAR REGISTER
         0000'CF    0C   A2   D0  0166               MOVL    IDCDAR(R2),T.DA                   ; READ IDCDAR REAIGTER
      0000'CF     0000'CF     D1  016C               CMPL    E.DA,T.DA                         ; ACTUAL = EXPECTED ?
                         23   13  0173               BEQL    20$                               ; BRANCH IF YES
                                  0175               $DS_ERRHARD_S    ,L$UNIT,MSG5_RG_FLT0,-
                                  0175                               REG_ERRPT,#MSG8_DA_FAIL,-
                                  0175                               E.DA,T.DA
                                  0191       ;;; TEST 6, SUBTEST 4, ERROR 1
                                  0198       20$:    $DS_CKLOOP       10$
             53    53    01   9C  01A0               ROTL    #1,R3,R3                          ; GET NEXT BIT POSITION TO TEST
                AD 53    14   E1  01A4               BBC     #20,R3,10$                        ; CONTINUE UNTIL ALL BITS TESTED
                                  01A8               $DS_ENDSUB
                                  01B1
                                  01B1               $DS_BGNSUB
                                  01BA               STARS
                                  01BA       ;;*****************************************************************
                                  01BA       ;++
                                  01BA       ; TEST IDCDAR WITH A READ MODIFY WRITE INSTRUCTION AND TEST THAT
                                  01BA       ; UNIBUS INIT WILL CLEAR THE IDCDAR REGISTER PROPERLY
                                  01BA       ;--
                                  01BA
                      0C A2   D4  01BA       10$:    CLRL    IDCDAR(R2)                        ; CLEAR IDCDAR REGISTER
    0C A2    FFFFFFFF    8F   C8  01BD               BISL2   #-1,IDCDAR(R2)                    ; WRITE ALL POSSIBLE IDCDAR BITS
  0000'CF    000FFFFF    8F   D0  01C5               MOVL    #^XFFFFFF,E.DA                    ; SET EXPECTED IDCDAR VALUE
        0000'CF    0C    A2   D0  01CE               MOVL    IDCDAR(R2),T.DA                   ; READ IDCDAR REGISTER
      0000'CF     0000'CF     D1  01D4               CMPL    E.DA,T.DA                         ; ACTUAL = EXPECTED ?
                         23   13  01DB               BEQL    20$                               ; BRANCH IF YES
                                  01DD               $DS_ERRHARD_S    ,L$UNIT,MSG9_RG_RMW,-
                                  01DD                               REG_ERRPT,#MSG8_DA_FAIL,-
                                  01DD                               E.DA,T.DA
                                  01F9       ;;; TEST 6, SUBTEST 5, ERROR 1
                                  0200       20$:    $DS_CKLOOP       10$
                                  0208               $DS_ENDSUB
                                  0211               $DS_ENDTEST
                                  021C               $DS_SBTTL <GET STATUS FUNCTION TEST>,ALIGN=LONG
                                  021C               .SBTTL  TEST 7:  GET STATUS FUNCTION TEST
                                  021C               $D2_SBTTL        007,<GET STATUS FUNCTION TEST>,<LONG>
                              00000000                       .PSECT  TEST_007, LONG, NOWRT
                                  001C
```

```
                               001C     728
                               001C     729 ;++
                               001C     730 ;
                               001C     731 ; TEST DESCRIPTION:
                               001C     732 ;
                               001C     733 ; THIS TEST WILL EXECUTE A GET STATUS COMMAND (WITH THE RST BIT
                               001C     734 ; SET IN THE IDCMPR REGISTER IF DRIVE IS AND RL02) AND CHECK THAT
                               001C     735 ; NO ERRORS OCCUR. THEN THE IDCMPR REGISTER CONTAINING THE DRIVE
                               001C     736 ; STATUS IS CHECKED FOR THE PROPER STATE.
                               001C     737 ;
                               001C     738 ; TEST STEPS:
                               001C     739 ;
                               001C     740 ; 1.      INITIALIZE SUBSYSTEM
                               001C     741 ; 2.      LOAD IDCMPR REGISTER FOR GET STATUS FUNCTION
                               001C     742 ; 3.      LOAD IDCCSR REGISTER AND EXECUTE GET STATUS COMMAND
                               001C     743 ; 4.      WAIT FOR COMMAND COMPLETION AND CHECK FOR ANY ERRORS -
                               001C     744 ;         REPORT ERROR TO OPERATOR
                               001C     745 ; 5.      CHECK DRIVE STATUS VIA THE IDCMPR REGISTER - REPORT ANY DISCREPANCY
                               001C     746 ;
                               001C     747 ; DEBUG:
                               001C     748 ;
                               001C     749 ;--
                               001C     750
                               001C     751         $DS_BGNDATA
       00000000 00000001       001C     752         .LONG   1,LUN0
       00000001 00000001       0024     753         .LONG   1,LUN1
       00000002 00000001       002C     754         .LONG   1,LUN2
       00000003 00000001       0034     755         .LONG   1,LUN3
                               003C     756         $DS_ENDDATA
                               0040     757
                               0040     758         $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
                               0042     759 1$:     GET_LUN -                               ; GET LUN INFO
                               0042     760           UNIT=         OFST_UNIT(AP)           ; UNIT TO TEST
              04 AC      DD    0042             PUSHL   OFST_UNIT(AP)
       0000'CF   01      FB    0045             CALLS   #1,GET_LUN$A
                               004A     761         BDEVFOUND       5$                      ; IF LUN AVAILABLE GOTO 5$
          06 50   00     E2    004A             BBSS    #0,R0,5$
             50   01     D0    004E                     MOVL    #1,R0           ; NOT AN ERROR
                008F     31    0051                     BRW     TEST_007_X      ; EXIT TEST 7
             0000'CF     D4    0054     762 5$:     CLRL    CONFIG                  ; CLEAR PROGRAM FLAGS
       53    0000'CF     D0    0058     763         MOVL    DRIVE_TBL,R3            ; GET ADR OF DRIVE INFO
        54    50 A3      D0    005D     764         MOVL    OFST_DRVRST(R3),R4      ; GET BIT MASK FOR DRIVE RESET CMD
        57    38 A3      D0    0061     765         MOVL    OFST_NMLSTS(R3),R7      ; GET EXPECTED NORMAL DRIVE STATUS
        56    40 A3      D0    0065     766         MOVL    OFST_MPBITS(R3),R6      ; GET ADDR OF MPR BIT MNEMONICS
                  54     DD    0069     767 10$:    PUSHL   R4                      ; IDCMPR DATA
                  00     DD    006B     768         PUSHL   #0                      ; IDCDAR DATA
                  00     DD    006D     769         PUSHL   #0                      ; IDCCNT DATA
                  00     DD    006F     770         PUSHL   #0                      ; IDCBAR DATA
                  04     DD    0071     771         PUSHL   #GETSTAT                ; PUSH COMMAND
       0000'CF    05     FB    0073     772         CALLS   #5,START_CMD            ; EXECUTE COMMAND
       0000'CF    00     FB    0078     773         CALLS   #0,STATUS_CHK           ; CHECK FUNCTION
           17 50         E8    007D     774         BLBS    R0,30$                  ; BR IF NO ERRORS
                               0080     775         $DS_ERRHARD_S   ,L$UNIT,MSG17_GETSTAT,-
                               0080     776                         PRINT_REG,R1
                               0090         ;;; TEST 7, SUBTEST 0, ERROR 1
                               0097     777 30$:    $DS_CKLOOP      10$
```

```
      0000'CF    00000040 8F   CA  009F   778            BICL2   #MPR_02$M_HS,T.MP                    ; MASK OUT RL02 HEAD SELECT BIT
           0000'CF     64 A3   CA  00A8   779            BICL2   OFST_STSDNC(R3),T.MP                 ; CLEAR OUT DONT CARE BITS
              0000'CF      57  D1  00AE   780            CMPL    R7,T.MP                              ; ACTUAL = EXPECTED ?
                           23  13  00B3   781            BEQL    40$                                  ; YES - BRANCH
                                   00B5   782            $DS_ERRHARD_S     ,L$UNIT,MSG17_GETSTAT,-
                                   00B5   783                              REG_ERRPT,#MSG27_MP_INC,-
                                   00B5   784                              R7,T.MP,R6
                                   00D1          ;;; TEST 7, SUBTEST 0, ERROR 2
                                   00D8   785 40$:       $DS_CKLOOP        10$
                                   00E0   786            $DS_ENDTEST
                                   00EB   787            $DS_SBTTL <NO-OP FUNCTION TEST>,ALIGN=LONG
                                   00EB                  .SBTTL  TEST 8:  NO-OP FUNCTION TEST
                                   00EB                  $D2_SBTTL         008,<NO-OP FUNCTION TEST>,<LONG>
                               00000000                            .PSECT  TEST_008, LONG, NOWRT
                                   0018
```

```
                           0018  ;++
                           0018  ;
                           0018  ; TEST DESCRIPTION:
                           0018  ;
                           0018  ; THIS TEST WILL VERIFY THE FUNCTIONALITY OF THE NO OP COMMAND BY
                           0018  ; LOADING THE IDCBAR, IDCCNT  AND IDCDAR REGISTERS WITH DATA.
                           0018  ; THEN EXECUTE THE NO OP COMMAND AND VERIFY THAT NO ERRORS OCCUR
                           0018  ; AND THE IDCBAR, IDCCNT AND IDCDAR REGISTERS REMAIN UNCHANGED.
                           0018  ;
                           0018  ; TEST STEPS:
                           0018  ;
                           0018  ; 1.     INITIALIZE SUBSYTEM
                           0018  ; 2.     LOAD IDCBAR, IDCCNT AND IDCDAR REGISTERS
                           0018  ; 3.     EXECUTE NO OP CMD
                           0018  ; 4.     WAIT FOR CRDY - IF NOT SET REPORT ERROR
                           0018  ; 5.     TEST FOR ANY ERRORS - REPORT ERROR IF ANY
                           0018  ; 6.     CHECK IDCBAR FOR ANY MODIFICATION - REPORT ERROR IF ANY
                           0018  ; 7.     CHECK IDCDAR FOR ANY MODIFICATION - REPORT ERROR IF ANY
                           0018  ; 8.     CHECK IDCCNT FOR ANY MODIFICATION - REPORT ERROR IF ANY
                           0018  ;
                           0018  ; DEBUG:
                           0018  ;
                           0018  ;        COMMAND DECODE LOGIC
                           0018  ;
                           0018  ;--
                           0018
                           0018          $DS_BGNDATA
 00000000 00000001         0018          .LONG   1,LUN0
 00000001 00000001         0020          .LONG   1,LUN1
 00000002 00000001         0028          .LONG   1,LUN2
 00000003 00000001         0030          .LONG   1,LUN3
                           0038          $DS_ENDDATA
                           003C
                           003C          $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
                           003E          GET_LUN -                               ; GET LUN INFO
                           003E            UNIT=          OFST_UNIT(AP)          ; LOGCICAL UNIT
        04 AC      DD      003E          PUSHL   OFST_UNIT(AP)
  0000'CF   01     FB      0041          CALLS   #1,GET_LUN$A
                           0046          BDEVFOUND        5$                     ; IF LUN AVAILABLE GOTO 5$
     06 50  00     E2      0046          BBSS    #0,R0,5$
        50  01     D0      004A                  MOVL    #1,R0         ; NOT AN ERROR
          0104     31      004D                  BRW     TEST_008_X    ; EXIT TEST 8
      0000'CF      D4      0050  5$:     CLRL    CONFIG                  ; CLEAR PROGRAM FLAGS
  0000'CF   00     FB      0054  10$:    CALLS   #0,SUBSYS_INIT          ; INIT CONTROLLER AND DRIVE
     17 50         E8      0059          BLBS    R0,20$                  ; BRANCH IF NO ERRORS
                           005C          $DS_ERRHARD_S    ,L$UNIT,MSG14_SYSINIT,-
                           005C                           PRINT_REG,R1
                           006C  ;;; TEST 8, SUBTEST 0, ERROR 1
                           0073  20$:    $DS_CKLOOP       10$
     53     01     D0      007B          MOVL    #1,R3                   ; SET EXPECTED IDCDAR
     54     02     D0      007E          MOVL    #2,R4                   ; SET EXPECTED IDCCNT
     55     03     D0      0081          MOVL    #3,R5                   ; SET EXPECTED IDCBAR
            00     DD      0084  25$:    PUSHL   #0                      ; PUSH IDCMPR DATA
            01     DD      0086          PUSHL   #1                      ; PUSH IDCDAR DATA
            02     DD      0088          PUSHL   #2                      ; PUSH IDCCNT DATA
            03     DD      008A          PUSHL   #3                      ; PUSH IDCBAR DATA
```

```
                  00   DD  008C   839           PUSHL   #NO_OP                          ; PUSH COMMAND
        0000'CF   05   FB  008E   840           CALLS   #5,START_CMD                    ; EXECUTE NOOP CMD
        0000'CF   10   D0  0093   841           MOVL    #NOCHK$M_DSTA,CONFIG            ; DONT GET DRIVE STATUS IN SUBR
        0000'CF   00   FB  0098   842           CALLS   #0,STATUS_CHK                   ; CHECK COMMAND COMPLETES OK
             17   50   E8  009D   843           BLBS    R0,30$                          ; BR IF NO ERRORS
                           00A0   844           $DS_ERRHARD_S    ,L$UNIT,MSG15_NOOP,-
                           00A0   845                            PRINT_REG,R1
                           00B0         ;;; TEST 8, SUBTEST 0, ERROR 2
                           00B7   846 30$:      $DS_CKLOOP       25$
        0000'CF   55   D1  00BF   847           CMPL    R5,T.BA                         ; CHECK IDCBAR REGISTER
                  21   13  00C4   848           BEQL    35$                             ; BRANCH IF OK
                           00C6   849           $DS_ERRHARD_S    ,L$UNIT,MSG15_NOOP,-
                           00C6   850                            REG_ERRPT,#MSG24_BA_MOD,-
                           00C6   851                            R5,T.BA
                           00E0         ;;; TEST 8, SUBTEST 0, ERROR 3
                           00E7   852 35$:      $DS_CKLOOP       25$
        0000'CF   54   D1  00EF   853           CMPL    R4,T.BC                         ; CHECK IDCCNT REGISTER
                  21   13  00F4   854           BEQL    40$                             ; BRANCH IF OK
                           00F6   855           $DS_ERRHARD_S    ,L$UNIT,MSG15_NOOP,-
                           00F6   856                            REG_ERRPT,#MSG24A_BC_MOD,-
                           00F6   857                            R4,T.BC
                           0110         ;;; TEST 8, SUBTEST 0, ERROR 4
                           0117   858 40$:      $DS_CKLOOP       25$
        0000'CF   53   D1  0120   859           CMPL             R3,T.DA                ; CHECK IDCDAR REGISTER
                  21   13  0125   860           BEQL    50$
                           0127   861           $DS_ERRHARD_S    ,L$UNIT,MSG15_NOOP,-
                           0127   862                            REG_ERRPT,#MSG25_DA_MOD,-
                           0127   863                            R3,T.DA
                           0141         ;;; TEST 8, SUBTEST 0, ERROR 5
                           0148   864 50$:      $DS_CKLOOP       25$
                           0151   865           $DS_ENDTEST
                           015C   866           $DS_SBTTL <INTERRUPT TEST>,ALIGN=LONG
                           015C                 .SBTTL  TEST 9:  INTERRUPT TEST
                           015C                 $D2_SBTTL        009,<INTERRUPT TEST>,<LONG>
                       00000000                          .PSECT  TEST_009, LONG, NOWRT
                           0014
```

```
                           0014   868
                           0014   869 ;++
                           0014   870 ;
                           0014   871 ; TEST DESCRIPTION:
                           0014   872 ;
                           0014   873 ; THIS TEST WILL VERIFY THAT THE IDC CONTROLLER IS ABLE TO COMPLETE
                           0014   874 ; THE INTERRUPT SEQUENCE PROPERLY.  THIS IS DONE BY EXECUTING A NO OP
                           0014   875 ; COMMAND WITH THE INTERRUPT ENABLE BIT (IE) SET IN THE CONTROL AND
                           0014   876 ; STATUS REGISTER.  AFTER INTERRUPT HAS OCCURED CHECK PROPER STATUS.
                           0014   877 ;
                           0014   878 ; TEST STEPS:
                           0014   879 ;
                           0014   880 ; 1.     INITIALIZE SUBSYSTEM
                           0014   881 ; 2.     CLEAR INTERRUPT STATUS AND ENABLE INTERRUPTS
                           0014   882 ; 3.     EXECUTE NO OP COMMAND
                           0014   883 ; 4.     WAIT FOR INTERRUPT
                           0014   884 ; 5.     IF INTERRUPT DID NOT OCCUR REPORT ERROR
                           0014   885 ; 6.     CHECK FOR ANY ERRORS FOLLOWING INTERRUPT
                           0014   886 ;        REPORT ANY FAILURES
                           0014   887 ; 7.     DISABLE INTERRUPTS
                           0014   888 ;
                           0014   889 ; DEBUG:
                           0014   890 ;
                           0014   891 ; IDC INTERRUPT LOGIC
                           0014   892 ;
                           0014   893 ;--
                           0014   894
                           0014   895         $DS_BGNDATA
       00000000 00000001   0014   896         .LONG   1,LUN0
       00000001 00000001   001C   897         .LONG   1,LUN1
       00000002 00000001   0024   898         .LONG   1,LUN2
       00000003 00000001   002C   899         .LONG   1,LUN3
                           0034   900         $DS_ENDDATA
                           0038   901
                           0038   902         $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
                           003A   903         GET_LUN -                                   ; GET LUN INFO
                           003A   904           UNIT=         OFST_UNIT(AP)               ; LOGICAL UNIT
            04 AC     DD   003A               PUSHL   OFST_UNIT(AP)
       0000'CF  01    FB   003D               CALLS   #1,GET_LUN$A
                           0042   905         BDEVFOUND       5$                          ; IF DEV AVAILABLE GOTO 5$
         06 50  00    E2   0042               BBSS    #0,R0,5$
            50  01    D0   0046                       MOVL    #1,R0         ; NOT AN ERROR
              016D    31   0049                       BRW     TEST_009_X    ; EXIT TEST 9
      54   0000'CF    D0   004C   906 5$:     MOVL    DRIVE_TBL,R4                ; GET DRIVE INFO TABLE
           0000'CF    D4   0051   907         CLRL    CONFIG                      ; CLEAR PROGRAM FLAGS
0000'CF  000000C1 8F  D0   0055   908         MOVL    #<CSR$M_CRDY!CSR$M_IE!NO_OP!-
                           005E   909                 CSR$M_DRDY>,E.CS            ; SET EXPECTED IDCCSR REG
   0000'CF   0000'CF  C8   005E   910         BISL2   P$UNIT,E.CS                 ; OR IN DRIVE NUMBER
     0000'CF   58 A4  C8   0065   911         BISL2   OFST_CSRTYP(R4),E.CS        ; OR IN EXPECTED DRIVE TYPE
       0000'CF   00   FB   006B   912 10$:    CALLS   #0,SUBSYS_INIT              ; INIT CONTROLLER AND DRIVE
            17 50     E8   0070   913         BLBS    R0,20$                      ; BRANCH IF NO ERRORS
                           0073   914         $DS_ERRHARD_S   ,L$UNIT,MSG14_SYSINIT,-
                           0073   915                         PRINT_REG,R1
                           0083       ;;; TEST 9, SUBTEST 0, ERROR 1
                           008A   916 20$:    $DS_CKLOOP      10$
           0000'CF    7C   0092   917 30$:    CLRQ    IDC_STAT$L_1                ; CLEAR INTERRUPT STATUS WORD
```

```
                  00   DD  0096   918          PUSHL   #0                               ; IDCMPR DATA
                  00   DD  0098   919          PUSHL   #0                               ; IDCDAR DATA
                  00   DD  009A   920          PUSHL   #0                               ; IDCCNT DATA
                  00   DD  009C   921          PUSHL   #0                               ; IDCBAR DATA
         00000040 8F   DD  009E   922          PUSHL   #<CSR$M_IE!NO_OP>                ; NOOP CMD + INTER ENABLE
          0000'CF  05  FB  00A4   923          CALLS   #5,START_CMD                     ; EXECUTE COMMAND
          53    FF 8F  9A  00A9   924          MOVZBL  #^XFF,R3                         ; SET WAIT TIME
             0000'CF   16  00AD   925 40$:     JSB     RD_IDCREGS                       ; UPDATE IDC REGISTERS
             0000'CF   D5  00B1   926          TSTL    IDC_STAT$W_2                     ; DID INTERRUPT OCCUR ?
                  02   13  00B5   927          BEQL    50$                              ; BR IF NOT
                  3A   11  00B7   928          BRB     70$                              ; INTERRUPT DID OCCUR - CHK STATS
               F1 53   F5  00B9   929 50$:     SOBGTR  R3,40$                           ; DECR AND TRY AGAIN
    62    00000040 8F  CA  00BC   930          BICL2   #CSR$M_IE,IDCCSR(R2)             ; TURN INTERRUPTS OFF
    62    01000000 8F  C8  00C3   931          BISL2   #CSR$M_IR,IDCCSR(R2)             ; CLEAR PENDING INTERRUPTS
                           00CA   932          $DS_ERRHARD_S     ,L$UNIT,MSG38_NOOP_IE,-
                           00CA   933                            PRINT_REG,#MSG39_NO_INT
                           00DE            ;;; TEST 9, SUBTEST 0, ERROR 2
                           00E5   934 60$:     $DS_CKLOOP        10$
                           00ED   935          $DS_EXIT          TEST                   ; EXIT TEST
    62    00000040 8F  CA  00F3   936 70$:     BICL2   #CSR$M_IE,IDCCSR(R2)             ; TURN INTERRUPTS OFF
    62    01000000 8F  C8  00FA   937          BISL2   #CSR$M_IR,IDCCSR(R2)             ; CLEAR PENDING INTERRUPTS
  0000'CF      0000'CF D1  0101   938          CMPL    E.CS,T.CS                        ; ACTUAL = EXPECTED ?
                  29   13  0108   939          BEQL    80$                              ; YES BRANCH
                           010A   940          $DS_ERRHARD_S     ,L$UNIT,MSG38_NOOP_IE,-
                           010A   941                            REG_ERRPT,#MSG23_CS_INC,-
                           010A   942                            E.CS,T.CS,#CSR_BITS
                           012C            ;;; TEST 9, SUBTEST 0, ERROR 3
                           0133   943 80$:     $DS_CKLOOP        10$
  0000'CF      03   02 EF  013C   944          EXTZV   #CHI$V_IPL,#3,IDC_STAT$W_2,-     ; GET ACTUAL BR LEVEL
               0000'CF     0142   945                  T.BR
  0000'CF      0000'CF D1  0145   946          CMPL    E.BR,T.BR                        ; EXPECTED = ACTUAL BR ?
                  23   13  014C   947          BEQL    90$                              ; BRANCH IF YES
                           014E   948          $DS_ERRHARD_S     ,L$UNIT,MSG38_NOOP_IE,-
                           014E   949                            REG_ERRPT,#MSG40_BAD_BR,-
                           014E   950                            E.BR,T.BR
                           016A            ;;; TEST 9, SUBTEST 0, ERROR 4
                           0171   951 90$:     $DS_CKLOOP        10$
  0000'CF      0000'CF 3C  017A   952          MOVZWL  IDC_STAT$W_VR,T.VEC              ; GET ACTUAL VECTOR
  0000'CF      0000'CF D1  0181   953          CMPL    E.VEC,T.VEC                      ; EXPECTED = ACTUAL VECTOR ?
                  23   13  0188   954          BEQL    100$                             ; BRANCH IF YES
                           018A   955          $DS_ERRHARD_S     ,L$UNIT,MSG38_NOOP_IE,-
                           018A   956                            REG_ERRPT,#MSG41_BAD_VEC,-
                           018A   957                            E.VEC,T.VEC
                           01A6            ;;; TEST 9, SUBTEST 0, ERROR 5
                           01AD   958 100$:    $DS_CKLOOP        10$
                           01B6   959          $DS_ENDTEST
                           01C1   960          $DS_SBTTL <READ HEADER AND HEADER CRC TEST>,ALIGN=LONG
                           01C1                .SBTTL  TEST 10:  READ HEADER AND HEADER CRC TEST
                           01C1                $D2_SBTTL         010,<READ HEADER AND HEADER CRC TEST>,<LONG>
                       00000000                          .PSECT  TEST_010, LONG, NOWRT
                           0024
```

```
                                0024
                                0024    ;++
                                0024    ;
                                0024    ; TEST DESCRIPTION:
                                0024    ;
                                0024    ; THIS TEST WILL VERIFY THE READ HEADER COMMAND BY EXECUTING 40 READ
                                0024    ; HEADER COMMANDS.  AFTER EACH READ HEADER CONTROLLER AND DRIVE STATUS
                                0024    ; ARE CHECKED FOR ANY ERRORS. HEADER WORDS ARE CHECKED FOR PROPER
                                0024    ; FORMAT AND THE HEADER WORD IS ALSO VALIDATED.
                                0024    ;
                                0024    ; TEST STEPS:
                                0024    ;
                                0024    ; 1.      INITIALIZE SUBSYSTEM
                                0024    ; 2.      EXECUTE READ HEADER COMMAND
                                0024    ; 3.      CHECK FOR ANY STATUS ERRORS
                                0024    ; 4.      TEST HDR WORD #1 ALWAYS YIELDS SAME CYLINDER
                                0024    ; 5.      TEST HDR WORD #1 ALWAYS YIELDS SAME HEAD
                                0024    ; 6.      TEST HDR WORD #2 IS EQUAL TO ZERO
                                0024    ; 7.      CALCULATE EXPECTED CRC WORD
                                0024    ; 8.      TEST THAT ACTUAL CRC IN HDR WORD #3 = EXPECTED CRC
                                0024    ;
                                0024    ; DEBUG:
                                0024    ; SINCE WE DO NOT KNOW THAT THE HEADER CRC CHECKING LOGIC IN THE
                                0024    ; CONTROLLER IS WORKING THE SOFTWARE CHECKING OF THE CRC SHOULD
                                0024    ; CONFIRM THE ABILITY OF THE SUBSYSTEM TO READ GOOD HEADERS.
                                0024    ;
                                0024    ;
                                0024    ;--
                                0024
                                0024
                                0024            $DS_BGNDATA
        00000000 00000001       0024            .LONG   1,LUN0
        00000001 00000001       002C            .LONG   1,LUN1
        00000002 00000001       0034            .LONG   1,LUN2
        00000003 00000001       003C            .LONG   1,LUN3
                                0044            $DS_ENDDATA
                                0048
                                0048            $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
                                004A            GET_LUN -                               ; GET LUN INFO
                                004A              UNIT=          OFST_UNIT(AP)          ; LOGICAL UNIT
              04 AC     DD      004A            PUSHL   OFST_UNIT(AP)
        0000'CF    01   FB      004D            CALLS   #1,GET_LUN$A
                                0052            BDEVFOUND       5$                      ; IF DEV AVAILABLE GOTO 5$
        06 50      00   E2      0052            BBSS    #0,R0,5$
           50      01   D0      0056                    MOVL    #1,R0           ; NOT AN ERROR
               011F     31      0059                    BRW     TEST_010_X      ; EXIT TEST 10
    54    0000'CF       D0      005C    5$:     MOVL    DRIVE_TBL,R4                    ; GET DRIVE INFO TABLE
     55      48 A4      D0      0061            MOVL    OFST_HDRTMP(R4),R5              ; GET ADDR OF HDR TEMPLATE
     56      44 A4      D0      0065            MOVL    OFST_HDBITS(R4),R6              ; GET ADDR OF HDR BIT MNEMONICS
                   53   D4      0069            CLRL    R3                              ; CLEAR ITERATION COUNTER
           0000'CF      D4      006B            CLRL    CONFIG                          ; CLEAR PROGRAM FLAGS
        0000'CF    00   FB      006F    10$:    CALLS   #0,SUBSYS_INIT                  ; INIT CONTROLLER AND DRIVE
              17 50     E8      0074            BLBS    R0,20$                          ; BRANCH IF NO ERRORS
                                0077            $DS_ERRHARD_S   ,L$UNIT,MSG14_SYSINIT,-
                                0077                            PRINT_REG,R1
                                0087    ;;; TEST 10, SUBTEST 0, ERROR 1
```

```
                              008E  20$:      $DS_CKLOOP     10$
                 00      DD   0096  30$:      PUSHL    #0                          ; IDCMPR DATA
                 00      DD   0098            PUSHL    #0                          ; IDCDAR DATA
                 00      DD   009A            PUSHL    #0                          ; IDCCNT DATA
                 00      DD   009C            PUSHL    #0                          ; IDCBAR DATA
                 08      DD   009E            PUSHL    #READH                      ; READ HEADER COMMAND
         0000'CF 05      FB   00A0            CALLS    #5,START_CMD                ; EXECUTE COMMAND
         0000'CF 00      FB   00A5            CALLS    #0,STATUS_CHK               ; CHECK FOR RDY ONLY
              17 50      E8   00AA            BLBS     R0,40$                      ; BRANCH IF NO ERRORS
                              00AD            $DS_ERRHARD_S   ,L$UNIT,MSG19_READH,-
                              00AD                            PRINT_REG,R1
                              00BD      ;;; TEST 10, SUBTEST 0, ERROR 2
                              00C4  40$:      $DS_CKLOOP     30$
                 53      D5   00CC            TSTL     R3                          ; IS THIS FIRST PASS
                 13      12   00CE            BNEQ     45$                         ; NO BRANCH
         59  0000'CF     D0   00D0            MOVL     T.HDR,R9                    ; ACTUAL WILL NOW BE THE EXPECTED
      0A A5     00       F0   00D5            INSV     #0,OFST_SECPOS(R5),-        ; CLEAR EXPECTED SECTOR BITS
         59   0E A5           00D9                     OFST_SECSIZ(R5),R9          ;
         59   54 A4      CA   00DC            BICL2    OFST_HDRDNC(R4),R9          ; CLEAR DONT CARE BITS
               003F      31   00E0            BRW      65$                         ; GO TEST HDR CRC
         57  0000'CF     D0   00E3  45$:      MOVL     T.HDR,R7                    ; GET CURRENT HEADER
      0A A5     00       F0   00E8            INSV     #0,OFST_SECPOS(R5),-        ; CLEAR EXPECTED SECTOR BITS
         57   0E A5           00EC                     OFST_SECSIZ(R5),R7          ;
         57   54 A4      CA   00EF            BICL2    OFST_HDRDNC(R4),R7          ; CLEAR DONT CARE BITS
            59   57      D1   00F3            CMPL     R7,R9                       ; ACTUAL = EXPECTED ?
                 21      13   00F6            BEQL     50$
                              00F8            $DS_ERRHARD_S   ,L$UNIT,MSG19_READH,-
                              00F8                            REG_ERRPT,#MSG42_RPT_CYL,-
                              00F8                            R9,R7,R6
                              0112      ;;; TEST 10, SUBTEST 0, ERROR 3
                              0119  50$:      $DS_CKLOOP     30$
                              0122            ;++
                              0122            ;
                              0122            ; CALCULATE THE CRC AND CHECK THAT IT COMPARES WITH THE
                              0122            ; CRC READ FROM THE DISK
                              0122            ;
                              0122            ;--
                              0122
                 3F      BB   0122  65$:      PUSHR    #^M<R0,R1,R2,R3,R4,R5>      ; SAVE REGISTERS
         57  0000'CF     DE   0124            MOVAL    WORK,R7                     ; GET ADDR OF TEMP LOCATION
         87  0000'CF     B0   0129            MOVW     T.MP1,(R7)+                 ; GET 1ST HEADER WORD
         67  0000'CF     B0   012E            MOVW     T.MP2,(R7)                  ; GET 2ND HEADER WORD
                 77      B5   0133            TSTW     -(R7)                       ; RESTORE ADDRESS
   67  04  00  0000'CF   0B   0135            CRC      CRC_TABLE,#0,#4,(R7)        ; CALC CRC FOR THOSE WORDS
            58   50      3C   013C            MOVZWL   R0,R8                       ; GET EXPECTED CRC
                 3F      BA   013F            POPR     #^M<R0,R1,R2,R3,R4,R5>      ; RESTORE REGISTERS
         0000'CF 58      D1   0141            CMPL     R8,T.MP3                    ; EXPECTED = ACTUAL ?
                 21      13   0146            BEQL     80$                         ; BRANCH IF YES
                              0148            $DS_ERRHARD_S   ,L$UNIT,MSG19_READH,-
                              0148                            REG_ERRPT,#MSG44_BAD_CRC,-
                              0148                            R8,T.MP3
                              0162      ;;; TEST 10, SUBTEST 0, ERROR 4
                              0169  80$:      $DS_CKLOOP     30$
      FF1E 53  01  27    F1   0172            ACBL     #39,#1,R3,30$               ; CONTINUE ITERATIONS
                              0178            $DS_ENDTEST
                              0183            $DS_SBTTL <RECAL FUNCTION TEST - R80>,ALIGN=LONG
```

```
                0183            .SBTTL  TEST 11:  RECAL FUNCTION TEST - R80
                0183            $D2_SBTTL       011,<RECAL FUNCTION TEST - R80>,<LONG>
            00000000                    .PSECT  TEST_011, LONG, NOWRT
                001C
```

```
                                  001C  1066
                                  001C  1067 ;++
                                  001C  1068 ;
                                  001C  1069 ; TEST DESCRIPTION:
                                  001C  1070 ;
                                  001C  1071 ; THIS TEST IS UTILIZED FOR THE R80 DISK DRIVE ONLY. THE PURPOSE
                                  001C  1072 ; OF THE TEST IS TO EXECTUTE A RECAL (SEEK TO CYLINDER 0) AND
                                  001C  1073 ; CHECK FOR ANY COMPORMISING CONTROLLER/DRIVE ERRORS.
                                  001C  1074 ;
                                  001C  1075 ; TEST STEPS:
                                  001C  1076 ;
                                  001C  1077 ; 1.     INITIALIZE SUBSYSTEM
                                  001C  1078 ; 2.     EXECUTE RECAL FUNCTION
                                  001C  1079 ; 3.     REPORT ANY ANY POSITIONING, DRIVE, OR CONTROLLER ERRORS
                                  001C  1080 ;
                                  001C  1081 ; DEBUG:
                                  001C  1082 ;
                                  001C  1083 ;       NONE
                                  001C  1084 ;
                                  001C  1085 ;--
                                  001C  1086
                                  001C  1087
                                  001C  1088          $DS_BGNDATA
       00000000 00000001          001C  1089          .LONG    1,LUN0
       00000001 00000001          0024  1090          .LONG    1,LUN1
       00000002 00000001          002C  1091          .LONG    1,LUN2
       00000003 00000001          0034  1092          .LONG    1,LUN3
                                  003C  1093          $DS_ENDDATA
                                  0040  1094
                                  0040  1095          $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
                                  0042  1096          GET_LUN -                                   ; GET LUN INFO
                                  0042  1097            UNIT=           OFST_UNIT(AP)             ; LOGICAL UNIT
                04 AC      DD     0042                PUSHL    OFST_UNIT(AP)
       0000'CF     01      FB     0045                CALLS    #1,GET_LUN$A
                                  004A  1098          BDEVFOUND         5$                        ; IF DEVICE AVAILABLE GOTO 5$
          06 50    00      E2     004A                BBSS     #0,R0,5$
             50    01      D0     004E                         MOVL     #1,R0           ; NOT AN ERROR
                 006D      31     0051                         BRW      TEST_011_X      ; EXIT TEST 11
       54    0000'CF       D0     0054  1099 5$:      MOVL     DRIVE_TBL,R4                       ; GET DRIVE INFO TABLE
          02       64      91     0059  1100          CMPB     OFST_DRVTYP(R4),#R80$T             ; IF DRIVE IS AN R80
                   06      13     005C  1101          BEQL     6$                                 ; THEN GOTO 6$
                                  005E  1102          $DS_EXIT          TEST                      ; ELSE EXIT
              0000'CF      D4     0064  1103 6$:      CLRL     CONFIG                             ; CLEAR ANY PROGRAM FLAGS
       0000'CF     00      FB     0068  1104 10$:     CALLS    #0,SUBSYS_INIT                     ; CLEAR CONTROLLER AND DRIVE
             1F 50         E8     006D  1105          BLBS     R0,20$                             ; IF NO ERROR GOTO 20$
                                  0070  1106          $DS_ERRHARD_S     ,L$UNIT,MSG14_SYSINIT,-
                                  0070  1107                            PRINT_REG,R1
                                  0080       ;;; TEST 11, SUBTEST 0, ERROR 1
                                  0087  1108          $DS_CKLOOP        10$
                   00      DD     008F  1109 20$:     PUSHL    #0                                 ; SELECT TRACK 0
          FFFFFFFF 8F      DD     0091  1110          PUSHL    #-1                                ; SELECT CYLINDER -1 FOR RECAL
       0000'CF     02      FB     0097  1111          CALLS    #2,POSITION_HDS                    ; EXECUTE RECAL
             17 50         E8     009C  1112          BLBS     R0,30$                             ; IF NO ERRORS GOTO 30$
                                  009F  1113          $DS_ERRHARD_S     ,L$UNIT,MS107_RECAL,-
                                  009F  1114                            PRINT_REG,R1
                                  00AF       ;;; TEST 11, SUBTEST 0, ERROR 2
```

```
    00B6  1115 30$:    $DS_CKLOOP      20$
    00BE  1116         $DS_ENDTEST
    00C9  1117         $DS_SBTTL <SEEK ERROR TEST - R80>,ALIGN=LONG
    00C9               .SBTTL  TEST 12:  SEEK ERROR TEST - R80
    00C9               $D2_SBTTL       012,<SEEK ERROR TEST - R80>,<LONG>
00000000                       .PSECT  TEST_012, LONG, NOWRT
    0018
```

```
                              0018  1119
                              0018  1120 ;++
                              0018  1121 ;
                              0018  1122 ; TEST DESCRIPTION:
                              0018  1123 ;
                              0018  1124 ; THIS TEST IS UTILIZED FOR THE R80 DISK DRIVE ONLY. THE PURPOSE
                              0018  1125 ; OF THIS TEST IS TO ISSUE A SEEK COMMAND TO THE R80 DISK DRIVE
                              0018  1126 ; USING CYLINDER 561 WHICH IS AN ILLEGAL CYLINDER.  THIS WILL CAUSE
                              0018  1127 ; THE R80 TO GENERATE A SKE ERROR IN THE DRIVE STATUS WORD. THEN
                              0018  1128 ; EXECUTE A RECAL FUNCTION AND CHECK THAT THE DRIVE RECOVERS PROPERLY.
                              0018  1129 ;
                              0018  1130 ; TEST STEPS:
                              0018  1131 ;
                              0018  1132 ; 1.        INITIALIZE SUBSYSTEM
                              0018  1133 ; 2.        ISSUE SEEK TO CYLINDER 561
                              0018  1134 ; 3.        CHECK DRIVE STATUS IS CORRECT (SKE)
                              0018  1135 ; 4.        EXECUTE RECAL FUNCTION
                              0018  1136 ; 5.        REPORT ANY POSITIONING, DRIVE, OR CONTROLLER ERRORS
                              0018  1137 ;
                              0018  1138 ; DEBUG:
                              0018  1139 ;
                              0018  1140 ;           NONE
                              0018  1141 ;
                              0018  1142 ;--
                              0018  1143
                              0018  1144
                              0018  1145            $DS_BGNDATA
     00000000 00000001        0018  1146            .LONG    1,LUN0
     00000001 00000001        0020  1147            .LONG    1,LUN1
     00000002 00000001        0028  1148            .LONG    1,LUN2
     00000003 00000001        0030  1149            .LONG    1,LUN3
                              0038  1150            $DS_ENDDATA
                              003C  1151
                              003C  1152            $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
                              003E  1153            GET_LUN -                                        ; GET LUN INFO
                              003E  1154              UNIT=           OFST_UNIT(AP)                  ; LOGICAL UNIT
            04 AC       DD    003E                  PUSHL    OFST_UNIT(AP)
     0000'CF    01      FB    0041                  CALLS    #1,GET_LUN$A
                              0046  1155            BDEVFOUND        5$                              ; IF DEVICE AVAILABLE GOTO 5$
       06 50    00      E2    0046                  BBSS     #0,R0,5$
          50    01      D0    004A                           MOVL     #1,R0          ; NOT AN ERROR
              00CA      31    004D                           BRW      TEST_012_X     ; EXIT TEST 12
   54     0000'CF       D0    0050  1156 5$:        MOVL     DRIVE_TBL,R4                            ; GET DRIVE INFO TABLE
          02    64      91    0055  1157            CMPB     OFST_DRVTYP(R4),#R80$T                  ; IF DRIVE IS AN R80
                06      13    0058  1158            BEQL     6$                                      ; THEN GOTO 6$
                              005A  1159            $DS_EXIT         TEST                            ; ELSE EXIT
        0000'CF         D4    0060  1160 6$:        CLRL     CONFIG                                  ; CLEAR ANY PROGRAM FLAGS
55     08 A4    01      C1    0064  1161            ADDL3    #1,OFST_LSTCYL(R4),R5                   ; CALC AN INVALID CYLINDER ADDRESS
       59    40 A4      D0    0069  1162            MOVL     OFST_MPBITS(R4),R9                      ; GET ADDRESS OF BIT MNEMONICS
     0000'CF    00      FB    006D  1163 10$:       CALLS    #0,SUBSYS_INIT                          ; CLEAR CONTROLLER AND DRIVE
          17 50         E8    0072  1164            BLBS     R0,20$                                  ; IF NO ERROR GOTO 20$
                              0075  1165            $DS_ERRHARD_S     ,L$UNIT,MSG14_SYSINIT,-
                              0075  1166                              PRINT_REG,R1
                              0085       ;;; TEST 12, SUBTEST 0, ERROR 1
                              008C  1167 20$:       $DS_CKLOOP        10$
                00      DD    0094  1168 25$:       PUSHL    #0                                      ; SELECT TRACK 0
```

```
                  55  DD  0096  1169          PUSHL   R5                          ; SPECIFY INVALID ADDRESS
          0000'CF  02  FB  0098  1170          CALLS   #2,POSITION_HDS             ; EXECUTE THE SEEK
       0000'CF  64 A4  CA  009D  1171          BICL2   OFST_STSDNC(R4),T.MP        ; RESET ANY DONT CARE BITS
       0000'CF  38 A4  D0  00A3  1172          MOVL    OFST_NMLSTS(R4),E.MP        ; START WITH NORMAL STATUS
  0000'CF  00000400 8F  C8  00A9  1173          BISL2   #MPR_80$M_SKE,E.MP          ; SET EXPECTED ERROR BIT
     0000'CF   0000'CF  D1  00B2  1174          CMPL    E.MP,T.MP                   ; IF STATUS WORD IS CORRECT
                  25  13  00B9  1175          BEQL    30$                         ; THEN GOTO 30$
                          00BB  1176          $DS_ERRHARD_S       ,L$UNIT,MS107A_SEEKERR,-
                          00BB  1177                              REG_ERRPT,#MSG47_DS_INC,-
                          00BB  1178                              E.MP,T.MP,R9
                          00D9       ;;; TEST 12, SUBTEST 0, ERROR 2
                          00E0  1179 30$:     $DS_CKLOOP          25$
                  00  DD  00E8  1180 35$:     PUSHL   #0                          ; SELECT TRACK 0
        FFFFFFFF 8F  DD  00EA  1181          PUSHL   #-1                         ; SELECT CYLINDER -1 FOR RECAL
          0000'CF  02  FB  00F0  1182          CALLS   #2,POSITION_HDS             ; EXECUTE RECAL
               17 50  E8  00F5  1183          BLBS    R0,40$                      ; IF NO ERRORS GOTO 30$
                          00F8  1184          $DS_ERRHARD_S       ,L$UNIT,MS107B_RECAL,-
                          00F8  1185                              PRINT_REG,R1
                          0108       ;;; TEST 12, SUBTEST 0, ERROR 3
                          010F  1186 40$:     $DS_CKLOOP          35$
                          0117  1187          $DS_ENDTEST
                          0122  1188          $DS_SBTTL <BASIC SEEK FUNCTION TEST>,ALIGN=LONG
                          0122                .SBTTL  TEST 13:  BASIC SEEK FUNCTION TEST
                          0122                $D2_SBTTL           013,<BASIC SEEK FUNCTION TEST>,<LONG>
                      00000000                          .PSECT  TEST_013, LONG, NOWRT
                          001C
```

```
                                             001C
                                             001C   ;++
                                             001C   ;
                                             001C   ; TEST DESCRIPTION:
                                             001C   ;
                                             001C   ; THIS TEST CHECKS THE BASIC POSITIONING LOGIC IN THE SUBSYSTEM
                                             001C   ; BUT WITH NO DRIVE MOVEMENT.  A READ HEADER COMMAND IS EXECUTED
                                             001C   ; AND THE CURRENT POSITION OF THE HEADS IS SAVED.  SEEK COMMANDS
                                             001C   ; ARE THEN ISSUED WITH A DIFFERENCE OF ZERO TO BOTH HEADS IN BOTH
                                             001C   ; DIRECTIONS.  AFTER EACH SEEK COMMAND A READ HEADER IS ISSUED
                                             001C   ; ONCE AGAIN.  THE HEADER READ BACK MUST INDICATE NO CHANGE IN
                                             001C   ; THE CURRENT POSITIONING OF THE HEADS AND THE PROPER HEAD MUST
                                             001C   ; BE SELECTED.  THE IDCMPR REGISTER CONTAINING DRIVE STATUS IS ALSO
                                             001C   ; CHECKED TO INSURE THAT IT INDICATES THAT THE PROPER HEAD WAS
                                             001C   ; SELECTED.
                                             001C   ;
                                             001C   ;
                                             001C   ; TEST STEPS:
                                             001C   ;
                                             001C   ; 1.      INITIALIZE SUBSYSTEM
                                             001C   ; 2.      EXECUTE READ HEADER CMD
                                             001C   ; 3.      CHECK FOR ANY ERRORS
                                             001C   ; 4.      SAVE CYLINDER ADDRESS FROM HEADER WORD
                                             001C   ; 5.      EXECUTE SEEK USING DATA FROM ARGUMENT TABLE
                                             001C   ; 6.      CHECK FOR ANY ERRORS
                                             001C   ; 7.      CHECK FOR PROPER DRIVE STATUS
                                             001C   ; 8.      EXECUTE READ HEADER COMMAND
                                             001C   ; 9.      CHECK FOR ANY ERRORS
                                             001C   ; 10.     CHECK HEADER WORD SHOWS NO DRIVE MOVEMENT TOOK PLACE
                                             001C   ; 11.     CHECK HEAD SELECT BIT IN HEADER WORD IS SAME AS HEAD
                                             001C   ;         USED IN SEEK COMMAND
                                             001C   ; 12.     CHECK HEAD SELECT BIT IN DRIVE STATUS WORD IS SAME AS
                                             001C   ;         HEAD USED IN SEEK COMMAND.
                                             001C   ; 13.     RETURN TO STEP ONE UNTIL ARGUMENT TABLE EXHAUSTED
                                             001C   ;
                                             001C   ; DEBUG:
                                             001C   ;
                                             001C   ;         DIFFERENCE REGISTER
                                             001C   ;         COUNT LOGIC
                                             001C   ;
                                             001C   ;--
                                             001C
                                             001C             $DS_BGNDATA
00000000 00000001 00000000 00000003          001C             .LONG    3,LUN0,RL02$T,HEAD_0
00000001 00000001 00000000 00000003          002C             .LONG    3,LUN0,RL02$T,HEAD_1
00000000 00000002 00000000 00000003          003C             .LONG    3,LUN0,R80$T,HEAD_0
00000000 00000001 00000001 00000003          004C             .LONG    3,LUN1,RL02$T,HEAD_0
00000001 00000001 00000001 00000003          005C             .LONG    3,LUN1,RL02$T,HEAD_1
00000000 00000002 00000001 00000003          006C             .LONG    3,LUN1,R80$T,HEAD_0
00000000 00000001 00000002 00000003          007C             .LONG    3,LUN2,RL02$T,HEAD_0
00000001 00000001 00000002 00000003          008C             .LONG    3,LUN2,RL02$T,HEAD_1
00000000 00000002 00000002 00000003          009C             .LONG    3,LUN2,R80$T,HEAD_0
00000000 00000001 00000003 00000003          00AC             .LONG    3,LUN3,RL02$T,HEAD_0
00000001 00000001 00000003 00000003          00BC             .LONG    3,LUN3,RL02$T,HEAD_1
00000000 00000002 00000003 00000003          00CC             .LONG    3,LUN3,R80$T,HEAD_0
                                             00DC             $DS_ENDDATA
```

```
                            00E0  1246
                            00E0  1247            $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
                            00E2  1248            GET_LUN -                                   ; GET LUN INFO
                            00E2  1249              UNIT= OFST_UNIT(AP)                       ; LOGCIAL UNIT
            04 AC     DD    00E2                  PUSHL   OFST_UNIT(AP)
     0000'CF    01    FB    00E5                  CALLS   #1,GET_LUN$A
                            00EA  1250            BDEVFOUND       1$                          ; IF DEV AVAILABLE GOTO 1$
       06 50    00    E2    00EA                  BBSS    #0,R0,1$
          50    01    D0    00EE                          MOVL    #1,R0          ; NOT AN ERROR
              018D    31    00F1                          BRW     TEST_013_X     ; EXIT TEST 13
     54   0000'CF     D0    00F4  1251 1$:        MOVL    DRIVE_TBL,R4                        ; GET DRIVE INFO TABLE
       08 AC    64    91    00F9  1252            CMPB    OFST_DRVTYP(R4),OFST_DRIV(AP)       ; IF SELECTED DRIVE = DATA TYPE
                06    13    00FD  1253            BEQL    2$                                  ; THEN GOTO 2$
                            00FF  1254            $DS_EXIT        TEST                        ; ELSE EXIT TEST
       55    48 A4    D0    0105  1255 2$:        MOVL    OFST_HDRTMP(R4),R5                  ; GET ADDR OF HDR TEMPLATE
       56    40 A4    D0    0109  1256            MOVL    OFST_MPBITS(R4),R6                  ; GET ADDR OF STATUS BIT MNEMONICS
       59    44 A4    D0    010D  1257            MOVL    OFST_HDBITS(R4),R9                  ; GET ADDR OF HDR BIT MNEMONICS
           0000'CF    D4    0111  1258            CLRL    CONFIG                              ; CLEAR PROGRAM FLAGS
           0000'CF    D4    0115  1259            CLRL    DIFF_WORD                           ; SET DIFFERENCE WRD FOR THIS TEST
     0000'CF    00    FB    0119  1260 10$:       CALLS   #0,SUBSYS_INIT                      ; CLEAR CONTROLLER AND DRIVE
          17 50       E8    011E  1261            BLBS    R0,20$                              ; BRANCH IF NO ERRORS
                            0121  1262            $DS_ERRHARD_S    ,L$UNIT,MSG14_SYSINIT,-
                            0121  1263                             PRINT_REG,R1
                            0131       ;;; TEST 13, SUBTEST 0, ERROR 1
                            0138  1264 20$:       $DS_CKLOOP      10$
                00    DD    0140  1265 25$:       PUSHL   #0                                  ; DEFINE FLAGS
     0000'CF    01    FB    0142  1266            CALLS   #1,GET_POSITION                     ; GET CURRENT DRIVE POSITION
          17 50       E8    0147  1267            BLBS    R0,30$                              ; BRANCH IF NO ERRORS
                            014A  1268            $DS_ERRHARD_S    ,L$UNIT,MSG19_READH,-
                            014A  1269                             PRINT_REG,R1
                            015A       ;;; TEST 13, SUBTEST 0, ERROR 2
                            0161  1270 30$:       $DS_CKLOOP      25$
                57    7C    0169  1271            CLRQ    R7                                  ; CLEAR REGS R7 AND R8
     65   0000'CF     F0    016B  1272            INSV    CURR_CYLN,OFST_CYLPOS(R5),-         ; SET EXPECTED CYLINDER
       57    04 A5          0170  1273                    OFST_CYLSIZ(R5),R7                  ;
   05 A5    0C AC     F0    0173  1274            INSV    OFST_SKHD(AP),OFST_TRKPOS(R5),-     ; SET EXPECTED TRACK
       57    09 A5          0178  1275                    OFST_TRKSIZ(R5),R7                  ;
58   10   10  0000'CF F0    017B  1276            INSV    CURR_CYLN,#16,#16,R8                ; LOAD DESTINATION CYLINDER
 58   08   08   0C AC F0    0182  1277            INSV    OFST_SKHD(AP),#8,#8,R8              ; LOAD DESTINATION HEAD
  0000'CF   0000'CF   D0    0188  1278            MOVL    CURR_CYLN,DEST_CYLN                 ; SET UP DESTINATION CYLINDER
                00    DD    018F  1279            PUSHL   #0                                  ; IDCMPR DATA
                58    DD    0191  1280            PUSHL   R8                                  ; IDCDAR DATA
                00    DD    0193  1281            PUSHL   #0                                  ; IDCCNT DATA
                00    DD    0195  1282            PUSHL   #0                                  ; IDCBAR DATA
                06    DD    0197  1283            PUSHL   #SEEK                               ; SEEK COMMAND
     0000'CF    05    FB    0199  1284            CALLS   #5,START_CMD                        ; EXECUTE SEEK COMMAND
     0000'CF    00    FB    019E  1285            CALLS   #0,STATUS_CHK                       ; CHECK FOR ANY ERRORS
          1C 50       E8    01A3  1286            BLBS    R0,40$                              ; BRANCH IF NO ERRORS
     0000'CF    08    C8    01A6  1287            BISL2   #PRT$M_EXT_1,RGLIST                 ; ADD MORE INFO TO ERR RPT
                            01AB  1288            $DS_ERRHARD_S    ,L$UNIT,MSG45_SK_ZERO,-
                            01AB  1289                             PRINT_REG,R1
                            01BB       ;;; TEST 13, SUBTEST 0, ERROR 3
                            01C2  1290 40$:       $DS_CKLOOP      25$
       5A    38 A4    3C    01CB  1291            MOVZWL  OFST_NMLSTS(R4),R10                 ; SET EXPECTED DRIVE STATUS
 5A   01   06   0C AC F0    01CF  1292            INSV    OFST_SKHD(AP),#6,#1,R10             ; OR IN EXPECTED HEAD
  0000'CF    64 A4    CA    01D5  1293            BICL2   OFST_STSDNC(R4),T.MP                ; CLEAR EXTRA BITS IN STATUS
```

```
          0000'CF    5A    D1  01DB  1294          CMPL     R10,T.MP                              ; DRIVE STATUS = EXPECTED ?
                     23    13  01E0  1295          BEQL     50$                                   ; BRANCH IF IT DOES
                               01E2  1296          $DS_ERRHARD_S     ,L$UNIT,MSG45_SK_ZERO,-
                               01E2  1297                            REG_ERRPT,#MSG47_DS_INC,-
                               01E2  1298                            R10,T.MP,R6
                               01FE           ;;; TEST 13, SUBTEST 0, ERROR 4
                               0205  1299 50$:     $DS_CKLOOP        25$
                     00    DD  020E  1300 55$:     PUSHL    #0                                    ; DEFINE FLAGS
          0000'CF    01    FB  0210  1301          CALLS    #1,GET_POSITION                       ; GET CURRENT DRIVE POSITION
                  17 50    E8  0215  1302          BLBS     R0,60$                                ; BRANCH IF NO ERRORS
                               0218  1303          $DS_ERRHARD_S     ,L$UNIT,MSG19_READH,-
                               0218  1304                            PRINT_REG,R1
                               0228           ;;; TEST 13, SUBTEST 0, ERROR 5
                               022F  1305 60$:     $DS_CKLOOP        55$
            0A A5    00    F0  0237  1306          INSV     #0,OFST_SECPOS(R5),-                  ; CLEAR OUT SECTOR BITS IN HDR
       0000'CF    0E A5        023B  1307                   OFST_SECSIZ(R5),T.HDR                 ;
       0000'CF    54 A4    CA  0240  1308          BICL2    OFST_HDRDNC(R4),T.HDR                 ; CLEAR DONT CARE BITS
          0000'CF    57    D1  0246  1309          CMPL     R7,T.HDR                              ; SEEK TO RIGHT CYLN AND HEAD ?
                     28    13  024B  1310          BEQL     70$                                   ; BRANCH IF IT DID
                     18    C8  024D  1311          BISL2    #<PRT$M_EXT_1!PRT$M_EXT_2>,-          ; ADD EXTENDED INFO TO ERR REPORT
             0000'CF           024F  1312                   RGLIST
                               0252  1313          $DS_ERRHARD_S     ,L$UNIT,MSG45_SK_ZERO,-
                               0252  1314                            REG_ERRPT,#MSG46_POS_ERR,-
                               0252  1315                            R7,T.HDR,R9
                               026E           ;;; TEST 13, SUBTEST 0, ERROR 6
                               0275  1316 70$:     $DS_CKLOOP        25$
                               027E  1317          $DS_ENDTEST
                               0289  1318          $DS_SBTTL <HEAD PROXIMITY TEST - RL02>,ALIGN=LONG
                               0289                .SBTTL  TEST 14:  HEAD PROXIMITY TEST - RL02
                               0289                $D2_SBTTL         014,<HEAD PROXIMITY TEST - RL02>,<LONG>
                           00000000                         .PSECT   TEST_014, LONG, NOWRT
                               0020
```

```
                             0020  1320 ;++
                             0020  1321 ;
                             0020  1322 ; TEST DESCRIPTION:
                             0020  1323 ;
                             0020  1324 ;       THIS TEST WILL CHECK HEAD PROXIMITY BY EXECUTING ZERO
                             0020  1325 ;       DIFFERENCE SEEKS TO BOTH HEADS AND VERIFY THAT THEY
                             0020  1326 ;       ARE ON THE SAME CYLINDER.
                             0020  1327 ;
                             0020  1328 ; TEST STEPS:
                             0020  1329 ;
                             0020  1330 ;       1.  INITIALIZE SUBSYSTEM
                             0020  1331 ;       2.  EXECUTE ZERO DIFFERENCE SEEK TO HEAD 0
                             0020  1332 ;       3.  RECORD CURRENT POSITION OF HEADS
                             0020  1333 ;       4.  EXECUTE ZERO DIFFERENCE SEEK TO HEAD 1
                             0020  1334 ;       5.  GET CURRENT POSITION OF HEADS
                             0020  1335 ;       6.  VERIFY THAT BOTH HEADS ARE FOLLOWING SAVE TRACK
                             0020  1336 ;
                             0020  1337 ; DEBUG
                             0020  1338 ;
                             0020  1339 ;       CHECK HEAD ALIGNMENT
                             0020  1340 ;
                             0020  1341 ;--
                             0020  1342
                             0020  1343
                             0020  1344         $DS_BGNDATA
     00000000 00000001       0020  1345         .LONG   1,LUN0
     00000001 00000001       0028  1346         .LONG   1,LUN1
     00000002 00000001       0030  1347         .LONG   1,LUN2
     00000003 00000001       0038  1348         .LONG   1,LUN3
                             0040  1349         $DS_ENDDATA
                             0044  1350
                             0044  1351         $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
                             0046  1352         GET_LUN -                               ; GET LUN INFO
                             0046  1353           UNIT=         OFST_UNIT(AP)           ; LOGICAL UNIT
            04 AC      DD    0046               PUSHL   OFST_UNIT(AP)
     0000'CF    01     FB    0049               CALLS   #1,GET_LUN$A
                             004E  1354         BDEVFOUND       5$                      ; IF DEV AVAILABLE GOTO 5$
       06 50    00     E2    004E               BBSS    #0,R0,5$
          50    01     D0    0052                       MOVL    #1,R0           ; NOT AN ERROR
              016A     31    0055                       BRW     TEST_014_X      ; EXIT TEST 14
  53    0000'CF        D0    0058  1355 5$:     MOVL    DRIVE_TBL,R3            ; GET DRIVE INFO TABLE
          01    63     91    005D  1356         CMPB    OFST_DRVTYP(R3),#RL02$T ; IF DRIVE IS AN RL02
                06     13    0060  1357         BEQL    6$                      ; THEN GOTO 6$
                             0062  1358         $DS_EXIT        TEST            ; ELSE EXIT
          0000'CF      D4    0068  1359 6$:     CLRL    CONFIG                  ; CLEAR PROGRAM FLAGS
     0000'CF    00     FB    006C  1360 10$:    CALLS   #0,SUBSYS_INIT          ; INIT RL02 SUBSYSTEM
          17 50        E8    0071  1361         BLBS    R0,20$                  ; BRANCH IF NO ERRORS
                             0074  1362         $DS_ERRHARD_S   ,L$UNIT,MSG14_SYSINIT,-
                             0074  1363                         PRINT_REG,R1
                             0084       ;;; TEST 14, SUBTEST 0, ERROR 1
                             008B  1364 20$:    $DS_CKLOOP      10$
                00     DD    0093  1365 22$:    PUSHL   #0                      ; DEFINE FLAGS
     0000'CF    01     FB    0095  1366         CALLS   #1,GET_POSITION         ; GET CURRENT DRIVE POSITION
          17 50        E8    009A  1367         BLBS    R0,25$                  ; BRANCH IF NO ERRORS
                             009D  1368         $DS_ERRHARD_S   ,L$UNIT,MSG19_READH,-
                             009D  1369                         PRINT_REG,R1
```

```
                        00AD        ;;; TEST 14, SUBTEST 0, ERROR 2
                        00B4  1370 25$:     $DS_CKLOOP      22$
              53    D4  00BC  1371          CLRL     R3                                    ; CLEAR R3
  53   10   10  0000'CF    F0  00BE  1372          INSV     CURR_CYLN,#16,#16,R3                  ; LOAD CYLINDER
              00    DD  00C5  1373          PUSHL    #0                                    ; IDCMPR DATA
              53    DD  00C7  1374          PUSHL    R3                                    ; IDCDAR CURR CYLN, HEAD 0
              00    DD  00C9  1375          PUSHL    #0                                    ; IDCNCR DATA
              00    DD  00CB  1376          PUSHL    #0                                    ; IDCBAR DATA
              06    DD  00CD  1377          PUSHL    #SEEK                                 ; DEFINE COMMAND
      0000'CF   05    FB  00CF  1378          CALLS    #5,START_CMD                          ; EXECUTE SEEK CMD
      0000'CF   00    FB  00D4  1379          CALLS    #0,STATUS_CHK                         ; CHECK FOR ANY ERRORS
           17 50    E8  00D9  1380          BLBS     R0,30$                                ; BRANCH IF NONE
                        00DC  1381          $DS_ERRHARD_S    ,L$UNIT,MSG45_SK_ZERO,-
                        00DC  1382                           PRINT_REG,R1
                        00EC        ;;; TEST 14, SUBTEST 0, ERROR 3
                        00F3  1383 30$:     $DS_CKLOOP      22$
              00    DD  00FB  1384 35$:     PUSHL    #0                                    ; DEFINE FLAGS
      0000'CF   01    FB  00FD  1385          CALLS    #1,GET_POSITION                       ; GET CURRENT DRIVE POSITION
           17 50    E8  0102  1386          BLBS     R0,40$                                ; BRANCH IF NO ERRORS
                        0105  1387          $DS_ERRHARD_S    ,L$UNIT,MSG19_READH,-
                        0105  1388                           PRINT_REG,R1
                        0115        ;;; TEST 14, SUBTEST 0, ERROR 4
                        011C  1389 40$:     $DS_CKLOOP      35$
        54  0000'CF    D0  0124  1390          MOVL     CURR_CYLN,R4                          ; SAVE CYLN FOR HEAD 0
   53   08   08   01    F0  0129  1391          INSV     #1,#8,#8,R3                           ; SELECT HEAD 1
              00    DD  012E  1392          PUSHL    #0                                    ; IDCMPR DATA
              53    DD  0130  1393          PUSHL    R3                                    ; IDCDAR = CURR CYLN, HEAD 1
              00    DD  0132  1394          PUSHL    #0                                    ; IDCCNT DATA
              00    DD  0134  1395          PUSHL    #0                                    ; IDCBAR DATA
              06    DD  0136  1396          PUSHL    #SEEK                                 ; DEFINE COMMAND
      0000'CF   05    FB  0138  1397          CALLS    #5,START_CMD                          ; EXECUTE SEEK CMD
      0000'CF   00    FB  013D  1398          CALLS    #0,STATUS_CHK                         ; CHECK FOR ANY ERRORS
           17 50    E8  0142  1399          BLBS     R0,50$                                ; BRANCH IF NONE
                        0145  1400          $DS_ERRHARD_S    ,L$UNIT,MSG45_SK_ZERO,-
                        0145  1401                           PRINT_REG,R1
                        0155        ;;; TEST 14, SUBTEST 0, ERROR 5
                        015C  1402 50$:     $DS_CKLOOP      22$
              00    DD  0165  1403 55$:     PUSHL    #0                                    ; DEFINE FLAGS
      0000'CF   01    FB  0167  1404          CALLS    #1,GET_POSITION                       ; GET CURRENT DRIVE POSITION
           17 50    E8  016C  1405          BLBS     R0,60$                                ; BRANCH IF NO ERRORS
                        016F  1406          $DS_ERRHARD_S    ,L$UNIT,MSG19_READH,-
                        016F  1407                           PRINT_REG,R1
                        017F        ;;; TEST 14, SUBTEST 0, ERROR 6
                        0186  1408 60$:     $DS_CKLOOP      55$
        54  0000'CF    D1  018E  1409          CMPL     CURR_CYLN,R4                          ; IS HEAD 1 ON SAME TRACK AS 0
              21    13  0193  1410          BEQL     70$                                   ; BRANCH IF IT IS
                        0195  1411          $DS_ERRHARD_S    ,L$UNIT,MSG19_READH,-
                        0195  1412                           REG_ERRPT,#MS45A_HD_PROX,-
                        0195  1413                           R4,CURR_CYLN
                        01AF        ;;; TEST 14, SUBTEST 0, ERROR 7
                        01B6  1414 70$:     $DS_CKLOOP      22$
                        01BF  1415          $DS_ENDTEST
                        01CA  1416          $DS_SBTTL <DIFFERENCE OF ONE SEEK TEST>,ALIGN=LONG
                        01CA                .SBTTL  TEST 15:  DIFFERENCE OF ONE SEEK TEST
                        01CA                $D2_SBTTL       015,<DIFFERENCE OF ONE SEEK TEST>,<LONG>
                    00000000                        .PSECT  TEST_015, LONG, NOWRT
```

```
0020
```

```
                                             0020
                                             0020  ;++
                                             0020  ;
                                             0020  ; TEST DESCRIPTION:
                                             0020  ;
                                             0020  ; THIS TEST CHECKS THE POSITIONING LOGIC IN THE SUBSYSTEM
                                             0020  ; WITH MINIMUM DRIVE MOVEMENT.  A READ HEADER COMMAND IS EXECUTED
                                             0020  ; AND THE CURRENT POSITION OF THE HEADS IS SAVED.  SEEK COMMANDS
                                             0020  ; ARE THEN ISSUED WITH A DIFFERENCE OF ONE IN BOTH
                                             0020  ; DIRECTIONS.  AFTER EACH SEEK COMMAND A READ HEADER IS ISSUED
                                             0020  ; ONCE AGAIN.  THE HEADER READ BACK MUST INDICATE A CHANGE IN
                                             0020  ; THE CURRENT POSITIONING OF THE HEADS AND THE PROPER HEAD MUST
                                             0020  ; BE SELECTED.
                                             0020  ;
                                             0020  ; NOTE: IF DRIVE IS AN RL02 THEN BOTH HEADS ARE TESTED HERE.
                                             0020  ;
                                             0020  ; TEST STEPS:
                                             0020  ;
                                             0020  ; 1.     INITIALIZE SUBSYSTEM
                                             0020  ; 2.     EXECUTE READ HEADER CMD
                                             0020  ; 3.     CHECK FOR ANY ERRORS
                                             0020  ; 4.     SAVE CYLINDER ADDRESS FROM HEADER WORD
                                             0020  ; 5.     EXECUTE SEEK USING DATA FROM ARGUMENT TABLE
                                             0020  ; 6.     CHECK FOR ANY ERRORS
                                             0020  ; 7.     CHECK THAT DRIVE READY IS RESET
                                             0020  ; 8.     CHECK THAT DRIVE STATUS INDICATES DRIVE IS SEEKING
                                             0020  ; 9.     WAIT FOR DRIVE READY
                                             0020  ; 10.    GET DRIVE STATUS AND CHECK THAT DRIVE IS LOCKED ON TRACK
                                             0020  ; 11.    EXECUTE READ HEADER COMMAND
                                             0020  ; 12.    CHECK FOR ANY ERRORS
                                             0020  ; 13.    CHECK THAT HEADER WORD #1 INDICATES DRIVE SEEKED TO RIGHT CYLINDER
                                             0020  ; 14.    RETURN TO STEP ONE UNTIL ARGUMENT TABLE EXHAUSTED
                                             0020  ;
                                             0020  ; DEBUG:
                                             0020  ;
                                             0020  ;        DIFFERENCE REGISTER
                                             0020  ;        COUNT LOGIC
                                             0020  ;        TRACK CROSSING DETECTION
                                             0020  ;
                                             0020  ;--
                                             0020
                                             0020          $DS_BGNDATA
00000000 00000001 00000000 00000003          0020          .LONG   3,LUN0,RL02$T,HEAD_0
00000001 00000001 00000000 00000003          0030          .LONG   3,LUN0,RL02$T,HEAD_1
00000000 00000002 00000000 00000003          0040          .LONG   3,LUN0,R80$T,HEAD_0
00000000 00000001 00000001 00000003          0050          .LONG   3,LUN1,RL02$T,HEAD_0
00000001 00000001 00000001 00000003          0060          .LONG   3,LUN1,RL02$T,HEAD_1
00000000 00000002 00000001 00000003          0070          .LONG   3,LUN1,R80$T,HEAD_0
00000000 00000001 00000002 00000003          0080          .LONG   3,LUN2,RL02$T,HEAD_0
00000001 00000001 00000002 00000003          0090          .LONG   3,LUN2,RL02$T,HEAD_1
00000000 00000002 00000002 00000003          00A0          .LONG   3,LUN2,R80$T,HEAD_0
00000000 00000001 00000003 00000003          00B0          .LONG   3,LUN3,RL02$T,HEAD_0
00000001 00000001 00000003 00000003          00C0          .LONG   3,LUN3,RL02$T,HEAD_1
00000000 00000002 00000003 00000003          00D0          .LONG   3,LUN3,R80$T,HEAD_0
                                             00E0          $DS_ENDDATA
                                             00E4
```

```
                            00E4  1474          $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
                            00E6  1475          GET_LUN -                              ; GET LUN INFO
                            00E6  1476            UNIT= OFST_UNIT(AP)                  ; LOGICAL UNIT
             04 AC     DD   00E6                PUSHL   OFST_UNIT(AP)
        0000'CF    01  FB   00E9                CALLS   #1,GET_LUN$A
                            00EE  1477          BDEVFOUND       1$                     ; IF DEV AVAILABLE GOTO 1$
          06 50    00  E2   00EE                BBSS    #0,R0,1$
             50    01  D0   00F2                        MOVL    #1,R0        ; NOT AN ERROR
                 029B  31   00F5                        BRW     TEST_015_X   ; EXIT TEST 15
       54   0000'CF    D0   00F8  1478 1$:      MOVL    DRIVE_TBL,R4                   ; GET ADDR OF DRIVE INFO TABLE
          08 AC    64  91   00FD  1479          CMPB    OFST_DRVTYP(R4),OFST_DRIV(AP)  ; IF DRIVE TYPE = DATA TYPE
                   06  13   0101  1480          BEQL    5$                             ; THEN GOTO 5$
                            0103  1481          $DS_EXIT        TEST                   ; ELSE EXIT TEST
         55     48 A4  D0   0109  1482 5$:      MOVL    OFST_HDRTMP(R4),R5             ; GET ADDR OF HEADER TEMPLATE
         56     40 A4  D0   010D  1483          MOVL    OFST_MPBITS(R4),R6             ; GET DRIVE STATUS BIT MNEMONICS
         59     44 A4  D0   0111  1484          MOVL    OFST_HDBITS(R4),R9             ; GET HDR BIT MNEMONICS
             0000'CF   D4   0115  1485          CLRL    CONFIG                         ; CLEAR PROGRAM FLAGS
        0000'CF    01  D0   0119  1486          MOVL    #1,DIFF_WORD                   ; SET DIFFERENCE WRD FOR THIS TEST
                   5A  D4   011E  1487          CLRL    R10                            ; INIT DIRECTION COUNTER
        0000'CF    00  FB   0120  1488 10$:     CALLS   #0,SUBSYS_INIT                 ; CLEAR CONTROLLER AND DRIVE
             17 50     E8   0125  1489          BLBS    R0,20$                         ; BRANCH IF NO ERRORS
                            0128  1490          $DS_ERRHARD_S   ,L$UNIT,MSG14_SYSINIT,-
                            0128  1491                          PRINT_REG,R1
                            0138       ;;; TEST 15, SUBTEST 0, ERROR 1
                            013F  1492 20$:     $DS_CKLOOP      10$
                   00  DD   0147  1493 25$:     PUSHL   #0                             ; DEFINE FLAGS
        0000'CF    01  FB   0149  1494          CALLS   #1,GET_POSITION                ; GET CURRENT DRIVE POSITION
             17 50     E8   014E  1495          BLBS    R0,30$                         ; BRANCH IF NO ERRORS
                            0151  1496          $DS_ERRHARD_S   ,L$UNIT,MSG19_READH,-
                            0151  1497                          PRINT_REG,R1
                            0161       ;;; TEST 15, SUBTEST 0, ERROR 2
                            0168  1498 30$:     $DS_CKLOOP      25$
             53    01  D0   0170  1499          MOVL    #1,R3                          ; SET INITIAL DIRECTION FORWARD
       08 A4   0000'CF D1   0173  1500 35$:     CMPL    CURR_CYLN,OFST_LSTCYL(R4)      ; IS DRIVE ON LAST CYL
                   03  12   0179  1501          BNEQ    40$                            ; BRANCH IF NOT
             53    53  CE   017B  1502          MNEGL   R3,R3                          ; ELSE SET DIR = REVERSE
             0000'CF   D5   017E  1503 40$:     TSTL    CURR_CYLN                      ; ARE WE AT CYLN ZERO ?
                   03  12   0182  1504          BNEQ    45$                            ; BRANCH IF NO
             53    01  D0   0184  1505          MOVL    #1,R3                          ; ELSE MAKE DIR FORWARD
  0000'CF    0000'CF   53  C1   0187  1506 45$:  ADDL3   R3,CURR_CYLN,DEST_CYLN      ; CALC DESTINATION CYLN
                   57  7C   018F  1507          CLRQ    R7                             ; CLEAR REGS R7 AND R8
         65    0000'CF F0   0191  1508          INSV    DEST_CYLN,OFST_CYLPOS(R5),-    ; SET EXECTED CYLINDER IN HDR
          57    04 A5       0196  1509                  OFST_CYLSIZ(R5),R7             ;
       05 A5    0C AC  F0   0199  1510          INSV    OFST_SKHD(AP),OFST_TRKPOS(R5),- ; SET EXPECTED TRACK IN HDR
          57    09 A5       019E  1511                  OFST_TRKSIZ(R5),R7             ;
  58   10   10  0000'CF F0  01A1  1512          INSV    DEST_CYLN,#16,#16,R8           ; LOAD DESTINATION CYLINDER
    58   08   08    0C AC  F0   01A8  1513      INSV    OFST_SKHD(AP),#8,#8,R8         ; LOAD DESTINATION TRACK
      0000'CF    86 8F  9A   01AE  1514         MOVZBL  #<CSR$M_CRDY!SEEK>,E.CS        ; SET EXPECTED IDCCSR REG
    0000'CF   0000'CF   C8   01B4  1515         BISL2   P$UNIT,E.CS                    ; OR IN DRIVE NUMBER
      0000'CF    58 A4  C8   01BB  1516         BISL2   OFST_CSRTYP(R4),E.CS           ; OR IN EXPECTED ATTENTION BIT
      0000'CF    3C A4  D0   01C1  1517         MOVL    OFST_SEKSTS(R4),E.MP           ; DRIVE STATUS = SEEKING
  0000'CF   01   06    0C AC  F0  01C7  1518    INSV    OFST_SKHD(AP),#6,#1,E.MP       ; SET EXPECTED RL HEAD SELECT BIT
                   00  DD   01CF  1519          PUSHL   #0                             ; IDCMPR DATA
                   58  DD   01D1  1520          PUSHL   R8                             ; IDCDAR DATA
                   00  DD   01D3  1521          PUSHL   #0                             ; IDCCNT DATA
                   00  DD   01D5  1522          PUSHL   #0                             ; IDCBAR DATA
```

```
                         06   DD  01D7           PUSHL   #SEEK                               ; SEEK COMMAND
              0000'CF    05   FB  01D9           CALLS   #5,START_CMD                        ; EXECUTE SEEK COMMAND
   0000'CF    00000084   8F   D0  01DE           MOVL    #<NOCHK$M_DRDY!NOCHK$M_ATTN>,CONFIG ; SET PROG FLAG FOR SUBR
              0000'CF    00   FB  01E7           CALLS   #0,STATUS_CHK                       ; CHECK FOR ANY ERRORS
                      1C 50   E8  01EC           BLBS    R0,80$                              ; BR IF NO ERRORS
              0000'CF    08   C8  01EF           BISL2   #PRT$M_EXT_1,RGLIST                 ; PNT EXTENDED INFO
                                  01F4           $DS_ERRHARD_S   ,L$UNIT,MSG48_SK_ONE,-
                                  01F4                           PRINT_REG,R1
                                  0204      ;;; TEST 15, SUBTEST 0, ERROR 3
                                  020B      80$:    $DS_CKLOOP      25$
         2E 0000'CF      00   E1  0214           BBC     #CSR$V_DRDY,T.CS,90$                ; BRANCH IF DRIVE READY CLEAR
              0000'CF    08   C8  021A           BISL2   #PRT$M_EXT_1,RGLIST                 ; PNT EXTENDED INFO
                                  021F           $DS_ERRHARD_S   ,L$UNIT,MSG48_SK_ONE,-
                                  021F                           REG_ERRPT,#MSG49_DRDY_NC,-
                                  021F                           E.CS,T.CS,#CSR_BITS
                                  0241      ;;; TEST 15, SUBTEST 0, ERROR 4
                                  0248      90$:    $DS_CKLOOP      25$
          0000'CF     64 A4   CA  0251           BICL2   OFST_STSDNC(R4),T.MP                ; CLEAR ANY DONT CARE BITS
        0000'CF     0000'CF   D1  0257      95$:  CMPL    E.MP,T.MP                          ; EXPECTED = ACTUAL ?
                         2A   13  025E           BEQL    100$                                ; BRANCH IF IT DOES
              0000'CF    08   C8  0260           BISL2   #PRT$M_EXT_1,RGLIST                 ; PNT EXTENDED INFO
                                  0265           $DS_ERRHARD_S   ,L$UNIT,MSG48_SK_ONE,-
                                  0265                           REG_ERRPT,#MSG47_DS_INC,-
                                  0265                           E.MP,T.MP,R6
                                  0283      ;;; TEST 15, SUBTEST 0, ERROR 5
                                  028A      100$:   $DS_CKLOOP      25$
              0000'CF    01   D0  0293           MOVL    #<NOCHK$M_CRDY>,CONFIG              ; SET PROG FLAG
              0000'CF    00   FB  0298           CALLS   #0,STATUS_CHK                       ; WAIT FOR DRIVE READY
                      1C 50   E8  029D           BLBS    R0,110$                             ; BRANCH IF NO ERRORS
              0000'CF    08   C8  02A0           BISL2   #PRT$M_EXT_1,RGLIST                 ; PNT EXTENDED INFO
                                  02A5           $DS_ERRHARD_S   ,L$UNIT,MSG48_SK_ONE,-
                                  02A5                           PRINT_REG,R1
                                  02B5      ;;; TEST 15, SUBTEST 0, ERROR 6
                                  02BC      110$:   $DS_CKLOOP      25$
          0000'CF     38 A4   D0  02C5           MOVL    OFST_NMLSTS(R4),E.MP                ; SET EXPECTED NORMAL STATUS
   0000'CF    01   06  0C AC  F0  02CB           INSV    OFST_SKHD(AP),#6,#1,E.MP            ; OR IN RL HEAD SELECT BIT
          0000'CF     64 A4   CA  02D3           BICL2   OFST_STSDNC(R4),T.MP                ; CLEAR ANY DONT CARE BITS
        0000'CF     0000'CF   D1  02D9           CMPL    E.MP,T.MP                           ; EXPECTED = ACTUAL ?
                         2A   13  02E0           BEQL    120$                                ; BRANCH IF IT DOES
              0000'CF    08   C8  02E2           BISL2   #PRT$M_EXT_1,RGLIST                 ; PNT EXTENDED INFO
                                  02E7           $DS_ERRHARD_S   ,L$UNIT,MSG48_SK_ONE,-
                                  02E7                           REG_ERRPT,#MSG47_DS_INC,-
                                  02E7                           E.MP,T.MP,R6
                                  0305      ;;; TEST 15, SUBTEST 0, ERROR 7
                                  030C      120$:   $DS_CKLOOP      25$
                         00   DD  0315      125$:   PUSHL   #0                           ; DEFINE FLAGS
              0000'CF    01   FB  0317           CALLS   #1,GET_POSITION                     ; GET CURRENT DRIVE POSITION
                      17 50   E8  031C           BLBS    R0,130$                             ; BRANCH IF NO ERRORS
                                  031F           $DS_ERRHARD_S   ,L$UNIT,MSG19_READH,-
                                  031F                           PRINT_REG,R1
                                  032F      ;;; TEST 15, SUBTEST 0, ERROR 8
                                  0336      130$:   $DS_CKLOOP      125$
              0A A5      00   F0  033E           INSV    #0,OFST_SECPOS(R5),-                ; CLEAR SECTOR BITS
          0000'CF     0E A5       0342                   OFST_SECSIZ(R5),T.HDR               ;
          0000'CF     54 A4   CA  0347           BICL2   OFST_HDRDNC(R4),T.HDR               ; CLEAR DONT CARE BITS
              0000'CF    57   D1  034D           CMPL    R7,T.HDR                            ; RIGHT HEAD AND CYLINDER ?
```

```
          28   13  0352  1573          BEQL    140$                              ; BRANCH IF YES
          18   C8  0354  1574          BISL2   #<PRT$M_EXT_1!PRT$M_EXT_2>,-      ; PRINT EXTENDED INFO IN ERR RPT
     0000'CF       0356  1575                  RGLIST
                   0359  1576          $DS_ERRHARD_S     ,L$UNIT,MSG48_SK_ONE,-
                   0359  1577                            REG_ERRPT,#MSG46_POS_ERR,-
                   0359  1578                            R7,T.HDR,R9
                   0375        ;;; TEST 15, SUBTEST 0, ERROR 9
                   037C  1579 140$:    $DS_CKLOOP        25$
          5A   D6  0385  1580          INCL    R10                               ; INCR DIRECTION CNT
       06 5A   E9  0387  1581          BLBC    R10,150$                          ; BR IF DONE
     53   53   CE  038A  1582          MNEGL   R3,R3                             ; ELSE SWAP DIRECTION
        FDE3   31  038D  1583          BRW     35$                               ; GO TEST THIS DIRECTION
                   0390  1584 150$:
                   0390  1585          $DS_ENDTEST
                   039B  1586          $DS_SBTTL <INCREMENTAL FORWARD AND REVERSE SEEK TEST>,ALIGN=LONG
                   039B                .SBTTL  TEST 16:  INCREMENTAL FORWARD AND REVERSE SEEK TEST
                   039B                $D2_SBTTL         016,<INCREMENTAL FORWARD AND REVERSE SEEK TEST>,<LONG>
              00000000                         .PSECT  TEST_016, LONG, NOWRT
                   002C
```

```
                                            002C
                                            002C   ;++
                                            002C   ;
                                            002C   ; TEST DESCRIPTION:
                                            002C   ;
                                            002C   ; TEST ONE TRACK SEEKS WITH BOTH HEADS ACROSS ALL TRACKS.  ALSO
                                            002C   ; INNER GUARD BAND DETECTION WILL BE CHECKED WHEN LAST CYLINDER
                                            002C   ; IS REACHED IF THE DRIVE UNDER TEST IS AN RL02.
                                            002C   ;
                                            002C   ; TEST STEPS:
                                            002C   ;
                                            002C   ; A.   SUBTEST 1
                                            002C   ;           1.  INITIALIZE SUBSYSTEM
                                            002C   ;           2.  POSITION HEADS TO CYLINDER 0
                                            002C   ;           3.  CHECK FOR ANY ERRORS
                                            002C   ;           4.  EXECUTE +1 SEEK CMD
                                            002C   ;           5.  CHECK FOR ANY ERRORS
                                            002C   ;           6.  VERIFY SEEK WITH READ HEADER
                                            002C   ;           7.  RETURN TO STEP 4. UNTIL LAST CYLINDER IS REACHED
                                            002C   ;
                                            002C   ; B.   SUBTEST 2
                                            002C   ;           1.  INITIALIZE SUBSYSTEM
                                            002C   ;           2.  POSITION HEADS TO LAST CYLINDER
                                            002C   ;           3.  CHECK FOR ERRORS
                                            002C   ;           4.  EXECUTE ANOTHER +1 SEEK CMD
                                            002C   ;           5.  CHECKK FOR ERRORS
                                            002C   ;           6.  VERIIFY DRIVE HAS REPOSITIONED TO LAST EVEN TRACK (510
                                            002C   ;               FOR RL02) OR THAT DRIVE POSITION DID NOT CHANGE (R80)
                                            002C   ;
                                            002C   ; C.   SUBTEST 3
                                            002C   ;           1.  INITIALIZE SUBSYSTEM
                                            002C   ;           2.  POSITION HEADS TO LAST CYLINDER
                                            002C   ;           3.  CHECK FOR ANY ERRORS
                                            002C   ;           4.  EXECUTE -1 SEEK CMD
                                            002C   ;           5.  CHECK FOR ANY ERRORS
                                            002C   ;           6.  VERIFY SEEK WITH READ HEADER
                                            002C   ;           7.  RETURN TO STEP 4. UNTIL CYLINDER 0 IS REACHED
                                            002C   ;
                                            002C   ; DEBUG:
                                            002C   ;
                                            002C   ;           DIFFERENCE LOGIC
                                            002C   ;           TRACK CROSSING LOGIC
                                            002C   ;
                                            002C   ;--
                                            002C
                                            002C              $DS_BGNDATA
00000000 00000001 00000000 00000003   002C              .LONG    3,LUN0,RL02$T,HEAD_0
00000001 00000001 00000000 00000003   003C              .LONG    3,LUN0,RL02$T,HEAD_1
00000000 00000002 00000000 00000003   004C              .LONG    3,LUN0,R80$T,HEAD_0
00000000 00000001 00000001 00000003   005C              .LONG    3,LUN1,RL02$T,HEAD_0
00000001 00000001 00000001 00000003   006C              .LONG    3,LUN1,RL02$T,HEAD_1
00000000 00000002 00000001 00000003   007C              .LONG    3,LUN1,R80$T,HEAD_0
00000000 00000001 00000002 00000003   008C              .LONG    3,LUN2,RL02$T,HEAD_0
00000001 00000001 00000002 00000003   009C              .LONG    3,LUN2,RL02$T,HEAD_1
00000000 00000002 00000002 00000003   00AC              .LONG    3,LUN2,R80$T,HEAD_0
00000000 00600001 00000003 00000003   00BC              .LONG    3,LUN3,RL02$T,HEAD_0
```

```
00000001 00000001 00000003 00000003  00CC  1644         .LONG   3,LUN3,RL02$T,HEAD_1
00000000 00000002 00000003 00000003  00DC  1645         .LONG   3,LUN3,R80$T,HEAD_0
                                     00EC  1646         $DS_ENDDATA
                                     00F0  1647
                                     00F0  1648         $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
                                     00F2  1649         GET_LUN -                               ; GET LUN INFO
                                     00F2  1650           UNIT= OFST_UNIT(AP)                   ; LOGICAL UNIT
                   04 AC    DD       00F2               PUSHL   OFST_UNIT(AP)
              0000'CF   01  FB       00F5               CALLS   #1,GET_LUN$A
                                     00FA  1651         BDEVFOUND       1$                      ; IF DEV AVAILABLE GOTO 1$
                06 50   00  E2       00FA               BBSS    #0,R0,1$
                   50   01  D0       00FE                       MOVL    #1,R0           ; NOT AN ERROR
                   036D 31           0101                       BRW     TEST_016_X      ; EXIT TEST 16
            54  0000'CF     D0       0104  1652 1$:     MOVL    DRIVE_TBL,R4                    ; GET DRIVE INFO TABLE
             08 AC      64  91       0109  1653         CMPB    OFST_DRVTYP(R4),OFST_DRIV(AP)   ; IF CURRENT DRIVE = DATA TYPE
                        06  13       010D  1654         BEQL    5$                              ; THEN GOTO 5$
                                     010F  1655         $DS_EXIT        TEST                    ; ELSE EXIT TEST
             55    48 A4    D0       0115  1656 5$:     MOVL    OFST_HDRTMP(R4),R5              ; GET ADDR OF HEADER TEMPLATE
             59    44 A4    D0       0119  1657         MOVL    OFST_HDBITS(R4),R9              ; GET ADDR OF HDR BIT MNEMONICS
                0000'CF     D4       011D  1658         CLRL    CONFIG                          ; CLEAR PROGRAM FLAGS
                                     0121  1659         $DS_BGNSUB
                                     012A  1660         STARS
                                     012A       ;;*****************************************************************
                                     012A  1661 ;++
                                     012A  1662 ; POSITION DRIVE BACK TO CYLINDER ZERO WITH -1 SEEKS.  THEN SEEK
                                     012A  1663 ; DRIVE FORWARD WITH +1 SEEKS UNTIL LAST CYLINDER IS ENCOUNTERED.
                                     012A  1664 ;--
                                     012A  1665
              0000'CF   00  FB       012A  1666 10$:    CALLS   #0,SUBSYS_INIT                  ; INIT SUBSYSTEM
                   17 50    E8       012F  1667         BLBS    R0,20$                          ; BRANCH IF NO ERRORS
                                     0132  1668         $DS_ERRHARD_S   ,L$UNIT,MSG14_SYSINIT,-
                                     0132  1669                         PRINT_REG,R1
                                     0142       ;;; TEST 16, SUBTEST 1, ERROR 1
                                     0149  1670 20$:    $DS_CKLOOP      10$
                        00  DD       0151  1671 25$:    PUSHL   #0                              ; DEFINE FLAGS
              0000'CF   01  FB       0153  1672         CALLS   #1,GET_POSITION                 ; GET CURRENT DRIVE POSITION
                   17 50    E8       0158  1673         BLBS    R0,30$                          ; BRANCH IF NO ERRORS
                                     015B  1674         $DS_ERRHARD_S   ,L$UNIT,MSG19_READH,-
                                     015B  1675                         PRINT_REG,R1
                                     016B       ;;; TEST 16, SUBTEST 1, ERROR 2
                                     0172  1676 30$:    $DS_CKLOOP      25$
                0000'CF     D5       017A  1677         TSTL    CURR_CYLN                       ; ARE WE ON CYLINDER 0
                        40  13       017E  1678         BEQL    60$                             ; BRANCH IF WE ARE
     0000'CF    0000'CF 01  C3       0180  1679 40$:    SUBL3   #1,CURR_CYLN,NEW_CYLN           ; SET UP DESTINATION
                   0C AC    DD       0188  1680 42$:    PUSHL   OFST_SKHD(AP)                   ; PUSH HEAD FROM ARG TABLE
                0000'CF     DD       018B  1681         PUSHL   NEW_CYLN                        ; PUSH DESTINATION
              0000'CF   02  FB       018F  1682         CALLS   #2,STEP_HDS                     ; DO THE SEEK
                   17 50    E8       0194  1683         BLBS    R0,50$                          ; BRANCH IF NO ERRORS
                                     0197  1684         $DS_ERRHARD_S   ,L$UNIT,@ERR_MSG,-
                                     0197  1685                         PRINT_REG,R1
                                     01A7       ;;; TEST 16, SUBTEST 1, ERROR 3
                                     01AE  1686 50$:    $DS_CKLOOP      42$
                0000'CF     D4       01B6  1687         CLRL    CONFIG                          ; CLEAR ANY PROGRAM FLAGS
                0000'CF     D5       01BA  1688         TSTL    CURR_CYLN                       ; ARE WE AT CYLINDER 0 NOW ?
                        C0  12       01BE  1689         BNEQ    40$                             ; NO - KEEP GOING
     0000'CF    0000'CF 01  C1       01C0  1690 60$:    ADDL3   #1,CURR_CYLN,NEW_CYLN           ; SET INCREMENT FORWARD BY 1
```

```
               0C AC    DD  01C8  62$:    PUSHL   OFST_SKHD(AP)                 ; PUSH DESIRED HEAD
             0000'CF    DD  01CB          PUSHL   NEW_CYLN                      ; PUSH DESTINATION
        0000'CF  02     FB  01CF          CALLS   #2,STEP_HDS                   ; EXECUTE THE SEEK
            17 50       E8  01D4          BLBS    R0,70$                        ; BRANCH IF NO ERRORS
                            01D7          $DS_ERRHARD_S    ,L$UNIT,@ERR_MSG,-
                            01D7                           PRINT_REG,R1
                            01E7          ;;; TEST 16, SUBTEST 1, ERROR 4
                            01EE  70$:    $DS_CKLOOP       62$
             0000'CF    D4  01F6          CLRL    CONFIG                        ; CLEAR ANY PROGRAM FLAGS
     08 A4   0000'CF    D1  01FA          CMPL    CURR_CYLN,OFST_LSTCYL(R4)     ; AT LAST CYLINDER
                  BE    12  0200          BNEQ    60$                           ; BRANCH IF NO
                            0202          $DS_ENDSUB
                            020B
                            020B          $DS_BGNSUB
                            0214          STARS
                            0214          ;;*****************************************************************
                            0214          ;++
                            0214          ; POSITION DRIVE TO LAST CYLINDER WITH +1 SEEKS.  IF DRIVE
                            0214          ; IS AN RL02 THEN SEEK DRIVE FORWARD INTO INNER GUARD BAND
                            0214          ; AND CHECK THAT DRIVE HAS REPOSITIONED BACK TO CYLINDER 510.
                            0214          ; IF DRIVE IS AN R80 THEN EXIT THIS SUBTEST.
                            0214          ;--
                            0214
            01 64       91  0214          CMPB    OFST_DRVTYP(R4),#RL02$T       ; IF DRIVE IS AN RL02
                  03    13  0217          BEQL    10$                           ; THEN GOTO 10$
                            0219          $DS_EXIT         SUB                  ; ELSE THIS SUBTEST
        0000'CF  00     FB  021C  10$:    CALLS   #0,SUBSYS_INIT                ; INITIALIZE SUBSYSTEM
            17 50       E8  0221          BLBS    R0,20$                        ; BRANCH IF NO ERRORS
                            0224          $DS_ERRHARD_S    ,L$UNIT,MSG14_SYSINIT,-
                            0224                           PRINT_REG,R1
                            0234          ;;; TEST 16, SUBTEST 2, ERROR 1
                            023B  20$:    $DS_CKLOOP       10$
                  00    DD  0243  25$:    PUSHL   #0                            ; DEFINE FLAGS
        0000'CF  01     FB  0245          CALLS   #1,GET_POSITION               ; GET CURRENT DRIVE POSITION
            17 50       E8  024A          BLBS    R0,30$                        ; BRANCH IF NO ERRORS
                            024D          $DS_ERRHARD_S    ,L$UNIT,MSG19_READH,-
                            024D                           PRINT_REG,R1
                            025D          ;;; TEST 16, SUBTEST 2, ERROR 2
                            0264  30$:    $DS_CKLOOP       25$
     08 A4   0000'CF    D1  026C          CMPL    CURR_CYLN,OFST_LSTCYL(R4)     ; AT LAST CYLINDER
                  42    13  0272          BEQL    60$                           ; BRANCH IF YES
0000'CF  0000'CF  01    C1  0274  40$:    ADDL3   #1,CURR_CYLN,NEW_CYLN         ; BUMP DESTINATION BY ONE
               0C AC    DD  027C  42$:    PUSHL   OFST_SKHD(AP)                 ; PUSH DESIRED HEAD
             0000'CF    DD  027F          PUSHL   NEW_CYLN                      ; PUSH DESTINATION
        0000'CF  02     FB  0283          CALLS   #2,STEP_HDS                   ; EXECUTE THE SEEK
            17 50       E8  0288          BLBS    R0,50$                        ; BRANCH IF NO ERRORS
                            028B          $DS_ERRHARD_S    ,L$UNIT,@ERR_MSG,-
                            028B                           PRINT_REG,R1
                            029B          ;;; TEST 16, SUBTEST 2, ERROR 3
                            02A2  50$:    $DS_CKLOOP       42$
             0000'CF    D4  02AA          CLRL    CONFIG                        ; CLEAR ANY PROGRAM FLAGS
     08 A4   0000'CF    D1  02AE          CMPL    CURR_CYLN,OFST_LSTCYL(R4)     ; AT LAST CYLN NOW ?
                  BE    12  02B4          BNEQ    40$                           ; BRANCH IF NOT
                  57    7C  02B6  60$:    CLRQ    R7                            ; CLEAR REGS R7 AND R8
        65    18 A4     F0  02B8          INSV    OFST_IGBCYL(R4),OFST_CYLPOS(R5),- ; SET UP EXPECTED CYLINDER
        57    04 A5         02BC                  OFST_CYLSIZ(R5),R7            ;
```

```
         05 A5   0C AC    F0  02BF         INSV    OFST_SKHD(AP),OFST_TRKPOS(R5),-  ; SET UP EXPECTED TRACK
            57   09 A5        02C4                 OFST_TRKSIZ(R5),R7               ;
      5A    08 A4   01    C1  02C7         ADDL3   #1,OFST_LSTCYL(R4),R10           ; GET LAST CYLN + 1
   58   10    10    5A    F0  02CC         INSV    R10,#16,#16,R8                   ; LOAD DESTINATION CYLINDER
58   08   08     0C AC    F0  02D1         INSV    OFST_SKHD(AP),#8,#8,R8           ; LOAD DESTINATION TRACK
                    00    DD  02D7         PUSHL   #0                               ; IDCMPR DATA
                    58    DD  02D9         PUSHL   R8                               ; IDCDAR DATA
                    00    DD  02DB         PUSHL   #0                               ; IDCCNT DATA
                    00    DD  02DD         PUSHL   #0                               ; IDCBAR DATA
                    06    DD  02DF         PUSHL   #SEEK                            ; DEFINE COMMAND
         0000'CF    05    FB  02E1         CALLS   #5,START_CMD                     ; EXECUTE SEEK TO INNR GB
         0000'CF    00    FB  02E6         CALLS   #0,STATUS_CHK                    ; CHECK FOR ANY ERRORS
               17   50    E8  02EB         BLBS    R0,70$                           ; BRANCH IF NONE
                              02EE         $DS_ERRHARD_S    ,L$UNIT,MSG51_SK_INNR,-
                              02EE                          PRINT_REG,R1
                              02FE    ;;; TEST 16, SUBTEST 2, ERROR 4
                              0305    70$:  $DS_CKLOOP      25$
                    00    DD  030E    75$:  PUSHL   #0                              ; DEFINE FLAGS
         0000'CF    01    FB  0310         CALLS   #1,GET_POSITION                  ; GET CURRENT DRIVE POSITION
               17   50    E8  0315         BLBS    R0,80$                           ; BRANCH IF NO ERRORS
                              0318         $DS_ERRHARD_S    ,L$UNIT,MSG19A_READH,-
                              0318                          PRINT_REG,R1
                              0328    ;;; TEST 16, SUBTEST 2, ERROR 5
                              032F    80$:  $DS_CKLOOP      75$
          0A A5     00    F0  0337         INSV    #0,OFST_SECPOS(R5),-             ; CLEAR SECTOR BITS
       0000'CF   0E A5        033B                 OFST_SECSIZ(R5),T.HDR            ;
       0000'CF   54 A4    CA  0340         BICL2   OFST_HDRDNC(R4),T.HDR            ; CLEAR OUT DONT CARE BITS
         0000'CF    57    D1  0346         CMPL    R7,T.HDR                         ; DID HDS POSITION CORRECTLY
                    23    13  034B         BEQL    90$                              ; BRANCH IF YES
                              034D         $DS_ERRHARD_S    ,L$UNIT,MSG51_SK_INNR,-
                              034D                          REG_ERRPT,#MSG46_POS_ERR,-
                              034D                          R7,T.HDR,R9
                              0369    ;;; TEST 16, SUBTEST 2, ERROR 6
                              0370    90$:  $DS_CKLOOP      25$
                              0379         $DS_ENDSUB
                              0382
                              0382         $DS_BGNSUB
                              038B         STARS
                              038B    ;;***********************************************************************
                              038B    ;++
                              038B    ; POSITION DRIVE TO LAST CYLINDER WITH +1 SEEKS.  THEN SEEK DRIVE
                              038B    ; REVERSE WITH -1 SEEKS UNTIL CYLINDER ZERO IS ENCOUNTERED.
                              038B    ;--
                              038B
         0000'CF    00    FB  038B    10$:  CALLS   #0,SUBSYS_INIT                  ; INITIALIZE SUBSYSTEM
               17   50    E8  0390         BLBS    R0,20$                           ; BRANCH IF NO ERRORS
                              0393         $DS_ERRHARD_S    ,L$UNIT,MSG14_SYSINIT,-
                              0393                          PRINT_REG,R1
                              03A3    ;;; TEST 16, SUBTEST 3, ERROR 1
                              03AA    20$:  $DS_CKLOOP      10$
                    00    DD  03B2    25$:  PUSHL   #0                              ; DEFINE FLAGS
         0000'CF    01    FB  03B4         CALLS   #1,GET_POSITION                  ; GET CURRENT DRIVE POSITION
               17   50    E8  03B9         BLBS    R0,30$                           ; BRANCH IF NO ERRORS
                              03BC         $DS_ERRHARD_S    ,L$UNIT,MSG19_READH,-
                              03BC                          PRINT_REG,R1
                              03CC    ;;; TEST 16, SUBTEST 3, ERROR 2
```

```
                              03D3  30$:     $DS_CKLOOP      25$
      08 A4   0000'CF    D1   03DB           CMPL    CURR_CYLN,OFST_LSTCYL(R4)       ; ARE WE AT LAST CYLINDER ?
                   42    13   03E1           BEQL    60$                             ; BRANCH IF WE ARE
0000'CF   0000'CF  01    C1   03E3  40$:     ADDL3   #1,CURR_CYLN,NEW_CYLN           ; POINT TO NEXT CYLN
                0C AC    DD   03EB  42$:     PUSHL   OFST_SKHD(AP)                   ; PUSH DESIRED HEAD
              0000'CF    DD   03EE           PUSHL   NEW_CYLN                        ; PUSH DESTINATION
          0000'CF  02    FB   03F2           CALLS   #2,STEP_HDS                     ; EXECUTE SEEK CMD
                17 50    E8   03F7           BLBS    R0,50$                          ; BRANCH IF NO ERRORS
                              03FA           $DS_ERRHARD_S      ,L$UNIT,@ERR_MSG,-
                              03FA                              PRINT_REG,R1
                              040A  ;;; TEST 16, SUBTEST 3, ERROR 3
                              0411  50$:     $DS_CKLOOP      42$
              0000'CF    D4   0419           CLRL    CONFIG                          ; CLEAR ANY PROGRAM FLAGS
      08 A4   0000'CF    D1   041D           CMPL    CURR_CYLN,OFST_LSTCYL(R4)       ; ARE WE AT LAST CYLN NOW ?
                   BE    12   0423           BNEQ    40$                             ; BRANCH IF NOT
0000'CF   0000'CF  01    C3   0425  60$:     SUBL3   #1,CURR_CYLN,NEW_CYLN           ; START REVERSE SEEKS NOW
                0C AC    DD   042D  62$:     PUSHL   OFST_SKHD(AP)                   ; PUSH DESIRED HEAD
              0000'CF    DD   0430           PUSHL   NEW_CYLN                        ; PUSH DESTINATION
          0000'CF  02    FB   0434           CALLS   #2,STEP_HDS                     ; EXECUTE SEEK CMD
                17 50    E8   0439           BLBS    R0,70$                          ; BRANCH IF NO ERRORS
                              043C           $DS_ERRHARD_S      ,L$UNIT,@ERR_MSG,-
                              043C                              PRINT_REG,R1
                              044C  ;;; TEST 16, SUBTEST 3, ERROR 4
                              0453  70$:     $DS_CKLOOP      62$
              0000'CF    D4   045B           CLRL    CONFIG                          ; CLEAR ANY PROGRAM FLAGS
              0000'CF    D5   045F           TSTL    CURR_CYLN                       ; ARE WE AT CYLINDER 0 ?
                   C0    12   0463           BNEQ    60$                             ; BRANCH IF NOT
                              0465           $DS_ENDSUB
                              046E           $DS_ENDTEST
                              0479           $DS_SBTTL <DIFFERENCE WORD TRANSMISSION TEST>,ALIGN=LONG
                              0479           .SBTTL  TEST 17:  DIFFERENCE WORD TRANSMISSION TEST
                              0479           $D2_SBTTL       017,<DIFFERENCE WORD TRANSMISSION TEST>,<LONG>
                          00000000                   .PSECT  TEST_017, LONG, NOWRT
                              0024
```

```
                                                0024
                                                0024  ;++
                                                0024  ;
                                                0024  ; TEST DESCRIPTION:
                                                0024  ;
                                                0024  ; THIS TEST WILL VERIFY THE TRANSMISSION AND DECODING OF THE DIFFERENCE
                                                0024  ; WORD.  EACH SEEK WILL START AT CYLINDER ZERO AND RETURN THERE WITH A
                                                0024  ; SEEK IN THE OPPOSIITE DIRECTION.  BOTH HEADS ARE TESTED IN THIS MANNER
                                                0024  ; IF THE DRIVE CURRENTLY UNDER TEST IS AN RL02.
                                                0024  ; PATTERNS USED FOR THE DIFFERENCE WORD ARE FLOATING ONES, FLOATING ZEROS,
                                                0024  ; GROWING ONES AND GROWING ZEROS.
                                                0024  ;
                                                0024  ; TEST STEPS:
                                                0024  ;
                                                0024  ; 1.     INITIALIZE SUBSYSTEM
                                                0024  ; 2.     GET CURRENT DRIVE POSITION
                                                0024  ; 3.     IF AT CYLN 0 GO TO STEP 5
                                                0024  ; 4.     SEEK HEADS BACK TO CYLN 0
                                                0024  ;        CHECK FOR ANY ERRORS
                                                0024  ; 5.     GET CYLINDER DESTINATION FROM CYLINDER LIST
                                                0024  ; 6.     EXECUTE SEEK AND CHECK FOR ERRORS
                                                0024  ; 7.     REVERSE DIRECTION AND SEEK BACK TO CYLN ZERO
                                                0024  ; 8.     CHECK FOR ANY ERRORS
                                                0024  ; 9.     IF NOT AT CYLN ZERO GO TO STEP 4
                                                0024  ;10.     GET NEXT CYLN FOR LIST AND GO TO STEP 5
                                                0024  ;
                                                0024  ; DEBUG:
                                                0024  ;
                                                0024  ;        DIFFERENCE LOGIC
                                                0024  ;        COUNT LOGIC
                                                0024  ;        VELOCITY ROM
                                                0024  ;
                                                0024  ;--
                                                0024
                                                0024           $DS_BGNDATA
00000000 00000001 00000000 00000003             0024           .LONG    3,LUN0,RL02$T,HEAD_0
00000001 00000001 00000000 00000003             0034           .LONG    3,LUN0,RL02$T,HEAD_1
00000000 00000002 00000000 00000003             0044           .LONG    3,LUN0,R80$T,HEAD_0
00000000 00000001 00000001 00000003             0054           .LONG    3,LUN1,RL02$T,HEAD_0
00000001 00000001 00000001 00000003             0064           .LONG    3,LUN1,RL02$T,HEAD_1
00000000 00000002 00000001 00000003             0074           .LONG    3,LUN1,R80$T,HEAD_0
00000000 00000001 00000002 00000003             0084           .LONG    3,LUN2,RL02$T,HEAD_0
00000001 00000001 00000002 00000003             0094           .LONG    3,LUN2,RL02$T,HEAD_1
00000000 00000002 00000002 00000003             00A4           .LONG    3,LUN2,R80$T,HEAD_0
00000000 00000001 00000003 00000003             00B4           .LONG    3,LUN3,RL02$T,HEAD_0
00000001 00000001 00000003 00000003             00C4           .LONG    3,LUN3,RL02$T,HEAD_1
00000000 00000002 00000003 00000003             00D4           .LONG    3,LUN3,R80$T,HEAD_0
                                                00E4           $DS_ENDDATA
                                                00E8
                                                00E8           $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
                                                00EA           GET_LUN -                          ; GET LUN INFO
                                                00EA             UNIT= OFST_UNIT(AP)              ; LOGICAL UNIT
                              04 AC     DD      00EA           PUSHL    OFST_UNIT(AP)
                        0000'CF    01     FB      00ED           CALLS    #1,GET_LUN$A
                                                00F2           BDEVFOUND        1$                ; IF DEV AVAILABLE GOTO 1$
                        06 50    00     E2      00F2           BBSS     #0,R0,1$
```

```
        50    01  D0 00F6                  MOVL    #1,R0                  ; NOT AN ERROR
            0122  31 00F9                  BRW     TEST_017_X             ; EXIT TEST 17
     53  0000'CF  D0 00FC  1$:     MOVL    DRIVE_TBL,R3                   ; GET DRIVE INFO TABLE
      08 AC   63  91 0101          CMPB    OFST_DRVTYP(R3),OFST_DRIV(AP)  ; IF DRIVE TYPE = DATA TYPE
              06  13 0105          BEQL    5$                             ; THEN GOTO 5$
                     0107          $DS_EXIT        TEST                   ; ELSE EXIT TEST
     55    08 A3  D0 010D  5$:     MOVL    OFST_LSTCYL(R3),R5             ; SAVE LAST CYLINCER NUMBER
         0000'CF  D4 0111          CLRL    CONFIG                         ; CLEAR ALL PROGRAM FLAGS
     54  0000'CF  DE 0115          MOVAL   CYLN_LIST,R4                   ; GET DATA FOR THIS TEST
 0000'CF      00  FB 011A  10$:    CALLS   #0,SUBSYS_INIT                 ; INITIALIZE SUBSYSTEM
           17 50  E8 011F          BLBS    R0,20$                         ; BRANCH IF NO ERRORS
                     0122          $DS_ERRHARD_S   ,L$UNIT,MSG14_SYSINIT,-
                     0122                          PRINT_REG,R1
                     0132  ;;; TEST 17, SUBTEST 0, ERROR 1
                     0139  20$:    $DS_CKLOOP      10$
                     0141          ;++
                     0141          ;
                     0141          ; POSITON DRIVE TO CYLINDER 0
                     0141          ;
                     0141          ;--
              00  DD 0141  25$:    PUSHL   #0                             ; DEFINE FLAGS
 0000'CF      01  FB 0143          CALLS   #1,GET_POSITION                ; GET CURRENT DRIVE POSITION
           17 50  E8 0148          BLBS    R0,30$                         ; BRANCH IF NO ERRORS
                     014B          $DS_ERRHARD_S   ,L$UNIT,MSG19_READH,-
                     014B                          PRINT_REG,R1
                     015B  ;;; TEST 17, SUBTEST 0, ERROR 2
                     0162  30$:    $DS_CKLOOP      25$
         0000'CF  D5 016A  35$:    TSTL    CURR_CYLN                      ; ARE WE AT CYLN 0
              38  13 016E          BEQL    60$                            ; BRANCH IF WE ARE
           0C AC  DD 0170  40$:    PUSHL   OFST_SKHD(AP)                  ; PUSH DESIRED HEAD
              00  DD 0173          PUSHL   #0                             ; PUSH DESTINATION CYLN
 0000'CF      02  FB 0175          CALLS   #2,POSITION_HDS                ; EXECUTE THE SEEK CMD
           17 50  E8 017A          BLBS    R0,50$                         ; BRANCH IF NO ERRORS
                     017D          $DS_ERRHARD_S   ,L$UNIT,@ERR_MSG,-
                     017D                          PRINT_REG,R1
                     018D  ;;; TEST 17, SUBTEST 0, ERROR 3
                     0194  50$:    $DS_CKLOOP      40$
         0000'CF  D5 019C          TSTL    CURR_CYLN                      ; ARE WE AT CYLN 0
              06  13 01A0          BEQL    60$                            ; BRANCH IF WE ARE
                     01A2          $DS_EXIT        TEST
                     01A8          ;++
                     01A8          ;
                     01A8          ; GET DESTINATION FROM CYLN_LIST AND DO FORWARD SEEK
                     01A8          ;
                     01A8          ;--
 0000'CF      64  3C 01A8  60$:    MOVZWL  (R4),NEW_CYLN                  ; GET DESTINATION CYLN
           0C AC  DD 01AD          PUSHL   OFST_SKHD(AP)                  ; PUSH DESIRED HEAD
         0000'CF  DD 01B0          PUSHL   NEW_CYLN                       ; PUSH DESTINATION CYLN
 0000'CF      02  FB 01B4          CALLS   #2,POSITION_HDS                ; EXECUTE THE SEEK CMD
           17 50  E8 01B9          BLBS    R0,70$                         ; BRANCH IF NO ERRORS
                     01BC          $DS_ERRHARD_S   ,L$UNIT,@ERR_MSG,-
                     01BC                          PRINT_REG,R1
                     01CC  ;;; TEST 17, SUBTEST 0, ERROR 4
                     01D3  70$:    $DS_CKLOOP      60$
                     01DB          ;++
                     01DB          ;
```

```
                            01DB  1924         ; CHANGE DIRECTION AND SEEK REVERSE TO CYLINDER ZERO
                            01DB  1925         ;
                            01DB  1926         ;--
          0C AC     DD      01DB  1927 75$:    PUSHL   OFST_SKHD(AP)                    ; PUSH DESIRED HEAD
             00     DD      01DE  1928         PUSHL   #0                               ; SET DESTINATION TO CYLN 0
      0000'CF 02    FB      01E0  1929         CALLS   #2,POSITION_HDS                  ; EXECUTE THE SEEK CMD
          17 50     E8      01E5  1930         BLBS    R0,80$                           ; BRANCH IF NO ERRORS
                            01E8  1931         $DS_ERRHARD_S    ,L$UNIT,@ERR_MSG,-
                            01E8  1932                          PRINT_REG,R1
                            01F8       ;;; TEST 17, SUBTEST 0, ERROR 5
                            01FF  1933 80$:    $DS_CKLOOP       75$
       54    02     C0      0207  1934 85$:    ADDL2   #2,R4                            ; POINT TO NEXT CYLINDER
00000000'8F  54     D1      020A  1935         CMPL    R4,#CYLN_LIST_X                  ; ARE SEEKS ALL DONE
             08     13      0211  1936         BEQL    90$                              ; BRANCH IF YES
       55    64     B1      0213  1937         CMPW    (R4),R5                          ; IF CYLINDER > LAST CYLN
             EF     14      0216  1938         BGTR    85$                              ; THEN GOTO 85$
           FF4F     31      0218  1939         BRW     35$                              ; GO DO NEXT SEEK
                            021B  1940 90$:
                            021B  1941         $DS_ENDTEST
                            0226  1942         $DS_ENDMOD
                            0004  1943         .END
```

```
$$$             = 00000000
$$E             = 00000001
$$M             = 00000005
$$N             = 00000001
$$S             = FFFFFFFF
$ENV            = 00000003  G
$ER             = FFFFFFFF
$MO             = 00000001  G
$S$             = 00000108 R      07
$ST             = 00000000
$TN             = 00000011
ALL             = 00000002  G
BASE_001          00000000 R      03
BASE_002          00000000 R      05
BASE_003          00000000 R      06
BASE_004          00000000 R      08
BASE_005          00000000 R      09
BASE_006          00000000 R      0A
BASE_007          00000000 R      0B
BASE_008          00000000 R      0C
BASE_009          00000000 R      0D
BASE_010          00000000 R      0E
BASE_011          00000000 R      0F
BASE_012          00000000 R      10
BASE_013          00000000 R      11
BASE_014          00000000 R      12
BASE_015          00000000 R      13
BASE_016          00000000 R      14
BASE_017          00000000 R      15
BASE_ADDR         ********    X   03
BIT...          = 00000028
BIT0            = 00000001
BIT1            = 00000002
BIT10           = 00000400
BIT11           = 00000800
BIT12           = 00001000
BIT13           = 00002000
BIT14           = 00004000
BIT15           = 00008000
BIT16           = 00010000
BIT17           = 00020000
BIT18           = 00040000
BIT19           = 00080000
BIT2            = 00000004
BIT20           = 00100000
BIT21           = 00200000
BIT22           = 00400000
BIT23           = 00800000
BIT24           = 01000000
BIT25           = 02000000
BIT26           = 04000000
BIT27           = 08000000
BIT28           = 10000000
BIT29           = 20000000
BIT3            = 00000008
BIT30           = 40000000

BIT31           = 80000000
BIT4            = 00000010
BIT5            = 00000020
BIT6            = 00000040
BIT7            = 00000080
BIT8            = 00000100
BIT9            = 00000200
CEP_FUNCTIONAL  = 00000000
CEP_REPAIR      = 00000001
CHC$_ABORT      = 00000002
CHC$_CLEAR      = 00000006
CHC$_CLRDFT     = 00000009
CHC$_DSINT      = 00000005
CHC$_ENINT      = 00000004
CHC$_INITA      = 00000000
CHC$_INITB      = 00000001
CHC$_PURGE      = 00000003
CHC$_SELF_TEST  = 0000000B
CHC$_SETDFT     = 00000008
CHC$_STATUS     = 00000007
CHC$_STOP       = 0000000A
CHI$M_CHNINT    = 00000001
CHI$M_DEVINT    = 00000002
CHI$M_IPL       = 0000007C
CHI$S_IPL       = 00000005
CHI$V_CHNINT    = 00000000
CHI$V_DEVINT    = 00000001
CHI$V_IPL       = 00000002
CHM$_FORWARD    = 00000000
CHM$_INVALIDATE = 00000001
CHM$_MAP        = 00000002
CHM$_MFWDN      = 00000002
CHM$_MFWDNO     = 0000000A
CHM$_MFWDV      = 00000003
CHM$_MFWDVO     = 0000000B
CHM$_MREVN      = 00000006
CHM$_MREVNO     = 0000000E
CHM$_MREVV      = 00000007
CHM$_MREVVO     = 0000000F
CHM$_NFWDN      = 00000000
CHM$_NREVN      = 00000004
CHM$_OFFSET     = 00000008
CHM$_REVERSE    = 00000004
CHS$M_BADBDP    = 20000000
CHS$M_BADPAR    = 40000000
CHS$M_BUSIC     = 00800000
CHS$M_BUSINIT   = 00400000
CHS$M_BUSNXM    = 08000000
CHS$M_BUSPDN    = 01000000
CHS$M_CHNDPE    = 00000040
CHS$M_CHNERR    = 00000002
CHS$M_CHNMPE    = 00000080
CHS$M_CHPFOT    = 00000100
CHS$M_DEVBUS    = 00000010
CHS$M_DEVERR    = 00000004
CHS$M_DEVTO     = 00000020

CHS$M_ERRANY    = 0000000F
CHS$M_MBAATN    = 00100000
CHS$M_MBACPE    = 00200000
CHS$M_MBADTB    = 00040000
CHS$M_MBADTC    = 00080000
CHS$M_MBAEXC    = 00010000
CHS$M_MBANED    = 00020000
CHS$M_MBAWCKLWR= 02000000
CHS$M_MBAWCKUPR= 04000000
CHS$M_PGMERR    = 00000008
CHS$M_PGMHDE    = 00000800
CHS$M_SYSERR    = 00000001
CHS$M_SYSMEM    = 00000200
CHS$M_SYSSBI    = 00000400
CHS$M_UIE       = 10000000
CHS$V_BADBDP    = 0000001D
CHS$V_BADPAR    = 0000001E
CHS$V_BUSIC     = 00000017
CHS$V_BUSINIT   = 00000016
CHS$V_BUSNXM    = 0000001B
CHS$V_BUSPDN    = 00000018
CHS$V_CHNDPE    = 00000006
CHS$V_CHNERR    = 00000001
CHS$V_CHNMPE    = 00000007
CHS$V_CHPFOT    = 00000008
CHS$V_DEVBUS    = 00000004
CHS$V_DEVERR    = 00000002
CHS$V_DEVTO     = 00000005
CHS$V_MBAATN    = 00000014
CHS$V_MBACPE    = 00000015
CHS$V_MBADTB    = 00000012
CHS$V_MBADTC    = 00000013
CHS$V_MBAEXC    = 00000010
CHS$V_MBANED    = 00000011
CHS$V_MBAWCKLWR= 00000019
CHS$V_MBAWCKUPR= 0000001A
CHS$V_PGMERR    = 00000003
CHS$V_PGMHDE    = 0000000B
CHS$V_SYSERR    = 00000000
CHS$V_SYSMEM    = 00000009
CHS$V_SYSSBI    = 0000000A
CHS$V_UIE       = 0000001C
CONFIG            ********    X   0B
CRC_TABLE         ********    X   0E
CSR$M_CRDY      = 00000080
CSR$M_DRDY      = 00000001
CSR$M_IE        = 00000040
CSR$M_IR        = 01000000
CSR$V_DRDY      = 00000000
CSR$V_FMT       = 0000001D
CSR_BITS          ********    X   06
CURR_CYLN         ********    X   11
CYLN_LIST         ********    X   15
CYLN_LIST_X       ********    X   15
DATA_001          0000001C R      03
DATA_002          00000020 R      05
```

```
DATA_003          00000018 R      06
DATA_004          00000018 R      08
DATA_005          00000018 R      09
DATA_006          00000018 R      0A
DATA_007          0000001C R      0B
DATA_008          00000018 R      0C
DATA_009          00000014 R      0D
DATA_010          00000024 R      0E
DATA_011          0000001C R      0F
DATA_012          00000018 R      10
DATA_013          0000001C R      11
DATA_014          00000020 R      12
DATA_015          00000020 R      13
DATA_016          0000002C R      14
DATA_017          00000024 R      15
DEFAULT         = 00000000  G
DEST_CYLN         ********    X   11
DIFF_WORD         ********    X   11
DRIVE_TBL         ********    X   0B
DS$ABORT          ********    X   03
DS$BGNSUB         ********    X   06
DS$BRANCH         ********    X   03
DS$BREAK          ********    X   03
DS$CHANNEL        ********    X   05
DS$CKLOOP         ********    X   03
DS$ENDSUB         ********    X   06
DS$ERRHARD        ********    X   06
DS$ERRSYS         ********    X   03
DS$K_BBC        = 0000001D  G
DS$K_BBCC       = 00000021  G
DS$K_BBCCI      = 00000023  G
DS$K_BBCS       = 0000001F  G
DS$K_BBS        = 0000001C  G
DS$K_BBSC       = 00000020  G
DS$K_BBSS       = 0000001E  G
DS$K_BBSSI      = 00000022  G
DS$K_BCC_M      = 0000001B  G
DS$K_BCS_M      = 0000001A  G
DS$K_BEQLU_B    = 00000005  G
DS$K_BEQLU_M    = 00000011  G
DS$K_BEQL_B     = 00000004  G
DS$K_BEQL_M     = 00000010  G
DS$K_BERROR     = 00000026  G
DS$K_BGEQU_B    = 00000007  G
DS$K_BGEQU_M    = 00000013  G
DS$K_BGEQ_B     = 00000006  G
DS$K_BGEQ_M     = 00000012  G
DS$K_BGTRU_B    = 00000009  G
DS$K_BGTRU_M    = 00000015  G
DS$K_BGTR_B     = 00000008  G
DS$K_BGTR_M     = 00000014  G
DS$K_BLBC       = 00000025  G
DS$K_BLBS       = 00000024  G
DS$K_BLEQU_B    = 00000003  G
DS$K_BLEQU_M    = 0000000F  G
DS$K_BLEQ_B     = 00000002  G

DS$K_BLEQ_M       = 0000000E  G
DS$K_BLSSU_B      = 00000001  G
DS$K_BLSSU_M      = 0000000D  G
DS$K_BLSS_B       = 00000000  G
DS$K_BLSS_M       = 0000000C  G
DS$K_BNEQU_B      = 0000000B  G
DS$K_BNEQU_M      = 00000017  G
DS$K_BNEQ_B       = 0000000A  G
DS$K_BNEQ_M       = 00000016  G
DS$K_BNERROR      = 00000027  G
DS$K_BVC_M        = 00000019  G
DS$K_BVS_M        = 00000018  G
DS$PROBE            ********    X   03
E.BA                ********    X   08
E.BC                ********    X   09
E.BR                ********    X   0D
E.CS                ********    X   06
E.DA                ********    X   0A
E.MP                ********    X   10
E.VEC               ********    X   0D
ENV$M_DOMAIN      = 00000002
ENV$M_LEVEL       = 00000001
ENV$M_SUPER       = 000003FC
ENV$S_DOMAIN      = 00000001
ENV$S_LEVEL       = 00000001
ENV$S_SUPER       = 00000008
ENV$V_DOMAIN      = 00000001
ENV$V_LEVEL       = 00000000
ENV$V_SUPER       = 00000002
ENV$_CPU          = 00000000
ENV$_FUNCTIONAL   = 00000000
ENV$_REPAIR       = 00000001
ENV$_SUPER        = 00000001
ENV$_SYSTEM       = 00000001
ERR_MSG             ********    X   14
GETSTAT           = 00000004
GET_LUN$A           ********    X   0B
GET_POSITION        ********    X   11
HEAD_0            = 00000000
HEAD_1            = 00000001
IDCBAR            = 00000004
IDCCNT            = 00000008
IDCCSR            = 00000000
IDCDAR            = 0000000C
IDCMPR            = 00000010
IDC_ADRFAIL         ********    X   03
IDC_SADRFAIL        ********    X   05
IDC_STAT$L_1        ********    X   0D
IDC_STAT$W_2        ********    X   0D
IDC_STAT$W_VR       ********    X   0D
L$UNIT              ********    X   03
LUN0              = 00000000
LUN1              = 00000001
LUN2              = 00000002
LUN3              = 00000003
MANUAL            = 00000001  G

MPR_02$M_HS       = 00000040
MPR_80$M_SKE      = 00000400
MS107A_SEEKERR      ********    X   10
MS107B_RECAL        ********    X   10
MS107_RECAL         ********    X   0F
MS45A_HD_PROX       ********    X   12
MSG14_SYSINIT       ********    X   0C
MSG15_NOOP          ********    X   0C
MSG17_GETSTAT       ********    X   0B
MSG19A_READH        ********    X   14
MSG19_READH         ********    X   0E
MSG23_CS_INC        ********    X   0D
MSG24A_BC_MOD       ********    X   0C
MSG24_BA_MOD        ********    X   0C
MSG25_DA_MOD        ********    X   0C
MSG27_MP_INC        ********    X   0B
MSG2_RG_ZERO        ********    X   06
MSG38_NOOP_IE       ********    X   0D
MSG39_NO_INT        ********    X   0D
MSG3_RG_ONE         ********    X   06
MSG40_BAD_BR        ********    X   0D
MSG41_BAD_VEC       ********    X   0D
MSG42_RPT_CYL       ********    X   0E
MSG44_BAD_CRC       ********    X   0E
MSG45_SK_ZERO       ********    X   11
MSG46_POS_ERR       ********    X   11
MSG47_DS_INC        ********    X   10
MSG48_SK_ONE        ********    X   13
MSG49_DRDY_NC       ********    X   13
MSG4_RG_FLT1        ********    X   06
MSG51_SK_INNR       ********    X   14
MSG5_RG_FLT0        ********    X   06
MSG6_CS_FAIL        ********    X   06
MSG7A_BC_FAIL       ********    X   09
MSG7_BA_FAIL        ********    X   08
MSG8_DA_FAIL        ********    X   0A
MSG9_RG_RMW         ********    X   06
MSG_001             00000002 R      03
MSG_002             00000002 R      05
MSG_003             00000002 R      06
MSG_004             00000002 R      08
MSG_005             00000002 R      09
MSG_006             00000002 R      0A
MSG_007             00000002 R      0B
MSG_008             00000002 R      0C
MSG_009             00000002 R      0D
MSG_010             00000002 R      0E
MSG_011             00000002 R      0F
MSG_012             00000002 R      10
MSG_013             00000002 R      11
MSG_014             00000002 R      12
MSG_015             00000002 R      13
MSG_016             00000002 R      14
MSG_017             00000002 R      15
NEW_CYLN            ********    X   14
NOCHK$M_ATTN      = 00000080
```

```
NOCHK$M_CRDY     = 00000001            RL02$T            = 00000001                T5_S5_X          000001F7 R      09
NOCHK$M_DRDY     = 00000004            SECD_ADDR           ********  X   05        T6_S1            00000027 RG     0A
NOCHK$M_DSTA     = 00000010            SEEK              = 00000006                T6_S1_X          0000008D R      0A
NO_OP            = 00000000            SEP_FUNCTIONAL    = 00000002                T6_S2            00000096 RG     0A
OFST_AVGTIM      = 00000080            SEP_REPAIR        = 00000003                T6_S2_X          000000E0 R      0A
OFST_CSRTYP      = 00000058            SIZ...            = 00000001                T6_S3            000000E9 RG     0A
OFST_CYLPOS      = 00000000            START_CMD           ********  X   0B        T6_S3_X          00000140 R      0A
OFST_CYLSIZ      = 00000004            STATUS_CHK          ********  X   0B        T6_S4            00000149 RG     0A
OFST_DEVMEM      = 0000001C            STEP_HDS            ********  X   14        T6_S4_X          000001A8 R      0A
OFST_DEVNAM      = 00000020            SUBSYS_INIT         ********  X   0C        T6_S5            000001B1 RG     0A
OFST_DRIV        = 00000008            T.BA                ********  X   08        T6_S5_X          00000208 R      0A
OFST_DRVERR      = 0000004C            T.BC                ********  X   09        TEST_001         00000020 RG     03
OFST_DRVRST      = 00000050            T.BR                ********  X   0D        TEST_001_X       00000074 RG     03
OFST_DRVTYP      = 00000000            T.CS                ********  X   06        TEST_002         00000024 RG     05
OFST_FMTLEN      = 00000030            T.DA                ********  X   0A        TEST_002_X       000000FE RG     05
OFST_GTSCMD      = 00000034            T.HDR               ********  X   0E        TEST_003         0000001C RG     06
OFST_HDBITS      = 00000044            T.MP                ********  X   0B        TEST_003_X       00000270 RG     06
OFST_HDRDNC      = 00000054            T.MP1               ********  X   0E        TEST_004         0000001C RG     08
OFST_HDRTMP      = 00000048            T.MP2               ********  X   0E        TEST_004_X       00000207 RG     08
OFST_HIGRPM      = 00000074            T.MP3               ********  X   0E        TEST_005         0000001C RG     09
OFST_HMBSEC      = 0000005C            T.VEC               ********  X   0D        TEST_005_X       00000203 RG     09
OFST_IGBCYL      = 00000018            T16_S1              00000121 RG   14        TEST_006         0000001C RG     0A
OFST_LOWRPM      = 00000070            T16_S1_X            00000202 R    14        TEST_006_X       00000214 RG     0A
OFST_LSTCYL      = 00000008            T16_S2              0000020B RG   14        TEST_007         00000040 RG     0B
OFST_LSTSEC      = 00000010            T16_S2_X            00000379 R    14        TEST_007_X       000000E3 RG     0B
OFST_LSTTRK      = 0000000C            T16_S3              00000382 RG   14        TEST_008         0000003C RG     0C
OFST_MAXTIM      = 0000007C            T16_S3_X            00000465 R    14        TEST_008_X       00000154 RG     0C
OFST_MPBITS      = 00000040            T3_S1               00000027 RG   06        TEST_009         00000038 RG     0D
OFST_NMLSTS      = 00000038            T3_S1_X             0000008D R    06        TEST_009_X       000001B9 RG     0D
OFST_ONETIM      = 00000078            T3_S2               00000096 RG   06        TEST_010         00000048 RG     0E
OFST_OP1STS      = 00000060            T3_S2_X             000000F1 R    06        TEST_010_X       0000017B RG     0E
OFST_PHYU        = 00000004            T3_S3               000000FA RG   06        TEST_011         00000040 RG     0F
OFST_RANCYL      = 0000006C            T3_S3_X             0000016C R    06        TEST_011_X       000000C1 RG     0F
OFST_SECLEN      = 00000028            T3_S4               00000175 RG   06        TEST_012         0000003C RG     10
OFST_SECPOS      = 0000000A            T3_S4_X             000001EF R    06        TEST_012_X       0000011A RG     10
OFST_SECSIZ      = 0000000E            T3_S5               000001F8 RG   06        TEST_013         000000E0 RG     11
OFST_SEKSTS      = 0000003C            T3_S5_X             00000264 R    06        TEST_013_X       00000281 RG     11
OFST_SKHD        = 0000000C            T4_S1               00000027 RG   08        TEST_014         00000044 RG     12
OFST_STSDNC      = 00000064            T4_S1_X             00000080 R    08        TEST_014_X       000001C2 RG     12
OFST_TRKLEN      = 0000002C            T4_S2               00000089 RG   08        TEST_015         000000E4 RG     13
OFST_TRKPOS      = 00000005            T4_S2_X             000000D3 R    08        TEST_015_X       00000393 RG     13
OFST_TRKSIZ      = 00000009            T4_S3               000000DC RG   08        TEST_016         000000F0 RG     14
OFST_TSTCYL      = 00000014            T4_S3_X             00000133 R    08        TEST_016_X       00000471 RG     14
OFST_UNIT        = 00000004            T4_S4               0000013C RG   08        TEST_017         000000E8 RG     15
OFST_WTPSTS      = 00000068            T4_S4_X             0000019B R    08        TEST_017_X       0000021E RG     15
P$UNIT             ********  X   0D    T4_S5               000001A4 RG   08        UBACLEAR_ERR     ********   X    05
POSITION_HDS       ********  X   0F    T4_S5_X             000001FB R    08        UBAINIT_ERR      ********   X    05
PRINT_REG          ********  X   0B    T5_S1               00000027 RG   09        WORK             ********   X    0E
PRT$M_EXT_1      = 00000008            T5_S1_X             0000007E R    09
PRT$M_EXT_2      = 00000010            T5_S2               00000087 RG   09
PRT$M_NONE       = 00000040            T5_S2_X             000000D1 R    09
R80$T            = 00000002            T5_S3               000000DA RG   09
RD_IDCREGS         ********  X   0D    T5_S3_X             00000131 R    09
READH            = 00000008            T5_S4               0000013A RG   09
REG_ERRPT          ********  X   06    T5_S4_X             00000199 R    09
RGLIST             ********  X   06    T5_S5               000001A2 RG   09
```

```
                                        +----------------+
                                        ! Psect synopsis !
                                        +----------------+

PSECT name                 Allocation         PSECT No.  Attributes
----------                 ----------         ---------  ----------
.  ABS  .                  00000000  (    0.)  00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
HEADER                     00000000  (    0.)  01 (  1.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC PAGE
$ABS$                      00000000  (    0.)  02 (  2.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
TEST_001                   0000007C  (  124.)  03 (  3.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
DISPATCH                   00000198  (  408.)  04 (  4.)  NOPIC   USR   CON   REL   LCL NOSHR NOEXE   RD   NOWRT NOVEC LONG
TEST_002                   00000106  (  262.)  05 (  5.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_003                   00000278  (  632.)  06 (  6.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
ARGLIST                    00000114  (  276.)  07 (  7.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_004                   0000020F  (  527.)  08 (  8.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_005                   0000020B  (  523.)  09 (  9.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_006                   0000021C  (  540.)  0A ( 10.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_007                   000000EB  (  235.)  0B ( 11.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_008                   0000015C  (  348.)  0C ( 12.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_009                   000001C1  (  449.)  0D ( 13.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_010                   00000183  (  387.)  0E ( 14.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_011                   000000C9  (  201.)  0F ( 15.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_012                   00000122  (  290.)  10 ( 16.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_013                   00000289  (  649.)  11 ( 17.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_014                   000001CA  (  458.)  12 ( 18.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_015                   0000039B  (  923.)  13 ( 19.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_016                   00000479  ( 1145.)  14 ( 20.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_017                   00000226  (  550.)  15 ( 21.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
$TSTCNT                    00000004  (    4.)  16 ( 22.)  NOPIC   USR   OVR   REL   LCL NOSHR NOEXE   RD   NOWRT NOVEC LONG
```

```
                                                +----------------------------+
                                                ! Symbol Cross Reference !
                                                +----------------------------+

SYMBOL          VALUE           DEFINITION      REFERENCES...
------          -----           ----------      -------------
$$$             =00000000       1819    (26)    1659    (26)    1703    (26)    1775    (26)    199     (5)     217     (5)
                                                235     (5)     259     (5)     283     (5)     350     (6)     367     (6)
                                                384     (6)     403     (6)     424     (6)     488     (7)     505     (7)
                                                522     (7)     541     (7)     562     (7)     628     (8)     650     (8)
                                                667     (8)     686     (8)     707     (8)
$$E             =00000001       1775    (26)    1701    (26)    1773    (26)    1817    (26)    215     (5)     233     (5)
                                                257     (5)     281     (5)     302     (5)     365     (6)     382     (6)
                                                401     (6)     422     (6)     440     (6)     503     (7)     520     (7)
                                                539     (7)     560     (7)     578     (7)     648     (8)     665     (8)
                                                684     (8)     705     (8)     724     (8)
$$M             =00000005       1932    (28)    1011    (15)    1022    (15)    103     (3)     1039    (15)    1060    (15)
                                                1095    (17)    1107    (17)    1114    (17)    1152    (19)    1166    (19)
                                                1178    (19)    1185    (19)    1247    (21)    1263    (21)    1269    (21)
                                                1289    (21)    1298    (21)    1304    (21)    1315    (21)    132     (4)
                                                1351    (22)    136     (4)     1363    (22)    1369    (22)    1382    (22)
                                                1388    (22)    1401    (22)    1407    (22)    141     (4)     1413    (22)
                                                1474    (24)    149     (4)     1491    (24)    1497    (24)    1530    (24)
                                                1536    (24)    1544    (24)    1551    (24)    1561    (24)    1567    (24)
                                                1578    (24)    1648    (26)    1669    (26)    1675    (26)    1685    (26)
                                                1696    (26)    1718    (26)    1724    (26)    1734    (26)    1756    (26)
                                                1762    (26)    1771    (26)    1785    (26)    1791    (26)    1801    (26)
                                                1812    (26)    1870    (28)    1884    (28)    1895    (28)    1904    (28)
                                                1920    (28)    1932    (28)    197     (5)     213     (5)     231     (5)
                                                253     (5)     277     (5)     298     (5)     348     (6)     363     (6)
                                                380     (6)     397     (6)     418     (6)     438     (6)     486     (7)
                                                501     (7)     518     (7)     535     (7)     556     (7)     576     (7)
                                                626     (8)     646     (8)     663     (8)     680     (8)     701     (8)
                                                722     (8)     758     (10)    776     (10)    784     (10)    822     (11)
                                                830     (11)    845     (11)    851     (11)    857     (11)    863     (11)
                                                902     (13)    915     (13)    933     (13)    942     (13)    95      (3)
                                                950     (13)    957     (13)    999     (15)
$$N             =00000001       1932    (28)    1011    (15)    1022    (15)    103     (3)     1039    (15)    1060    (15)
                                                1107    (17)    1114    (17)    1166    (19)    1178    (19)    1185    (19)
                                                1263    (21)    1269    (21)    1289    (21)    1298    (21)    1304    (21)
                                                1315    (21)    136     (4)     1363    (22)    1369    (22)    1382    (22)
                                                1388    (22)    1401    (22)    1407    (22)    141     (4)     1413    (22)
                                                149     (4)     1491    (24)    1497    (24)    1530    (24)    1536    (24)
                                                1544    (24)    1551    (24)    1561    (24)    1567    (24)    1578    (24)
                                                1669    (26)    1675    (26)    1685    (26)    1696    (26)    1718    (26)
                                                1724    (26)    1734    (26)    1756    (26)    1762    (26)    1771    (26)
                                                1785    (26)    1791    (26)    1801    (26)    1812    (26)    1884    (28)
                                                1895    (28)    1904    (28)    1920    (28)    1932    (28)    213     (5)
                                                231     (5)     253     (5)     277     (5)     298     (5)     363     (6)
                                                380     (6)     397     (6)     418     (6)     438     (6)     501     (7)
                                                518     (7)     535     (7)     556     (7)     576     (7)     646     (8)
                                                663     (8)     67      (2)     680     (8)     701     (8)     722     (8)
                                                776     (10)    784     (10)    830     (11)    845     (11)    851     (11)
                                                857     (11)    863     (11)    915     (13)    933     (13)    942     (13)
                                                950     (13)    957     (13)
```

```
$$S              =FFFFFFFF     1941    (28)      1064    (15)     108     (3)      1095    (17)     1117    (17)     1152    (19)
                                                 1188    (19)     1247    (21)     1318    (21)     132     (4)      1351    (22)
                                                 1416    (22)     1474    (24)     153     (4)      1586    (24)     1648    (26)
                                                 1659    (26)     1703    (26)     1775    (26)     1819    (26)     1870    (28)
                                                 197     (5)      199     (5)      217     (5)      235     (5)      259     (5)
                                                 283     (5)      304     (5)      348     (6)      350     (6)      367     (6)
                                                 384     (6)      403     (6)      424     (6)      442     (6)      486     (7)
                                                 488     (7)      505     (7)      522     (7)      541     (7)      562     (7)
                                                 580     (7)      626     (8)      628     (8)      650     (8)      667     (8)
                                                 686     (8)      707     (8)      726     (8)      75      (2)      758     (10)
                                                 787     (10)     822     (11)     866     (11)     902     (13)     95      (3)
                                                 960     (13)     999     (15)
$ENV             =00000003     65      (2)
$ER              =FFFFFFFF     1941    (28)    #-1011    (15)   #-102     (3)    #-1022    (15)   #-103     (3)    #-1039    (15)
                                               #-1060    (15)   #-1107    (17)   #-1114    (17)   #-1166    (19)   #-1178    (19)
                                               #-1185    (19)   #-1263    (21)   #-1269    (21)   #-1289    (21)   #-1298    (21)
                                               #-1304    (21)   #-1315    (21)   #-135     (4)    #-136     (4)    #-1363    (22)
                                               #-1369    (22)   #-1382    (22)   #-1388    (22)   #-140     (4)    #-1401    (22)
                                               #-1407    (22)   #-141     (4)    #-1413    (22)   #-148     (4)    #-149     (4)
                                               #-1491    (24)   #-1497    (24)   #-1530    (24)   #-1536    (24)   #-1544    (24)
                                               #-1551    (24)   #-1561    (24)   #-1567    (24)   #-1578    (24)     1659    (26)
                                               #-1669    (26)   #-1675    (26)   #-1685    (26)   #-1696    (26)     1703    (26)
                                               #-1718    (26)   #-1724    (26)   #-1734    (26)   #-1756    (26)   #-1762    (26)
                                               #-1771    (26)     1775    (26)   #-1785    (26)   #-1791    (26)   #-1801    (26)
                                               #-1812    (26)   #-1884    (28)   #-1895    (28)   #-1904    (28)   #-1920    (28)
                                               #-1932    (28)     199     (5)    #-213     (5)      217     (5)    #-231     (5)
                                                 235     (5)    #-253     (5)      259     (5)    #-277     (5)      283     (5)
                                               #-298     (5)      350     (6)    #-363     (6)      367     (6)    #-380     (6)
                                                 384     (6)    #-397     (6)      403     (6)    #-418     (6)      424     (6)
                                               #-438     (6)      488     (7)    #-501     (7)      505     (7)    #-518     (7)
                                                 522     (7)    #-535     (7)      541     (7)    #-556     (7)      562     (7)
                                               #-576     (7)      628     (8)    #-646     (8)      650     (8)    #-663     (8)
                                                 667     (8)    #-680     (8)      686     (8)    #-701     (8)      707     (8)
                                               #-722     (8)    #-776     (10)   #-784     (10)   #-830     (11)   #-845     (11)
                                               #-851     (11)   #-857     (11)   #-863     (11)   #-915     (13)   #-933     (13)
                                               #-942     (13)   #-950     (13)   #-957     (13)
$MO              =00000001     1942    (28)      1942    (28)
$S$              =00000108-R   1775    (26)      1659    (26)     1701    (26)     1703    (26)     1773    (26)     1775    (26)
                                                 1817    (26)     199     (5)      215     (5)      217     (5)      233     (5)
                                                 235     (5)      257     (5)      259     (5)      281     (5)      283     (5)
                                                 302     (5)      350     (6)      365     (6)      367     (6)      382     (6)
                                                 384     (6)      401     (6)      403     (6)      422     (6)      424     (6)
                                                 440     (6)      488     (7)      503     (7)      505     (7)      520     (7)
                                                 522     (7)      539     (7)      541     (7)      560     (7)      562     (7)
                                                 578     (7)      628     (8)      648     (8)      650     (8)      665     (8)
                                                 667     (8)      684     (8)      686     (8)      705     (8)      707     (8)
                                                 724     (8)
$ST              =00000000     1941    (28)      1011    (15)     1012    (15)     1022    (15)     1023    (15)     103     (3)
                                                 1039    (15)     104     (3)      1040    (15)     1060    (15)     1061    (15)
                                                 1063    (15)     107     (3)      1095    (17)     1107    (17)     1108    (17)
                                                 1114    (17)     1115    (17)     1116    (17)     1152    (19)     1166    (19)
                                                 1167    (19)     1178    (19)     1179    (19)     1185    (19)     1186    (19)
                                                 1187    (19)     1247    (21)     1263    (21)     1264    (21)     1269    (21)
                                                 1270    (21)     1289    (21)     1290    (21)     1298    (21)     1299    (21)
                                                 1304    (21)     1305    (21)     1315    (21)     1316    (21)     1317    (21)
                                                 132     (4)      1351    (22)     136     (4)      1363    (22)     1364    (22)
```

```
                                          1369  (22)    137   (4)     1370  (22)    1382  (22)    1383  (22)
                                          1388  (22)    1389  (22)    1401  (22)    1402  (22)    1407  (22)
                                          1408  (22)    141   (4)     1413  (22)    1414  (22)    1415  (22)
                                          142   (4)     1474  (24)    149   (4)     1491  (24)    1492  (24)
                                          1497  (24)    1498  (24)    150   (4)     152   (4)     1530  (24)
                                          1531  (24)    1536  (24)    1537  (24)    1544  (24)    1545  (24)
                                          1551  (24)    1552  (24)    1561  (24)    1562  (24)    1567  (24)
                                          1568  (24)    1578  (24)    1579  (24)    1585  (24)    1648  (26)
                                          1659  (26)    1669  (26)    1670  (26)    1675  (26)    1676  (26)
                                          1685  (26)    1686  (26)    1696  (26)    1697  (26)    1701  (26)
                                          1703  (26)    1714  (26)    1718  (26)    1719  (26)    1724  (26)
                                          1725  (26)    1734  (26)    1735  (26)    1756  (26)    1757  (26)
                                          1762  (26)    1763  (26)    1771  (26)    1772  (26)    1773  (26)
                                          1775  (26)    1785  (26)    1786  (26)    1791  (26)    1792  (26)
                                          1801  (26)    1802  (26)    1812  (26)    1813  (26)    1817  (26)
                                          1818  (26)    1870  (28)    1884  (28)    1885  (28)    1895  (28)
                                          1896  (28)    1904  (28)    1905  (28)    1920  (28)    1921  (28)
                                          1932  (28)    1933  (28)    1941  (28)    197   (5)     199   (5)
                                          213   (5)     214   (5)     215   (5)     217   (5)     231   (5)
                                          232   (5)     233   (5)     235   (5)     253   (5)     254   (5)
                                          257   (5)     259   (5)     277   (5)     278   (5)     281   (5)
                                          283   (5)     298   (5)     301   (5)     302   (5)     303   (5)
                                          348   (6)     350   (6)     363   (6)     364   (6)     365   (6)
                                          367   (6)     380   (6)     381   (6)     382   (6)     384   (6)
                                          397   (6)     398   (6)     401   (6)     403   (6)     418   (6)
                                          419   (6)     422   (6)     424   (6)     438   (6)     439   (6)
                                          440   (6)     441   (6)     486   (7)     488   (7)     501   (7)
                                          502   (7)     503   (7)     505   (7)     518   (7)     519   (7)
                                          520   (7)     522   (7)     535   (7)     536   (7)     539   (7)
                                          541   (7)     556   (7)     557   (7)     560   (7)     562   (7)
                                          576   (7)     577   (7)     578   (7)     579   (7)     626   (8)
                                          628   (8)     646   (8)     647   (8)     648   (8)     650   (8)
                                          663   (8)     664   (8)     665   (8)     667   (8)     680   (8)
                                          681   (8)     684   (8)     686   (8)     701   (8)     702   (8)
                                          705   (8)     707   (8)     722   (8)     723   (8)     724   (8)
                                          725   (8)     758   (10)    776   (10)    777   (10)    784   (10)
                                          785   (10)    786   (10)    822   (11)    830   (11)    831   (11)
                                          845   (11)    846   (11)    851   (11)    852   (11)    857   (11)
                                          858   (11)    863   (11)    864   (11)    865   (11)    902   (13)
                                          915   (13)    916   (13)    933   (13)    934   (13)    942   (13)
                                          943   (13)    95    (3)     950   (13)    951   (13)    957   (13)
                                          958   (13)    959   (13)    999   (15)
$TN              =00000011    1942  (28)  1002  (15)    1011  (15)    1022  (15)    1039  (15)    1060  (15)
                                          1063  (15)    1064  (15)    107   (3)     108   (3)     1088  (17)
                                          1095  (17)    1098  (17)    1102  (17)    1107  (17)    1114  (17)
                                          1116  (17)    1117  (17)    1145  (19)    1152  (19)    1155  (19)
                                          1159  (19)    1166  (19)    1178  (19)    1185  (19)    1187  (19)
                                          1188  (19)    1232  (21)    1247  (21)    1250  (21)    1254  (21)
                                          1263  (21)    1269  (21)    1289  (21)    1298  (21)    1304  (21)
                                          1315  (21)    1317  (21)    1318  (21)    132   (4)     1344  (22)
                                          1351  (22)    1354  (22)    1358  (22)    1363  (22)    1369  (22)
                                          1382  (22)    1388  (22)    1401  (22)    1407  (22)    1413  (22)
                                          1415  (22)    1416  (22)    1459  (24)    1474  (24)    1477  (24)
                                          1481  (24)    1491  (24)    1497  (24)    152   (4)     153   (4)
                                          1530  (24)    1536  (24)    1544  (24)    1551  (24)    1561  (24)
                                          1567  (24)    1578  (24)    1585  (24)    1586  (24)    1633  (26)
```

```
                                                   1648    (26)        1651    (26)        1655    (26)        1669    (26)        1675    (26)
                                                   1685    (26)        1696    (26)        1718    (26)        1724    (26)        1734    (26)
                                                   1756    (26)        1762    (26)        1771    (26)        1785    (26)        1791    (26)
                                                   1801    (26)        1812    (26)        1818    (26)        1819    (26)        1855    (28)
                                                   1870    (28)        1873    (28)        1877    (28)        1884    (28)        1895    (28)
                                                   1904    (28)        1908    (28)        1920    (28)        1932    (28)        1941    (28)
                                                   1942    (28)        197     (5)         213     (5)         231     (5)         253     (5)
                                                   277     (5)         298     (5)         303     (5)         304     (5)         348     (6)
                                                   363     (6)         380     (6)         397     (6)         418     (6)         438     (6)
                                                   441     (6)         442     (6)         486     (7)         501     (7)         518     (7)
                                                   535     (7)         556     (7)         576     (7)         579     (7)         580     (7)
                                                   626     (8)         646     (8)         663     (8)         680     (8)         701     (8)
                                                   722     (8)         725     (8)         726     (8)         75      (2)         751     (10)
                                                   758     (10)        761     (10)        776     (10)        784     (10)        786     (10)
                                                   787     (10)        815     (11)        822     (11)        825     (11)        830     (11)
                                                   845     (11)        851     (11)        857     (11)        863     (11)        865     (11)
                                                   866     (11)        895     (13)        902     (13)        905     (13)        915     (13)
                                                   933     (13)        935     (13)        942     (13)        95      (3)         950     (13)
                                                   957     (13)        959     (13)        960     (13)        992     (15)        999     (15)
ALL                =00000002      67      (2)      1095    (17)        1152    (19)        1247    (21)        132     (4)         1351    (22)
                                                   1474    (24)        1648    (26)        1870    (28)        197     (5)         348     (6)
                                                   486     (7)         626     (8)         758     (10)        822     (11)        902     (13)
                                                   95      (3)         999     (15)
BASE_001            00000000-R    75      (2)      95      (3)
BASE_002            00000000-R    108     (3)      132     (4)
BASE_003            00000000-R    153     (4)      197     (5)
BASE_004            00000000-R    304     (5)      348     (6)
BASE_005            00000000-R    442     (6)      486     (7)
BASE_006            00000000-R    580     (7)      626     (8)
BASE_007            00000000-R    726     (8)      758     (10)
BASE_008            00000000-R    787     (10)     822     (11)
BASE_009            00000000-R    866     (11)     902     (13)
BASE_010            00000000-R    960     (13)     999     (15)
BASE_011            00000000-R    1064    (15)     1095    (17)
BASE_012            00000000-R    1117    (17)     1152    (19)
BASE_013            00000000-R    1188    (19)     1247    (21)
BASE_014            00000000-R    1318    (21)     1351    (22)
BASE_015            00000000-R    1416    (22)     1474    (24)
BASE_016            00000000-R    1586    (24)     1648    (26)
BASE_017            00000000-R    1819    (26)     1870    (28)
BASE_ADDR           00000000-XR                   #-204    (5)        #-355    (6)        #-493    (7)        #-633    (8)        #-96     (3)
BIT...             =00000028      148     (4)      102     (3)          135    (4)          140    (4)          148    (4)          65     (2)
                                                   68      (2)          70     (2)
BIT0               =00000001      66      (2)     #-389    (6)        #-409    (6)        #-672    (8)        #-692    (8)
BIT1               =00000002      66      (2)     #-241    (5)        #-265    (5)        #-527    (7)        #-547    (7)
BIT10              =00000400      66      (2)
BIT11              =00000800      66      (2)
BIT12              =00001000      66      (2)
BIT13              =00002000      66      (2)
BIT14              =00004000      66      (2)
BIT15              =00008000      66      (2)
BIT16              =00010000      66      (2)
BIT17              =00020000      66      (2)
BIT18              =00040000      66      (2)
BIT19              =00080000      66      (2)
BIT2               =00000004      66      (2)
```

```
BIT20           =00100000    66     (2)
BIT21           =00200000    66     (2)
BIT22           =00400000    66     (2)
BIT23           =00800000    66     (2)
BIT24           =01000000    66     (2)
BIT25           =02000000    66     (2)
BIT26           =04000000    66     (2)
BIT27           =08000000    66     (2)
BIT28           =10000000    66     (2)
BIT29           =20000000    66     (2)
BIT3            =00000008    66     (2)
BIT30           =40000000    66     (2)
BIT31           =80000000    66     (2)
BIT4            =00000010    66     (2)
BIT5            =00000020    66     (2)
BIT6            =00000040    66     (2)
BIT7            =00000080    66     (2)
BIT8            =00000100    66     (2)
BIT9            =00000200    66     (2)
CEP_FUNCTIONAL  =00000000    65     (2)
CEP_REPAIR      =00000001    65     (2)
CHC$_ABORT      =00000002    68     (2)
CHC$_CLEAR      =00000006    68     (2)    #-139     (4)
CHC$_CLRDFT     =00000009    68     (2)
CHC$_DSINT      =00000005    68     (2)
CHC$_ENINT      =00000004    68     (2)
CHC$_INITA      =00000000    68     (2)    #-134     (4)
CHC$_INITB      =00000001    68     (2)
CHC$_PURGE      =00000003    68     (2)
CHC$_SELF_TEST  =0000000B    68     (2)
CHC$_SETDFT     =00000008    68     (2)
CHC$_STATUS     =00000007    68     (2)
CHC$_STOP       =0000000A    68     (2)
CHI$M_CHNINT    =00000001    68     (2)
CHI$M_DEVINT    =00000002    68     (2)
CHI$M_IPL       =0000007C    68     (2)
CHI$S_IPL       =00000005    68     (2)
CHI$V_CHNINT    =00000000    68     (2)
CHI$V_DEVINT    =00000001    68     (2)
CHI$V_IPL       =00000002    68     (2)    #-944     (13)
CHM$_FORWARD    =00000000    68     (2)      68      (2)
CHM$_INVALIDATE=00000001     68     (2)      68      (2)
CHM$_MAP        =00000002    68     (2)      68      (2)
CHM$_MFWDN      =00000002    68     (2)
CHM$_MFWDNO     =0000000A    68     (2)
CHM$_MFWDV      =00000003    68     (2)
CHM$_MFWDVO     =0000000B    68     (2)
CHM$_MREVN      =00000006    68     (2)
CHM$_MREVNO     =0000000E    68     (2)
CHM$_MREVV      =00000007    68     (2)
CHM$_MREVVO     =0000000F    68     (2)
CHM$_NFWDN      =00000000    68     (2)
CHM$_NREVN      =00000004    68     (2)
CHM$_OFFSET     =00000008    68     (2)      68      (2)
CHM$_REVERSE    =00000004    68     (2)      68      (2)
CHS$M_BADBDP    =20000000    68     (2)
```

```
CHS$M_BADPAR   =40000000       68      (2)
CHS$M_BUSIC    =00800000       68      (2)
CHS$M_BUSINIT  =00400000       68      (2)
CHS$M_BUSNXM   =08000000       68      (2)
CHS$M_BUSPDN   =01000000       68      (2)
CHS$M_CHNDPE   =00000040       68      (2)
CHS$M_CHNERR   =00000002       68      (2)      68      (2)
CHS$M_CHNMPE   =00000080       68      (2)
CHS$M_CHPFOT   =00000100       68      (2)
CHS$M_DEVBUS   =00000010       68      (2)
CHS$M_DEVERR   =00000004       68      (2)      68      (2)
CHS$M_DEVTO    =00000020       68      (2)
CHS$M_ERRANY   =0000000F       68      (2)
CHS$M_MBAATN   =00100000       68      (2)
CHS$M_MBACPE   =00200000       68      (2)
CHS$M_MBADTB   =00040000       68      (2)
CHS$M_MBADTC   =00080000       68      (2)
CHS$M_MBAEXC   =00010000       68      (2)
CHS$M_MBANED   =00020000       68      (2)
CHS$M_MBAWCKLWR=02000000       68      (2)
CHS$M_MBAWCKUPR=04000000       68      (2)
CHS$M_PGMERR   =00000008       68      (2)      68      (2)
CHS$M_PGMHDE   =00000800       68      (2)
CHS$M_SYSERR   =00000001       68      (2)      68      (2)
CHS$M_SYSMEM   =00000200       68      (2)
CHS$M_SYSSBI   =00000400       68      (2)
CHS$M_UIE      =10000000       68      (2)
CHS$V_BADBDP   =0000001D       68      (2)
CHS$V_BADPAR   =0000001E       68      (2)
CHS$V_BUSIC    =00000017       68      (2)
CHS$V_BUSINIT  =00000016       68      (2)
CHS$V_BUSNXM   =0000001B       68      (2)
CHS$V_BUSPDN   =00000018       68      (2)
CHS$V_CHNDPE   =00000006       68      (2)
CHS$V_CHNERR   =00000001       68      (2)
CHS$V_CHNMPE   =00000007       68      (2)
CHS$V_CHPFOT   =00000008       68      (2)
CHS$V_DEVBUS   =00000004       68      (2)
CHS$V_DEVERR   =00000002       68      (2)
CHS$V_DEVTO    =00000005       68      (2)
CHS$V_MBAATN   =00000014       68      (2)
CHS$V_MBACPE   =00000015       68      (2)
CHS$V_MBADTB   =00000012       68      (2)
CHS$V_MBADTC   =00000013       68      (2)
CHS$V_MBAEXC   =00000010       68      (2)
CHS$V_MBANED   =00000011       68      (2)
CHS$V_MBAWCKLWR=00000019       68      (2)
CHS$V_MBAWCKUPR=0000001A       68      (2)
CHS$V_PGMERR   =00000003       68      (2)
CHS$V_PGMHDE   =0000000B       68      (2)
CHS$V_SYSERR   =00000000       68      (2)
CHS$V_SYSMEM   =00000009       68      (2)
CHS$V_SYSSBI   =0000000A       68      (2)
CHS$V_UIE      =0000001C       68      (2)
CONFIG          00000000-XR                     #-1007  (15)    #-1103  (17)    #-1160  (19)    #-1258  (21)    #-1359  (22)
                                                #-1485  (24)    #-1525  (24)    #-1546  (24)    #-1658  (26)    #-1687  (26)
```

```
                                              #-1698   (26)   #-1736   (26)   #-1803   (26)   #-1814   (26)   #-1879   (28)
                                              #-762    (10)   #-826    (11)   #-841    (11)   #-907    (13)
CRC_TABLE        00000000-XR                   1053    (15)
CSR$M_CRDY      =00000080                     #-1514   (24)   #-224    (5)    #-243    (5)    #-268    (5)    #-289    (5)
                                              #-634    (8)    #-908    (13)
CSR$M_DRDY      =00000001                     #-909    (13)
CSR$M_IE        =00000040                     #-908    (13)   #-922    (13)   #-930    (13)   #-936    (13)
CSR$M_IR        =01000000                     #-931    (13)   #-937    (13)
CSR$V_DRDY      =00000000                     #-1532   (24)
CSR$V_FMT       =0000001D                     #-256    (5)    #-280    (5)
CSR_BITS         00000000-XR                  #-1536   (24)   #-213    (5)    #-231    (5)    #-253    (5)    #-277    (5)
                                              #-298    (5)    #-942    (13)
CURR_CYLN        00000000-XR                  #-1272   (21)   #-1276   (21)   #-1278   (21)   #-1372   (22)   #-1390   (22)
                                              #-1409   (22)   #-1413   (22)   #-1500   (24)   #-1503   (24)   #-1506   (24)
                                              #-1677   (26)   #-1679   (26)   #-1688   (26)   #-1690   (26)   #-1699   (26)
                                              #-1726   (26)   #-1728   (26)   #-1737   (26)   #-1793   (26)   #-1795   (26)
                                              #-1804   (26)   #-1806   (26)   #-1815   (26)   #-1897   (28)   #-1906   (28)
CYLN_LIST        00000000-XR                   1880    (28)
CYLN_LIST_X      00000000-XR                  #-1935   (28)
DATA_001         0000001C-R   95      (3)      95      (3)
DATA_002         00000020-R   132     (4)      132     (4)
DATA_003         00000018-R   197     (5)      197     (5)
DATA_004         00000018-R   348     (6)      348     (6)
DATA_005         00000018-R   486     (7)      486     (7)
DATA_006         00000018-R   626     (8)      626     (8)
DATA_007         0000001C-R   751     (10)     758     (10)
DATA_008         00000018-R   815     (11)     822     (11)
DATA_009         00000014-R   895     (13)     902     (13)
DATA_010         00000024-R   992     (15)     999     (15)
DATA_011         0000001C-R   1088    (17)     1095    (17)
DATA_012         00000018-R   1145    (19)     1152    (19)
DATA_013         0000001C-R   1232    (21)     1247    (21)
DATA_014         00000020-R   1344    (22)     1351    (22)
DATA_015         00000020-R   1459    (24)     1474    (24)
DATA_016         0000002C-R   1633    (26)     1648    (26)
DATA_017         00000024-R   1855    (28)     1870    (28)
DEFAULT         =00000000     67      (2)      1095    (17)    1152    (19)    1247    (21)    132     (4)     1351    (22)
                                               1474    (24)    1648    (26)    1870    (28)    197     (5)     348     (6)
                                               486     (7)     626     (8)     758     (10)    822     (11)    902     (13)
                                               95      (3)     999     (15)
DEST_CYLN        00000000-XR                  #-1278   (21)   #-1506   (24)   #-1508   (24)   #-1512   (24)
DIFF_WORD        00000000-XR                  #-1259   (21)   #-1486   (24)
DRIVE_TBL        00000000-XR                  #-1003   (15)   #-1099   (17)   #-1156   (19)   #-1251   (21)   #-1355   (22)
                                              #-1478   (24)   #-1652   (26)   #-1874   (28)   #-763    (10)   #-906    (13)
DS$ABORT         00000000-XR                   105     (3)     138     (4)     143     (4)     151     (4)
DS$BGNSUB        00000000-XR                   1659    (26)    1703    (26)    1775    (26)    199     (5)     217     (5)
                                               235     (5)     259     (5)     283     (5)     350     (6)     367     (6)
                                               384     (6)     403     (6)     424     (6)     488     (7)     505     (7)
                                               522     (7)     541     (7)     562     (7)     628     (8)     650     (8)
                                               667     (8)     686     (8)     707     (8)
DS$BRANCH        00000000-XR                   102     (3)     135     (4)     140     (4)     148     (4)
DS$BREAK         00000000-XR                   1063    (15)    107     (3)     1116    (17)    1187    (19)    1317    (21)
                                               1415    (22)    152     (4)     1585    (24)    1818    (26)    1941    (28)
                                               303     (5)     441     (6)     579     (7)     725     (8)     786     (10)
                                               865     (11)    959     (13)
DS$CHANNEL       00000000-XR                   134     (4)     139     (4)
```

```
DS$CKLOOP         00000000-XR                 1012    (15)      1023    (15)      104     (3)       1040    (15)      1061    (15)
                                              1108    (17)      1115    (17)      1167    (19)      1179    (19)      1186    (19)
                                              1264    (21)      1270    (21)      1290    (21)      1299    (21)      1305    (21)
                                              1316    (21)      1364    (22)      137     (4)       1370    (22)      1383    (22)
                                              1389    (22)      1402    (22)      1408    (22)      1414    (22)      142     (4)
                                              1492    (24)      1498    (24)      150     (4)       1531    (24)      1537    (24)
                                              1545    (24)      1552    (24)      1562    (24)      1568    (24)      1579    (24)
                                              1670    (26)      1676    (26)      1686    (26)      1697    (26)      1719    (26)
                                              1725    (26)      1735    (26)      1757    (26)      1763    (26)      1772    (26)
                                              1786    (26)      1792    (26)      1802    (26)      1813    (26)      1885    (28)
                                              1896    (28)      1905    (28)      1921    (28)      1933    (28)      214     (5)
                                              232     (5)       254     (5)       278     (5)       301     (5)       364     (6)
                                              381     (6)       398     (6)       419     (6)       439     (6)       502     (7)
                                              519     (7)       536     (7)       557     (7)       577     (7)       647     (8)
                                              664     (8)       681     (8)       702     (8)       723     (8)       777     (10)
                                              785     (10)      831     (11)      846     (11)      852     (11)      858     (11)
                                              864     (11)      916     (13)      934     (13)      943     (13)      951     (13)
                                              958     (13)
DS$ENDSUB         00000000-XR                 1701    (26)      1773    (26)      1817    (26)      215     (5)       233     (5)
                                              257     (5)       281     (5)       302     (5)       365     (6)       382     (6)
                                              401     (6)       422     (6)       440     (6)       503     (7)       520     (7)
                                              539     (7)       560     (7)       578     (7)       648     (8)       665     (8)
                                              684     (8)       705     (8)       724     (8)
DS$ERRHARD        00000000-XR                 1011    (15)      1022    (15)      1039    (15)      1060    (15)      1107    (17)
                                              1114    (17)      1166    (19)      1178    (19)      1185    (19)      1263    (21)
                                              1269    (21)      1289    (21)      1298    (21)      1304    (21)      1315    (21)
                                              1363    (22)      1369    (22)      1382    (22)      1388    (22)      1401    (22)
                                              1407    (22)      1413    (22)      1491    (24)      1497    (24)      1530    (24)
                                              1536    (24)      1544    (24)      1551    (24)      1561    (24)      1567    (24)
                                              1578    (24)      1669    (26)      1675    (26)      1685    (26)      1696    (26)
                                              1718    (26)      1724    (26)      1734    (26)      1756    (26)      1762    (26)
                                              1771    (26)      1785    (26)      1791    (26)      1801    (26)      1812    (26)
                                              1884    (28)      1895    (28)      1904    (28)      1920    (28)      1932    (28)
                                              213     (5)       231     (5)       253     (5)       277     (5)       298     (5)
                                              363     (6)       380     (6)       397     (6)       418     (6)       438     (6)
                                              501     (7)       518     (7)       535     (7)       556     (7)       576     (7)
                                              646     (8)       663     (8)       680     (8)       701     (8)       722     (8)
                                              776     (10)      784     (10)      830     (11)      845     (11)      851     (11)
                                              857     (11)      863     (11)      915     (13)      933     (13)      942     (13)
                                              950     (13)      957     (13)
DS$ERRSYS         00000000-XR                 103     (3)       136     (4)       141     (4)       149     (4)
DS$K_BBC         =0000001D      148     (4)
DS$K_BBCC        =00000021      148     (4)
DS$K_BBCCI       =00000023      148     (4)
DS$K_BBCS        =0000001F      148     (4)
DS$K_BBS         =0000001C      148     (4)
DS$K_BBSC        =00000020      148     (4)
DS$K_BBSS        =0000001E      148     (4)
DS$K_BBSSI       =00000022      148     (4)
DS$K_BCC_M       =0000001B      148     (4)
DS$K_BCS_M       =0000001A      148     (4)
DS$K_BEQLU_B     =00000005      148     (4)
DS$K_BEQLU_M     =00000011      148     (4)
DS$K_BEQL_B      =00000004      148     (4)
DS$K_BEQL_M      =00000010      148     (4)
DS$K_BERROR      =00000026      148     (4)
```

```
DS$K_BGEQU_B    =00000007      148     (4)
DS$K_BGEQU_M    =00000013      148     (4)
DS$K_BGEQ_B     =00000006      148     (4)
DS$K_BGEQ_M     =00000012      148     (4)
DS$K_BGTRU_B    =00000009      148     (4)
DS$K_BGTRU_M    =00000015      148     (4)
DS$K_BGTR_B     =00000008      148     (4)
DS$K_BGTR_M     =00000014      148     (4)
DS$K_BLBC       =00000025      148     (4)
DS$K_BLBS       =00000024      148     (4)
DS$K_BLEQU_B    =00000003      148     (4)
DS$K_BLEQU_M    =0000000F      148     (4)
DS$K_BLEQ_B     =00000002      148     (4)
DS$K_BLEQ_M     =0000000E      148     (4)
DS$K_BLSSU_B    =00000001      148     (4)
DS$K_BLSSU_M    =0000000D      148     (4)
DS$K_BLSS_B     =00000000      148     (4)
DS$K_BLSS_M     =0000000C      148     (4)
DS$K_BNEQU_B    =0000000B      148     (4)
DS$K_BNEQU_M    =00000017      148     (4)
DS$K_BNEQ_B     =0000000A      148     (4)
DS$K_BNEQ_M     =00000016      148     (4)
DS$K_BNERROR    =00000027      148     (4)    #-102    (3)    #-135    (4)    #-140    (4)    #-148    (4)
DS$K_BVC_M      =00000019      148     (4)
DS$K_BVS_M      =00000018      148     (4)
DS$PROBE         00000000-XR                   101    (3)      147    (4)
E.BA             00000000-XR                  #-357    (6)    #-359    (6)    #-363    (6)    #-374    (6)    #-376    (6)
                                              #-380    (6)    #-391    (6)    #-393    (6)    #-397    (6)    #-412    (6)
                                              #-414    (6)    #-418    (6)    #-432    (6)    #-434    (6)    #-438    (6)
E.BC             00000000-XR                  #-495    (7)    #-497    (7)    #-501    (7)    #-512    (7)    #-514    (7)
                                              #-518    (7)    #-529    (7)    #-531    (7)    #-535    (7)    #-550    (7)
                                              #-552    (7)    #-556    (7)    #-570    (7)    #-572    (7)    #-576    (7)
E.BR             00000000-XR                  #-946    (13)   #-950    (13)
E.CS             00000000-XR                  #-1514   (24)   #-1515   (24)   #-1516   (24)   #-1536   (24)   #-206    (5)
                                              #-209    (5)    #-213    (5)    #-224    (5)    #-227    (5)    #-231    (5)
                                              #-246    (5)    #-249    (5)    #-253    (5)    #-270    (5)    #-273    (5)
                                              #-277    (5)    #-291    (5)    #-294    (5)    #-298    (5)    #-909    (13)
                                              #-910    (13)   #-911    (13)   #-938    (13)   #-942    (13)
E.DA             00000000-XR                  #-640    (8)    #-642    (8)    #-646    (8)    #-657    (8)    #-659    (8)
                                              #-663    (8)    #-674    (8)    #-676    (8)    #-680    (8)    #-695    (8)
                                              #-697    (8)    #-701    (8)    #-716    (8)    #-718    (8)    #-722    (8)
E.MP             00000000-XR                  #-1172   (19)   #-1173   (19)   #-1174   (19)   #-1178   (19)   #-1517   (24)
                                               1518    (24)   #-1539   (24)   #-1544   (24)   #-1553   (24)    1554    (24)
                                              #-1556   (24)   #-1561   (24)
E.VEC            00000000-XR                  #-953    (13)   #-957    (13)
ENV$M_DOMAIN    =00000002      65      (2)
ENV$M_LEVEL     =00000001      65      (2)
ENV$M_SUPER     =000003FC      65      (2)
ENV$S_DOMAIN    =00000001      65      (2)
ENV$S_LEVEL     =00000001      65      (2)
ENV$S_SUPER     =00000008      65      (2)
ENV$V_DOMAIN    =00000001      65      (2)     65     (2)
ENV$V_LEVEL     =00000000      65      (2)     65     (2)
ENV$V_SUPER     =00000002      65      (2)
ENV$_CPU        =00000000      65      (2)     65     (2)
ENV$_FUNCTIONAL =00000000      65      (2)     65     (2)
```

```
ENV$_REPAIR      =00000001     65      (2)       65      (2)
ENV$_SUPER       =00000001     65      (2)
ENV$_SYSTEM      =00000001     65      (2)       65      (2)
ERR_MSG           00000000-XR                   1685     (26)     1696     (26)     1734     (26)     1801     (26)     1812     (26)
                                                1904     (28)     1920     (28)     1932     (28)
GETSTAT          =00000004                     #-771     (10)
GET_LUN$A         00000000-XR                   1001     (15)     1097     (17)     1154     (19)     1249     (21)     1353     (22)
                                                1476     (24)     1650     (26)     1872     (28)     760      (10)     824      (11)
                                                904      (13)
GET_POSITION      00000000-XR                   1266     (21)     1301     (21)     1366     (22)     1385     (22)     1404     (22)
                                                1494     (24)     1564     (24)     1672     (26)     1721     (26)     1759     (26)
                                                1788     (26)     1892     (28)
HEAD_0           =00000000                      1233     (21)     1235     (21)     1236     (21)     1238     (21)     1239     (21)
                                                1241     (21)     1242     (21)     1244     (21)     1460     (24)     1462     (24)
                                                1463     (24)     1465     (24)     1466     (24)     1468     (24)     1469     (24)
                                                1471     (24)     1634     (26)     1636     (26)     1637     (26)     1639     (26)
                                                1640     (26)     1642     (26)     1643     (26)     1645     (26)     1856     (28)
                                                1858     (28)     1859     (28)     1861     (28)     1862     (28)     1864     (28)
                                                1865     (28)     1867     (28)
HEAD_1           =00000001                      1234     (21)     1237     (21)     1240     (21)     1243     (21)     1461     (24)
                                                1464     (24)     1467     (24)     1470     (24)     1635     (26)     1638     (26)
                                                1641     (26)     1644     (26)     1857     (28)     1860     (28)     1863     (28)
                                                1866     (28)
IDCBAR           =00000004                     #-356     (6)     #-358     (6)     #-373     (6)     #-375     (6)     #-390     (6)
                                               #-392     (6)     #-411     (6)     #-413     (6)     #-430     (6)     #-431     (6)
                                               #-433     (6)     #-637     (8)
IDCCNT           =00000008                     #-494     (7)     #-496     (7)     #-511     (7)     #-513     (7)     #-528     (7)
                                               #-530     (7)     #-549     (7)     #-551     (7)     #-568     (7)     #-569     (7)
                                               #-571     (7)     #-638     (8)
IDCCSR           =00000000                     #-205     (5)     #-207     (5)     #-223     (5)     #-225     (5)     #-245     (5)
                                               #-247     (5)     #-269     (5)     #-271     (5)     #-289     (5)     #-290     (5)
                                               #-292     (5)     #-299     (5)     #-634     (8)     #-930     (13)    #-931     (13)
                                               #-936     (13)    #-937     (13)
IDCDAR           =0000000C                     #-639     (8)     #-641     (8)     #-656     (8)     #-658     (8)     #-675     (8)
                                               #-696     (8)     #-714     (8)     #-715     (8)     #-717     (8)
IDCMPR           =00000010                     #-673     (8)     #-694     (8)
IDC_ADRFAIL       00000000-XR                   103      (3)
IDC_SADRFAIL      00000000-XR                   149      (4)
IDC_STAT$L_1      00000000-XR                  #-917     (13)
IDC_STAT$W_2      00000000-XR                  #-926     (13)      944     (13)
IDC_STAT$W_VR     00000000-XR                  #-952     (13)
L$UNIT            00000000-XR                  #-101     (3)     #-1011    (15)    #-1022    (15)    #-103     (3)     #-1039    (15)
                                               #-1060    (15)    #-1107    (17)    #-1114    (17)    #-1166    (19)    #-1178    (19)
                                               #-1185    (19)    #-1263    (21)    #-1269    (21)    #-1289    (21)    #-1298    (21)
                                               #-1304    (21)    #-1315    (21)    #-134     (4)     #-136     (4)     #-1363    (22)
                                               #-1369    (22)    #-1382    (22)    #-1388    (22)    #-139     (4)     #-1401    (22)
                                               #-1407    (22)    #-141     (4)     #-1413    (22)    #-147     (4)     #-149     (4)
                                               #-1491    (24)    #-1497    (24)    #-1530    (24)    #-1536    (24)    #-1544    (24)
                                               #-1551    (24)    #-1561    (24)    #-1567    (24)    #-1578    (24)    #-1669    (26)
                                               #-1675    (26)    #-1685    (26)    #-1696    (26)    #-1718    (26)    #-1724    (26)
                                               #-1734    (26)    #-1756    (26)    #-1762    (26)    #-1771    (26)    #-1785    (26)
                                               #-1791    (26)    #-1801    (26)    #-1812    (26)    #-1884    (28)    #-1895    (28)
                                               #-1904    (28)    #-1920    (28)    #-1932    (28)    #-213     (5)     #-231     (5)
                                               #-253     (5)     #-277     (5)     #-298     (5)     #-363     (6)     #-380     (6)
                                               #-397     (6)     #-418     (6)     #-438     (6)     #-501     (7)     #-518     (7)
                                               #-535     (7)     #-556     (7)     #-576     (7)     #-646     (8)     #-663     (8)
```

```
                                               #-680    (8)    #-701    (8)    #-722    (8)    #-776    (10)   #-784    (10)
                                               #-830    (11)   #-845    (11)   #-851    (11)   #-857    (11)   #-863    (11)
                                               #-915    (13)   #-933    (13)   #-942    (13)   #-950    (13)   #-957    (13)
LUN0             =00000000                      1089    (17)    1146    (19)    1233    (21)    1234    (21)    1235    (21)
                                                1345    (22)    1460    (24)    1461    (24)    1462    (24)    1634    (26)
                                                1635    (26)    1636    (26)    1856    (28)    1857    (28)    1858    (28)
                                                752     (10)    816     (11)    896     (13)    993     (15)
LUN1             =00000001                      1090    (17)    1147    (19)    1236    (21)    1237    (21)    1238    (21)
                                                1346    (22)    1463    (24)    1464    (24)    1465    (24)    1637    (26)
                                                1638    (26)    1639    (26)    1859    (28)    1860    (28)    1861    (28)
                                                753     (10)    817     (11)    897     (13)    994     (15)
LUN2             =00000002                      1091    (17)    1148    (19)    1239    (21)    1240    (21)    1241    (21)
                                                1347    (22)    1466    (24)    1467    (24)    1468    (24)    1640    (26)
                                                1641    (26)    1642    (26)    1862    (28)    1863    (28)    1864    (28)
                                                754     (10)    818     (11)    898     (13)    995     (15)
LUN3             =00000003                      1092    (17)    1149    (19)    1242    (21)    1243    (21)    1244    (21)
                                                1348    (22)    1469    (24)    1470    (24)    1471    (24)    1643    (26)
                                                1644    (26)    1645    (26)    1865    (28)    1866    (28)    1867    (28)
                                                755     (10)    819     (11)    899     (13)    996     (15)
MANUAL           =00000001       67      (2)
MPR_02$M_HS      =00000040                     #-778    (10)
MPR_80$M_SKE     =00000400                     #-1173   (19)
MS107A_SEEKERR    00000000-XR                   1178    (19)
MS107B_RECAL      00000000-XR                   1185    (19)
MS107_RECAL       00000000-XR                   1114    (17)
MS45A_HD_PROX     00000000-XR                  #-1413   (22)
MSG14_SYSINIT     00000000-XR                   1011    (15)    1107    (17)    1166    (19)    1263    (21)    1363    (22)
                                                1491    (24)    1669    (26)    1718    (26)    1785    (26)    1884    (28)
                                                830     (11)    915     (13)
MSG15_NOOP        00000000-XR                   845     (11)    851     (11)    857     (11)    863     (11)
MSG17_GETSTAT     00000000-XR                   776     (10)    784     (10)
MSG19A_READH      00000000-XR                   1762    (26)
MSG19_READH       00000000-XR                   1022    (15)    1039    (15)    1060    (15)    1269    (21)    1304    (21)
                                                1369    (22)    1388    (22)    1407    (22)    1413    (22)    1497    (24)
                                                1567    (24)    1675    (26)    1724    (26)    1791    (26)    1895    (28)
MSG23_CS_INC      00000000-XR                  #-942    (13)
MSG24A_BC_MOD     00000000-XR                  #-857    (11)
MSG24_BA_MOD      00000000-XR                  #-851    (11)
MSG25_DA_MOD      00000000-XR                  #-863    (11)
MSG27_MP_INC      00000000-XR                  #-784    (10)
MSG2_RG_ZERO      00000000-XR                   231     (5)     380     (6)     518     (7)     663     (8)
MSG38_NOOP_IE     00000000-XR                   933     (13)    942     (13)    950     (13)    957     (13)
MSG39_NO_INT      00000000-XR                  #-933    (13)
MSG3_RG_ONE       00000000-XR                   213     (5)     363     (6)     501     (7)     646     (8)
MSG40_BAD_BR      00000000-XR                  #-950    (13)
MSG41_BAD_VEC     00000000-XR                  #-957    (13)
MSG42_RPT_CYL     00000000-XR                  #-1039   (15)
MSG44_BAD_CRC     00000000-XR                  #-1060   (15)
MSG45_SK_ZERO     00000000-XR                   1289    (21)    1298    (21)    1315    (21)    1382    (22)    1401    (22)
MSG46_POS_ERR     00000000-XR                  #-1315   (21)   #-1578   (24)   #-1771   (26)
MSG47_DS_INC      00000000-XR                  #-1178   (19)   #-1298   (21)   #-1544   (24)   #-1561   (24)
MSG48_SK_ONE      00000000-XR                   1530    (24)    1536    (24)    1544    (24)    1551    (24)    1561    (24)
                                                1578    (24)
MSG49_DRDY_NC     00000000-XR                  #-1536   (24)
MSG4_RG_FLT1      00000000-XR                   253     (5)     397     (6)     535     (7)     680     (8)
MSG51_SK_INNR     00000000-XR                   1756    (26)    1771    (26)
```

```
MSG5_RG_FLT0     00000000-XR                277     (5)      418     (6)      556     (7)      701     (8)
MSG6_CS_FAIL     00000000-XR              #-213     (5)    #-231     (5)    #-253     (5)    #-277     (5)    #-298     (5)
MSG7A_BC_FAIL    00000000-XR              #-501     (7)    #-518     (7)    #-535     (7)    #-556     (7)    #-576     (7)
MSG7_BA_FAIL     00000000-XR              #-363     (6)    #-380     (6)    #-397     (6)    #-418     (6)    #-438     (6)
MSG8_DA_FAIL     00000000-XR              #-646     (8)    #-663     (8)    #-680     (8)    #-701     (8)    #-722     (8)
MSG9_RG_RMW      00000000-XR                298     (5)      438     (6)      576     (7)      722     (8)
MSG_001          00000002-R   75      (2)   95      (3)
MSG_002          00000002-R   108     (3)   132     (4)
MSG_003          00000002-R   153     (4)   197     (5)
MSG_004          00000002-R   304     (5)   348     (6)
MSG_005          00000002-R   442     (6)   486     (7)
MSG_006          00000002-R   580     (7)   626     (8)
MSG_007          00000002-R   726     (8)   758     (10)
MSG_008          00000002-R   787     (10)  822     (11)
MSG_009          00000002-R   866     (11)  902     (13)
MSG_010          00000002-R   960     (13)  999     (15)
MSG_011          00000002-R   1064    (15)  1095    (17)
MSG_012          00000002-R   1117    (17)  1152    (19)
MSG_013          00000002-R   1188    (19)  1247    (21)
MSG_014          00000002-R   1318    (21)  1351    (22)
MSG_015          00000002-R   1416    (22)  1474    (24)
MSG_016          00000002-R   1586    (24)  1648    (26)
MSG_017          00000002-R   1819    (26)  1870    (28)
NEW_CYLN         00000000-XR              #-1679    (26)   #-1681    (26)   #-1690    (26)   #-1692    (26)   #-1728    (26)
                                          #-1730    (26)   #-1795    (26)   #-1797    (26)   #-1806    (26)   #-1808    (26)
                                          #-1914    (28)   #-1916    (28)
NOCHK$M_ATTN    =00000080                 #-1525    (24)
NOCHK$M_CRDY    =00000001                 #-1546    (24)
NOCHK$M_DRDY    =00000004                 #-1525    (24)
NOCHK$M_DSTA    =00000010                 #-841     (11)
NO_OP           =00000000                 #-839     (11)   #-908     (13)   #-922     (13)
OFST_AVGTIM     =00000080     70      (2)
OFST_CSRTYP     =00000058     70      (2) #-1516    (24)   #-911     (13)
OFST_CYLPOS     =00000000                 #-1272    (21)   #-1508    (24)   #-1740    (26)
OFST_CYLSIZ     =00000004                 #-1273    (21)   #-1509    (24)   #-1741    (26)
OFST_DEVMEM     =0000001C     70      (2)
OFST_DEVNAM     =00000020     70      (2)
OFST_DRIV       =00000008                 #-1252    (21)   #-1479    (24)   #-1653    (26)   #-1875    (28)
OFST_DRVERR     =0000004C     70      (2)
OFST_DRVRST     =00000050     70      (2) #-764     (10)
OFST_DRVTYP     =00000000     70      (2) #-1100    (17)   #-1157    (19)   #-1252    (21)   #-1356    (22)   #-1479    (24)
                                          #-1653    (26)   #-1712    (26)   #-1875    (28)
OFST_FMTLEN     =00000030     70      (2)
OFST_GTSCMD     =00000034     70      (2)
OFST_HDBITS     =00000044     70      (2) #-1005    (15)   #-1257    (21)   #-1484    (24)   #-1657    (26)
OFST_HDRDNC     =00000054     70      (2) #-1029    (15)   #-1034    (15)   #-1308    (21)   #-1571    (24)   #-1766    (26)
OFST_HDRTMP     =00000048     70      (2) #-1004    (15)   #-1255    (21)   #-1482    (24)   #-1656    (26)
OFST_HIGRPM     =00000074     70      (2)
OFST_HMBSEC     =0000005C     70      (2)
OFST_IGBCYL     =00000018     70      (2) #-1740    (26)
OFST_LOWRPM     =00000070     70      (2)
OFST_LSTCYL     =00000008     70      (2) #-1161    (19)   #-1500    (24)   #-1699    (26)   #-1726    (26)   #-1737    (26)
                                          #-1744    (26)   #-1793    (26)   #-1804    (26)   #-1878    (28)
OFST_LSTSEC     =00000010     70      (2)
OFST_LSTTRK     =0000000C     70      (2)
OFST_MAXTIM     =0000007C     70      (2)
```

```
OFST_MPBITS      =00000040    70    (2)   #-1162  (19)  #-1256  (21)  #-1483  (24)  #-766   (10)
OFST_NMLSTS      =00000038    70    (2)   #-1172  (19)  #-1291  (21)  #-1553  (24)  #-765   (10)
OFST_ONETIM      =00000078    70    (2)
OFST_OP1STS      =00000060    70    (2)
OFST_PHYU        =00000004    70    (2)
OFST_RANCYL      =0000006C    70    (2)
OFST_SECLEN      =00000028    70    (2)
OFST_SECPOS      =0000000A                #-1027  (15)  #-1032  (15)  #-1306  (21)  #-1569  (24)  #-1764  (26)
OFST_SECSIZ      =0000000E                #-1028  (15)  #-1033  (15)  #-1307  (21)  #-1570  (24)  #-1765  (26)
OFST_SEKSTS      =0000003C    70    (2)   #-1517  (24)
OFST_SKHD        =0000000C                #-1274  (21)  #-1277  (21)  #-1292  (21)  #-1510  (24)  #-1513  (24)
                                          #-1518  (24)  #-1554  (24)  #-1680  (26)  #-1691  (26)  #-1729  (26)
                                          #-1742  (26)  #-1746  (26)  #-1796  (26)  #-1807  (26)  #-1899  (28)
                                          #-1915  (28)  #-1927  (28)
OFST_STSDNC      =00000064    70    (2)   #-1171  (19)  #-1293  (21)  #-1538  (24)  #-1555  (24)  #-779   (10)
OFST_TRKLEN      =0000002C    70    (2)
OFST_TRKPOS      =00000005                #-1274  (21)  #-1510  (24)  #-1742  (26)
OFST_TRKSIZ      =00000009                #-1275  (21)  #-1511  (24)  #-1743  (26)
OFST_TSTCYL      =00000014    70    (2)
OFST_UNIT        =00000004                #-1001  (15)  #-1097  (17)  #-1154  (19)  #-1249  (21)  #-1353  (22)
                                          #-1476  (24)  #-1650  (26)  #-1872  (28)  #-760   (10)  #-824   (11)
                                          #-904   (13)
OFST_WTPSTS      =00000068    70    (2)
P$UNIT            00000000-XR             #-1515  (24)  #-910   (13)
POSITION_HDS      00000000-XR               1111  (17)    1170  (19)    1182  (19)    1901  (28)    1917  (28)
                                            1929  (28)
PRINT_REG         00000000-XR               1011  (15)    1022  (15)    1107  (17)    1114  (17)    1166  (19)
                                            1185  (19)    1263  (21)    1269  (21)    1289  (21)    1304  (21)
                                            1363  (22)    1369  (22)    1382  (22)    1388  (22)    1401  (22)
                                            1407  (22)    1491  (24)    1497  (24)    1530  (24)    1551  (24)
                                            1567  (24)    1669  (26)    1675  (26)    1685  (26)    1696  (26)
                                            1718  (26)    1724  (26)    1734  (26)    1756  (26)    1762  (26)
                                            1785  (26)    1791  (26)    1801  (26)    1812  (26)    1884  (28)
                                            1895  (28)    1904  (28)    1920  (28)    1932  (28)    776   (10)
                                            830   (11)    845   (11)    915   (13)    933   (13)
PRT$M_EXT_1      =00000008                #-1287  (21)  #-1311  (21)  #-1528  (24)  #-1533  (24)  #-1541  (24)
                                          #-1549  (24)  #-1558  (24)  #-1574  (24)
PRT$M_EXT_2      =00000010                #-1311  (21)  #-1574  (24)
PRT$M_NONE       =00000040                #-198   (5)   #-349   (6)   #-487   (7)   #-627   (8)
R80$T            =00000002                #-1100  (17)  #-1157  (19)    1235  (21)    1238  (21)    1241  (21)
                                            1244  (21)    1462  (24)    1465  (24)    1468  (24)    1471  (24)
                                            1636  (26)    1639  (26)    1642  (26)    1645  (26)    1858  (28)
                                            1861  (28)    1864  (28)    1867  (28)
RD_IDCREGS        00000000-XR               925   (13)
READH            =00000008                #-1017  (15)
REG_ERRPT         00000000-XR               1039  (15)    1060  (15)    1178  (19)    1298  (21)    1315  (21)
                                            1413  (22)    1536  (24)    1544  (24)    1561  (24)    1578  (24)
                                            1771  (26)    213   (5)     231   (5)     253   (5)     277   (5)
                                            298   (5)     363   (6)     380   (6)     397   (6)     418   (6)
                                            438   (6)     501   (7)     518   (7)     535   (7)     556   (7)
                                            576   (7)     646   (8)     663   (8)     680   (8)     701   (8)
                                            722   (8)     784   (10)    851   (11)    857   (11)    863   (11)
                                            942   (13)    950   (13)    957   (13)
RGLIST            00000000-XR             #-1287  (21)  #-1312  (21)  #-1528  (24)  #-1533  (24)  #-1541  (24)
                                          #-1549  (24)  #-1558  (24)  #-1575  (24)  #-198   (5)   #-349   (6)
                                          #-487   (7)   #-627   (8)
```

```
RL02$T          =00000001                 1233   (21)       1234   (21)       1236   (21)       1237   (21)       1239   (21)
                                          1240   (21)       1242   (21)       1243   (21)     #-1356   (22)       1460   (24)
                                          1461   (24)       1463   (24)       1464   (24)       1466   (24)       1467   (24)
                                          1469   (24)       1470   (24)       1634   (26)       1635   (26)       1637   (26)
                                          1638   (26)       1640   (26)       1641   (26)       1643   (26)       1644   (26)
                                        #-1712   (26)       1856   (28)       1857   (28)       1859   (28)       1860   (28)
                                          1862   (28)       1863   (28)       1865   (28)       1866   (28)
SECD_ADDR        00000000-XR            #-133    (4)
SEEK            =00000006               #-1283   (21)     #-1377   (22)     #-1396   (22)     #-1514   (24)     #-1523   (24)
                                        #-1751   (26)
SEP_FUNCTIONAL  =00000002     65     (2)
SEP_REPAIR      =00000003     65     (2)  65     (2)
SIZ...          =00000001     68     (2)  65     (2)        68     (2)
START_CMD        00000000-XR              1018   (15)       1284   (21)       1378   (22)       1397   (22)       1524   (24)
                                          1752   (26)       772    (10)       840    (11)       923    (13)
STATUS_CHK       00000000-XR              1019   (15)       1285   (21)       1379   (22)       1398   (22)       1526   (24)
                                          1547   (24)       1753   (26)       773    (10)       842    (11)
STEP_HDS         00000000-XR              1682   (26)       1693   (26)       1731   (26)       1798   (26)       1809   (26)
SUBSYS_INIT      00000000-XR              1008   (15)       1104   (17)       1163   (19)       1260   (21)       1360   (22)
                                          1488   (24)       1666   (26)       1715   (26)       1782   (26)       1881   (28)
                                          827    (11)       912    (13)
T.BA             00000000-XR            #-358    (6)      #-359    (6)      #-363    (6)      #-375    (6)      #-376    (6)
                                        #-380    (6)      #-392    (6)      #-393    (6)      #-397    (6)      #-413    (6)
                                        #-414    (6)      #-418    (6)      #-433    (6)      #-434    (6)      #-438    (6)
                                        #-847    (11)     #-851    (11)
T.BC             00000000-XR            #-496    (7)      #-497    (7)      #-501    (7)      #-513    (7)      #-514    (7)
                                        #-518    (7)      #-530    (7)      #-531    (7)      #-535    (7)      #-551    (7)
                                        #-552    (7)      #-556    (7)      #-571    (7)      #-572    (7)      #-576    (7)
                                        #-853    (11)     #-857    (11)
T.BR             00000000-XR            #-945    (13)     #-946    (13)     #-950    (13)
T.CS             00000000-XR              1532   (24)     #-1536   (24)     #-207    (5)      #-208    (5)      #-209    (5)
                                        #-213    (5)      #-225    (5)      #-226    (5)      #-227    (5)      #-231    (5)
                                        #-247    (5)      #-248    (5)      #-249    (5)      #-253    (5)      #-271    (5)
                                        #-272    (5)      #-273    (5)      #-277    (5)      #-292    (5)      #-293    (5)
                                        #-294    (5)      #-298    (5)      #-938    (13)     #-942    (13)
T.DA             00000000-XR            #-641    (8)      #-642    (8)      #-646    (8)      #-658    (8)      #-659    (8)
                                        #-663    (8)      #-675    (8)      #-676    (8)      #-680    (8)      #-696    (8)
                                        #-697    (8)      #-701    (8)      #-717    (8)      #-718    (8)      #-722    (8)
                                        #-859    (11)     #-863    (11)
T.HDR            00000000-XR            #-1026   (15)     #-1031   (15)       1307   (21)     #-1308   (21)     #-1309   (21)
                                        #-1315   (21)       1570   (24)     #-1571   (24)     #-1572   (24)     #-1578   (24)
                                          1765   (26)     #-1766   (26)     #-1767   (26)     #-1771   (26)
T.MP             00000000-XR            #-1171   (19)     #-1174   (19)     #-1178   (19)     #-1293   (21)     #-1294   (21)
                                        #-1298   (21)     #-1538   (24)     #-1539   (24)     #-1544   (24)     #-1555   (24)
                                        #-1556   (24)     #-1561   (24)     #-778    (10)     #-779    (10)     #-780    (10)
                                        #-784    (10)
T.MP1            00000000-XR            #-1050   (15)
T.MP2            00000000-XR            #-1051   (15)
T.MP3            00000000-XR            #-1056   (15)     #-1060   (15)
T.VEC            00000000-XR            #-952    (13)     #-953    (13)     #-957    (13)
T16_S1           00000121-R   1659   (26)
T16_S1_X         00000202-R   1701   (26)
T16_S2           0000020B-R   1703   (26)
T16_S2_X         00000379-R   1773   (26)  #-1714   (26)
T16_S3           00000382-R   1775   (26)
T16_S3_X         00000465-R   1817   (26)
```

```
T3_S1           00000027-R   199   (5)
T3_S1_X         0000008D-R   215   (5)
T3_S2           00000096-R   217   (5)
T3_S2_X         000000F1-R   233   (5)
T3_S3           000000FA-R   235   (5)
T3_S3_X         0000016C-R   257   (5)
T3_S4           00000175-R   259   (5)
T3_S4_X         000001EF-R   281   (5)
T3_S5           000001F8-R   283   (5)
T3_S5_X         00000264-R   302   (5)
T4_S1           00000027-R   350   (6)
T4_S1_X         00000080-R   365   (6)
T4_S2           00000089-R   367   (6)
T4_S2_X         000000D3-R   382   (6)
T4_S3           000000DC-R   384   (6)
T4_S3_X         00000133-R   401   (6)
T4_S4           0000013C-R   403   (6)
T4_S4_X         0000019B-R   422   (6)
T4_S5           000001A4-R   424   (6)
T4_S5_X         000001FB-R   440   (6)
T5_S1           00000027-R   488   (7)
T5_S1_X         0000007E-R   503   (7)
T5_S2           00000087-R   505   (7)
T5_S2_X         000000D1-R   520   (7)
T5_S3           000000DA-R   522   (7)
T5_S3_X         00000131-R   539   (7)
T5_S4           0000013A-R   541   (7)
T5_S4_X         00000199-R   560   (7)
T5_S5           000001A2-R   562   (7)
T5_S5_X         000001F7-R   578   (7)
T6_S1           00000027-R   628   (8)
T6_S1_X         0000008D-R   648   (8)
T6_S2           00000096-R   650   (8)
T6_S2_X         000000E0-R   665   (8)
T6_S3           000000E9-R   667   (8)
T6_S3_X         00000140-R   684   (8)
T6_S4           00000149-R   686   (8)
T6_S4_X         000001A8-R   705   (8)
T6_S5           000001B1-R   707   (8)
T6_S5_X         00000208-R   724   (8)
TEST_001        00000020-R   95    (3)     95     (3)
TEST_001_X      00000074-R   107   (3)
TEST_002        00000024-R   132   (4)     132    (4)
TEST_002_X      000000FE-R   152   (4)
TEST_003        0000001C-R   197   (5)     197    (5)
TEST_003_X      00000270-R   303   (5)
TEST_004        0000001C-R   348   (6)     348    (6)
TEST_004_X      00000207-R   441   (6)
TEST_005        0000001C-R   486   (7)     486    (7)
TEST_005_X      00000203-R   579   (7)
TEST_006        0000001C-R   626   (8)     626    (8)
TEST_006_X      00000214-R   725   (8)
TEST_007        00000040-R   758   (10)    758    (10)
TEST_007_X      000000E3-R   786   (10)   #-761   (10)
TEST_008        0000003C-R   822   (11)    822    (11)
TEST_008_X      00000154-R   865   (11)   #-825   (11)
```

```
TEST_009        00000038-R   902    (13)    902    (13)
TEST_009_X      000001B9-R   959    (13)   #-905   (13)   #-935    (13)
TEST_010        00000048-R   999    (15)    999    (15)
TEST_010_X      0000017B-R   1063   (15)   #-1002  (15)
TEST_011        00000040-R   1095   (17)    1095   (17)
TEST_011_X      000000C1-R   1116   (17)   #-1098  (17)   #-1102   (17)
TEST_012        0000003C-R   1152   (19)    1152   (19)
TEST_012_X      0000011A-R   1187   (19)   #-1155  (19)   #-1159   (19)
TEST_013        000000E0-R   1247   (21)    1247   (21)
TEST_013_X      00000281-R   1317   (21)   #-1250  (21)   #-1254   (21)
TEST_014        00000044-R   1351   (22)    1351   (22)
TEST_014_X      000001C2-R   1415   (22)   #-1354  (22)   #-1358   (22)
TEST_015        000000E4-R   1474   (24)    1474   (24)
TEST_015_X      00000393-R   1585   (24)   #-1477  (24)   #-1481   (24)
TEST_016        000000F0-R   1648   (26)    1648   (26)
TEST_016_X      00000471-R   1818   (26)   #-1651  (26)   #-1655   (26)
TEST_017        000000E8-R   1870   (28)    1870   (28)
TEST_017_X      0000021E-R   1941   (28)   #-1873  (28)   #-1877   (28)   #-1908   (28)
UBACLEAR_ERR    00000000-XR                 141    (4)
UBAINIT_ERR     00000000-XR                 136    (4)
WORK            00000000-XR                 1049   (15)
```

```
                                      +--------------------------+
                                      ! Macros Cross Reference !
                                      +--------------------------+

MACRO             SIZE       DEFINITION       REFERENCES...
------            ----       ----------       -------------
$D1_ABPROGRAM     1          105     (3)      105    (3)      138    (4)      143    (4)      151    (4)
$D1_BDATA         1          751     (10)     1088   (17)     1145   (19)     1232   (21)     1344   (22)     1459   (24)
                                              1633   (26)     1855   (28)     751    (10)     815    (11)     895    (13)
                                              992    (15)
$D1_BSUBT         1          199     (5)      1659   (26)     1703   (26)     1775   (26)     199    (5)      217    (5)
                                              235    (5)      259    (5)      283    (5)      350    (6)      367    (6)
                                              384    (6)      403    (6)      424    (6)      488    (7)      505    (7)
                                              522    (7)      541    (7)      562    (7)      628    (8)      650    (8)
                                              667    (8)      686    (8)      707    (8)
$D1_BTEST         1          95      (3)      1095   (17)     1152   (19)     1247   (21)     132    (4)      1351   (22)
                                              1474   (24)     1648   (26)     1870   (28)     197    (5)      348    (6)
                                              486    (7)      626    (8)      758    (10)     822    (11)     902    (13)
                                              95     (3)      999    (15)
$D1_ERR           1          103     (3)      1011   (15)     1022   (15)     103    (3)      1039   (15)     1060   (15)
                                              1107   (17)     1114   (17)     1166   (19)     1178   (19)     1185   (19)
                                              1263   (21)     1269   (21)     1289   (21)     1298   (21)     1304   (21)
                                              1315   (21)     136    (4)      1363   (22)     1369   (22)     1382   (22)
                                              1388   (22)     1401   (22)     1407   (22)     141    (4)      1413   (22)
                                              149    (4)      1491   (24)     1497   (24)     1530   (24)     1536   (24)
                                              1544   (24)     1551   (24)     1561   (24)     1567   (24)     1578   (24)
                                              1669   (26)     1675   (26)     1685   (26)     1696   (26)     1718   (26)
                                              1724   (26)     1734   (26)     1756   (26)     1762   (26)     1771   (26)
                                              1785   (26)     1791   (26)     1801   (26)     1812   (26)     1884   (28)
                                              1895   (28)     1904   (28)     1920   (28)     1932   (28)     213    (5)
                                              231    (5)      253    (5)      277    (5)      298    (5)      363    (6)
                                              380    (6)      397    (6)      418    (6)      438    (6)      501    (7)
                                              518    (7)      535    (7)      556    (7)      576    (7)      646    (8)
                                              663    (8)      680    (8)      701    (8)      722    (8)      776    (10)
                                              784    (10)     830    (11)     845    (11)     851    (11)     857    (11)
                                              863    (11)     915    (13)     933    (13)     942    (13)     950    (13)
                                              957    (13)
$D1_ERR_S         1          103     (3)      1011   (15)     1022   (15)     103    (3)      1039   (15)     1060   (15)
                                              1107   (17)     1114   (17)     1166   (19)     1178   (19)     1185   (19)
                                              1263   (21)     1269   (21)     1289   (21)     1298   (21)     1304   (21)
                                              1315   (21)     136    (4)      1363   (22)     1369   (22)     1382   (22)
                                              1388   (22)     1401   (22)     1407   (22)     141    (4)      1413   (22)
                                              149    (4)      1491   (24)     1497   (24)     1530   (24)     1536   (24)
                                              1544   (24)     1551   (24)     1561   (24)     1567   (24)     1578   (24)
                                              1669   (26)     1675   (26)     1685   (26)     1696   (26)     1718   (26)
                                              1724   (26)     1734   (26)     1756   (26)     1762   (26)     1771   (26)
                                              1785   (26)     1791   (26)     1801   (26)     1812   (26)     1884   (28)
                                              1895   (28)     1904   (28)     1920   (28)     1932   (28)     213    (5)
                                              231    (5)      253    (5)      277    (5)      298    (5)      363    (6)
                                              380    (6)      397    (6)      418    (6)      438    (6)      501    (7)
                                              518    (7)      535    (7)      556    (7)      576    (7)      646    (8)
                                              663    (8)      680    (8)      701    (8)      722    (8)      776    (10)
                                              784    (10)     830    (11)     845    (11)     851    (11)     857    (11)
                                              863    (11)     915    (13)     933    (13)     942    (13)     950    (13)
                                              957    (13)
```

```
$D1_ESUBT          1            215     (5)      1701    (26)     1773    (26)     1817    (26)     215     (5)      233     (5)
                                                 257     (5)      281     (5)      302     (5)      365     (6)      382     (6)
                                                 401     (6)      422     (6)      440     (6)      503     (7)      520     (7)
                                                 539     (7)      560     (7)      578     (7)      648     (8)      665     (8)
                                                 684     (8)      705     (8)      724     (8)
$D1_ETEST          1            107     (3)      1063    (15)     107     (3)      1116    (17)     1187    (19)     1317    (21)
                                                 1415    (22)     152     (4)      1585    (24)     1818    (26)     1941    (28)
                                                 303     (5)      441     (6)      579     (7)      725     (8)      786     (10)
                                                 865     (11)     959     (13)
$D1_EXSUB          1            1714    (26)     1714    (26)
$D1_EXTEST         1            761     (10)     1002    (15)     1098    (17)     1102    (17)     1155    (19)     1159    (19)
                                                 1250    (21)     1254    (21)     1354    (22)     1358    (22)     1477    (24)
                                                 1481    (24)     1651    (26)     1655    (26)     1873    (28)     1877    (28)
                                                 1908    (28)     761     (10)     825     (11)     905     (13)     935     (13)
$D1_SBTTL          2            75      (2)      1064    (15)     108     (3)      1117    (17)     1188    (19)     1318    (21)
                                                 1416    (22)     153     (4)      1586    (24)     1819    (26)     304     (5)
                                                 442     (6)      580     (7)      726     (8)      75      (2)      787     (10)
                                                 866     (11)     960     (13)
$D2_BDATA          1            95      (3)      1088    (17)     1145    (19)     1232    (21)     132     (4)      1344    (22)
                                                 1459    (24)     1633    (26)     1855    (28)     197     (5)      348     (6)
                                                 486     (7)      626     (8)      751     (10)     815     (11)     895     (13)
                                                 95      (3)      992     (15)
$D2_BTEST          1            95      (3)      1095    (17)     1152    (19)     1247    (21)     132     (4)      1351    (22)
                                                 1474    (24)     1648    (26)     1870    (28)     197     (5)      348     (6)
                                                 486     (7)      626     (8)      758     (10)     822     (11)     902     (13)
                                                 95      (3)      999     (15)
$D2_SBTTL          1            75      (2)      1064    (15)     108     (3)      1117    (17)     1188    (19)     1318    (21)
                                                 1416    (22)     153     (4)      1586    (24)     1819    (26)     304     (5)
                                                 442     (6)      580     (7)      726     (8)      75      (2)      787     (10)
                                                 866     (11)     960     (13)
$DEF               1            148     (4)
$DEFINI            1            71      (2)      71      (2)      72      (2)      73      (2)
$DS_ABORT          1            105     (3)      105     (3)      138     (4)      143     (4)      151     (4)
$DS_BGNDATA        1            751     (10)     1088    (17)     1145    (19)     1232    (21)     1344    (22)     1459    (24)
                                                 1633    (26)     1855    (28)     751     (10)     815     (11)     895     (13)
                                                 992     (15)
$DS_BGNMOD         1            65      (2)      65      (2)
$DS_BGNSUB         1            199     (5)      1659    (26)     1703    (26)     1775    (26)     199     (5)      217     (5)
                                                 235     (5)      259     (5)      283     (5)      350     (6)      367     (6)
                                                 384     (6)      403     (6)      424     (6)      488     (7)      505     (7)
                                                 522     (7)      541     (7)      562     (7)      628     (8)      650     (8)
                                                 667     (8)      686     (8)      707     (8)
$DS_BGNTEST        1            95      (3)      1095    (17)     1152    (19)     1247    (21)     132     (4)      1351    (22)
                                                 1474    (24)     1648    (26)     1870    (28)     197     (5)      348     (6)
                                                 486     (7)      626     (8)      758     (10)     822     (11)     902     (13)
                                                 95      (3)      999     (15)
$DS_BITDEF         2            66      (2)      66      (2)
$DS_BNERROR        1            102     (3)      102     (3)      135     (4)      140     (4)      148     (4)
$DS_BREAK          1            107     (3)      1063    (15)     107     (3)      1116    (17)     1187    (19)     1317    (21)
                                                 1415    (22)     152     (4)      1585    (24)     1818    (26)     1941    (28)
                                                 303     (5)      441     (6)      579     (7)      725     (8)      786     (10)
                                                 865     (11)     959     (13)
$DS_CHANNEL_S      1            134     (4)      134     (4)      139     (4)
$DS_CHCDEF         1            68      (2)      68      (2)
$DS_CHDEF          1            68      (2)      68      (2)
$DS_CHIDEF         1            68      (2)      68      (2)
```

$DS_CHMDEF	2	68	(2)	68	(2)								
$DS_CHSDEF	3	68	(2)	68	(2)								
$DS_CKLOOP	1	104	(3)	1012	(15)	1023	(15)	104	(3)	1040	(15)	1061	(15)
				1108	(17)	1115	(17)	1167	(19)	1179	(19)	1186	(19)
				1264	(21)	1270	(21)	1290	(21)	1299	(21)	1305	(21)
				1316	(21)	1364	(22)	137	(4)	1370	(22)	1383	(22)
				1389	(22)	1402	(22)	1408	(22)	1414	(22)	142	(4)
				1492	(24)	1498	(24)	150	(4)	1531	(24)	1537	(24)
				1545	(24)	1552	(24)	1562	(24)	1568	(24)	1579	(24)
				1670	(26)	1676	(26)	1686	(26)	1697	(26)	1719	(26)
				1725	(26)	1735	(26)	1757	(26)	1763	(26)	1772	(26)
				1786	(26)	1792	(26)	1802	(26)	1813	(26)	1885	(28)
				1896	(28)	1905	(28)	1921	(28)	1933	(28)	214	(5)
				232	(5)	254	(5)	278	(5)	301	(5)	364	(6)
				381	(6)	398	(6)	419	(6)	439	(6)	502	(7)
				519	(7)	536	(7)	557	(7)	577	(7)	647	(8)
				664	(8)	681	(8)	702	(8)	723	(8)	777	(10)
				785	(10)	831	(11)	846	(11)	852	(11)	858	(11)
				864	(11)	916	(13)	934	(13)	943	(13)	951	(13)
				958	(13)								
$DS_ENDDATA	1	95	(3)	1093	(17)	1150	(19)	1245	(21)	132	(4)	1349	(22)
				1472	(24)	1646	(26)	1868	(28)	197	(5)	348	(6)
				486	(7)	626	(8)	756	(10)	820	(11)	900	(13)
				95	(3)	997	(15)						
$DS_ENDMOD	1	1942	(28)	1942	(28)								
$DS_ENDSUB	1	215	(5)	1701	(26)	1773	(26)	1817	(26)	215	(5)	233	(5)
				257	(5)	281	(5)	302	(5)	365	(6)	382	(6)
				401	(6)	422	(6)	440	(6)	503	(7)	520	(7)
				539	(7)	560	(7)	578	(7)	648	(8)	665	(8)
				684	(8)	705	(8)	724	(8)				
$DS_ENDTEST	1	107	(3)	1063	(15)	107	(3)	1116	(17)	1187	(19)	1317	(21)
				1415	(22)	152	(4)	1585	(24)	1818	(26)	1941	(28)
				303	(5)	441	(6)	579	(7)	725	(8)	786	(10)
				865	(11)	959	(13)						
$DS_ENVDEF	2	65	(2)	65	(2)								
$DS_ERRHARD_S	1	211	(5)	1010	(15)	1021	(15)	1037	(15)	1058	(15)	1106	(17)
				1113	(17)	1165	(19)	1176	(19)	1184	(19)	1262	(21)
				1268	(21)	1288	(21)	1296	(21)	1303	(21)	1313	(21)
				1362	(22)	1368	(22)	1381	(22)	1387	(22)	1400	(22)
				1406	(22)	1411	(22)	1490	(24)	1496	(24)	1529	(24)
				1534	(24)	1542	(24)	1550	(24)	1559	(24)	1566	(24)
				1576	(24)	1668	(26)	1674	(26)	1684	(26)	1695	(26)
				1717	(26)	1723	(26)	1733	(26)	1755	(26)	1761	(26)
				1769	(26)	1784	(26)	1790	(26)	1800	(26)	1811	(26)
				1883	(28)	1894	(28)	1903	(28)	1919	(28)	1931	(28)
				211	(5)	229	(5)	251	(5)	275	(5)	296	(5)
				361	(6)	378	(6)	395	(6)	416	(6)	436	(6)
				499	(7)	516	(7)	533	(7)	554	(7)	574	(7)
				644	(8)	661	(8)	678	(8)	699	(8)	720	(8)
				775	(10)	782	(10)	829	(11)	844	(11)	849	(11)
				855	(11)	861	(11)	914	(13)	932	(13)	940	(13)
				948	(13)	955	(13)						
$DS_ERRSYS_S	1	103	(3)	103	(3)	136	(4)	141	(4)	149	(4)		
$DS_EXIT	1	761	(10)	1002	(15)	1098	(17)	1102	(17)	1155	(19)	1159	(19)
				1250	(21)	1254	(21)	1354	(22)	1358	(22)	1477	(24)
				1481	(24)	1651	(26)	1655	(26)	1714	(26)	1873	(28)

```
                                            1877    (28)      1908    (28)      761     (10)      825     (11)      905     (13)
                                            935     (13)
$DS_PROBE_S        1         99      (3)    145     (4)       99      (3)
$DS_QADEF          2         102     (3)    102     (3)       135     (4)       140     (4)       148     (4)
$DS_SBTTL          1         75      (2)    1064    (15)      108     (3)       1117    (17)      1188    (19)      1318    (21)
                                            1416    (22)      153     (4)       1586    (24)      1819    (26)      304     (5)
                                            442     (6)       580     (7)       726     (8)       75      (2)       787     (10)
                                            866     (11)      960     (13)
$DS_SECDEF         1         67      (2)    67      (2)
$ENRKA_SYMDEF      7         73      (2)    73      (2)
$EQU               1         148     (4)    102     (3)       135     (4)       140     (4)       148     (4)       65      (2)
                                            66      (2)       68      (2)       70      (2)
$EQULS1            1         148     (4)    102     (3)       135     (4)       140     (4)       148     (4)       65      (2)
                                            68      (2)       70      (2)
$EQULST            1         65      (2)    102     (3)       135     (4)       140     (4)       148     (4)       65      (2)
                                            68      (2)       70      (2)
$GBLINI            2         65      (2)    102     (3)       135     (4)       140     (4)       148     (4)       65      (2)
                                            66      (2)       68      (2)       70      (2)
$HDR_LAYOUTDEF     2         72      (2)    72      (2)
$IDC_REGDEF        10        71      (2)    71      (2)
$PUSHADR           1         103     (3)    1011    (15)      1022    (15)      103     (3)       1039    (15)      1060    (15)
                                            1107    (17)      1114    (17)      1166    (19)      1178    (19)      1185    (19)
                                            1263    (21)      1269    (21)      1289    (21)      1298    (21)      1304    (21)
                                            1315    (21)      134     (4)       136     (4)       1363    (22)      1369    (22)
                                            1382    (22)      1388    (22)      139     (4)       1401    (22)      1407    (22)
                                            141     (4)       1413    (22)      149     (4)       1491    (24)      1497    (24)
                                            1530    (24)      1536    (24)      1544    (24)      1551    (24)      1561    (24)
                                            1567    (24)      1578    (24)      1669    (26)      1675    (26)      1685    (26)
                                            1696    (26)      1718    (26)      1724    (26)      1734    (26)      1756    (26)
                                            1762    (26)      1771    (26)      1785    (26)      1791    (26)      1801    (26)
                                            1812    (26)      1884    (28)      1895    (28)      1904    (28)      1920    (28)
                                            1932    (28)      213     (5)       231     (5)       253     (5)       277     (5)
                                            298     (5)       363     (6)       380     (6)       397     (6)       418     (6)
                                            438     (6)       501     (7)       518     (7)       535     (7)       556     (7)
                                            576     (7)       646     (8)       663     (8)       680     (8)       701     (8)
                                            722     (8)       776     (10)      784     (10)      830     (11)      845     (11)
                                            851     (11)      857     (11)      863     (11)      915     (13)      933     (13)
                                            942     (13)      950     (13)      957     (13)
$VIELD             1         65      (2)    65      (2)       68      (2)
$VIELD1            1         148     (4)    65      (2)       68      (2)
BDEVFOUND          1         761     (10)   1002    (15)      1098    (17)      1155    (19)      1250    (21)      1354    (22)
                                            1477    (24)      1651    (26)      1873    (28)      761     (10)      825     (11)
                                            905     (13)
DRV_TBL            3         70      (2)    70      (2)
ENRKADEF           1         69      (2)    69      (2)
GET_LUN            1         760     (10)   1001    (15)      1097    (17)      1154    (19)      1249    (21)      1353    (22)
                                            1476    (24)      1650    (26)      1872    (28)      760     (10)      824     (11)
                                            904     (13)
STARS              1         200     (5)    1660    (26)      1704    (26)      1776    (26)      200     (5)       218     (5)
                                            236     (5)       260     (5)       284     (5)       351     (6)       368     (6)
                                            385     (6)       404     (6)       425     (6)       489     (7)       506     (7)
                                            523     (7)       542     (7)       563     (7)       629     (8)       651     (8)
                                            668     (8)       687     (8)       708     (8)
```

```
                              +--------------------------+
                              ! Performance indicators !
                              +--------------------------+

Phase                      Page faults   CPU Time      Elapsed Time
-----                      -----------   --------      ------------
Initialization                     32    00:00:00.13   00:00:00.57
Command processing                879    00:00:00.92   00:00:01.84
Pass 1                           2922    00:00:51.66   00:01:27.81
Symbol table sort                   8    00:00:01.17   00:00:01.17
Pass 2                            675    00:00:12.23   00:00:36.75
Symbol table output                20    00:00:00.35   00:00:03.05
Psect synopsis output               3    00:00:00.10   00:00:00.30
Cross-reference output            315    00:00:07.09   00:00:21.19
Assembler run totals             4858    00:01:13.68   00:02:32.85

The working set limit was 1024 pages.
455348 bytes (890 pages) of virtual memory were used to buffer the intermediate code.
There were 40 pages of symbol table space allocated to hold 701 non-local and 185 local symbols.
1943 source lines were read in Pass 1, producing 94 object records in Pass 2.
122 pages of virtual memory were used to define 49 macros.

                             +----------------------------+
                             ! Macro library statistics !
                             +----------------------------+

Macro library name                           Macros defined
------------------                           --------------
$1$DUA9:[BOWSER.TAPE]ENRKA.MLB;16                         8
SYS$COMMON:[SYSLIB]LIB.MLB;3                              0
$1$DUA9:[DIAGLIB.LIBRARY.BLISS]DIAG.MLB;1                41
SYS$COMMON:[SYSLIB]STARLET.MLB;3                          6
TOTALS (all libraries)                                   55

982 GETS were required to define 55 macros.

There were no errors, warnings or information messages.

MACRO/LIST/CROSS ENRKA1
```

Table of contents

```
    0000
    0000            .TITLE  ENRKA2  IDC DISK FUNCTIONAL DIAGNOSTIC - SECTION 2
    0000            .IDENT  /1.1/
    0000            .LIST   MEB
    0000            .NLIST  CND
00000000            .PSECT  HEADER, PAGE, NOWRT
    0000
    0000    ;++
    0000    ; COPYRIGHT (C) 1982, 1986
    0000    ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS 01754
    0000    ;
    0000    ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON  A
    0000    ; SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLU-
    0000    ; SION OF THE ABOVE COPYRIGHT NOTICE.  THIS  SOFTWARE,  OR  ANY
    0000    ; OTHER  COPIES  THEREOF, MAY NOT BE PROVIDED OR OTHERWISE MADE
    0000    ; AVAILABLE TO ANY OTHER PERSON EXCEPT FOR USE ON  SUCH  SYSTEM
    0000    ; AND  TO  ONE WHO AGREES TO THESE LICENSE TERMS.  TITLE TO AND
    0000    ; OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN DEC.
    0000    ;
    0000    ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT
    0000    ; NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL
    0000    ; EQUIPMENT CORPORATION.
    0000    ;
    0000    ; DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF
    0000    ; ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
    0000    ;
    0000    ;++
    0000    ; FACILITY: VAX-11 DIAGNOSTIC LIBRARY
    0000    ;
    0000    ; ABSTRACT: THIS SECTION CONTAINS IDC SUBSYSTEM TESTS 16 THRU 29
    0000    ;
    0000    ; ENVIRONMENT: VAX-11 DIAGNOSTIC SUPERVISOR
    0000    ;
    0000    ;--
    0000    ;++
    0000    ; AUTHOR: JOHN L. WALSH
    0000    ;
    0000    ; CREATION DATE:  MARCH 1982
    0000    ;
    0000    ; MODIFIED BY:
    0000    ;       V1.1 NO CHANGES TO THIS MODULE
    0000    ;
    0000    ;--
```

```
        0000              .SBTTL  DECLARATIONS
        0000      ;++
        0000      ; INCLUDE FILES
        0000      ;--
        0000              .LIBRARY /DISK$DIAG:[DIAGLIB.LIBRARY.BLISS]DIAG.MLB/
        0000              .LIBRARY /SYS$LIBRARY:LIB.MLB/
        0000              .LIBRARY /DISK$DIAG:[BOWSER.TAPE]ENRKA.MLB/
        0000
        0000      ;++
        0000      ; EQUATED SYMBOLS
        0000      ;--
        0000
        0000      ;
        0000      ; CREATE SYMBOLS FOR THE PROGRAM/SUPERVISOR INTERFACE
        0000      ;
        0000
        0000              $DS_BGNMOD      <SEP_REPAIR>,18
        0000      ;;;     Macro-32 Diagnostic macro library Version 8.2 "DIAG.MLB (233)"
        0000              $DS_BITDEF
        0000              $DS_SECDEF      MANUAL,ALL
        0000              ENRKADEF
        0000      ;;;     ENRKA MACRO LIBRARY VERSION     V1.1
        0000              $DS_CHDEF
        0000              DRV_TBL
        0000              $IDC_REGDEF
        0000              .SAVE   LOCAL_BLOCK
    00000000              .RESTORE
        0000              $HDR_LAYOUTDEF
        0000              .SAVE   LOCAL_BLOCK
    00000000              .RESTORE
        0000              $ENRKA_SYMDEF
        0000              .SAVE   LOCAL_BLOCK
    00000000              .RESTORE
        0000              .NLIST  MEB
        0000              $DS_SBTTL <CONVERGING - DIVERGING SEEK TEST>,ALIGN=LONG
        0000              .SBTTL  TEST 18:  CONVERGING - DIVERGING SEEK TEST
        0000              $D2_SBTTL       018,<CONVERGING - DIVERGING SEEK TEST>,<LONG>
    00000000                      .PSECT  TEST_018, LONG, NOWRT
        0024
```

```
                                                  0024
                                                  0024  ;++
                                                  0024  ;
                                                  0024  ; TEST DESCRIPTION:
                                                  0024  ;
                                                  0024  ; THIS TEST WILL EXERCISE THE IDC DISK DRIVE WITH CONVERGING SEEKS
                                                  0024  ; AND THEN WITH DIVERGING SEEKS. THIS IS A MEANS OF CHECKING ALL
                                                  0024  ; POSSIBLE DIFFERENCE WORD COMBINATIONS.
                                                  0024  ;
                                                  0024  ;
                                                  0024  ; TEST STEPS:
                                                  0024  ;
                                                  0024  ;        1.  INITIALIZE SUBSYSTEM
                                                  0024  ;        2.  GET CURRENT DRIVE POSITION
                                                  0024  ;        3.  IF ALREADY AT CYLINDER 0 GO TO STEP 6.
                                                  0024  ;        4.  SEEK HEADS BACK TO CYLINDER 0
                                                  0024  ;        5.  VERIFY SEEK IF NOT AT CYLINDER 0 EXIT TEST
                                                  0024  ;        6.  START CONVERGING SEEKS
                                                  0024  ;        7.  WHEN UPPER LIMIT AND LOWER CROSS OVER -
                                                  0024  ;            THEN START DIVERGING SEEKS.
                                                  0024  ;        8.  CONTINUE UNTIL CURRENT CYLN RETURN TO ZERO
                                                  0024  ;
                                                  0024  ; DEBUG:
                                                  0024  ;
                                                  0024  ;        DIFFERNECE LOGIC
                                                  0024  ;        COUNT LOGIC
                                                  0024  ;        VELOCITY ROM
                                                  0024  ;
                                                  0024  ;--
                                                  0024
                                                  0024           $DS_BGNDATA
00000000 00000001 00000000 00000003  0024           .LONG    3,LUN0,RL02$T,HEAD_0
00000001 00000001 00000000 00000003  0034           .LONG    3,LUN0,RL02$T,HEAD_1
00000000 00000002 00000000 00000003  0044           .LONG    3,LUN0,R80$T,HEAD_0
00000000 00000001 00000001 00000003  0054           .LONG    3,LUN1,RL02$T,HEAD_0
00000001 00000001 00000001 00000003  0064           .LONG    3,LUN1,RL02$T,HEAD_1
00000000 00000002 00000001 00000003  0074           .LONG    3,LUN1,R80$T,HEAD_0
00000000 00000001 00000002 00000003  0084           .LONG    3,LUN2,RL02$T,HEAD_0
00000001 00000001 00000002 00000003  0094           .LONG    3,LUN2,RL02$T,HEAD_1
00000000 00000002 00000002 00000003  00A4           .LONG    3,LUN2,R80$T,HEAD_0
00000000 00000001 00000003 00000003  00B4           .LONG    3,LUN3,RL02$T,HEAD_0
00000001 00000001 00000003 00000003  00C4           .LONG    3,LUN3,RL02$T,HEAD_1
00000000 00000002 00000003 00000003  00D4           .LONG    3,LUN3,R80$T,HEAD_0
                                     00E4           $DS_ENDDATA
                                     00E8
                                     00E8           $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
                                     00EA           GET_LUN -                                  ; GET LUN INFO
                                     00EA             UNIT= OFST_UNIT(AP)                      ; LOGICAL UNIT
                   04 AC       DD    00EA           PUSHL    OFST_UNIT(AP)
            00000000'EF   01   FB    00ED           CALLS    #1,GET_LUN$A
                                     00F4           BDEVFOUND        1$                        ; IF DEV AVAILABLE GOTO 1$
                06 50     00   E2    00F4           BBSS     #0,R0,1$
                   50     01   D0    00F8                    MOVL     #1,R0          ; NOT AN ERROR
                 0143     31         00FB                    BRW      TEST_018_X     ; EXIT TEST 18
      54    00000000'EF        D0    00FE     1$:   MOVL     DRIVE_TBL,R4              ; GET DRIVE INFO TABLE
                08 AC     64   91    0105           CMPB     OFST_DRVTYP(R4),OFST_DRIV(AP)  ; IF DRIVE TYPE = DATA TYPE
```

```
                    06   13  0109              BEQL    5$                          ; THEN GOTO 5$
                             010B              $DS_EXIT        TEST                ; ELSE EXIT TEST
           00000000'EF   D4  0111      5$:     CLRL    CONFIG                      ; CLEAR ALL PROGRAM FLAGS
                             0117              STARS
                             0117      ;;*****************************************************************
                             0117      ;++
                             0117      ; POSITION HEADS TO CYLINDER ZERO AND VERIFY THE SEEK THEN
                             0117      ; DO CONVERGING SEEKS UNTIL UPPER LIMIT = LOWER LIMIT.
                             0117      ; MID WAY THRU THE TEST UPPER LIMIT AND LOWER LIMIT
                             0117      ; CROSS OVER EACH OTHER AND SEEKS BEGIN DIVERGING.
                             0117      ;--
                             0117
        57    08 A4    01    C1  0117          ADDL3   #1,OFST_LSTCYL(R4),R7       ; CALC LAST CYLN + 1
                 59    01    CE  011C          MNEGL   #1,R9                       ; R9 <-- LOWER CYLINDER LIMIT
          00000000'EF  00    FB  011F  10$:    CALLS   #0,SUBSYS_INIT              ; INITIALIZE SUBSYSTEM
                    1D 50    E8  0126          BLBS    R0,20$                      ; BRANCH IF NO ERRORS
                             0129              $DS_ERRHARD_S   ,L$UNIT,MSG14_SYSINIT,-
                             0129                              PRINT_REG,R1
                             013F      ;;; TEST 18, SUBTEST 0, ERROR 1
                             0146      20$:    $DS_CKLOOP      10$
                       00    DD  014E  25$:    PUSHL   #0                          ; DEFINE FLAGS
          00000000'EF  01    FB  0150          CALLS   #1,GET_POSITION             ; GET CURRENT DRIVE POSITION
                    1D 50    E8  0157          BLBS    R0,30$                      ; BRANCH IF NO ERRORS
                             015A              $DS_ERRHARD_S   ,L$UNIT,MSG19_READH,-
                             015A                              PRINT_REG,R1
                             0170      ;;; TEST 18, SUBTEST 0, ERROR 2
                             0177      30$:    $DS_CKLOOP      25$
           00000000'EF   D5  017F      35$:    TSTL    CURR_CYLN                   ; ARE WE AT CYLN 0
                    42   13  0185              BEQL    60$                         ; BRANCH IF WE ARE
                 0C AC   DD  0187      40$:    PUSHL   OFST_SKHD(AP)               ; PUSH DESIRED HEAD
                    00   DD  018A              PUSHL   #0                          ; PUSH DESTINATION CYLN
          00000000'EF  02    FB  018C          CALLS   #2,POSITION_HDS             ; EXECUTE THE SEEK CMD
                    1D 50    E8  0193          BLBS    R0,50$                      ; BRANCH IF NO ERRORS
                             0196              $DS_ERRHARD_S   ,L$UNIT,@ERR_MSG,-
                             0196                              PRINT_REG,R1
                             01AC      ;;; TEST 18, SUBTEST 0, ERROR 3
                             01B3      50$:    $DS_CKLOOP      40$
           00000000'EF   D5  01BB              TSTL    CURR_CYLN                   ; ARE WE AT CYLN 0
                    06   13  01C1              BEQL    60$                         ; BRANCH IF WE ARE
                             01C3              $DS_EXIT        TEST
                 57    01    C2  01C9  60$:    SUBL2   #1,R7                       ; CALCULATE DESTINATION
                 0C AC   DD  01CC      65$:    PUSHL   OFST_SKHD(AP)               ; PUSH DESIRED HEAD
                    57   DD  01CF              PUSHL   R7                          ; PUSH DESTINATION CYLN
          00000000'EF  02    FB  01D1          CALLS   #2,POSITION_HDS             ; EXECUTE THE SEEK CMD
                    1D 50    E8  01D8          BLBS    R0,70$                      ; BRANCH IF NO ERRORS
                             01DB              $DS_ERRHARD_S   ,L$UNIT,@ERR_MSG,-
                             01DB                              PRINT_REG,R1
                             01F1      ;;; TEST 18, SUBTEST 0, ERROR 4
                             01F8      70$:    $DS_CKLOOP      65$
                 59    01    C0  0200          ADDL2   #1,R9                       ; CALCULATE NEXT SEEK
                 0C AC   DD  0203      75$:    PUSHL   OFST_SKHD(AP)               ; PUSH DESIRED HEAD
                    59   DD  0206              PUSHL   R9                          ; SET DESTINATION TO CYLN 0
          00000000'EF  02    FB  0208          CALLS   #2,POSITION_HDS             ; EXECUTE THE SEEK CMD
                    1D 50    E8  020F          BLBS    R0,80$                      ; BRANCH IF NO ERRORS
                             0212              $DS_ERRHARD_S   ,L$UNIT,@ERR_MSG,-
                             0212                              PRINT_REG,R1
```

```
                      0228          ;;; TEST 18, SUBTEST 0, ERROR 5
                      022F     175 80$:    $DS_CKLOOP      75$
           57   D5    0237     176         TSTL    R7                          ; SEEKS ALL DONE ?
           03   13    0239     177         BEQL    90$                         ; BRANCH IF YES
         FF8B   31    023B     178         BRW     60$                         ; GO DO NEXT FWD AND REV SEEK
                      023E     179 90$:
                      023E     180         $DS_ENDTEST
                      0249     181         $DS_SBTTL <RANDOM RECAL TEST - R80>,ALIGN=LONG
                      0249                 .SBTTL  TEST 19:  RANDOM RECAL TEST - R80
                      0249                 $D2_SBTTL       019,<RANDOM RECAL TEST - R80>,<LONG>
                  00000000                         .PSECT  TEST_019, LONG, NOWRT
                      001C
```

```
                              001C     ;++
                              001C     ;
                              001C     ; TEST DESCRIPTION:
                              001C     ;
                              001C     ; THIS TEST IS DESIGNED TO EXERCISE THE RECAL FUNCTION FOR THE R80
                              001C     ; DISK DRIVE BY SEEKING TO RANDOM CYLINDERS AND EXECUTING A RECAL
                              001C     ; FUNCTION (SEEK TO 0) FROM THAT CYLINDER.
                              001C     ;
                              001C     ; TEST STEPS:
                              001C     ;
                              001C     ;           1.  INITIALIZE SUBSYSTEM
                              001C     ;           2.  SEEK HEADS TO RANDOM CYLINDER
                              001C     ;           3.  REPORT ANY ERRORS
                              001C     ;           4.  EXECUTE RECAL FUNCTION (SEEK TO ZERO)
                              001C     ;           5.  REPORT ANY ERRORS
                              001C     ;           6.  CONTINUE UNTIL 64 RECALS HAVE BEEN COMPLETED
                              001C     ;
                              001C     ; DEBUG:
                              001C     ;
                              001C     ;           DIFFERNECE LOGIC
                              001C     ;           GUARD BAND DETECTION LOGIC
                              001C     ;           VELOCITY ROM
                              001C     ;
                              001C     ;--
                              001C
                              001C
                              001C                 $DS_BGNDATA
       00000000 00000001      001C                 .LONG     1,LUN0
       00000001 00000001      0024                 .LONG     1,LUN1
       00000002 00000001      002C                 .LONG     1,LUN2
       00000003 00000001      0034                 .LONG     1,LUN3
                              003C                 $DS_ENDDATA
                              0040
                              0040                 $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
                              0042                 GET_LUN -                                   ; GET LUN INFO
                              0042                   UNIT=         OFST_UNIT(AP)               ; LOGICAL UNIT
                04 AC    DD   0042                 PUSHL     OFST_UNIT(AP)
     00000000'EF    01    FB  0045                 CALLS     #1,GET_LUN$A
                              004C                 BDEVFOUND           5$                      ; IF DEV AVAILABLE GOTO 5$
          06 50    00    E2   004C                 BBSS      #0,R0,5$
                50    01    D0  0050                         MOVL      #1,R0          ; NOT AN ERROR
              00DF    31      0053                         BRW       TEST_019_X     ; EXIT TEST 19
54     00000000'EF    D0      0056       5$:       MOVL      DRIVE_TBL,R4              ; GET DRIVE INFO TABLE
             02     64    91  005D                 CMPB      OFST_DRVTYP(R4),#R80$T    ; IF DRIVE TYPE = R80
                    06    13  0060                 BEQL      6$                        ; THEN GOTO 6$
                              0062                 $DS_EXIT            TEST            ; ELSE EXIT
       00000000'EF    D4      0068       6$:       CLRL      CONFIG                    ; CLEAR ALL PROGRAM FLAGS
                    53    D4  006E                 CLRL      R3                        ; RESET RECAL COUNTER
                    56    7C  0070                 CLRQ      R6                        ; CLEAR WORK REGISTERS
   58    08 A4    01    C1    0072                 ADDL3     #1,OFST_LSTCYL(R4),R8     ; CLAC NU8MBER OF CYLINDERS
     00000000'EF    00    FB  0077       10$:      CALLS     #0,SUBSYS_INIT            ; INITIALIZE SUBSYSTEM
             1D 50    E8      007E                 BLBS      R0,20$                    ; BRANCH IF NO ERRORS
                              0081                 $DS_ERRHARD_S       ,L$UNIT,MSG14_SYSINIT,-
                              0081                                     PRINT_REG,R1
                              0097             ;;; TEST 19, SUBTEST 0, ERROR 1
                              009E       20$:      $DS_CKLOOP          10$
```

```
          00000000'EF   16  00A6  25$:    JSB     GET_RANDOM                  ; GET A RANDOM NUMBER
                55  51  D0  00AC          MOVL    R1,R5                       ; AND SAVE IT
                    56  D4  00AF          CLRL    R6                          ; CLEAR UPPER HALF OF DIVIDEND
      57   56   55  58  7B  00B1          EDIV    R8,R5,R6,R7                 ; MAKE SURE CYLINDER IS IN RANGE
                            00B6          STARS
                            00B6  ;;**************************************************************
                            00B6  ;++
                            00B6  ; SEEK DRIVE TO A RANDOM CYLINDER
                            00B6  ;--
                            00B6
                    00  DD  00B6  30$:    PUSHL   #0                          ; PUSH DESIRED HEAD
                    57  DD  00B8          PUSHL   R7                          ; PUSH DESTINATION CYLN
        00000000'EF 02  FB  00BA          CALLS   #2,POSITION_HDS             ; EXECUTE THE SEEK CMD
                 1D 50  E8  00C1          BLBS    R0,40$                      ; BRANCH IF NO ERRORS
                            00C4          $DS_ERRHARD_S    ,L$UNIT,@ERR_MSG,-
                            00C4                           PRINT_REG,R1
                            00DA  ;;; TEST 19, SUBTEST 0, ERROR 2
                            00E1  40$:    $DS_CKLOOP       30$
          00000000'EF   D4  00E9          CLRL    CONFIG                      ; CLEAR ANY PROGRAM FLAGS
                            00EF          STARS
                            00EF  ;;**************************************************************
                            00EF  ;++
                            00EF  ; EXECUTE RECAL FROM RANDOM CYLINDER
                            00EF  ;--
                            00EF
                    00  DD  00EF  50$:    PUSHL   #0                          ; PUSH DESIRED HEAD
          FFFFFFFF 8F   DD  00F1          PUSHL   #-1                         ; PUSH DESTINATION CYLN
        00000000'EF 02  FB  00F7          CALLS   #2,POSITION_HDS             ; EXECUTE THE SEEK CMD
                 1D 50  E8  00FE          BLBS    R0,60$                      ; BRANCH IF NO ERRORS
                            0101          $DS_ERRHARD_S    ,L$UNIT,@ERR_MSG,-
                            0101                           PRINT_REG,R1
                            0117  ;;; TEST 19, SUBTEST 0, ERROR 3
                            011E  60$:    $DS_CKLOOP       50$
          00000000'EF   D4  0126          CLRL    CONFIG                      ; CLEAR ANY PROGRAM FLAGS
      FF74 53   01  3F  3D  012C          ACBW    #63,#1,R3,25$               ; GET NEXT RANDOM CYLINDER
                            0132          $DS_ENDTEST
                            013D          $DS_SBTTL <RANDOM SEEK TEST>,ALIGN=LONG
                            013D          .SBTTL  TEST 20:  RANDOM SEEK TEST
                            013D          $D2_SBTTL        020,<RANDOM SEEK TEST>,<LONG>
                        00000000                  .PSECT  TEST_020, LONG, NOWRT
                            0014
```

```
                                  0014   267 ;++
                                  0014   268 ;
                                  0014   269 ; TEST DESCRIPTION:
                                  0014   270 ;
                                  0014   271 ; THIS TEST EXERCISES THE POSITIONING LOGIC IN THE DRIVE
                                  0014   272 ; UNDER TEST WITH RANDOM SEEKS. 1024 SEEKS ARE PERFORMED
                                  0014   273 ; DURING THIS TEST. IF THE TEST DRIVE IS AN RL02 THEN
                                  0014   274 ; RANDOM TRACKS ARE ALSO USED.
                                  0014   275 ;
                                  0014   276 ; TEST STEPS:
                                  0014   277 ;
                                  0014   278 ;          1.  INITIALIZE SUBSYSTEM
                                  0014   279 ;          2.  SEEK HEADS TO RANDOM CYLINDER
                                  0014   280 ;          3.  REPORT ANY ERRORS
                                  0014   281 ;          4.  CONTINUE UNTIL 1024 SEEKS HAVE BEEN COMPLETED
                                  0014   282 ;
                                  0014   283 ; DEBUG:
                                  0014   284 ;
                                  0014   285 ;          DIFFERNECE LOGIC
                                  0014   286 ;          TRACK COUNTING LOGIC
                                  0014   287 ;          VELOCITY ROM
                                  0014   288 ;
                                  0014   289 ;--
                                  0014   290
                                  0014   291
                                  0014   292           $DS_BGNDATA
          00000000 00000001       0014   293           .LONG   1,LUN0
          00000001 00000001       001C   294           .LONG   1,LUN1
          00000002 00000001       0024   295           .LONG   1,LUN2
          00000003 00000001       002C   296           .LONG   1,LUN3
                                  0034   297           $DS_ENDDATA
                                  0038   298
                                  0038   299           $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
                                  003A   300           GET_LUN -                                  ; GET LUN INFO
                                  003A   301             UNIT=           OFST_UNIT(AP)            ; LOGICAL UNIT
                04 AC       DD    003A                 PUSHL   OFST_UNIT(AP)
       00000000'EF     01   FB    003D                 CALLS   #1,GET_LUN$A
                                  0044   302           BDEVFOUND       5$                         ; IF DEV AVAILABLE GOTO 5$
           06 50    00      E2    0044                 BBSS    #0,R0,5$
                50    01    D0    0048                         MOVL    #1,R0          ; NOT AN ERROR
                  00AA      31    004B                         BRW     TEST_020_X     ; EXIT TEST 20
  54     00000000'EF        D0    004E   303 5$:       MOVL    DRIVE_TBL,R4                       ; GET DRIVE INFO TABLE
         00000000'EF        D4    0055   304           CLRL    CONFIG                             ; CLEAR ALL PROGRAM FLAGS
                      53    D4    005B   305           CLRL    R3                                 ; RESET SEEK COUNTER
                      56    7C    005D   306           CLRQ    R6                                 ; CLEAR WORK REGISTERS
                      5A    D4    005F   307           CLRL    R10                                ; RESET DESIRED TRACK TO 0
  58     08 A4        01    C1    0061   308           ADDL3   #1,OFST_LSTCYL(R4),R8              ; CALC NU8MBER OF CYLINDERS
  59     0C A4        01    C1    0066   309           ADDL3   #1,OFST_LSTTRK(R4),R9              ; CALC NUMBER OF TRACKS
       00000000'EF    00    FB    006B   310 10$:      CALLS   #0,SUBSYS_INIT                     ; INITIALIZE SUBSYSTEM
              1D 50         E8    0072   311           BLBS    R0,20$                             ; BRANCH IF NO ERRORS
                                  0075   312           $DS_ERRHARD_S     ,L$UNIT,MSG14_SYSINIT,-
                                  0075   313                             PRINT_REG,R1
                                  008B       ;;; TEST 20, SUBTEST 0, ERROR 1
                                  0092   314 20$:      $DS_CKLOOP      10$
         00000000'EF  16          009A   315 25$:      JSB     GET_RANDOM                         ; GET A RANDOM NUMBER
                55    51    D0    00A0   316           MOVL    R1,R5                              ; AND SAVE IT
```

```
                       56   D4  00A3   317          CLRL    R6                       ; CLEAR UPPER HALF OF DIVIDEND
         57    56   55   58   7B  00A5   318          EDIV    R8,R5,R6,R7              ; MAKE SURE CYLINDER IS IN RANGE
                    02   64   91  00AA   319          CMPB    OFST_DRVTYP(R4),#R80$T   ; IF DRIVE IS AN R80
                    05   13   00AD       320          BEQL    30$                      ; THEN GOTO 30$
         5A    57   01   00   EF  00AF   321          EXTZV   #0,#1,R7,R10             ; ELSE GET RANDOM TRACK FOR RL
                              00B4   322
                              00B4   323          STARS
                              00B4       ;;*********************************************************************
                              00B4   324 ;++
                              00B4   325 ; SEEK DRIVE TO A RANDOM CYLINDER
                              00B4   326 ;--
                              00B4   327
                       5A   DD  00B4   328 30$:   PUSHL   R10                      ; PUSH DESIRED HEAD
                       57   DD  00B6   329          PUSHL   R7                       ; PUSH DESTINATION CYLN
          00000000'EF  02   FB  00B8   330          CALLS   #2,POSITION_HDS          ; EXECUTE THE SEEK CMD
                    1D 50   E8  00BF   331          BLBS    R0,40$                   ; BRANCH IF NO ERRORS
                              00C2   332          $DS_ERRHARD_S    ,L$UNIT,@ERR_MSG,-
                              00C2   333                           PRINT_REG,R1
                              00D8       ;;; TEST 20, SUBTEST 0, ERROR 2
                              00DF   334 40$:   $DS_CKLOOP       30$
           00000000'EF      D4  00E7   335          CLRL    CONFIG                   ; CLEAR ANY PROGRAM FLAGS
     A5 53    00000400 8F   F2  00ED   336          AOBLSS  #1024,R3,25$             ; GET NEXT RANDOM CYLINDER
                              00F5   337          $DS_ENDTEST
                              0100   338          $DS_SBTTL <HEADER NOT FOUND TEST>,ALIGN=LONG
                              0100              .SBTTL  TEST 21:  HEADER NOT FOUND TEST
                              0100              $D2_SBTTL        021,<HEADER NOT FOUND TEST>,<LONG>
                          00000000                      .PSECT  TEST_021, LONG, NOWRT
                              0018
```

```
                             0018
                             0018      ;++
                             0018      ;
                             0018      ; TEST DESCRIPTION:
                             0018      ;
                             0018      ; THIS TEST WILL CHECK THAT A HEADER NOT FOUND ERROR CAN BE DETECTED
                             0018      ; USING A READ DATA COMMAND WITH THE IDCDAR REGISTER SPECIFING AN
                             0018      ; ILLEGAL ADDRESS.
                             0018      ;
                             0018      ; TEST STEPS:
                             0018      ;
                             0018      ; A.  SUBTEST 1
                             0018      ;
                             0018      ;          1.  INITIALIZE SUBSYSTEM
                             0018      ;          2.  GET CURRENT POSITION OF DRIVE
                             0018      ;          3.  IF AT CYLN ZERO GO TO SUBTEST 2
                             0018      ;          4.  EXECUTE SEEK BACK TO ZERO
                             0018      ;          5.  IF NOT AT CYLN ZERO NOW EXIT TEST
                             0018      ;
                             0018      ; B.  SUBTEST 2
                             0018      ;
                             0018      ;          1.  SET UP IDCDAR REGISTER FOR SECTO 40
                             0018      ;          2.  EXECUTE READ DATA COMMAND
                             0018      ;          3.  CHECK FOR ANY UNEXPECTED ERRORS
                             0018      ;          4.  CHECK THAT HEADER NOT FOUND IS SET IN IDCCSR
                             0018      ;
                             0018      ; DEBUG:
                             0018      ;
                             0018      ;          HEADER COMPARASIN LOGIC
                             0018      ;
                             0018      ;--
                             0018
                             0018
                             0018              $DS_BGNDATA
   00000000 00000001         0018              .LONG   1,LUN0
   00000001 00000001         0020              .LONG   1,LUN1
   00000002 00000001         0028              .LONG   1,LUN2
   00000003 00000001         0030              .LONG   1,LUN3
                             0038              $DS_ENDDATA
                             003C
                             003C              $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
                             003E              GET_LUN -                               ; GET LUN INFO
                             003E                UNIT=        OFST_UNIT(AP)            ; LOGICAL UNIT
         04 AC     DD        003E              PUSHL   OFST_UNIT(AP)
00000000'EF  01    FB        0041              CALLS   #1,GET_LUN$A
                             0048              BDEVFOUND       5$                      ; ID DEV AVAILABLE GOTO 5$
      06 50  00    E2        0048              BBSS    #0,R0,5$
         50  01    D0        004C                      MOVL    #1,R0         ; NOT AN ERROR
           029D    31        004F                      BRW     TEST_021_X    ; EXIT TEST 21
    00000000'EF    D4        0052      5$:     CLRL    CONFIG                          ; CLEAR ANY PROGRAM FLAGS
                             0058              $DS_BGNSUB
                             0063              STARS
                             0063      ;;*********************************************************************
                             0063      ;++
                             0063      ; POSITION DRIVE TO CYLINDER 0 AND VERIFY THE SEEK
                             0063      ;--
```

```
                                  0063
        00000000'EF   00    FB    0063   10$:    CALLS   #0,SUBSYS_INIT                 ; INITIALIZE SUBSYSTEM
                 1D 50      E8    006A           BLBS    R0,20$                         ; BRANCH IF NO ERRORS
                                  006D           $DS_ERRHARD_S    ,L$UNIT,MSG14_SYSINIT,-
                                  006D                            PRINT_REG,R1
                                  0083       ;;; TEST 21, SUBTEST 1, ERROR 1
                                  008A   20$:    $DS_CKLOOP       10$
                      00    DD    0092   25$:    PUSHL   #0                             ; DEFINE FLAGS
        00000000'EF   01    FB    0094           CALLS   #1,GET_POSITION                ; GET CURRENT DRIVE POSITION
                 1D 50      E8    009B           BLBS    R0,30$                         ; BRANCH IF NO ERRORS
                                  009E           $DS_ERRHARD_S    ,L$UNIT,MSG19_READH,-
                                  009E                            PRINT_REG,R1
                                  00B4       ;;; TEST 21, SUBTEST 1, ERROR 2
                                  00BB   30$:    $DS_CKLOOP       25$
            00000000'EF     D5    00C3           TSTL    CURR_CYL_HD                    ; ARE WE AT CYLN 0
                      44    13    00C9           BEQL    50$                            ; BRANCH IF WE ARE
                      00    DD    00CB   35$:    PUSHL   #0                             ; PUSH HEAD TO USE
                      00    DD    00CD           PUSHL   #0                             ; PUSH DESTINATION CYLN
        00000000'EF   02    FB    00CF           CALLS   #2,POSITION_HDS                ; EXECUTE THE SEEK
                 1D 50      E8    00D6           BLBS    R0,40$                         ; BRANCH IF NO ERRORS
                                  00D9           $DS_ERRHARD_S    ,L$UNIT,@ERR_MSG,-
                                  00D9                            PRINT_REG,R1
                                  00EF       ;;; TEST 21, SUBTEST 1, ERROR 3
                                  00F6   40$:    $DS_CKLOOP       35$
                                  00FE           $DS_ESCAPE       TEST                  ; IF ERROR EXIT TEST
            00000000'EF     D4    0109           CLRL    CONFIG                         ; CLEAR ANY PROGRAM FLAGS
                                  010F   50$:    $DS_ENDSUB
                                  011A           $DS_BGNSUB
                                  0125           STARS
                                  0125       ;;*******************************************************************
                                  0125       ;++
                                  0125       ; FORCE A HEADER NOT FOUND BY USING AN ILLEGAL ADDRESS
                                  0125       ; IN THE IDCDAR AND EXECUTING A READ DATA COMMAND.
                                  0125       ; ALSO TIME THE SETTING OF THE OPI TIMEOUT LOGIC.
                                  0125       ;--
                                  0125
     54     00000000'EF     D0    0125           MOVL    DRIVE_TBL,R4                   ; GET DRIVE INFO TABLE
           56     28 A4     D0    012C           MOVL    OFST_SECLEN(R4),R6             ; GET DRIVE SECTOR LENGTH
                      55    D4    0130           CLRL    R5                             ; RESET WORK REGISTER
 55   10    10    14 A4     F0    0132           INSV    OFST_TSTCYL(R4),#16,#16,R5     ; SETUP CYLINDER ADDRESS
   55   08    08    01      F0    0138           INSV    #1,#8,#8,R5                    ; LOAD UNSELECTED TRACK ADDRESS
        00000000'EF   00    FB    013D   10$:    CALLS   #0,SUBSYS_INIT                 ; INITIALIZE SUBSYSTEM
                 1D 50      E8    0144           BLBS    R0,20$                         ; BRANCH IF NO ERRORS
                                  0147           $DS_ERRHARD_S    ,L$UNIT,MSG14_SYSINIT,-
                                  0147                            PRINT_REG,R1
                                  015D       ;;; TEST 21, SUBTEST 2, ERROR 1
                                  0164   20$:    $DS_CKLOOP       10$
                      56    DD    016C           PUSHL   R6                             ; GET BYTE COUNT
            00000000'EF     7F    016E           PUSHAQ  READ_BUFFER                    ; GET BUFFER DESCRIPTOR
        00000000'EF   02    FB    0174           CALLS   #2,$GS_SETMAP                  ; MAP THE UBA
               53     1D    3C    017B   25$:    MOVZWL  #29,R3                         ; SET OPI WAIT LIMIT
                      00    DD    017E           PUSHL   #0                             ; IDCMPR DATA
                      55    DD    0180           PUSHL   R5                             ; IDCDAR = ILLEGAL ADDRESS
               7E     56    CE    0182           MNEGL   R6,-(SP)                       ; IDCCNT (WORD COUNT)
                      00    DD    0185           PUSHL   #0                             ; IDCBAR DATA
                      0C    DD    0187           PUSHL   #READ                          ; DEFINE COMMAND
```

```
          00000000'EF    05   FB  0189             CALLS   #5,START_CMD                        ; EXECUTE READ DATA CMD
                                  0190      27$:    $DS_WAITMS_S   #1                          ; WAIT 10 MILLI SECOND
          62   00000400 8F    D3  019B             BITL    #CSR$M_OPI,IDCCSR(R2)               ; IF OPI IS SET
                         2A   12  01A2             BNEQ    30$                                 ; THEN GOTO 30$
                      E9 53   F5  01A4             SOBGTR  R3,27$                              ; ELSE DECREMENT WAIT TIME
             00000000'EF      16  01A7             JSB     RD_IDCREGS                          ; SNAPSHOT DEVICE REGISTERS
                                  01AD             $DS_ERRHARD_S     ,L$UNIT,MSG52_RD_HNF,-
                                  01AD                               PRINT_REG,#MSG35_OPI_TO
                                  01C7      ;;; TEST 21, SUBTEST 2, ERROR 2
                                  01CE      30$:    $DS_CKLOOP       25$
                                  01D6             $DS_ESCAPE        TEST                      ; LEAVE TEST IF ERROR
          00000000'EF    03   D0  01E1             MOVL    #<NOCHK$M_CERR!NOCHK$M_CRDY>,CONFIG ; SET STATUS FLAGS
          00000000'EF    00   FB  01E8             CALLS   #0,STATUS_CHK                       ; CHECK DRIVE STATUS
                      1D 50   E8  01EF             BLBS    R0,35$                              ; BRANCH IF NO ERRORS
                                  01F2             $DS_ERRHARD_S     ,L$UNIT,MSG52_RD_HNF,-
                                  01F2                               PRINT_REG,R1
                                  0208      ;;; TEST 21, SUBTEST 2, ERROR 3
                                  020F      35$:    $DS_CKLOOP       25$
                              3C  0218             MOVZWL  #<CSR$M_ERR!CSR$M_HNF!CSR$M_OPI!-   ; SET EXPECTED IDCCSR REGISTER
                                  0219                     CSR$M_CRDY!READ!CSR$M_DRDY>,-       ;
       00000000'EF    948D 8F     0219                     E.CS                                ;
00000000'EF   00000000'EF     C8  0221             BISL2   P$UNIT,E.CS                         ; OR IN THE DRIVE NUMBER
     00000000'EF      58 A4   C8  022C             BISL2   OFST_CSRTYP(R4),E.CS                ; OR IN DRIVE TYPE BIT
00000000'EF   00100000 8F     CA  0234             BICL2   #CSR$M_ECS0,T.CS                    ; RESET ANY ECC STATUS BITS
00000000'EF   00000000'EF     D1  023F             CMPL    E.CS,T.CS                           ; EXPECTED = ACTUAL ?
                         33   13  024A             BEQL    40$                                 ; BRANCH IF YES
                                  024C             $DS_ERRHARD_S     ,L$UNIT,MSG52_RD_HNF,-
                                  024C                               REG_ERRPT,#MSG23_CS_INC,-
                                  024C                               E.CS,T.CS,#CSR_BITS
                                  0278      ;;; TEST 21, SUBTEST 2, ERROR 4
                                  027F      40$:    $DS_CKLOOP       25$
             53    1E    53   C3  0288             SUBL3   R3,#30,R3                           ; CALC ACTUAL TIME FOR OPI TO SET
                   14    53   D1  028C             CMPL    R3,#20                              ; IF ACTUAL NOT GTR THEN MAX
                   2A    15       028F             BLEQ    50$                                 ; THEN GOTO 50$
                                  0291             $DS_ERRHARD_S     ,L$UNIT,MSG52_RD_HNF,-
                                  0291                               PRINT_REG,#MSG37_OPI_LAT
                                  02AB      ;;; TEST 21, SUBTEST 2, ERROR 5
                                  02B2             $DS_CKLOOP        25$
                   0A    53   D1  02BB      50$:    CMPL    R3,#10                             ; IF ACTUAL NOT LESS THEN MIN
                         21   14  02BE             BGTR    60$                                 ; THEN GOTO 60$
                                  02C0             $DS_ERRHARD_S     ,L$UNIT,MSG52_RD_HNF,-
                                  02C0                               PRINT_REG,#MSG36_OPI_RLY
                                  02DA      ;;; TEST 21, SUBTEST 2, ERROR 6
                                  02E1      60$:    $DS_ENDSUB
                                  02EC             $DS_ENDTEST
                                  02F7             $DS_SBTTL <NON-EXISTANT MEMORY ERROR TEST>,ALIGN=LONG
                                  02F7             .SBTTL  TEST 22:  NON-EXISTANT MEMORY ERROR TEST
                                  02F7             $D2_SBTTL         022,<NON-EXISTANT MEMORY ERROR TEST>,<LONG>
                        00000000                           .PSECT  TEST_022, LONG, NOWRT
                                  0024
```

```
                              0024
                              0024     ;++
                              0024     ;
                              0024     ; TEST DESCRIPTION:
                              0024     ;
                              0024     ; THIS TEST WILL VERIFY THAT A NON-EXISTANT MEMORY ERROR WILL SET
                              0024     ; BY MAPPING THE UBA SUCH THAT A TRANSFER WILL OCCUR ACROSS AN
                              0024     ; UNMAPPED PAGE BOUNDRY THUS CAUSING A NON-EXISTANT MEMORY FAILURE.
                              0024     ; A NON-EXISTANAT MEMORY FAILURE SHOULD OCCUR USING ALL BUS ADDRESS BITS
                              0024     ; 9 THRU 17.
                              0024     ;
                              0024     ; TEST STEPS:
                              0024     ;
                              0024     ; A.  SUBTEST 1
                              0024     ;
                              0024     ;         1.  INITIALIZE SUBSYSTEM
                              0024     ;         2.  GET CURRENT DRIVE POSITION
                              0024     ;         3.  IF AT CYLN ZERO GO TO SUBTEST 2
                              0024     ;         4.  EXECUTE SEEK BACK TO ZERO
                              0024     ;         5.  IF NOT AT CYLN ZERO NOW EXIT TEST
                              0024     ;
                              0024     ; B.  SUBTEST 2
                              0024     ;
                              0024     ;         1.  INITIALIZE SUBSYSTEM
                              0024     ;         2.  MAP UBA FOR AN ILLEGAL ADDRESS
                              0024     ;         3.  EXECUTE A WRITE DATA COMMAND
                              0024     ;         4.  CHECK FOR ANY UNEXPECTED ERRORS
                              0024     ;         5.  CHECK THAT NON-EXISTANT MEMORY ERROR IS SET IN IDCCSR
                              0024     ;         6.  RETURN TO STEP 1 UNTIL BA BITS 9 THRU 17 HAVE BEEN TESTED.
                              0024     ;
                              0024     ; DEBUG:
                              0024     ;
                              0024     ;         BUS SLAVE SYNC TIME OUT LOGIC
                              0024     ;         BUS ADDRESS REGISTER
                              0024     ;         BUS ADDRESS DRIVERS
                              0024     ;
                              0024     ;--
                              0024
                              0024
                              0024              $DS_BGNDATA
       00000000 00000001      0024              .LONG   1,LUN0
       00000001 00000001      002C              .LONG   1,LUN1
       00000002 00000001      0034              .LONG   1,LUN2
       00000003 00000001      003C              .LONG   1,LUN3
                              0044              $DS_ENDDATA
                              0048
                              0048              $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
                              004A              GET_LUN -                                   ; GET LUN INFO
                              004A                UNIT=           OFST_UNIT(AP)             ; LOGICAL UNIT
            04 AC       DD    004A              PUSHL   OFST_UNIT(AP)
    00000000'EF   01    FB    004D              CALLS   #1,GET_LUN$A
                              0054              BDEVFOUND       5$                          ; ID DEV AVAILABLE GOTO 5$
       06 50   00       E2    0054              BBSS    #0,R0,5$
           50  01       D0    0058                      MOVL    #1,R0           ; NOT AN ERROR
             01C7       31    005B                      BRW     TEST_022_X      ; EXIT TEST 22
      00000000'EF       D4    005E         5$:  CLRL    CONFIG                      ; CLEAR ANY PROGRAM FLAGS
```

```
       54   00000000'EF  D0  0064               MOVL    DRIVE_TBL,R4                    ; GET DRIVE INFO
             57    14 A4  D0  006B               MOVL    OFST_TSTCYL(R4),R7              ; GET TEST CYLINDER
                      58  D4  006F               CLRL    R8                              ; RESET WORK REGISTER
    58    10    10    57  F0  0071               INSV    R7,#16,#16,R8                   ; SET UP DAR REGISTER
                              0076               $DS_BGNSUB
                              0081               STARS
                              0081      ;;*****************************************************************
                              0081      ;++
                              0081      ; POSITION DRIVE TO TEST CYLINDER 0 AND VERIFY THE SEEK
                              0081      ;--
                              0081
        00000000'EF    00  FB  0081      10$:     CALLS   #0,SUBSYS_INIT                  ; INITIALIZE SUBSYSTEM
                  1D 50  E8  0088               BLBS    R0,20$                          ; BRANCH IF NO ERRORS
                              008B               $DS_ERRHARD_S    ,L$UNIT,MSG14_SYSINIT,-
                              008B                                PRINT_REG,R1
                              00A1      ;;; TEST 22, SUBTEST 1, ERROR 1
                              00A8      20$:     $DS_CKLOOP      10$
                      00  DD  00B0      35$:     PUSHL   #0                              ; PUSH HEAD TO USE
                      57  DD  00B2               PUSHL   R7                              ; PUSH DESTINATION CYLN
        00000000'EF    02  FB  00B4               CALLS   #2,POSITION_HDS                 ; EXECUTE THE SEEK
                  1D 50  E8  00BB               BLBS    R0,40$                          ; BRANCH IF NO ERRORS
                              00BE               $DS_ERRHARD_S    ,L$UNIT,@ERR_MSG,-
                              00BE                                PRINT_REG,R1
                              00D4      ;;; TEST 22, SUBTEST 1, ERROR 2
                              00DB      40$:     $DS_CKLOOP      35$
                              00E3               $DS_ESCAPE      TEST                    ; EXIT TEST IF ERROR AND NO LOOP
             00000000'EF  D4  00EE               CLRL    CONFIG                          ; CLEAR ANY PROGRAM FLAGS
             00000000'EF  D5  00F4               TSTL    CURR_CYLN                       ; ARE WE AT CYLN 0 NOW ?
                  06  13  00FA               BEQL    50$                             ; BRANCH IF WE ARE
                              00FC               $DS_EXIT        TEST
                              0102      50$:     $DS_ENDSUB
                              010D               $DS_BGNSUB
                              0118               STARS
                              0118      ;;*****************************************************************
                              0118      ;++
                              0118      ; FORCE A NON EXISTANT MEMORY ERROR BY MAPPING AN INCORRECT MEMORY
                              0118      ; ADDRESS AND THEN EXECUTING A WRITE DATA COMMAND
                              0118      ; TEST ALL BUS ADDRESS BITS 9 THRU 17
                              0118      ;--
                              0118
         55   0200 8F  3C  0118               MOVZWL  #^X200,R5                       ; SET UP INITIAL BUS ADDR REG
        00000000'EF    00  FB  011D      10$:     CALLS   #0,SUBSYS_INIT                  ; INITIALIZE SUBSYSTEM
                  1D 50  E8  0124               BLBS    R0,20$                          ; BRANCH IF NO ERRORS
                              0127               $DS_ERRHARD_S    ,L$UNIT,MSG14_SYSINIT,-
                              0127                                PRINT_REG,R1
                              013D      ;;; TEST 22, SUBTEST 2, ERROR 1
                              0144      20$:     $DS_CKLOOP      10$
                      02  DD  014C               PUSHL   #2                              ; GET BYTE COUNT
             00000000'EF  7F  014E               PUSHAQ  WRITE_BUFFER                    ; GET BUFFER ADDRS
        00000000'EF    02  FB  0154               CALLS   #2,$GS_SETMAP                   ; MAP UBA
                      00  DD  015B               PUSHL   #0                              ; IDCMPR DATA
                      58  DD  015D               PUSHL   R8                              ; IDCDAR (SECTOR 0)
             FFFFFFFF 8F  DD  015F               PUSHL   #-1                             ; IDCCNT (BYTE COUNT)
                      55  DD  0165               PUSHL   R5                              ; SET BA TO FLOP OVR BOUNDRY
                      0A  DD  0167               PUSHL   #WRITE                          ; SET UP FOR WRITE DATA FUNCTION
        00000000'EF    05  FB  0169               CALLS   #5,START_CMD                    ; EXECUTE WRITE DATA CMD
```

```
          00000000'EF   02   D0  0170   585         MOVL    #NOCHK$M_CERR,CONFIG              ; DONT CHECK ANY IDCCSR ERRORS
          00000000'EF   00   FB  0177   586         CALLS   #0,STATUS_CHK                     ; CHECK FOR ANY OTHER ERRORS
                     1D 50   E8  017E   587         BLBS    R0,30$                            ; BRANCH IF NONE
                             0181   588         $DS_ERRHARD_S    ,L$UNIT,MSG53_WRT_NXM,-
                             0181   589                          PRINT_REG,R1
                             0197       ;;; TEST 22, SUBTEST 2, ERROR 2
                             019E   590 30$:    $DS_CKLOOP       10$
     00000000'EF     208B 8F  3C  01A7   591         MOVZWL  #<CSR$M_NXM!CSR$M_CRDY!-          ; NOTE:NXM DOES NOT SET CERR
                             01B0   592                 WRITE!CSR$M_DRDY>,E.CS            ; SET EXPECTED IDCCSR REGISTER
 00000000'EF    00000000'EF   C8  01B0   593         BISL2   P$UNIT,E.CS                       ; OR IN THE DRIVE NUMBER
       00000000'EF     58 A4   C8  01BB   594         BISL2   OFST_CSRTYP(R4),E.CS              ; OR IN DRIVE TYPE BIT
 00000000'EF    00000000'EF   D1  01C3   595         CMPL    E.CS,T.CS                         ; EXPECTED = ACTUAL ?
                        33   13  01CE   596         BEQL    40$                               ; BRANCH IF YES
                             01D0   597         $DS_ERRHARD_S    ,L$UNIT,MSG53_WRT_NXM,-
                             01D0   598                          REG_ERRPT,#MSG23_CS_INC,-
                             01D0   599                          E.CS,T.CS,#CSR_BITS
                             01FC       ;;; TEST 22, SUBTEST 2, ERROR 3
                             0203   600 40$:    $DS_CKLOOP       10$
               55   55   01   9C  020C   601         ROTL    #1,R5,R5                          ; POINT TO NEXT BA BIT
                  03 55   12   E0  0210   602         BBS     #18,R5,50$                        ; IF ALL ADDRESS BITS TESTED GOTO  -
cont> 42$
                      FF06   31  0214   603         BRW     10$                               ; ELSE TEST NEXT BIT
                             0217   604 50$:    $DS_ENDSUB
                             0222   605         $DS_ENDTEST
                             022D   606         $DS_SBTTL <LOOK AHEAD REGISTER TEST - R80>,ALIGN=LONG
                             022D               .SBTTL  TEST 23:  LOOK AHEAD REGISTER TEST - R80
                             022D               $D2_SBTTL        023,<LOOK AHEAD REGISTER TEST - R80>,<LONG>
                         00000000                        .PSECT  TEST_023, LONG, NOWRT
                             0024
```

```
                               0024
                               0024      ;++
                               0024      ;
                               0024      ; TEST DESCRIPTION:
                               0024      ;
                               0024      ; THIS TEST VERIFIES THAT THE LOOK AHEAD SECTOR BITS IN
                               0024      ; R80 DRIVE STATUS WORD ARE COUNTING CORRECTLY.
                               0024      ;
                               0024      ; TEST STEPS:
                               0024      ;
                               0024      ; 1.  INITIALIZE SUBSYSTEM
                               0024      ; 2.  PERFORM GET STATUS FUNCTION
                               0024      ; 3.  RETRUN TO STEP 2 UNTILL COUNTER LOOKAHEAD REACHES 0
                               0024      ; 4.  DISPLAY ERROR IF PROGRAM TIMES OUT WAITING FOR
                               0024      ;     SECTOR 0.
                               0024      ; 5.  PERFORM CONTINIOUS GET STATUS FUNCTION WHILE MONITORING
                               0024      ;     THE COUNT OF THE LOOKAHEAD BITS
                               0024      ; 6.  REPORT ERROR IF COUNT IS NOT THE EXPECTED VALUE.
                               0024      ;
                               0024      ; DEBUG:
                               0024      ;
                               0024      ;           DRIVE SECTOR COUNTER
                               0024      ;
                               0024      ;--
                               0024
                               0024
                               0024               $DS_BGNDATA
      00000000 00000001        0024               .LONG   1,LUN0
      00000001 00000001        002C               .LONG   1,LUN1
      00000002 00000001        0034               .LONG   1,LUN2
      00000003 00000001        003C               .LONG   1,LUN3
                               0044               $DS_ENDDATA
                               0048
                               0048               $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
                               004A               GET_LUN -                                   ; GET LUN INFO
                               004A                 UNIT= OFST_UNIT(AP)                       ; LOGICAL UNIT
           04 AC     DD        004A               PUSHL   OFST_UNIT(AP)
    00000000'EF  01  FB        004D               CALLS   #1,GET_LUN$A
                               0054               BDEVFOUND       5$                          ; IF DEV AVAILABLE GOTO 5$
        06 50  00    E2        0054               BBSS    #0,R0,5$
           50  01    D0        0058                       MOVL    #1,R0           ; NOT AN ERROR
             01C6    31        005B                       BRW     TEST_023_X      ; EXIT TEST 23
54    00000000'EF    D0        005E      5$:      MOVL    DRIVE_TBL,R4                        ; GET DRIVE INFO TABLE
           02  64    91        0065               CMPB    OFST_DRVTYP(R4),#R80$T              ; IF DRIVE TYPE = R80
               06    13        0068               BEQL    6$                                  ; THEN GOTO 6$
                               006A               $DS_EXIT        TEST                        ; ELSE EXIT TEST
      00000000'EF    D4        0070      6$:      CLRL    CONFIG                              ; CLEAR PROGRAM FLAGS
    00000000'EF  00  FB        0076      10$:     CALLS   #0,SUBSYS_INIT                      ; CLEAR SUBSYSTEM
           1D  50    E8        007D               BLBS    R0,20$                              ; IF NO ERRORS GOTO 20$
                               0080               $DS_ERRHARD_S   ,L$UNIT,MSG14_SYSINIT,-
                               0080                               PRINT_REG,R1
                               0096          ;;; TEST 23, SUBTEST 0, ERROR 1
                               009D      20$:     $DS_CKLOOP      10$
               53    D4        00A5               CLRL    R3                                  ; RESET ERROR RETRY COUNTER
               55    D4        00A7      23$:     CLRL    R5                                  ; RESET WAIT FOR ZERO COUNTER
               57    D4        00A9               CLRL    R7                                  ; SET EXPECTED SECTOR NUMBER
```

```
              58    04    9A  00AB   658          MOVZBL   #GETSTAT,R8                        ; SAVE COMMAND
       58  00000000'EF    C8  00AE   659          BISL2    P$UNIT,R8                          ; OR IN DRIVE NUMBER
                    5B    D4  00B5   660 25$:     CLRL     R11                                ; CLEAR CRDY COUNTER COUNTER
                    00    DD  00B7   661          PUSHL    #0                                 ; IDCMPR DATA
                    00    DD  00B9   662          PUSHL    #0                                 ; IDCDAR DATA
                    00    DD  00BB   663          PUSHL    #0                                 ; IDCCNT DATA
                    00    DD  00BD   664          PUSHL    #0                                 ; IDCBAR DATA
                    04    DD  00BF   665          PUSHL    #GETSTAT                           ; FUNCTION = GET STATUS
        00000000'EF    05    FB  00C1   666          CALLS    #5,START_CMD                       ; INITIATE GET STATUS COMMAND
       62  00000080 8F    D3  00C8   667 27$:     BITL     #CSR$M_CRDY,IDCCSR(R2)             ; IF READY IS SET
                    42    12  00CF   668          BNEQ     30$                                ; THEN GOTO 30$
    EF 5B  00000064 8F    F2  00D1   669          AOBLSS   #100,R11,27$                       ; TRY AGAIN
           00000000'EF    16  00D9   670          JSB      RD_IDCREGS                         ; SNAPSHOT REGISTERS
                              00DF   671          $DS_ERRHARD_S     ,L$UNIT,MSG17_GETSTAT,-
                              00DF   672                            PRINT_REG,#MSG30_CRDY
                              00F9       ;;; TEST 23, SUBTEST 0, ERROR 2
                              0100   673          $DS_CKLOOP        25$
                              0108   674          $DS_ESCAPE        TEST
      00000000'EF    10 A2    D0  0113   675 30$:     MOVL     IDCMPR(R2),T.MP                    ; GET STATUS WORD FROM DRIVE
  56  00000000'EF    05 00    EF  011B   676          EXTZV    #0,#5,T.MP,R6                      ; EXTRACT LOOK AHEAD BITS
                    43    13  0124   677          BEQL     40$                                ; IF AT SECTOR 0 GOTO 40$
    87 55  00000064 8F    F2  0126   678          AOBLSS   #100,R5,25$                        ; TRY AGAIN TO FIND SECTOR 0
           00000000'EF    16  012E   679          JSB      RD_IDCREGS                         ; SNAPSHOT REGISTERS
                              0134   680          $DS_ERRHARD_S     ,L$UNIT,MS108_SECTNZ,-
                              0134   681                            REG_ERRPT,#0,R7,R6
                              014E       ;;; TEST 23, SUBTEST 0, ERROR 3
                              0155   682          $DS_CKLOOP        23$
                              015E   683          $DS_ESCAPE        TEST                      ; EXIT TEST IF ERROR
              57    01    D0  0169   684 40$:     MOVL     #1,R7                              ; SET EXPECTED SECTOR COUNT
                    55    D4  016C   685 45$:     CLRL     R5                                 ; RESET RETRY COUNTER
                    5B    D4  016E   686 47$:     CLRL     R11                                ; RESET COUNTER
              62    58    D0  0170   687          MOVL     R8,IDCCSR(R2)                      ; LOAD GET STATUS COMMAND
       62  00000080 8F    D3  0173   688 48$:     BITL     #CSR$M_CRDY,IDCCSR(R2)             ; IF READY IS SET
                    43    12  017A   689          BNEQ     50$                                ; THEN GOTO 50$
    EF 5B  00000064 8F    F2  017C   690          AOBLSS   #100,R11,48$                       ; TRY AGAIN
           00000000'EF    16  0184   691          JSB      RD_IDCREGS                         ; SNAPSHOT REGISTERS
                              018A   692          $DS_ERRHARD_S     ,L$UNIT,MSG17_GETSTAT,-
                              018A   693                            PRINT_REG,#MSG30_CRDY
                              01A4       ;;; TEST 23, SUBTEST 0, ERROR 4
                              01AB   694          $DS_CKLOOP        23$
                              01B4   695          $DS_ESCAPE        TEST
      00000000'EF    10 A2    D0  01BF   696 50$:     MOVL     IDCMPR(R2),T.MP                    ; GET DRIVE STATUS WORD
  56  00000000'EF    05 00    EF  01C7   697          EXTZV    #0,#5,T.MP,R6                      ; EXTRACT LOOK AHEAD BITS
              57    56    D1  01D0   698          CMPL     R6,R7                              ; EXPECTED SECTOR REACHED
                    45    13  01D3   699          BEQL     60$                                ; BRANCH IF YES
           95 55    0A    F2  01D5   700          AOBLSS   #10,R5,47$                         ; TRY AGAIN
     FED6 53    01    01    3D  01D9   701          ACBW     #1,#1,R3,25$                       ; MAYBE WE JUST LOST SYNC SO -
                              01DF   702                                                     ; LETS TRY IT ALL OVER AGAIN -
                              01DF   703                                                     ; JUST TO MAKE SURE
           00000000'EF    16  01DF   704          JSB      RD_IDCREGS                         ; SNAPSHOT REGISTERS
                              01E5   705          $DS_ERRHARD_S     ,L$UNIT,MS109_BADCNT,-
                              01E5   706                            REG_ERRPT,#0,R7,R6
                              01FF       ;;; TEST 23, SUBTEST 0, ERROR 5
                              0206   707          $DS_CKLOOP        23$
                              020F   708          $DS_ESCAPE        TEST                      ; EXIT TEST IF ERROR
   FF4B 57    01    10 A4    9D  021A   709 60$:     ACBB     OFST_LSTSEC(R4),#1,R7,45$          ; CHECK NEXT COUNT
```

```
          0221   710        $DS_ENDTEST
          022C   711        $DS_SBTTL <HEAD SELECT LOGIC TEST - R80>,ALIGN=LONG
          022C              .SBTTL  TEST 24:  HEAD SELECT LOGIC TEST - R80
          022C              $D2_SBTTL       024,<HEAD SELECT LOGIC TEST - R80>,<LONG>
      00000000                        .PSECT  TEST_024, LONG, NOWRT
          0020
```

```
                                0020      ;++
                                0020      ;
                                0020      ; TEST DESCRIPTION:
                                0020      ;
                                0020      ; THIS TEST WILL VERIFY THAT ALL HEADS IN THE R80 DRIVE CAN BE
                                0020      ; SELECTED USING A SEEK COMMAND. ALL HEADS ARE TESTED BY EXECUTING
                                0020      ; SEEKS AND VERIFYING HEAD SELECTED WITH A READ HEADER COMMAND.
                                0020      ;
                                0020      ; TEST STEPS:
                                0020      ;
                                0020      ;        1.  INITIALIZE SUBSYSTEM
                                0020      ;        2.  SEEK HEADS TO CYLINDER ONE
                                0020      ;        3.  REPORT ANY ERRORS ESPECIALLY HEAD SELECT ERRORS
                                0020      ;        6.  CONTINUE UNTIL ALL HEADS HAVE BEEN TESTED
                                0020      ;
                                0020      ; DEBUG:
                                0020      ;
                                0020      ;        HEAD SELECT LOGIC
                                0020      ;
                                0020      ;--
                                0020
                                0020
                                0020               $DS_BGNDATA
       00000000 00000001        0020               .LONG   1,LUN0
       00000001 00000001        0028               .LONG   1,LUN1
       00000002 00000001        0030               .LONG   1,LUN2
       00000003 00000001        0038               .LONG   1,LUN3
                                0040               $DS_ENDDATA
                                0044
                                0044               $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
                                0046               GET_LUN -                                   ; GET LUN  INFO
                                0046                 UNIT= OFST_UNIT(AP)                       ; LOGICAL UNIT
             04 AC     DD       0046               PUSHL   OFST_UNIT(AP)
   00000000'EF    01   FB       0049               CALLS   #1,GET_LUN$A
                                0050               BDEVFOUND        5$                         ; IF DEV AVAILABLE GOTO 5$
          06 50   00   E2       0050               BBSS    #0,R0,5$
             50   01   D0       0054                       MOVL     #1,R0          ; NOT AN ERROR
           008A   31            0057                       BRW      TEST_024_X     ; EXIT TEST 24
 54  00000000'EF       D0       005A      5$:      MOVL    DRIVE_TBL,R4                        ; GET DRIVE INFO TABLE
          02  64       91       0061               CMPB    OFST_DRVTYP(R4),#R80$T              ; IF DRIVE TYPE = R80
              06       13       0064               BEQL    6$                                  ; THEN GOTO 6$
                                0066               $DS_EXIT         TEST                       ; ELSE EXIT TEST
      00000000'EF      D4       006C      6$:      CLRL    CONFIG                              ; CLEAR ALL PROGRAM FLAGS
              53       D4       0072               CLRL    R3                                  ; RESET HEAD POINTER
   00000000'EF    00   FB       0074      10$:     CALLS   #0,SUBSYS_INIT                      ; INITIALIZE SUBSYSTEM
           1D 50       E8       007B               BLBS    R0,20$                              ; BRANCH IF NO ERRORS
                                007E               $DS_ERRHARD_S    ,L$UNIT,MSG14_SYSINIT,-
                                007E                                PRINT_REG,R1
                                0094          ;;; TEST 24, SUBTEST 0, ERROR 1
                                009B      20$:     $DS_CKLOOP       10$
              53       DD       00A3      30$:     PUSHL   R3                                  ; PUSH DESIRED HEAD
              01       DD       00A5               PUSHL   #1                                  ; PUSH DESTINATION CYLN
   00000000'EF    02   FB       00A7               CALLS   #2,POSITION_HDS                     ; EXECUTE THE SEEK CMD
           1D 50       E8       00AE               BLBS    R0,40$                              ; BRANCH IF NO ERRORS
                                00B1               $DS_ERRHARD_S    ,L$UNIT,@ERR_MSG,-
                                00B1                                PRINT_REG,R1
```

```
                          00C7          ;;; TEST 24, SUBTEST 0, ERROR 2
                          00CE    763 40$:    $DS_CKLOOP       30$
        00000000'EF   D4  00D6    764         CLRL    CONFIG                      ; CLEAR ANY PROGRAM FLAGS
    C2 53     0C A4   F3  00DC    765         AOBLEQ  OFST_LSTTRK(R4),R3,30$      ; GET NEXT RANDOM CYLINDER
                          00E1    766         $DS_ENDTEST
                          00EC    767         $DS_SBTTL <BASIC DATA TRANSFER TEST>,ALIGN=LONG
                          00EC                .SBTTL  TEST 25:  BASIC DATA TRANSFER TEST
                          00EC                $D2_SBTTL        025,<BASIC DATA TRANSFER TEST>,<LONG>
                      00000000                         .PSECT  TEST_025, LONG, NOWRT
                          001C
```

```
                                             001C
                                             001C    ;++
                                             001C    ;
                                             001C    ; TEST DESCRIPTION:
                                             001C    ;
                                             001C    ; THIS TEST WILL CHECK THAT READ WRITE LOOP CAPABILITY OF THE IDC
                                             001C    ; SUBSYSTEM BY EXECUTING A ONE SECTOR WRITE DATA COMMAND.
                                             001C    ; AND THEN EXECUTE A ONE SECTOR READ DATA COMMAND. VERIFY THAT THE
                                             001C    ; DATA READ BACK IS CORRECT.
                                             001C    ;
                                             001C    ; TEST STEPS:
                                             001C    ;
                                             001C    ; A.  SUBTEST 1
                                             001C    ;
                                             001C    ;         1.  INITIALIZE SUBSYSTEM
                                             001C    ;         2.  EXECUTE SEEK TO TEST CYLINDER
                                             001C    ;         3.  IF ERROR THEN ESCAPE TEST
                                             001C    ;
                                             001C    ; B.  SUBTEST 2
                                             001C    ;
                                             001C    ;         1.  INITIALIZE SUBSYSTEM
                                             001C    ;         2.  EXECUTE WRITE DATA COMMAND
                                             001C    ;         3.  CHECK FOR ANY ERRORS
                                             001C    ;
                                             001C    ; C.  SUBTEST 3
                                             001C    ;
                                             001C    ;         1.  INITIALIZE SUBSYSTEM
                                             001C    ;         2.  EXECUTE READ DATA COMMAND
                                             001C    ;         3.  CHECK FOR ANY ERRORS
                                             001C    ;         4.  VERIFY DATA IS CORRECT - REPORT ANY DESCREPENCIES
                                             001C    ;
                                             001C    ; DEBUG:
                                             001C    ;
                                             001C    ;         PHASE LOCK LOOP CKT
                                             001C    ;         RL02 R/W MODULE
                                             001C    ;         UPPER OR LOWER HEAD
                                             001C    ;
                                             001C    ;--
                                             001C
                                             001C             $DS_BGNDATA
00000000 00000001 00000000 00000003          001C             .LONG     3,LUN0,RL02$T,HEAD_0
00000001 00000001 00000000 00000003          002C             .LONG     3,LUN0,RL02$T,HEAD_1
00000000 00000002 00000000 00000003          003C             .LONG     3,LUN0,R80$T,HEAD_0
00000000 00000001 00000001 00000003          004C             .LONG     3,LUN1,RL02$T,HEAD_0
00000001 00000001 00000001 00000003          005C             .LONG     3,LUN1,RL02$T,HEAD_1
00000000 00000002 00000001 00000003          006C             .LONG     3,LUN1,R80$T,HEAD_0
00000000 00000001 00000002 00000003          007C             .LONG     3,LUN2,RL02$T,HEAD_0
00000001 00000001 00000002 00000003          008C             .LONG     3,LUN2,RL02$T,HEAD_1
00000000 00000002 00000002 00000003          009C             .LONG     3,LUN2,R80$T,HEAD_0
00000000 00000001 00000003 00000003          00AC             .LONG     3,LUN3,RL02$T,HEAD_0
00000001 00000001 00000003 00000003          00BC             .LONG     3,LUN3,RL02$T,HEAD_1
00000000 00000002 00000003 00000003          00CC             .LONG     3,LUN3,R80$T,HEAD_0
                                             00DC             $DS_ENDDATA
                                             00E0
                                             00E0             $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
                                             00E2             GET_LUN -                              ; GET LUN INFO
```

```
                           00E2   825            UNIT= OFST_UNIT(AP)                        ; LOGICAL UNIT
              04 AC    DD  00E2             PUSHL   OFST_UNIT(AP)
       00000000'EF     01  FB  00E5             CALLS   #1,GET_LUN$A
                           00EC   826            BDEVFOUND          5$                     ; IF DEV AVAILABLE GOTO 5$
           06 50   00  E2  00EC             BBSS    #0,R0,5$
              50   01  D0  00F0                     MOVL     #1,R0          ; NOT AN ERROR
                 0279  31  00F3                     BRW      TEST_025_X     ; EXIT TEST 25
     54   00000000'EF  D0  00F6   827 5$:       MOVL    DRIVE_TBL,R4                       ; GET DRIVE INFO TABLE
        08 AC      64  91  00FD   828           CMPB    OFST_DRVTYP(R4),OFST_DRIV(AP)      ; IF DRIVE TYPE = DATA TYPE
                   06  13  0101   829           BEQL    6$                                 ; THEN GOTO 6$
                           0103   830           $DS_EXIT         TEST                      ; ELSE EXIT TEST
          00000000'EF  D4  0109   831 6$:       CLRL    CONFIG                             ; CLEAR ANY PROGRAM FLAGS
        57     28 A4   D0  010F   832           MOVL    OFST_SECLEN(R4),R7                 ; GET BYTES PER SECTOR
        58     14 A4   D0  0113   833           MOVL    OFST_TSTCYL(R4),R8                 ; GET TEST CYLINDER
                           0117   834           $DS_BGNSUB
                           0122   835           STARS
                           0122       ;;********************************************************************
                           0122   836 ;++
                           0122   837 ; POSITION DRIVE TO TEST CYLINDER AND VERIFY THE SEEK
                           0122   838 ;--
                           0122   839
       00000000'EF     00  FB  0122   840 10$:     CALLS   #0,SUBSYS_INIT                     ; INITIALIZE SUBSYSTEM
              1D 50    E8  0129   841           BLBS    R0,20$                             ; BRANCH IF NO ERRORS
                           012C   842           $DS_ERRHARD_S    ,L$UNIT,MSG14_SYSINIT,-
                           012C   843                            PRINT_REG,R1
                           0142       ;;; TEST 25, SUBTEST 1, ERROR 1
                           0149   844 20$:      $DS_CKLOOP       10$
              0C AC    DD  0151   845 35$:      PUSHL   OFST_SKHD(AP)                      ; PUSH HEAD TO USE
                   58  DD  0154   846           PUSHL   R8                                 ; PUSH DESTINATION CYLN
       00000000'EF     02  FB  0156   847           CALLS   #2,POSITION_HDS                    ; EXECUTE THE SEEK
              1D 50    E8  015D   848           BLBS    R0,40$                             ; BRANCH IF NO ERRORS
                           0160   849           $DS_ERRHARD_S    ,L$UNIT,@ERR_MSG,-
                           0160   850                            PRINT_REG,R1
                           0176       ;;; TEST 25, SUBTEST 1, ERROR 2
                           017D   851 40$:      $DS_CKLOOP       35$
                           0185   852           $DS_ESCAPE       TEST                      ; IF ERROR EXIT TEST
          00000000'EF  D4  0190   853           CLRL    CONFIG                             ; CLEAR ANY PROGRAM FLAGS
                           0196   854 50$:      $DS_ENDSUB
                           01A1   855           $DS_BGNSUB
                           01AC   856           STARS
                           01AC       ;;********************************************************************
                           01AC   857 ;++
                           01AC   858 ; DO A WRITE DATA COMMAND TO SECTOR 0
                           01AC   859 ;--
                           01AC   860
                   53  D4  01AC   861           CLRL    R3                                 ; INIT INDEX
  6BD6BD6B D6BD6BD6 8F 7D  01AE   862 5$:       MOVQ    #^X6BD6BD6BD6BD6BD6,-
        00000000'FF43      01B8   863                   @WRITE_BUFFER[R3]                  ; LOAD WRITE BUFFER WITH DATA
        00000000'FF43  7C  01BE   864           CLRQ    @READ_BUFFER[R3]                   ; CLEAR READ BUFFER
   E1 53    00000040 8F F2 01C5   865           AOBLSS  #64,R3,5$                          ; DO 256 WORDS
                   57  DD  01CD   866           PUSHL   R7                                 ; GET BYTE COUNT
          00000000'EF  7F  01CF   867           PUSHAQ  WRITE_BUFFER                       ; GET BUFFER ADDRESS
       00000000'EF     02  FB  01D5   868           CALLS   #2,$GS_SETMAP                      ; MAP UBA
                   59  D4  01DC   869           CLRL    R9                                 ; CLEAR REG
     59    10    10    58  F0  01DE   870           INSV    R8,#16,#16,R9                      ; SET DESTINATION CYLINDER
  59    08    08    0C AC  F0  01E3   871           INSV    OFST_SKHD(AP),#8,#8,R9             ; SET DESTINATION HEAD
```

```
                  00    DD   01E9  25$:      PUSHL   #0                           ; IDCMPR DATA
                  59    DD   01EB            PUSHL   R9                           ; IDCDAR DATA
            7E    57    CE   01ED            MNEGL   R7,-(SP)                     ; IDCCNT (BYTE COUNT)
                  00    DD   01F0            PUSHL   #0                           ; IDCBAR DATA
                  0A    DD   01F2            PUSHL   #WRITE                       ; DEFINE COMMAND
      00000000'EF 05    FB   01F4            CALLS   #5,START_CMD                 ; EXECUTE WRITE DATA CMD
      00000000'EF 00    FB   01FB            CALLS   #0,STATUS_CHK                ; CHECK FOR ANY ERRORS
               1D 50    E8   0202            BLBS    R0,30$                       ; BRANCH IF NONE
                             0205            $DS_ERRHARD_S    ,L$UNIT,MSG20_WRITE,-
                             0205                             PRINT_REG,R1
                             021B      ;;; TEST 25, SUBTEST 2, ERROR 1
                             0222  30$:      $DS_CKLOOP       25$
       5A   59    01    C1   022A            ADDL3   #1,R9,R10                    ; SET EXPECT IDCDAR
                  5A    DD   022E            PUSHL   R10                          ; GET EXPECTED DAR DATA
      00000000'EF 01    FB   0230            CALLS   #1,CHANGE_DAR$A              ; CHANGE DAR TO EXPECTED READ BACK -
cont>  VALUE
            5A    51    D0   0237            MOVL    R1,R10                       ; LOAD NEW DAR DATA
      00000000'EF 5A    D1   023A            CMPL    R10,T.DA                     ; DID IDCDAR UPDATE OK
                  2F    13   0241            BEQL    40$                          ; BRANCH IF IT DID
                             0243            $DS_ERRHARD_S    ,L$UNIT,MSG20_WRITE,-
                             0243                             REG_ERRPT,#MSG54_DA_UPD,-
                             0243                             R10,T.DA,DAR_MNEM
                             026B      ;;; TEST 25, SUBTEST 2, ERROR 2
                             0272  40$:      $DS_CKLOOP       25$
                             027B            $DS_ENDSUB
                             0286            $DS_BGNSUB
                             0291            STARS
                             0291      ;;*******************************************************************
                             0291      ;++
                             0291      ; EXECUTE A READ DATA COMMAND FOR SECTOR 0 AND VERIFY THE DATA
                             0291      ;--
                             0291
                  57    DD   0291            PUSHL   R7                           ; GET BYTE COUNT
          00000000'EF   7F   0293            PUSHAQ  READ_BUFFER                  ; GET ADRS OF READ BUFFER
      00000000'EF 02    FB   0299            CALLS   #2,$GS_SETMAP                ; MAP UBA
                  00    DD   02A0  25$:      PUSHL   #0                           ; IDCMPR DATA
                  59    DD   02A2            PUSHL   R9                           ; IDCDAR = SECTOR 0
            7E    57    CE   02A4            MNEGL   R7,-(SP)                     ; IDCCNT (BYTE COUNT)
                  00    DD   02A7            PUSHL   #0                           ; IDCBAR DATA
                  0C    DD   02A9            PUSHL   #READ                        ; DEFINE COMMAND
      00000000'EF 05    FB   02AB            CALLS   #5,START_CMD                 ; EXECUTE READ DATA CMD
      00000000'EF 00    FB   02B2            CALLS   #0,STATUS_CHK                ; CHECK FOR ANY ERRORS
               1D 50    E8   02B9            BLBS    R0,30$                       ; BRANCH IF NONE
                             02BC            $DS_ERRHARD_S    ,L$UNIT,MSG21_READ,-
                             02BC                             PRINT_REG,R1
                             02D2      ;;; TEST 25, SUBTEST 3, ERROR 1
                             02D9  30$:      $DS_CKLOOP       25$
      00000000'EF 5A    D1   02E1            CMPL    R10,T.DA                     ; DID IDCDAR UPDATE OK ?
                  2F    13   02E8            BEQL    40$                          ; BRANCH IF IT DID
                             02EA            $DS_ERRHARD_S    ,L$UNIT,MSG21_READ,-
                             02EA                             REG_ERRPT,#MSG54_DA_UPD,-
                             02EA                             R10,T.DA,DAR_MNEM
                             0312      ;;; TEST 25, SUBTEST 3, ERROR 2
                             0319  40$:      $DS_CKLOOP       25$
          00000000'EF   DD   0321            PUSHL   WRITE_BUFFER                 ; GET GOOD DATA
          00000000'EF   DD   0327            PUSHL   READ_BUFFER                  ; GET DATA JUST READ
          00000000'EF   DD   032D            PUSHL   WORD_COUNT                   ; GET WORD COUNT
```

```
           00000000'EF   03   FB  0333   923          CALLS    #3,CMPDATA                        ; CHECK THE DATA
                    1B 50     E8  033A   924          BLBS     R0,50$                            ; BRANCH IF DATA OK
                                  033D   925          $DS_ERRHARD_S    ,L$UNIT,MSG55_DAT_CP1,-
                                  033D   926                           PRINT_DATA_ERR
                                  0351        ;;; TEST 25, SUBTEST 3, ERROR 3
                                  0358   927 50$:     $DS_CKLOOP       25$
                                  0361   928          $DS_ENDSUB
                                  036C   929          $DS_ENDTEST
                                  0377   930          $DS_SBTTL <WRITE CHECK ERROR TEST>,ALIGN=LONG
                                  0377                .SBTTL  TEST 26:  WRITE CHECK ERROR TEST
                                  0377                $D2_SBTTL        026,<WRITE CHECK ERROR TEST>,<LONG>
                              00000000                        .PSECT  TEST_026, LONG, NOWRT
                                  001C
```

```
                            001C
                            001C      ;++
                            001C      ;
                            001C      ; TEST DESCRIPTION:
                            001C      ;
                            001C      ; THIS STEST WILL CHECK THAT WRITE CHECK ERROR WILL SET BY WRITING
                            001C      ; A PATTERN ON THE DISK, THEN CHANGE ONE WORD IN MEMORY AND EXECUTE
                            001C      ; A WRITE CHECK COMMAND.
                            001C      ;
                            001C      ; TEST STEPS:
                            001C      ;
                            001C      ; A.  SUBTEST 1
                            001C      ;
                            001C      ;           1. INITIALIZE SUBSYSTEM
                            001C      ;           2.  EXECUTE SEEK TO TEST CYLINDER
                            001C      ;           3.  IF ERROR THEN ESCAPE TEST
                            001C      ;
                            001C      ; B.  SUBTEST 2
                            001C      ;
                            001C      ;           1. INITIALIZE SUBSYSTEM
                            001C      ;           2. EXECUTE A WRITE DATA COMMAND OF ONE SECTOR
                            001C      ;           3. CHECK FOR ANY ERRORS
                            001C      ;
                            001C      ; C.        SUBTEST 3
                            001C      ;
                            001C      ;           1. INITIALIZE SUBSYSTEM
                            001C      ;           2. MODIFY A WORD IN THE WRITE BUFFER
                            001C      ;           3. EXECUTE A WRITE CHECK COMMAND
                            001C      ;           4. CHECK FOR ANY UNEXPECTED ERRORS
                            001C      ;           5. CHECK THAT WRITE CHECK ERROR DID SET IN IDCCSR
                            001C      ;
                            001C      ; DEBUG:
                            001C      ;
                            001C      ;           WRITE CHECK LOGIC
                            001C      ;
                            001C      ;--
                            001C
                            001C                  $DS_BGNDATA
       00000000 00000001    001C                  .LONG   1,LUN0
       00000001 00000001    0024                  .LONG   1,LUN1
       00000002 00000001    002C                  .LONG   1,LUN2
       00000003 00000001    0034                  .LONG   1,LUN3
                            003C                  $DS_ENDDATA
                            0040
                            0040                  $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
                            0042                  GET_LUN -                               ; GET LUN INFO
                            0042                    UNIT= OFST_UNIT(AP)                   ; LOGICAL UNIT
              04 AC    DD   0042                  PUSHL   OFST_UNIT(AP)
   00000000'EF    01   FB   0045                  CALLS   #1,GET_LUN$A
                            004C                  BDEVFOUND       1$                      ; IF DEV AVAILABLE GOTO 1$
          06 50   00   E2   004C                  BBSS    #0,R0,1$
             50   01   D0   0050                          MOVL    #1,R0           ; NOT AN ERROR
                0201   31   0053                          BRW     TEST_026_X      ; EXIT TEST 26
54   00000000'EF       D0   0056      1$:         MOVL    DRIVE_TBL,R4                    ; GET DRIVE INFO TABLE
        55    14 A4    D0   005D                  MOVL    OFST_TSTCYL(R4),R5              ; GET DRIVE TEST CYLINDER
        56    28 A4    D0   0061                  MOVL    OFST_SECLEN(R4),R6              ; GET BYTES PER SECTOR
```

```
          00000000'EF    D4  0065    983          CLRL    CONFIG                              ; CLEAR ANY PROGRAM FLAGS
                   53    D4  006B    984          CLRL    R3                                  ; CLEAR INDEX
    FFFFFFFF FFFFFFFF 8F 7D  006D    985 5$:      MOVQ    #^XFFFFFFFFFFFFFFFF,-
         00000000'FF43       0077    986                  @WRITE_BUFFER[R3]                   ; LOAD BUFFER WITH DATA
  E8 53    00000040 8F   F2  007D    987          AOBLSS  #64,R3,5$                           ; DO 256 WORDS
                             0085    988          $DS_BGNSUB
                             0090    989          STARS
                             0090         ;;**********************************************************************
                             0090    990  ;++
                             0090    991  ; POSITION DRIVE TO TEST CYLINDER AND VERIFY THE SEEK
                             0090    992  ;--
                             0090    993
       00000000'EF   00  FB  0090    994 10$:     CALLS   #0,SUBSYS_INIT                      ; INITIALIZE SUBSYSTEM
                1D 50    E8  0097    995          BLBS    R0,20$                              ; BRANCH IF NO ERRORS
                             009A    996          $DS_ERRHARD_S    ,L$UNIT,MSG14_SYSINIT,-
                             009A    997                           PRINT_REG,R1
                             00B0         ;;; TEST 26, SUBTEST 1, ERROR 1
                             00B7    998 20$:     $DS_CKLOOP       10$
                   00    DD  00BF    999 35$:     PUSHL   #0                                  ; PUSH HEAD TO USE
                   55    DD  00C1   1000          PUSHL   R5                                  ; PUSH DESTINATION CYLN
       00000000'EF   02  FB  00C3   1001          CALLS   #2,POSITION_HDS                     ; EXECUTE THE SEEK
                1D 50    E8  00CA   1002          BLBS    R0,40$                              ; BRANCH IF NO ERRORS
                             00CD   1003          $DS_ERRHARD_S    ,L$UNIT,@ERR_MSG,-
                             00CD   1004                           PRINT_REG,R1
                             00E3         ;;; TEST 26, SUBTEST 1, ERROR 2
                             00EA   1005 40$:     $DS_CKLOOP       35$
                             00F2   1006          $DS_ESCAPE       TEST                       ; EXIT TEST IF ERROR
          00000000'EF    D4  00FD   1007          CLRL    CONFIG                              ; CLEAR ANY PROGRAM FLAGS
                             0103   1008 50$:     $DS_ENDSUB
                             010E   1009          $DS_BGNSUB
                             0119   1010          STARS
                             0119         ;;**********************************************************************
                             0119   1011  ;++
                             0119   1012  ; EXECUTE A WRITE DATA COMMAND TO SECTOR ZERO
                             0119   1013  ;--
                             0119   1014
                   56    DD  0119   1015          PUSHL   R6                                  ; GET BYTE COUNT
          00000000'EF    7F  011B   1016          PUSHAQ  WRITE_BUFFER                        ; GET BUFFER ADDRESS
       00000000'EF   02  FB  0121   1017          CALLS   #2,$GS_SETMAP                       ; MAP UBA
                   57    D4  0128   1018          CLRL    R7                                  ; INIT REGISTER
    57    10    10   55  F0  012A   1019          INSV    R5,#16,#16,R7                       ; LOAD DESTINATION CYLINDER
                   00    DD  012F   1020 25$:     PUSHL   #0                                  ; IDCMPR DATA
                   57    DD  0131   1021          PUSHL   R7                                  ; IDCDAR (SECTOR 0)
              7E   56    CE  0133   1022          MNEGL   R6,-(SP)                            ; IDCCNT (BYTE COUNT)
                   00    DD  0136   1023          PUSHL   #0                                  ; IDCBAR DATA
                   0A    DD  0138   1024          PUSHL   #WRITE                              ; DEFINE COMMAND
       00000000'EF   05  FB  013A   1025          CALLS   #5,START_CMD                        ; EXECUTE WRITE DATA CMD
       00000000'EF   00  FB  0141   1026          CALLS   #0,STATUS_CHK                       ; CHECK FOR ANY ERRORS
                1D 50    E8  0148   1027          BLBS    R0,30$                              ; BRANCH IF NONE
                             014B   1028          $DS_ERRHARD_S    ,L$UNIT,MSG20_WRITE,-
                             014B   1029                           PRINT_REG,R1
                             0161         ;;; TEST 26, SUBTEST 2, ERROR 1
                             0168   1030 30$:     $DS_CKLOOP       25$
                             0170   1031          $DS_ENDSUB
                             017B   1032          $DS_BGNSUB
                             0186   1033          STARS
```

```
                                 0186        ;;*********************************************************************
                                 0186  1034  ;++
                                 0186  1035  ; MODIFY ONE WORD IN THE WRITE BUFFER AND EXECUTE A WRITE CHECK DATA
                                 0186  1036  ; COMMAND AND VERIFY THE WRITE CHECK ERROR IS SET (DCRC DURING A
                                 0186  1037  ; WRITE CHECK OPERATION).
                                 0186  1038  ;--
                                 0186  1039
                             3C  0186  1040           MOVZWL  #<CSR$M_ERR!CSR$M_DCK!CSR$M_CRDY!- ;
                                 0187  1041                   WRTCHK!CSR$M_DRDY>,-                ; SET EXPECTED IDCCSR REGISTER
       00000000'EF   8883 8F     0187  1042                   E.CS                                ;
         00000000'EF   58 A4  C8  018F  1043          BISL2   OFST_CSRTYP(R4),E.CS                ; OR IN EXPECTED DRIVE TYPE BIT
   00000000'EF   00000000'EF  C8  0197  1044          BISL2   P$UNIT,E.CS                         ; OR IN DRIVE NUMBER
           00000000'EF   01  CA  01A2  1045 10$:      BICL2   #1,BUFFER1                          ; CHANGE A BIT IN THE BUFFER
                         56  DD  01A9  1046           PUSHL   R6                                  ; GET BYTE COUNT
                00000000'EF  7F  01AB  1047           PUSHAQ  WRITE_BUFFER                        ; GET ADRS OF WRITE BUFFER
           00000000'EF   02  FB  01B1  1048           CALLS   #2,$GS_SETMAP                       ; MAP UBA
                         00  DD  01B8  1049 20$:      PUSHL   #0                                  ; IDCMPR DATA
                         57  DD  01BA  1050           PUSHL   R7                                  ; IDCDAR = SECTOR 0
                    7E   56  CE  01BC  1051           MNEGL   R6,-(SP)                            ; IDCCNT (BYTE COUNT)
                         00  DD  01BF  1052           PUSHL   #0                                  ; IDCBAR DATA
                         02  DD  01C1  1053           PUSHL   #WRTCHK                             ; DEFINE COMMAND
           00000000'EF   05  FB  01C3  1054           CALLS   #5,START_CMD                        ; EXECUTE WRITE CHECK CMD
           00000000'EF   02  D0  01CA  1055           MOVL    #NOCHK$M_CERR,CONFIG                ; DONT CHECK ANY CONTROLLER ERRS
           00000000'EF   00  FB  01D1  1056           CALLS   #0,STATUS_CHK                       ; CHECK FOR ANY ERRORS
                      1D 50  E8  01D8  1057           BLBS    R0,30$                              ; BRANCH IF NONE
                                 01DB  1058           $DS_ERRHARD_S    ,L$UNIT,MSG57_WCK_FRC,-
                                 01DB  1059                            PRINT_REG,R1
                                 01F1        ;;; TEST 26, SUBTEST 3, ERROR 1
                                 01F8  1060 30$:      $DS_CKLOOP       20$
   00000000'EF   00000000'EF  D1  0200  1061          CMPL    E.CS,T.CS                           ; ACTUAL = EXPECTED ?
                         33  13  020B  1062           BEQL    40$                                 ; BRANCH IF IT DOES
                                 020D  1063           $DS_ERRHARD_S    ,L$UNIT,MSG57_WCK_FRC,-
                                 020D  1064                            REG_ERRPT,#MSG23_CS_INC,-
                                 020D  1065                            E.CS,T.CS,#CSR_BITS
                                 0239        ;;; TEST 26, SUBTEST 3, ERROR 2
                                 0240  1066 40$:      $DS_CKLOOP       20$
                                 0249  1067           $DS_ENDSUB
                                 0254  1068           $DS_ENDTEST
                                 025F  1069           $DS_SBTTL <WRITE CHECK DATA TEST>,ALIGN=LONG
                                 025F                 .SBTTL  TEST 27:  WRITE CHECK DATA TEST
                                 025F                 $D2_SBTTL        027,<WRITE CHECK DATA TEST>,<LONG>
                             00000000                          .PSECT  TEST_027, LONG, NOWRT
                                 0018
```

```
                          0018
                          0018   ;++
                          0018   ;
                          0018   ; TEST DESCRIPTION:
                          0018   ;
                          0018   ; CHECK THAT ABILITY OF THE IDC SUBSYSTEM TO WRITE CHECK DATA ERROR FREE
                          0018   ; BY EXECUTING A ONE SECTOR WRITE DATA COMMAND.  THEN READ THE DATA BACK
                          0018   ; AND CHECK THAT IT IS CORRECT.  THEN EXECUTE A WRITE CHECK COMMAND AND
                          0018   ; CHECK THAT NO ERROR OCCURS.
                          0018   ;
                          0018   ; TEST STEPS:
                          0018   ;
                          0018   ; A.  SUBTEST 1
                          0018   ;
                          0018   ;          1. INITIALIZE SUBSYSTEM
                          0018   ;          2.  EXECUTE SEEK TO TEST CYLINDER
                          0018   ;          3.  IF ERROR THEN ESCAPE TEST
                          0018   ;
                          0018   ; B.  SUBTEST 2
                          0018   ;
                          0018   ;          1. INITIALIZE SUBSYSTEM
                          0018   ;          2. EXECUTE WRITE DATA COMMAND
                          0018   ;          3. CHECK FOR ANY ERRORS
                          0018   ;
                          0018   ; C.  SUBTEST 3
                          0018   ;
                          0018   ;          1. INITIALIZE SUBSYSTEM
                          0018   ;          2. EXECUTE READ DATA COMMAND
                          0018   ;          3. CHECK FOR ANY ERRORS
                          0018   ;          4. VERIFY DATA WAS WRITTEN CORRECTLY
                          0018   ;
                          0018   ; D.  SUBTEST 4
                          0018   ;
                          0018   ;          1. INITIAZLIZE SUBSYSTEM
                          0018   ;          2. EXECUTE WRITE CHECK COMMAND
                          0018   ;          3. CHECK FOR ANY ERRORS
                          0018   ;
                          0018   ; DEBUG:
                          0018   ;
                          0018   ;          WRITE CHECK LOGIC
                          0018   ;
                          0018   ;--
                          0018
                          0018
                          0018
                          0018             $DS_BGNDATA
00000000 00000001         0018             .LONG   1,LUN0
00000001 00000001         0020             .LONG   1,LUN1
00000002 00000001         0028             .LONG   1,LUN2
00000003 00000001         0030             .LONG   1,LUN3
                          0038             $DS_ENDDATA
                          003C
                          003C             $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
                          003E             GET_LUN -                              ; GET LUN INFO
                          003E               UNIT= OFST_UNIT(AP)                  ; LOGICAL UNIT
         04 AC    DD      003E             PUSHL   OFST_UNIT(AP)
```

```
       00000000'EF    01   FB  0041              CALLS   #1,GET_LUN$A
                               0048  1126        BDEVFOUND        1$                        ; IF DEV AVAILABLE GOTO 1$
             06 50    00   E2  0048              BBSS    #0,R0,1$
                50    01   D0  004C                      MOVL     #1,R0        ; NOT AN ERROR
                  0233    31   004F                      BRW      TEST_027_X   ; EXIT TEST 27
     54   00000000'EF     D0   0052  1127 1$:    MOVL    DRIVE_TBL,R4                       ; GET DRIVE INFO TABLE
             55    14 A4  D0   0059  1128        MOVL    OFST_TSTCYL(R4),R5                 ; GET TEST CYLINDER
             56    28 A4  D0   005D  1129        MOVL    OFST_SECLEN(R4),R6                 ; GET BYTES PER SECTOR
                      57  D4   0061  1130        CLRL    R7                                 ; INIT REGISTER
    57    10    10    55  F0   0063  1131        INSV    R5,#16,#16,R7                      ; SET UP DESTINATION CYLN
          00000000'EF     D4   0068  1132        CLRL    CONFIG                             ; CLEAR ANY PROGRAM FLAGS
                      53  D4   006E  1133        CLRL    R3                                 ; CLEAR INDEX
 AAAA5555 AAAA5555 8F     7D   0070  1134 5$:    MOVQ    #^XAAAA5555AAAA5555,-
       00000000'FF43           007A  1135                @WRITE_BUFFER[R3]                  ; LOAD DATA INTO BUFFER
   E8 53   00000040 8F    F2   0080  1136        AOBLSS  #64,R3,5$                          ; DO 256 WORDS
                               0088  1137        $DS_BGNSUB
                               0093  1138        STARS
                               0093       ;;*******************************************************************
                               0093  1139 ;++
                               0093  1140 ; POSITION DRIVE TO TEST CYLINDER AND VERIFY THE SEEK
                               0093  1141 ;--
                               0093  1142
       00000000'EF    00   FB  0093  1143 10$:   CALLS   #0,SUBSYS_INIT                     ; INITIALIZE SUBSYSTEM
                1D 50      E8  009A  1144        BLBS    R0,20$                             ; BRANCH IF NO ERRORS
                               009D  1145        $DS_ERRHARD_S     ,L$UNIT,MSG14_SYSINIT,-
                               009D  1146                          PRINT_REG,R1
                               00B3       ;;; TEST 27, SUBTEST 1, ERROR 1
                               00BA  1147 20$:   $DS_CKLOOP        10$
                      00   DD  00C2  1148 35$:   PUSHL   #0                                 ; PUSH HEAD TO USE
                      55   DD  00C4  1149        PUSHL   R5                                 ; PUSH DESTINATION CYLN
       00000000'EF    02   FB  00C6  1150        CALLS   #2,POSITION_HDS                    ; EXECUTE THE SEEK
                1D 50      E8  00CD  1151        BLBS    R0,40$                             ; BRANCH IF NO ERRORS
                               00D0  1152        $DS_ERRHARD_S     ,L$UNIT,@ERR_MSG,-
                               00D0  1153                          PRINT_REG,R1
                               00E6       ;;; TEST 27, SUBTEST 1, ERROR 2
                               00ED  1154 40$:   $DS_CKLOOP        35$
                               00F5  1155        $DS_ESCAPE        TEST                     ; EXIT TEST IF ERROR
          00000000'EF     D4   0100  1156        CLRL    CONFIG                             ; CLEAR ANY PROGRAM FLAGS
                               0106  1157 50$:   $DS_ENDSUB
                               0111  1158        $DS_BGNSUB
                               011C  1159        STARS
                               011C       ;;*******************************************************************
                               011C  1160 ;++
                               011C  1161 ; EXECUTE A WRITE DATA COMMAND TO SECTOR 0
                               011C  1162 ;--
                               011C  1163
                      56   DD  011C  1164        PUSHL   R6                                 ; GET BYTE COUNT
          00000000'EF     7F   011E  1165        PUSHAQ  WRITE_BUFFER                       ; GET BUFFER ADDRESS
       00000000'EF    02   FB  0124  1166        CALLS   #2,$GS_SETMAP                      ; MAP UBA
                      00   DD  012B  1167 20$:   PUSHL   #0                                 ; IDCMPR DATA
                      57   DD  012D  1168        PUSHL   R7                                 ; IDCDAR (SECTOR 0)
                7E    56   CE  012F  1169        MNEGL   R6,-(SP)                           ; IDCCNT (BYTE COUNT)
                      00   DD  0132  1170        PUSHL   #0                                 ; IDCBAR DATA
                      0A   DD  0134  1171        PUSHL   #WRITE                             ; DEFINE COMMAND
       00000000'EF    05   FB  0136  1172        CALLS   #5,START_CMD                       ; EXECUTE WRITE DATA CMD
       00000000'EF    00   FB  013D  1173        CALLS   #0,STATUS_CHK                      ; CHECK FOR ANY ERRORS
```

```
                 1D 50    E8  0144   1174            BLBS    R0,30$                          ; BRANCH IF NONE
                              0147   1175            $DS_ERRHARD_S    ,L$UNIT,MSG20_WRITE,-
                              0147   1176                             PRINT_REG,R1
                              015D          ;;; TEST 27, SUBTEST 2, ERROR 1
                              0164   1177 30$:       $DS_CKLOOP       20$
                              016C   1178            $DS_ENDSUB
                              0177   1179            $DS_BGNSUB
                              0182   1180            STARS
                              0182          ;;*******************************************************************
                              0182   1181 ;++
                              0182   1182 ; EXECUTE A READ DATA COMMAND AND VERIFY THE DATA IS CORRECT
                              0182   1183 ;--
                              0182   1184
                    56    DD  0182   1185            PUSHL   R6                                  ; GET BYTE COUNT
            00000000'EF   7F  0184   1186            PUSHAQ  READ_BUFFER                         ; GET ADRS OF READ BUFFER
   00000000'EF      02    FB  018A   1187            CALLS   #2,$GS_SETMAP                       ; MAP UBA
                    00    DD  0191   1188 20$:       PUSHL   #0                                  ; IDCMPR DATA
                    57    DD  0193   1189            PUSHL   R7                                  ; IDCDAR = SECTOR 0
             7E     56    CE  0195   1190            MNEGL   R6,-(SP)                            ; IDCCNT (BYTE COUNT)
                    00    DD  0198   1191            PUSHL   #0                                  ; IDCBAR DATA
                    0C    DD  019A   1192            PUSHL   #READ                               ; DEFINE COMMAND
   00000000'EF      05    FB  019C   1193            CALLS   #5,START_CMD                        ; EXECUTE READ DATA CMD
   00000000'EF      00    FB  01A3   1194            CALLS   #0,STATUS_CHK                       ; CHECK FOR ANY ERRORS
                 1D 50    E8  01AA   1195            BLBS    R0,30$                              ; BRANCH IF NONE
                              01AD   1196            $DS_ERRHARD_S    ,L$UNIT,MSG21_READ,-
                              01AD   1197                             PRINT_REG,R1
                              01C3          ;;; TEST 27, SUBTEST 3, ERROR 1
                              01CA   1198 30$:       $DS_CKLOOP       20$
            00000000'EF   DD  01D2   1199            PUSHL   WRITE_BUFFER                        ; GET GOOD DATA
            00000000'EF   DD  01D8   1200            PUSHL   READ_BUFFER                         ; GET DATA JUST READ
            00000000'EF   DD  01DE   1201            PUSHL   WORD_COUNT                          ; GET WORD COUNT
   00000000'EF      03    FB  01E4   1202            CALLS   #3,CMPDATA                          ; CHECK THE DATA
                 1B 50    E8  01EB   1203            BLBS    R0,50$                              ; BRANCH IF DATA OK
                              01EE   1204            $DS_ERRHARD_S    ,L$UNIT,MSG55_DAT_CP1,-
                              01EE   1205                             PRINT_DATA_ERR
                              0202          ;;; TEST 27, SUBTEST 3, ERROR 2
                              0209   1206 50$:       $DS_CKLOOP       20$
                              0211   1207            $DS_ENDSUB
                              021C   1208            $DS_BGNSUB
                              0227   1209            STARS
                              0227          ;;*******************************************************************
                              0227   1210 ;++
                              0227   1211 ; EXECUTE A WRITE CHECK DATA COMMAND AND CHECK THAT NO ERROR IS SET.
                              0227   1212 ;--
                              0227   1213
                    56    DD  0227   1214            PUSHL   R6                                  ; GET BYTE COUNT
            00000000'EF   7F  0229   1215            PUSHAQ  WRITE_BUFFER                        ; GET BUFFER ADDRESS
   00000000'EF      02    FB  022F   1216            CALLS   #2,$GS_SETMAP                       ; MAP UBA
                    00    DD  0236   1217 20$:       PUSHL   #0                                  ; IDCMPR DATA
                    57    DD  0238   1218            PUSHL   R7                                  ; IDCDAR = SECTOR 0
             7E     56    CE  023A   1219            MNEGL   R6,-(SP)                            ; IDCCNT (BYTE COUNT)
                    00    DD  023D   1220            PUSHL   #0                                  ; IDCBAR DATA
                    02    DD  023F   1221            PUSHL   #WRTCHK                             ; DEFINE COMMAND
   00000000'EF      05    FB  0241   1222            CALLS   #5,START_CMD                        ; EXECUTE WRITE CHECK CMD
   00000000'EF      00    FB  0248   1223            CALLS   #0,STATUS_CHK                       ; CHECK FOR ANY ERRORS
                 1D 50    E8  024F   1224            BLBS    R0,30$                              ; BRANCH IF NONE
```

```
                 0252  1225          $DS_ERRHARD_S    ,L$UNIT,MSG16_WRTCHK,-
                 0252  1226                            PRINT_REG,R1
                 0268        ;;; TEST 27, SUBTEST 4, ERROR 1
                 026F  1227 30$:     $DS_CKLOOP       20$
                 0277  1228          $DS_ENDSUB
                 0282  1229          $DS_ENDTEST
                 028D  1230          $DS_SBTTL <PARTIAL SECTOR DATA TRANSFER TEST>,ALIGN=LONG
                 028D                .SBTTL  TEST 28:  PARTIAL SECTOR DATA TRANSFER TEST
                 028D                $D2_SBTTL        028,<PARTIAL SECTOR DATA TRANSFER TEST>,<LONG>
             00000000                         .PSECT  TEST_028, LONG, NOWRT
                 0024
```

```
0024
0024    ;++
0024    ;
0024    ; TEST DESCRIPTION:
0024    ;
0024    ; THIS TEST WILL VERIFY THAT WITH A BYTE COUNT OF LESS THAN ONE SECTOR
0024    ; (1/2 SECTOR IS USED HERE) A WRITE DATA COMMAND WILL CAUSE A SECTOR TO
0024    ; ZERO FILLED.  ALSO CHECK THAT A PARTIAL SECTOR READ WILL ONLY TRANSFER
0024    ; THAT NUMBER OF WORDS SPECIFIED, AND THAT A PARTIAL SECTOR WRITE CHECK
0024    ; CAN BE PERFORMED ERROR FREE.
0024    ;
0024    ; TEST STEPS:
0024    ;
0024    ; A.  SUBTEST 1
0024    ;
0024    ;           1. INITIALIZE SUBSYSTEM
0024    ;           2.  EXECUTE SEEK TO TEST CYLINDER
0024    ;           3.  IF ERROR THEN ESCAPE TEST
0024    ;
0024    ; B.  SUBTEST 2
0024    ;
0024    ;           1. INITIALIZE SUBSYSTEM
0024    ;           2. EXECUTE A FULL SECTOR WRITE DATA COMMAND
0024    ;           3. CHECK FOR ANY ERRORS
0024    ;           4. EXECUTE A PARTIAL SECTOR WRITE COMMAND
0024    ;           5. CHECK FOR ANY ERRORS
0024    ;
0024    ; C.  SUBTEST 3
0024    ;
0024    ;           1. INITIALIZE SUBSYSTEM
0024    ;           2. EXECUTE A FULL SECTOR READ
0024    ;           3. CHECK FOR ANY ERRORS
0024    ;           4. VERIFY DATA AND CHECK THAT SECTOR WAS ZERO FILLED
0024    ;
0024    ; D. SUBTEST 4
0024    ;
0024    ;           1. INITIALIZE SUBSYSTEM
0024    ;           2. EXECUTE A PARTIAL READ DATA CMD
0024    ;           3. CHECK FOR ANY ERRORS
0024    ;           4. VERIFY 1ST HALF WORDS OF READ BUFFER IS CORRECT
0024    ;           5. VERIFY THAT TOP HALF OF READ BUFFER REMAINED UNCHANGED
0024    ;
0024    ; E.  SUBTEST 5
0024    ;
0024    ;           1. INITIALIZE SUBSYSTEM
0024    ;           2. EXECUTE A PARTIAL SECTOR WRITE CHECK COMMAND
0024    ;           3. CHECK FOR ANY ERRORS
0024    ;
0024    ; DEBUG:
0024    ;
0024    ;           SECTOR FORMAT CONTROL
0024    ;
0024    ;--
0024
0024
0024
```

```
                                  0024   1288             $DS_BGNDATA
        00000000 00000001         0024   1289             .LONG    1,LUN0
        00000001 00000001         002C   1290             .LONG    1,LUN1
        00000002 00000001         0034   1291             .LONG    1,LUN2
        00000003 00000001         003C   1292             .LONG    1,LUN3
                                  0044   1293             $DS_ENDDATA
                                  0048   1294
                                  0048   1295             $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
                                  004A   1296             GET_LUN -                                 ; GET LUN INFO
                                  004A   1297               UNIT= OFST_UNIT(AP)                     ; LOGICAL UNIT
                   04 AC    DD    004A                    PUSHL    OFST_UNIT(AP)
      00000000'EF     01    FB    004D                    CALLS    #1,GET_LUN$A
                                  0054   1298             BDEVFOUND        1$                       ; IF DEV AVAILABLE GOTO 1$
             06 50    00    E2    0054                    BBSS     #0,R0,1$
                50    01    D0    0058                             MOVL     #1,R0        ; NOT AN ERROR
                    03A1    31    005B                             BRW      TEST_028_X   ; EXIT TEST 28
   54    00000000'EF        D0    005E   1299 1$:         MOVL     DRIVE_TBL,R4                     ; GET DRIVE INFO TABLE
         55    14 A4        D0    0065   1300             MOVL     OFST_TSTCYL(R4),R5               ; GET TEST CYLINDER TO USE
         56    28 A4        D0    0069   1301             MOVL     OFST_SECLEN(R4),R6               ; GET BYTES PER SECTOR
                      57    D4    006D   1302             CLRL     R7                               ; INIT REGISTER
  57     10    10     55    F0    006F   1303             INSV     R5,#16,#16,R7                    ; LOAD DESTINATION CYLINDER
         58    56     02    C7    0074   1304             DIVL3    #2,R6,R8                         ; CALC 1/2 SECTOR BYTE COUNT
         00000000'EF        D4    0078   1305             CLRL     CONFIG                           ; CLEAR ANY PROGRAM FLAGS
                                  007E   1306             $DS_BGNSUB
                                  0089   1307             STARS
                                  0089        ;;********************************************************************
                                  0089   1308 ;++
                                  0089   1309 ; POSITION DRIVE TO TEST CYLINDER AND VERIFY THE SEEK
                                  0089   1310 ;__
                                  0089   1311
      00000000'EF     00    FB    0089   1312 10$:         CALLS    #0,SUBSYS_INIT                  ; INITIALIZE SUBSYSTEM
                1D 50       E8    0090   1313             BLBS     R0,20$                           ; BRANCH IF NO ERRORS
                                  0093   1314             $DS_ERRHARD_S        ,L$UNIT,MSG14_SYSINIT,-
                                  0093   1315                                  PRINT_REG,R1
                                  00A9        ;;; TEST 28, SUBTEST 1, ERROR 1
                                  00B0   1316 20$:        $DS_CKLOOP           10$
                      00    DD    00B8   1317 35$:        PUSHL    #0                               ; PUSH HEAD TO USE
                      55    DD    00BA   1318             PUSHL    R5                               ; PUSH DESTINATION CYLN
      00000000'EF     02    FB    00BC   1319             CALLS    #2,POSITION_HDS                  ; EXECUTE THE SEEK
                1D 50       E8    00C3   1320             BLBS     R0,40$                           ; BRANCH IF NO ERRORS
                                  00C6   1321             $DS_ERRHARD_S        ,L$UNIT,@ERR_MSG,-
                                  00C6   1322                                  PRINT_REG,R1
                                  00DC        ;;; TEST 28, SUBTEST 1, ERROR 2
                                  00E3   1323 40$:        $DS_CKLOOP           35$
                                  00EB   1324             $DS_ESCAPE           TEST
         00000000'EF        D4    00F6   1325             CLRL     CONFIG                           ; CLEAR ANY PROGRAM FLAGS
                                  00FC   1326 50$:        $DS_ENDSUB
                                  0107   1327             $DS_BGNSUB
                                  0112   1328             STARS
                                  0112        ;;********************************************************************
                                  0112   1329 ;++
                                  0112   1330 ; DO A FULL SECTOR WRITE DATA COMMAND TO SECTOR ZERO.
                                  0112   1331 ; THEN DO A PARTIAL SECTOR WRITE TO SECTOR ZERO.
                                  0112   1332 ;__
                                  0112   1333
                      53    D4    0112   1334             CLRL     R3                               ; CLEAR INDEX
```

```
FFFFFFFF FFFFFFFF 8F    7D   0114  1335 5$:    MOVQ    #^XFFFFFFFFFFFFFFFF,-
          00000000'FF43      011E  1336            @WRITE_BUFFER[R3]          ; LOAD BUFFER WITH DATA
   E8 53   00000040 8F  F2   0124  1337            AOBLSS  #64,R3,5$          ; DO 256 WORDS
                    56  DD   012C  1338            PUSHL   R6                 ; GET BYTE COUNT
           00000000'EF  7F   012E  1339            PUSHAQ  WRITE_BUFFER       ; GET ADDRS OF BUFFER
     00000000'EF    02  FB   0134  1340            CALLS   #2,$GS_SETMAP      ; MAP UBA
                    00  DD   013B  1341 20$:   PUSHL   #0                 ; IDCMPR DATA
                    57  DD   013D  1342            PUSHL   R7                 ; IDCDAR (SECTOR 0)
             7E     56  CE   013F  1343            MNEGL   R6,-(SP)           ; IDCCNT (BYTE COUNT)
                    00  DD   0142  1344            PUSHL   #0                 ; IDCBAR DATA
                    0A  DD   0144  1345            PUSHL   #WRITE             ; DEFINE COMMAND
     00000000'EF    05  FB   0146  1346            CALLS   #5,START_CMD       ; EXECUTE WRITE DATA CMD
     00000000'EF    00  FB   014D  1347            CALLS   #0,STATUS_CHK      ; CHECK FOR ANY ERRORS
              1D    50  E8   0154  1348            BLBS    R0,30$             ; BRANCH IF NONE
                             0157  1349            $DS_ERRHARD_S    ,L$UNIT,MSG20_WRITE,-
                             0157  1350                             PRINT_REG,R1
                             016D        ;;; TEST 28, SUBTEST 2, ERROR 1
                             0174  1351 30$:   $DS_CKLOOP      20$
                    58  DD   017C  1352            PUSHL   R8                 ; GET BYTE COUNT
           00000000'EF  7F   017E  1353            PUSHAQ  WRITE_BUFFER       ; GET ADDRS OF BUFFER
     00000000'EF    02  FB   0184  1354            CALLS   #2,$GS_SETMAP      ; MAP UBA
                    00  DD   018B  1355 35$:   PUSHL   #0                 ; IDCMPR DATA
                    57  DD   018D  1356            PUSHL   R7                 ; IDCDAR (SECTOR 0)
             7E     58  CE   018F  1357            MNEGL   R8,-(SP)           ; IDCCNT (BYTE COUNT)
                    00  DD   0192  1358            PUSHL   #0                 ; IDCBAR DATA
                    0A  DD   0194  1359            PUSHL   #WRITE             ; DEFINE COMMAND
     00000000'EF    05  FB   0196  1360            CALLS   #5,START_CMD       ; EXECUTE WRITE DATA CMD
     00000000'EF    00  FB   019D  1361            CALLS   #0,STATUS_CHK      ; CHECK FOR ANY ERRORS
              1D    50  E8   01A4  1362            BLBS    R0,40$             ; BRANCH IF NONE
                             01A7  1363            $DS_ERRHARD_S    ,L$UNIT,MSG58_PAR_WRT,-
                             01A7  1364                             PRINT_REG,R1
                             01BD        ;;; TEST 28, SUBTEST 2, ERROR 2
                             01C4  1365 40$:   $DS_CKLOOP      35$
                             01CC  1366            $DS_ENDSUB
                             01D7  1367            $DS_BGNSUB
                             01E2  1368            STARS
                             01E2        ;;*******************************************************************
                             01E2  1369  ;++
                             01E2  1370  ; EXECUTE A FULL SECTOR READ AND VERIFY THE THE SECTOR WAS ZERO
                             01E2  1371  ; FILLED.
                             01E2  1372  ;--
                             01E2  1373
              53    58  D0   01E2  1374            MOVL    R8,R3              ; SET INDEX HALF WAY INTO WRT BUF
          00000000'FF43 94   01E5  1375 5$:    CLRB    @WRITE_BUFFER[R3]  ; TOP HALF SHOULD BE ZERO NOW
   F1 53   00000200 8F  F2   01EC  1376            AOBLSS  #512,R3,5$         ; DO ALL OF UPPER HALF
                    56  DD   01F4  1377            PUSHL   R6                 ; GET BYTE COUNT
           00000000'EF  7F   01F6  1378            PUSHAQ  READ_BUFFER        ; GET ADRS OF READ BUFFER
     00000000'EF    02  FB   01FC  1379            CALLS   #2,$GS_SETMAP      ; MAP UBA
                    00  DD   0203  1380 20$:   PUSHL   #0                 ; IDCMPR DATA
                    57  DD   0205  1381            PUSHL   R7                 ; IDCDAR = SECTOR 0
             7E     56  CE   0207  1382            MNEGL   R6,-(SP)           ; IDCCNT (BYTE COUNT)
                    00  DD   020A  1383            PUSHL   #0                 ; IDCBAR DATA
                    0C  DD   020C  1384            PUSHL   #READ              ; DEFINE COMMAND
     00000000'EF    05  FB   020E  1385            CALLS   #5,START_CMD       ; EXECUTE READ DATA CMD
     00000000'EF    00  FB   0215  1386            CALLS   #0,STATUS_CHK      ; CHECK FOR ANY ERRORS
              1D    50  E8   021C  1387            BLBS    R0,30$             ; BRANCH IF NONE
```

```
                                    021F       $DS_ERRHARD_S    ,L$UNIT,MSG60_WRT_PRD,-
                                    021F                        PRINT_REG,R1
                                    0235  ;;; TEST 28, SUBTEST 3, ERROR 1
                                    023C  30$:    $DS_CKLOOP       20$
            00000000'EF      DD     0244          PUSHL   WRITE_BUFFER                  ; GET GOOD DATA
            00000000'EF      DD     024A          PUSHL   READ_BUFFER                   ; GET DATA JUST READ
            00000000'EF      DD     0250          PUSHL   WORD_COUNT                    ; GET WORD COUNT
        00000000'EF   03     FB     0256          CALLS   #3,CMPDATA                    ; CHECK THE DATA
                 1B   50     E8     025D          BLBS    R0,50$                        ; BRANCH IF DATA OK
                                    0260          $DS_ERRHARD_S    ,L$UNIT,MSG61_DAT_PAR,-
                                    0260                           PRINT_DATA_ERR
                                    0274  ;;; TEST 28, SUBTEST 3, ERROR 2
                                    027B  50$:    $DS_CKLOOP       20$
                                    0283          $DS_ENDSUB
                                    028E          $DS_BGNSUB
                                    0299          STARS
                                    0299  ;;****************************************************************
                                    0299  ;++
                                    0299  ; EXECUTE A PARTIAL SECTOR READ DATA COMMAND AND VERIFY THE DATA.
                                    0299  ; ALSO MAKE SURE ONLY THE CORRECT AMOUNT OF WORDS WERE TRANSFERED
                                    0299  ; TO MEMORY.
                                    0299  ;--
                                    0299
                 53   58     D0     0299          MOVL    R8,R3                         ; INDEX TO TOP HALF OF BUFFER
                 FF   8F     90     029C  5$:     MOVB    #^XFF,-
            00000000'FF43           029F                  @WRITE_BUFFER[R3]             ; CHANGE TOP HALF OF SECTOR
00000000'FF43  00000000'FF43  90    02A5          MOVB    @WRITE_BUFFER[R3],-
                                    02B2                  @READ_BUFFER[R3]              ; CHANGE READ BUFFER ALSO
       E2 53   00000200 8F   F2     02B2          AOBLSS  #512,R3,5$                    ; ALL ONES
                      58     DD     02BA          PUSHL   R8                            ; GET BYTE COUNT
            00000000'EF      7F     02BC          PUSHAQ  READ_BUFFER                   ; GET BUFFER ADDDRESS
        00000000'EF   02     FB     02C2          CALLS   #2,$GS_SETMAP                 ; MAP DATA PATH #1
                      00     DD     02C9  20$:    PUSHL   #0                            ; IDCMPR DATA
                      57     DD     02CB          PUSHL   R7                            ; IDCDAR = SECTOR 0
                 7E   58     CE     02CD          MNEGL   R8,-(SP)                      ; IDCCNT (BYTE COUNT)
                      00     DD     02D0          PUSHL   #0                            ; IDCBAR DATA
                      0C     DD     02D2          PUSHL   #READ                         ; DEFINE COMMAND
        00000000'EF   05     FB     02D4          CALLS   #5,START_CMD                  ; EXECUTE PARTIAL SECTOR READ
        00000000'EF   00     FB     02DB          CALLS   #0,STATUS_CHK                 ; CHECK FOR ANY ERRORS
                 1D   50     E8     02E2          BLBS    R0,30$                        ; BRANCH IF NONE
                                    02E5          $DS_ERRHARD_S    ,L$UNIT,MSG59_PAR_RD,-
                                    02E5                           PRINT_REG,R1
                                    02FB  ;;; TEST 28, SUBTEST 4, ERROR 1
                                    0302  30$:    $DS_CKLOOP       20$
            00000000'EF      DD     030A          PUSHL   WRITE_BUFFER                  ; GET GOOD DATA
            00000000'EF      DD     0310          PUSHL   READ_BUFFER                   ; GET UNKNOWN DATA
            00000000'EF      DD     0316          PUSHL   WORD_COUNT                    ; GET WORD COUNT
        00000000'EF   03     FB     031C          CALLS   #3,CMPDATA                    ; CHECK BOTTOM 64 WORDS
                 1B   50     E8     0323          BLBS    R0,40$                        ; BRANCH IF NO ERRORS
                                    0326          $DS_ERRHARD_S    ,L$UNIT,MSG62_DC_PAR,-
                                    0326                           PRINT_DATA_ERR
                                    033A  ;;; TEST 28, SUBTEST 4, ERROR 2
                                    0341  40$:    $DS_CKLOOP       20$
          00000000'EF48      9F     0349          PUSHAB  BUFFER1[R8]                   ; GET TOP HALF OF WRT BUFF
          00000000'EF48      9F     0350          PUSHAB  BUFFER2[R8]                   ; GET TOP HALF OF RD BUF
            00000000'EF      DD     0357          PUSHL   WORD_COUNT                    ; GET WORD COUNT
```

```
          00000000'EF   03    FB  035D  1439           CALLS   #3,CMPDATA                          ; CHECK TOP HALF OF BUFFERS
                   1B 50    E8  0364  1440           BLBS    R0,50$                              ; BRANCH IF NO ERRORS
                                0367  1441           $DS_ERRHARD_S    ,L$UNIT,MSG63_PRD_TMW,-
                                0367  1442                            PRINT_DATA_ERR
                                037B        ;;; TEST 28, SUBTEST 4, ERROR 3
                                0382  1443 50$:      $DS_CKLOOP       20$
                                038B  1444           $DS_ENDSUB
                                0396  1445           $DS_BGNSUB
                                03A1  1446           STARS
                                03A1       ;;*****************************************************************
                                03A1  1447 ;++
                                03A1  1448 ; EXECUTE A PARTIAL SECTOR WRITE CHECK DATA COMMAND.
                                03A1  1449 ;--
                                03A1  1450
                         58   DD  03A1  1451           PUSHL   R8                                  ; GET BYTE COUNT
                 00000000'EF  7F  03A3  1452           PUSHAQ  WRITE_BUFFER                        ; GET BUFFER ADDRESS
          00000000'EF   02    FB  03A9  1453           CALLS   #2,$GS_SETMAP                       ; MAP UBA
                         00   DD  03B0  1454 20$:      PUSHL   #0                                  ; IDCMPR DATA
                         57   DD  03B2  1455           PUSHL   R7                                  ; IDCDAR = SECTOR 0
                   7E    58   CE  03B4  1456           MNEGL   R8,-(SP)                            ; IDCCNT (BYTE COUNT)
                         00   DD  03B7  1457           PUSHL   #0                                  ; IDCBAR DATA
                         02   DD  03B9  1458           PUSHL   #WRTCHK                             ; DEFINE COMMAND
          00000000'EF   05    FB  03BB  1459           CALLS   #5,START_CMD                        ; EXECUTE WRITE CHECK CMD
          00000000'EF   00    FB  03C2  1460           CALLS   #0,STATUS_CHK                       ; CHECK FOR ANY ERRORS
                   1D 50    E8  03C9  1461           BLBS    R0,30$                              ; BRANCH IF NONE
                                03CC  1462           $DS_ERRHARD_S    ,L$UNIT,MSG64_WCK_PAR,-
                                03CC  1463                            PRINT_REG,R1
                                03E2        ;;; TEST 28, SUBTEST 5, ERROR 1
                                03E9  1464 30$:      $DS_CKLOOP       20$
                                03F1  1465           $DS_ENDSUB
                                03FC  1466           $DS_ENDTEST
                                0407  1467           $DS_SBTTL <MULTI SECTOR DATA TRANSFER TEST>,ALIGN=LONG
                                0407                 .SBTTL  TEST 29:  MULTI SECTOR DATA TRANSFER TEST
                                0407                 $D2_SBTTL        029,<MULTI SECTOR DATA TRANSFER TEST>,<LONG>
                            00000000                         .PSECT  TEST_029, LONG, NOWRT
                                0024
```

```
                                             0024
                                             0024  ;++
                                             0024  ;
                                             0024  ; TEST DESCRIPTION:
                                             0024  ;
                                             0024  ; THIS TEST WILL CHECK THAT THE IDC SUBSYSTEM CAN PERFORM DATA
                                             0024  ; TRANSFERS GREATER THANONE SECTOR BY EXECUTING A WRITE, READ, AND A
                                             0024  ; WRITE CHECK OF TWO SECTORS. ALL ERRORS ARE CHECKED
                                             0024  ; AND THE IDCDAR REGISTER IS CHECKED FOR PROPER INCREMENT.
                                             0024  ;
                                             0024  ;
                                             0024  ; TEST STEPS:
                                             0024  ;
                                             0024  ; A.  SUBTEST 1
                                             0024  ;
                                             0024  ;        1.  INITIALIZE SUBSYSTEM
                                             0024  ;        2.  POSITION DRIVE TO TEST CYLINDER
                                             0024  ;        3.  IF ERROR THEN ESCAPE TEST
                                             0024  ;
                                             0024  ; B.  SUBTEST 2
                                             0024  ;
                                             0024  ;        1.  INITIALIZE SUBSYSTEM
                                             0024  ;        2.  EXECUTE WRITE DATA COMMAND (2 SECTORS)
                                             0024  ;        3.  CHECK FOR ANY ERRORS
                                             0024  ;        4.  CHECK THAT THE IDCDAR REG INCREMENTED CORRECTLY
                                             0024  ;
                                             0024  ; C.  SUBTEST 3
                                             0024  ;
                                             0024  ;        1.  INITIALIZE SUBSYSTEM
                                             0024  ;        2.  EXECUTE READ DATA COMMAND (2 SECTORS)
                                             0024  ;        3.  CHECK FOR ANY ERRORS
                                             0024  ;        4.  CHECK THAT THE IDCDAR REG INCREMENTED CORRECTLY
                                             0024  ;        5.  VERIFY DATA IS CORRECT - REPORT ANY DESCREPENCIES
                                             0024  ;
                                             0024  ; D.  SUBTEST 4
                                             0024  ;
                                             0024  ;        1. INITIALIZE SUBSYSTEM
                                             0024  ;        2. EXECUTE WRITE CHECK COMMAND (2 SECTORS)
                                             0024  ;        3. CHECK FOR ANY ERRORS
                                             0024  ;
                                             0024  ; DEBUG:
                                             0024  ;
                                             0024  ;        SECTOR FORMAT CONTROL
                                             0024  ;        MICRO SEQUENCER
                                             0024  ;
                                             0024  ;--
                                             0024
                                             0024           $DS_BGNDATA
00000000 00000001 00000000 00000003          0024           .LONG   3,LUN0,RL02$T,HEAD_0
00000001 00000001 00000000 00000003          0034           .LONG   3,LUN0,RL02$T,HEAD_1
00000000 00000002 00000000 00000003          0044           .LONG   3,LUN0,R80$T,HEAD_0
00000000 00000001 00000001 00000003          0054           .LONG   3,LUN1,RL02$T,HEAD_0
00000001 00000001 00000001 00000003          0064           .LONG   3,LUN1,RL02$T,HEAD_1
00000000 00000002 00000001 00000003          0074           .LONG   3,LUN1,R80$T,HEAD_0
00000000 00000001 00000002 00000003          0084           .LONG   3,LUN2,RL02$T,HEAD_0
00000001 00000001 00000002 00000003          0094           .LONG   3,LUN2,RL02$T,HEAD_1
```

```
00000000 00000002 00000002 00000003  00A4  1525           .LONG   3,LUN2,R80$T,HEAD_0
00000000 00000001 00000003 00000003  00B4  1526           .LONG   3,LUN3,RL02$T,HEAD_0
00000001 00000001 00000003 00000003  00C4  1527           .LONG   3,LUN3,RL02$T,HEAD_1
00000000 00000002 00000003 00000003  00D4  1528           .LONG   3,LUN3,R80$T,HEAD_0
                                     00E4  1529           $DS_ENDDATA
                                     00E8  1530
                                     00E8  1531           $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
                                     00EA  1532           GET_LUN -                                      ; GET LUN INFO
                                     00EA  1533             UNIT= OFST_UNIT(AP)                          ; LOGICAL UNIT
                         04 AC   DD  00EA                 PUSHL   OFST_UNIT(AP)
             00000000'EF    01   FB  00ED                 CALLS   #1,GET_LUN$A
                                     00F4  1534           BDEVFOUND       5$                             ; IF DEV AVAILABLE GOTO 5$
                   06 50    00   E2  00F4                 BBSS    #0,R0,5$
                      50    01   D0  00F8                         MOVL    #1,R0           ; NOT AN ERROR
                          02E0   31  00FB                         BRW     TEST_029_X      ; EXIT TEST 29
          54    00000000'EF   D0  00FE  1535 5$:          MOVL    DRIVE_TBL,R4                           ; GET DRIVE INFO TABLE
                   08 AC    64   91  0105  1536           CMPB    OFST_DRVTYP(R4),OFST_DRIV(AP)          ; IF DRIVE TYPE = DATA TYPE
                            06   13  0109  1537           BEQL    6$                                     ; THEN GOTO 6$
                                     010B  1538           $DS_EXIT        TEST                           ; ELSE EXIT TEST
                  00000000'EF   D4  0111  1539 6$:          CLRL    CONFIG                                 ; CLEAR ANY PROGRAM FLAGS
                   55    14 A4   D0  0117  1540           MOVL    OFST_TSTCYL(R4),R5                     ; GET TEST CYLINDER TO USE
            56    28 A4    02   C5  011B  1541           MULL3   #2,OFST_SECLEN(R4),R6                  ; CALC 2 SECTOR BYTE COUNT
                            57   D4  0120  1542           CLRL    R7                                     ; INIT REGISTER
         57    10    10    55   F0  0122  1543           INSV    R5,#16,#16,R7                          ; LOAD DESTINATION CYLINDER
   57    08    08    0C AC   F0  0127  1544           INSV    OFST_SKHD(AP),#8,#8,R7                 ; LOAD DESTINATION TRACK
                                     012D  1545           $DS_BGNSUB
                                     0138  1546           STARS
                                     0138        ;;*********************************************************************
                                     0138  1547  ;++
                                     0138  1548  ; POSITION DRIVE TO TEST CYLINDER AND VERIFY THE SEEK
                                     0138  1549  ;--
                                     0138  1550
             00000000'EF    00   FB  0138  1551 10$:      CALLS   #0,SUBSYS_INIT                         ; INITIALIZE SUBSYSTEM
                      1D 50      E8  013F  1552           BLBS    R0,20$                                 ; BRANCH IF NO ERRORS
                                     0142  1553           $DS_ERRHARD_S     ,L$UNIT,MSG14_SYSINIT,-
                                     0142  1554                             PRINT_REG,R1
                                     0158        ;;; TEST 29, SUBTEST 1, ERROR 1
                                     015F  1555 20$:      $DS_CKLOOP      10$
                         0C AC   DD  0167  1556 35$:      PUSHL   OFST_SKHD(AP)                          ; PUSH HEAD TO USE
                            55   DD  016A  1557           PUSHL   R5                                     ; PUSH DESTINATION CYLN
             00000000'EF    02   FB  016C  1558           CALLS   #2,POSITION_HDS                        ; EXECUTE THE SEEK
                      1D 50      E8  0173  1559           BLBS    R0,40$                                 ; BRANCH IF NO ERRORS
                                     0176  1560           $DS_ERRHARD_S     ,L$UNIT,@ERR_MSG,-
                                     0176  1561                             PRINT_REG,R1
                                     018C        ;;; TEST 29, SUBTEST 1, ERROR 2
                                     0193  1562 40$:      $DS_CKLOOP      35$
                                     019B  1563           $DS_ESCAPE      TEST                           ; EXIT TEST IF ERROR
                  00000000'EF   D4  01A6  1564           CLRL    CONFIG                                 ; CLEAR ANY PROGRAM FLAGS
                                     01AC  1565 50$:      $DS_ENDSUB
                                     01B7  1566           $DS_BGNSUB
                                     01C2  1567           STARS
                                     01C2        ;;*********************************************************************
                                     01C2  1568  ;++
                                     01C2  1569  ; DO A WRITE DATA COMMAND TO SECTOR 0 WITH BYTE COUNT = 2 SECTORS
                                     01C2  1570  ;--
                                     01C2  1571
```

```
                    53   D4  01C2          CLRL    R3                              ; INIT INDEX
 6BD6BD6B D6BD6BD6 8F    7D  01C4  5$:     MOVQ    #^X6BD6BD6BD6BD6BD6,-
         00000000'FF43       01CE                  @WRITE_BUFFER[R3]               ; LOAD WRITE BUFFER WITH DATA
         00000000'FF43   7C  01D4          CLRQ    @READ_BUFFER[R3]                ; CLEAR READ BUFFER
    E1 53    00000080 8F F2  01DB          AOBLSS  #128,R3,5$                      ; DO 512 WORDS
                    56   DD  01E3          PUSHL   R6                              ; GET BYTE COUNT
          00000000'EF    7F  01E5          PUSHAQ  WRITE_BUFFER                    ; GET BUFFER ADDRESS
       00000000'EF  02   FB  01EB          CALLS   #2,$GS_SETMAP                   ; MAP UBA
                    00   DD  01F2  20$:    PUSHL   #0                              ; IDCMPR DATA
                    57   DD  01F4          PUSHL   R7                              ; IDCDAR (SECTOR 0)
                7E  56   CE  01F6          MNEGL   R6,-(SP)                        ; IDCCNT (BYTE COUNT)
                    00   DD  01F9          PUSHL   #0                              ; IDCBAR DATA
                    0A   DD  01FB          PUSHL   #WRITE                          ; DEFINE COMMAND
       00000000'EF  05   FB  01FD          CALLS   #5,START_CMD                    ; EXECUTE WRITE DATA CMD
       00000000'EF  00   FB  0204          CALLS   #0,STATUS_CHK                   ; CHECK FOR ANY ERRORS
                 1D 50   E8  020B          BLBS    R0,30$                          ; BRANCH IF NONE
                             020E          $DS_ERRHARD_S    ,L$UNIT,MSG20A_WRITE,-
                             020E                           PRINT_REG,R1
                             0224  ;;; TEST 29, SUBTEST 2, ERROR 1
                             022B  30$:    $DS_CKLOOP       20$
          5A    57  02   C1  0233          ADDL3   #2,R7,R10                       ; SET EXPECTED IDCDAR REGISTER
                    5A   DD  0237          PUSHL   R10                             ; GET EXPECTED DAR DATA
       00000000'EF  01   FB  0239          CALLS   #1,CHANGE_DAR$A                 ; CHANGE DAR TO EXPECTED READ BACK -
cont>  VALUE
                5A  51   D0  0240          MOVL    R1,R10                          ; LOAD NEW DAR DATA
       00000000'EF  5A   D1  0243          CMPL    R10,T.DA                        ; DID IDCDAR UPDATE OK
                    2F   13  024A          BEQL    40$                             ; BRANCH IF IT DID
                             024C          $DS_ERRHARD_S    ,L$UNIT,MSG20A_WRITE,-
                             024C                           REG_ERRPT,#MSG54_DA_UPD,-
                             024C                           R10,T.DA,DAR_MNEM
                             0274  ;;; TEST 29, SUBTEST 2, ERROR 2
                             027B  40$:    $DS_CKLOOP       20$
                             0284          $DS_ENDSUB
                             028F          $DS_BGNSUB
                             029A          STARS
                             029A  ;;*****************************************************************
                             029A  ;++
                             029A  ; EXECUTE A 2 SECTOR READ DATA COMMAND AND VERIFY THE DATA
                             029A  ;--
                             029A
                    56   DD  029A          PUSHL   R6                              ; GET BYTE COUNT
          00000000'EF    7F  029C          PUSHAQ  READ_BUFFER                     ; GET ADRS OF READ BUFFER
       00000000'EF  02   FB  02A2          CALLS   #2,$GS_SETMAP                   ; MAP UBA
                    00   DD  02A9  20$:    PUSHL   #0                              ; IDCMPR DATA
                    57   DD  02AB          PUSHL   R7                              ; IDCDAR = SECTOR 0
                7E  56   CE  02AD          MNEGL   R6,-(SP)                        ; IDCCNT (BYTE COUNT)
                    00   DD  02B0          PUSHL   #0                              ; IDCBAR DATA
                    0C   DD  02B2          PUSHL   #READ                           ; DEFINE COMMAND
       00000000'EF  05   FB  02B4          CALLS   #5,START_CMD                    ; EXECUTE READ DATA CMD
       00000000'EF  00   FB  02BB          CALLS   #0,STATUS_CHK                   ; CHECK FOR ANY ERRORS
                 1D 50   E8  02C2          BLBS    R0,30$                          ; BRANCH IF NONE
                             02C5          $DS_ERRHARD_S    ,L$UNIT,MSG21A_READ,-
                             02C5                           PRINT_REG,R1
                             02DB  ;;; TEST 29, SUBTEST 3, ERROR 1
                             02E2  30$:    $DS_CKLOOP       20$
       00000000'EF  5A   D1  02EA          CMPL    R10,T.DA                        ; DID IDCDAR UPDATE OK ?
                    2F   13  02F1          BEQL    40$                             ; BRANCH IF IT DID
```

```
                                02F3  1624          $DS_ERRHARD_S    ,L$UNIT,MSG21A_READ,-
                                02F3  1625                           REG_ERRPT,#MSG54_DA_UPD,-
                                02F3  1626                           R10,T.DA,DAR_MNEM
                                031B       ;;; TEST 29, SUBTEST 3, ERROR 2
                                0322  1627 40$:     $DS_CKLOOP       20$
          00000000'EF     DD    032A  1628          PUSHL   WRITE_BUFFER                  ; GET GOOD DATA
          00000000'EF     DD    0330  1629          PUSHL   READ_BUFFER                   ; GET DATA JUST READ
          00000000'EF     DD    0336  1630          PUSHL   WORD_COUNT                    ; GET WORD COUNT
      00000000'EF    03   FB    033C  1631          CALLS   #3,CMPDATA                    ; CHECK THE DATA
               1B    50   E8    0343  1632          BLBS    R0,50$                        ; BRANCH IF DATA OK
                                0346  1633          $DS_ERRHARD_S    ,L$UNIT,MSG65_DAT_CP2,-
                                0346  1634                           PRINT_DATA_ERR
                                035A       ;;; TEST 29, SUBTEST 3, ERROR 3
                                0361  1635 50$:     $DS_CKLOOP       20$
                                036A  1636          $DS_ENDSUB
                                0375  1637          $DS_BGNSUB
                                0380  1638          STARS
                                0380       ;;*************************************************************
                                0380  1639 ;++
                                0380  1640 ; EXECUTE A WRITE CHECK COMMAND FOR 2 SECTORS
                                0380  1641 ;--
                                0380  1642
                     56   DD    0380  1643          PUSHL   R6                            ; GET BYTE COUNT
          00000000'EF     7F    0382  1644          PUSHAQ  WRITE_BUFFER                  ; GET BUFFER ADDRESS
      00000000'EF    02   FB    0388  1645          CALLS   #2,$GS_SETMAP                 ; MAP DATA PATH #1
                     00   DD    038F  1646 20$:     PUSHL   #0                            ; IDCMPR DATA
                     57   DD    0391  1647          PUSHL   R7                            ; IDCDAR = SECTOR 0
               7E    56   CE    0393  1648          MNEGL   R6,-(SP)                      ; IDCCNT (BYTE COUNT)
                     00   DD    0396  1649          PUSHL   #0                            ; IDCBAR DATA
                     02   DD    0398  1650          PUSHL   #WRTCHK                       ; DEFINE COMMAND
      00000000'EF    05   FB    039A  1651          CALLS   #5,START_CMD                  ; EXECUTE WRITE CHECK
      00000000'EF    00   FB    03A1  1652          CALLS   #0,STATUS_CHK                 ; CHECK FOR ANY ERRORS
               1D    50   E8    03A8  1653          BLBS    R0,30$                        ; BRANCH IF NONE
                                03AB  1654          $DS_ERRHARD_S    ,L$UNIT,MSG16A_WRTCHK,-
                                03AB  1655                           PRINT_REG,R1
                                03C1       ;;; TEST 29, SUBTEST 4, ERROR 1
                                03C8  1656 30$:     $DS_CKLOOP       20$
                                03D0  1657          $DS_ENDSUB
                                03DB  1658          $DS_ENDTEST
                                03E6  1659          $DS_SBTTL <SECTOR ADDRESSING TEST>,ALIGN=LONG
                                03E6                .SBTTL  TEST 30:  SECTOR ADDRESSING TEST
                                03E6                $D2_SBTTL        030,<SECTOR ADDRESSING TEST>,<LONG>
                            00000000                        .PSECT   TEST_030, LONG, NOWRT
                                001C
```

```
                             001C
                             001C   ;++
                             001C   ;
                             001C   ; TEST DESCRIPTION:
                             001C   ;
                             001C   ; THIS TEST WILL CHECK THAT THE SECTOR BITS CAN COMPARE CORRECTLY
                             001C   ; BY WRITING ALL SECTORS OF A TRACK.  DATA FOR EACH SECTOR WILL
                             001C   ; BE ITS OWN ADDRESS.  EACH SECTOR IS THEN READ TO VERIFY THE SECTOR
                             001C   ; CONTAINS THE CORRECT DATA.  IF DATA IS NOT CORRECT THEN SECTOR BITS
                             001C   ; ARE NOT COMPARING CORRECTLY.
                             001C   ;
                             001C   ; TEST STEPS:
                             001C   ;
                             001C   ; A.  SUBTEST 1
                             001C   ;
                             001C   ;       1.  INITIALIZE SUBSYSTEM
                             001C   ;       2.  POSITION DRIVE TO TEST CYLINDER
                             001C   ;       3.  IF ERROR THEN ESCAPE TEST
                             001C   ;
                             001C   ; B.  SUBTEST 2
                             001C   ;
                             001C   ;       1.  INITIALIZE SUBSYSTEM
                             001C   ;       2.  EXECUTE WRITE DATA COMMAND
                             001C   ;       3.  CHECK FOR ANY ERRORS
                             001C   ;       4.  CHECK THAT IDCDAR INCREMENTED CORRECTLY
                             001C   ;       5.  GOTO TO STEP 2 UNTIL ALL SECTORS WRITTEN
                             001C   ;
                             001C   ; C.  SUBTEST 3
                             001C   ;
                             001C   ;       1.  INITIALIZE SUBSYSTEM
                             001C   ;       2.  EXECUTE READ DATA COMMAND
                             001C   ;       3.  CHECK FOR ANY ERRORS
                             001C   ;       4.  CHECK THAT IDCDAR REGISTER INCREMENTED CORRECTLY
                             001C   ;       5.  VERIFY DATA IS CORRECT - REPORT ANY DESCREPENCIES
                             001C   ;       6.  GOTO STEP 2 UNTIL ALL SECTORS READ
                             001C   ;
                             001C   ; DEBUG:
                             001C   ;
                             001C   ;       HEADER FOUND LOGIC
                             001C   ;
                             001C   ;--
                             001C
                             001C           $DS_BGNDATA
     00000000 00000001       001C           .LONG    1,LUN0
     00000001 00000001       0024           .LONG    1,LUN1
     00000002 00000001       002C           .LONG    1,LUN2
     00000003 00000001       0034           .LONG    1,LUN3
                             003C           $DS_ENDDATA
                             0040
                             0040           $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
                             0042           GET_LUN -                                       ; GET LUN INFO
                             0042             UNIT= OFST_UNIT(AP)                           ; LOGICAL UNIT
              04 AC     DD   0042           PUSHL    OFST_UNIT(AP)
   00000000'EF   01     FB   0045           CALLS    #1,GET_LUN$A
                             004C           BDEVFOUND        5$                             ; IF DEV AVAILABLE GOTO 5$
         06 50   00     E2   004C           BBSS     #0,R0,5$
```

```
            50     01   D0  0050                      MOVL    #1,R0            ; NOT AN ERROR
                 029F   31  0053                      BRW     TEST_030_X       ; EXIT TEST 30
     54   00000000'EF   D0  0056  5$:       MOVL    DRIVE_TBL,R4
          00000000'EF   D4  005D  6$:       CLRL    CONFIG                            ; CLEAR ANY PROGRAM FLAGS
            55  14 A4   D0  0063            MOVL    OFST_TSTCYL(R4),R5                ; GET TEST CYLINDER TO USE
            56  28 A4   D0  0067            MOVL    OFST_SECLEN(R4),R6                ; GET BYTES PER SECTOR
            5B  10 A4   D0  006B            MOVL    OFST_LSTSEC(R4),R11               ; GET LAST SECTOR
                   57   D4  006F            CLRL    R7                                ; INIT REGISTER
   57   10    10   55   F0  0071            INSV    R5,#16,#16,R7                     ; LOAD DESTINATION CYLN
                            0076            $DS_BGNSUB
                            0081            STARS
                            0081  ;;*******************************************************************
                            0081  ;++
                            0081  ; POSITION DRIVE TO TEST CYLINDER AND VERIFY THE SEEK
                            0081  ;--
                            0081
   00000000'EF     00   FB  0081  10$:      CALLS   #0,SUBSYS_INIT                    ; INITIALIZE SUBSYSTEM
              1D   50   E8  0088            BLBS    R0,20$                            ; BRANCH IF NO ERRORS
                            008B            $DS_ERRHARD_S    ,L$UNIT,MSG14_SYSINIT,-
                            008B                             PRINT_REG,R1
                            00A1  ;;; TEST 30, SUBTEST 1, ERROR 1
                            00A8  20$:      $DS_CKLOOP       10$
                   00   DD  00B0  35$:      PUSHL   #0                                ; PUSH HEAD TO USE
                   55   DD  00B2            PUSHL   R5                                ; PUSH DESTINATION CYLN
   00000000'EF     02   FB  00B4            CALLS   #2,POSITION_HDS                   ; EXECUTE THE SEEK
              1D   50   E8  00BB            BLBS    R0,40$                            ; BRANCH IF NO ERRORS
                            00BE            $DS_ERRHARD_S    ,L$UNIT,@ERR_MSG,-
                            00BE                             PRINT_REG,R1
                            00D4  ;;; TEST 30, SUBTEST 1, ERROR 2
                            00DB  40$:      $DS_CKLOOP       35$
                            00E3            $DS_ESCAPE       TEST                     ; EXIT TEST IF ERROR
          00000000'EF   D4  00EE            CLRL    CONFIG                            ; CLEAR ANY PROGRAM FLAGS
                            00F4  50$:      $DS_ENDSUB
                            00FF            $DS_BGNSUB
                            010A            STARS
                            010A  ;;*******************************************************************
                            010A  ;++
                            010A  ; DO A WRITE DATA COMMAND TO TO EACH SECTOR
                            010A  ;--
                            010A
                   58   D4  010A            CLRL    R8                                ; R8 <-- CURRENT SECTOR
                   53   D4  010C  1$:       CLRL    R3                                ; INIT INDEX
  00000000'FF43    58   B0  010E  5$:       MOVW    R8,@WRITE_BUFFER[R3]              ; LOAD BUF WITH SECTOR ADDRS
 F0 53   00000100  8F   F2  0116            AOBLSS  #256,R3,5$                        ; DO 256 WORDS
   57   08    00   58   F0  011E            INSV    R8,#0,#8,R7                       ; LOAD DESIRED SECTOR
                   56   DD  0123            PUSHL   R6                                ; GET BYTE COUNT
          00000000'EF   7F  0125            PUSHAQ  WRITE_BUFFER                      ; GET BUFFER ADDRESS
   00000000'EF     02   FB  012B            CALLS   #2,$GS_SETMAP                     ; MAP UBA
        5A    57   01   C1  0132  10$:      ADDL3   #1,R7,R10                         ; SET EXPECTED IDCDAR
                   00   DD  0136            PUSHL   #0                                ; IDCMPR DATA
                   57   DD  0138            PUSHL   R7                                ; IDCDAR = CURRENT SECTOR
              7E   56   CE  013A            MNEGL   R6,-(SP)                          ; IDCCNT (BYTE COUNT)
                   00   DD  013D            PUSHL   #0                                ; IDCBAR DATA
                   0A   DD  013F            PUSHL   #WRITE                            ; DEFINE COMMAND
   00000000'EF     05   FB  0141            CALLS   #5,START_CMD                      ; EXECUTE WRITE DATA CMD
   00000000'EF     00   FB  0148            CALLS   #0,STATUS_CHK                     ; CHECK FOR ANY ERRORS
```

```
                1D 50   E8  014F  1764          BLBS    R0,30$                          ; BRANCH IF NONE
                            0152  1765          $DS_ERRHARD_S   ,L$UNIT,MSG20_WRITE,-
                            0152  1766                          PRINT_REG,R1
                            0168          ;;; TEST 30, SUBTEST 2, ERROR 1
                            016F  1767 30$:     $DS_CKLOOP      10$
                   5A   DD  0177  1768          PUSHL   R10                             ; GET EXPECTED DAR DATA
         00000000'EF   01   FB  0179  1769      CALLS   #1,CHANGE_DAR$A                 ; CHANGE DAR TO EXPECTED READ BACK -
cont>  VALUE
               5A   51  D0  0180  1770          MOVL    R1,R10                          ; LOAD NEW DAR DATA
      00000000'EF   5A  D1  0183  1771          CMPL    R10,T.DA                        ; DID IDCDAR UPDATE OK
                   2F   13  018A  1772          BEQL    40$                             ; BRANCH IF IT DID
                            018C  1773          $DS_ERRHARD_S   ,L$UNIT,MSG20_WRITE,-
                            018C  1774                          REG_ERRPT,#MSG54_DA_UPD,-
                            018C  1775                          R10,T.DA,DAR_MNEM
                            01B4          ;;; TEST 30, SUBTEST 2, ERROR 2
                            01BB  1776 40$:     $DS_CKLOOP      10$
      FF42 58    01    5B   3D  01C4  1777      ACBW    R11,#1,R8,1$                    ; WRITE ALL SECTORS
                            01CA  1778 50$:     $DS_ENDSUB
                            01D5  1779          $DS_BGNSUB
                            01E0  1780          STARS
                            01E0          ;;*******************************************************************
                            01E0  1781 ;++
                            01E0  1782 ; EXECUTE A READ DATA COMMAND FOR EACH SECTOR AND VERIFY THE DATA
                            01E0  1783 ; READ IS EQUAL TO THE SECTOR ADDRESS.
                            01E0  1784 ;--
                            01E0  1785
                   58   D4  01E0  1786          CLRL    R8                              ; SET POINTER TO SECTOR 0
                   53   D4  01E2  1787 1$:      CLRL    R3                              ; INIT INDEX
    00000000'FF43  58   B0  01E4  1788 5$:      MOVW    R8,@WRITE_BUFFER[R3]            ; SET EXPECTED DATA
        00000000'FF43   B4  01EC  1789          CLRW    @READ_BUFFER[R3]                ; COMPLEMENT READ BUFFER
  E9 53    00000100 8F  F2  01F3  1790          AOBLSS  #256,R3,5$                      ; DO 256 WORDS
   57    08    00   58  F0  01FB  1791          INSV    R8,#0,#8,R7                     ; LOAD DESIRED SECTOR
         5A    57   01  C1  0200  1792          ADDL3   #1,R7,R10                       ; SET EXPECTED IDCDAR
                   56   DD  0204  1793          PUSHL   R6                              ; GET BYTE COUNT
          00000000'EF   7F  0206  1794          PUSHAQ  READ_BUFFER                     ; GET ADRS OF READ BUFFER
      00000000'EF   02  FB  020C  1795          CALLS   #2,$GS_SETMAP                   ; MAP UBA
                   00   DD  0213  1796 10$:     PUSHL   #0                              ; IDCMPR DATA
                   57   DD  0215  1797          PUSHL   R7                              ; IDCDAR = CURRENT SECTOR #
              7E   56   CE  0217  1798          MNEGL   R6,-(SP)                        ; IDCCNT (BYTE COUNT)
                   00   DD  021A  1799          PUSHL   #0                              ; IDCBAR DATA
                   0C   DD  021C  1800          PUSHL   #READ                           ; DEFINE COMMAND
      00000000'EF   05  FB  021E  1801          CALLS   #5,START_CMD                    ; EXECUTE READ DATA CMD
      00000000'EF   00  FB  0225  1802          CALLS   #0,STATUS_CHK                   ; CHECK FOR ANY ERRORS
                1D 50   E8  022C  1803          BLBS    R0,30$                          ; BRANCH IF NONE
                            022F  1804          $DS_ERRHARD_S   ,L$UNIT,MSG21_READ,-
                            022F  1805                          PRINT_REG,R1
                            0245          ;;; TEST 30, SUBTEST 3, ERROR 1
                            024C  1806 30$:     $DS_CKLOOP      10$
                   5A   DD  0254  1807          PUSHL   R10                             ; PUSH DAR DATA
      00000000'EF   01  FB  0256  1808          CALLS   #1,CHANGE_DAR$A                 ; SET EXPECTED DAR REGISTER
               5A   51  D0  025D  1809          MOVL    R1,R10                          ; RESTORE R10
      00000000'EF   5A  D1  0260  1810          CMPL    R10,T.DA                        ; DID IDCDAR UPDATE OK ?
                   2F   13  0267  1811          BEQL    40$                             ; BRANCH IF IT DID
                            0269  1812          $DS_ERRHARD_S   ,L$UNIT,MSG21_READ,-
                            0269  1813                          REG_ERRPT,#MSG54_DA_UPD,-
                            0269  1814                          R10,T.DA,DAR_MNEM
                            0291          ;;; TEST 30, SUBTEST 3, ERROR 2
```

```
                         0298  1815 40$:    $DS_CKLOOP        10$
           00000000'EF   DD  02A1  1816         PUSHL   WRITE_BUFFER              ; GET GOOD DATA
           00000000'EF   DD  02A7  1817         PUSHL   READ_BUFFER               ; GET DATA JUST READ
           00000000'EF   DD  02AD  1818         PUSHL   WORD_COUNT                ; GET WORD COUNT
       00000000'EF   03   FB  02B3  1819         CALLS   #3,CMPDATA                ; CHECK THE DATA
                 1B 50   E8  02BA  1820         BLBS    R0,50$                    ; BRANCH IF DATA OK
                         02BD  1821         $DS_ERRHARD_S   ,L$UNIT,MSG67_DAT_ES,-
                         02BD  1822                         PRINT_DATA_ERR
                         02D1       ;;; TEST 30, SUBTEST 3, ERROR 3
                         02D8  1823 50$:    $DS_CKLOOP        10$
     FEFB 58   01   5B   3D  02E1  1824         ACBW    R11,#1,R8,1$              ; READ ALL SECTORS
                         02E7  1825 60$:    $DS_ENDSUB
                         02F2  1826         $DS_ENDTEST
                         02FD  1827         $DS_SBTTL <MAXIMUM DATA TRANSFER TEST>,ALIGN=LONG
                         02FD               .SBTTL  TEST 31:  MAXIMUM DATA TRANSFER TEST
                         02FD               $D2_SBTTL       031,<MAXIMUM DATA TRANSFER TEST>,<LONG>
                     00000000                   .PSECT  TEST_031, LONG, NOWRT
                         0020
```

```
                                            0020
                                            0020  ;++
                                            0020  ;
                                            0020  ; TEST DESCRIPTION:
                                            0020  ;
                                            0020  ; THIS TEST WILL VERIFY THAT THE IDC SUBSYSTEM IS CAPABLE OF DOING
                                            0020  ; WRITE, READ, AND WRITE CHECK DATA FUNCTIONS WITH A MAXIMUM WORD
                                            0020  ; COUNT.
                                            0020  ;
                                            0020  ; TEST STEPS:
                                            0020  ;
                                            0020  ; A.  SUBTEST 1
                                            0020  ;
                                            0020  ;         1. INITIALIZE SUBSYSTEM
                                            0020  ;         2.  POSITION DRIVE TO TEST CYLINDER
                                            0020  ;         3.  IF ERROR THEN ESCAPE TEST
                                            0020  ;
                                            0020  ; B.  SUBTEST 2
                                            0020  ;
                                            0020  ;         1. INITIALIZE SUBSYSTEM
                                            0020  ;         2. EXECUTE WRITE DATA COMMAND
                                            0020  ;         3. CHECK FOR ANY ERRORS
                                            0020  ;
                                            0020  ; C.  SUBTEST 3
                                            0020  ;
                                            0020  ;         1. INITIALIZE SUBSYSTEM
                                            0020  ;         2. EXECUTE READ DATA COMMAND
                                            0020  ;         3. CHECK FOR ANY ERRORS
                                            0020  ;         4. VERIFY DATA WAS WRITTEN CORRECTLY
                                            0020  ;
                                            0020  ; D.  SUBTEST 4
                                            0020  ;
                                            0020  ;         1. INITIAZLIZE SUBSYSTEM
                                            0020  ;         2. EXECUTE WRITE CHECK COMMAND
                                            0020  ;         3. CHECK FOR ANY ERRORS
                                            0020  ;
                                            0020  ; DEBUG:
                                            0020  ;
                                            0020  ;--
                                            0020
                                            0020
                                            0020            $DS_BGNDATA
00000000 00000001 00000000 00000003         0020            .LONG   3,LUN0,RL02$T,HEAD_0
00000001 00000001 00000000 00000003         0030            .LONG   3,LUN0,RL02$T,HEAD_1
00000000 00000002 00000000 00000003         0040            .LONG   3,LUN0,R80$T,HEAD_0
00000000 00000001 00000001 00000003         0050            .LONG   3,LUN1,RL02$T,HEAD_0
00000001 00000001 00000001 00000003         0060            .LONG   3,LUN1,RL02$T,HEAD_1
00000000 00000002 00000001 00000003         0070            .LONG   3,LUN1,R80$T,HEAD_0
00000000 00000001 00000002 00000003         0080            .LONG   3,LUN2,RL02$T,HEAD_0
00000001 00000001 00000002 00000003         0090            .LONG   3,LUN2,RL02$T,HEAD_1
00000000 00000002 00000002 00000003         00A0            .LONG   3,LUN2,R80$T,HEAD_0
00000000 00000001 00000003 00000003         00B0            .LONG   3,LUN3,RL02$T,HEAD_0
00000001 00000001 00000003 00000003         00C0            .LONG   3,LUN3,RL02$T,HEAD_1
00000000 00000002 00000003 00000003         00D0            .LONG   3,LUN3,R80$T,HEAD_0
                                            00E0            $DS_ENDDATA
                                            00E4
```

```
                                   00E4  1885          $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
                                   00E6  1886          GET_LUN -                                  ; GET LUN INFO
                                   00E6  1887            UNIT= OFST_UNIT(AP)                      ; LOGICAL UNIT
                        04 AC  DD  00E6                PUSHL   OFST_UNIT(AP)
              00000000'EF   01  FB  00E9                CALLS   #1,GET_LUN$A
                                   00F0  1888          BDEVFOUND        1$                        ; IF DEV AVAILABLE GOTO 1$
                  06 50   00  E2  00F0                BBSS    #0,R0,1$
                     50   01  D0  00F4                        MOVL    #1,R0           ; NOT AN ERROR
                       02C5   31  00F7                        BRW     TEST_031_X      ; EXIT TEST 31
         54   00000000'EF  D0  00FA  1889 1$:     MOVL    DRIVE_TBL,R4                   ; GET DRIVE INFO TABLE
                08 AC   64  91  0101  1890          CMPB    OFST_DRVTYP(R4),OFST_DRIV(AP)   ; IF DRIVE TYPE = DATA TYPE
                        06  13  0105  1891          BEQL    4$                             ; THEN GOTO 4$
                                   0107  1892          $DS_EXIT         TEST                      ; ELSE EXIT TEST
                 00000000'EF  D4  010D  1893 4$:     CLRL    CONFIG                         ; CLEAR ANY PROGRAM FLAGS
                  55    14 A4  D0  0113  1894          MOVL    OFST_TSTCYL(R4),R5             ; GET DESTINATION CYLN
                  56    2C A4  D0  0117  1895          MOVL    OFST_TRKLEN(R4),R6             ; GET TRACK BYTE COUNT
                           57  D4  011B  1896          CLRL    R7                             ; INIT REGISTER
            57   10    10   55  F0  011D  1897          INSV    R5,#16,#16,R7                  ; LOAD DESTINATION CYLN
       57   08    08    0C AC  F0  0122  1898          INSV    OFST_SKHD(AP),#8,#8,R7         ; LOAD DESTINATION TRACK
      58   56   00000200 8F  C7  0128  1899          DIVL3   #512,R6,R8                     ; CALC NUMBER OF PAGES IN MEM
                59    58   02  C5  0130  1900          MULL3   #2,R8,R9                       ; GET NUMBER OF BUFFERS ALLOCATED
                                   0134  1901          $DS_GETBUF_S     R8,VIR_WRT_BUF,-          ; GET R8 PAGES OF WRT MEMORY
                                   0134  1902                           PHY_WRT_BUF
                                   014B  1903          $DS_BERROR       2$
                                   015B  1904          $DS_GETBUF_S     R8,VIR_RD_BUF,-           ; GET R8 PAGES OF RD MEMORY
                                   015B  1905                           PHY_RD_BUF
                                   0172  1906          $DS_BNERROR      3$
                                   0182  1907 2$:     $DS_PRINTB_S     IDCGETBUF
                                   018F  1908          $DS_ABORT        TEST                      ; NO MEMORY AVAILABLE SO DIE
                           3F  BB  0192  1909 3$:     PUSHR   #^M<R0,R1,R2,R3,R4,R5>          ; SAVE REGISTERS
    00000000'EF   000000BD 8F  D0  0194  1910          MOVL    #^XBD,WORK                     ; LOAD SOURCE DATA
 56   BD 8F   00000000'EF   01  2C  019F  1911          MOVC5   #1,WORK,#^XBD,R6,@PHY_WRT_BUF   ; LOAD BUFFER
                 00000000'FF       01A9
                           3F  BA  01AE  1912          POPR    #^M<R0,R1,R2,R3,R4,R5>          ; RESTORE REGISTERS
                                   01B0  1913          $DS_BGNSUB
                                   01BB  1914          STARS
                                   01BB       ;;*******************************************************************
                                   01BB  1915 ;++
                                   01BB  1916 ; POSITION DRIVE TO TEST CYLINDER AND VERIFY THE SEEK
                                   01BB  1917 ;--
                                   01BB  1918
              00000000'EF   00  FB  01BB  1919 10$:    CALLS   #0,SUBSYS_INIT                 ; INITIALIZE SUBSYSTEM
                       1D 50  E8  01C2  1920          BLBS    R0,20$                         ; BRANCH IF NO ERRORS
                                   01C5  1921          $DS_ERRHARD_S    ,L$UNIT,MSG14_SYSINIT,-
                                   01C5  1922                           PRINT_REG,R1
                                   01DB       ;;; TEST 31, SUBTEST 1, ERROR 1
                                   01E2  1923 20$:    $DS_CKLOOP       10$
                        0C AC  DD  01EA  1924 35$:    PUSHL   OFST_SKHD(AP)                  ; PUSH HEAD TO USE
                           55  DD  01ED  1925          PUSHL   R5                             ; PUSH DESTINATION CYLN
              00000000'EF   02  FB  01EF  1926          CALLS   #2,POSITION_HDS                ; EXECUTE THE SEEK
                       1D 50  E8  01F6  1927          BLBS    R0,40$                         ; BRANCH IF NO ERRORS
                                   01F9  1928          $DS_ERRHARD_S    ,L$UNIT,@ERR_MSG,-
                                   01F9  1929                           PRINT_REG,R1
                                   020F       ;;; TEST 31, SUBTEST 1, ERROR 2
                                   0216  1930 40$:    $DS_CKLOOP       35$
                                   021E  1931          $DS_ESCAPE       TEST                      ; EXIT TEST IF ERROR
```

```
        00000000'EF   D4  0229  1932           CLRL    CONFIG                          ; CLEAR ANY PROGRAM FLAGS
                          022F  1933 50$:      $DS_ENDSUB
                          023A  1934           $DS_BGNSUB
                          0245  1935           STARS
                          0245       ;;***********************************************************************
                          0245  1936 ;++
                          0245  1937 ; EXECUTE A WRITE DATA COMMAND WITH MAXIMUM WORD COUNT
                          0245  1938 ;--
                          0245  1939
                 56   DD  0245  1940           PUSHL   R6                              ; GET BYTE COUNT
        00000000'EF   7F  0247  1941           PUSHAQ  PHY_WRT_BUF                     ; GET BUFFER ADDRESS
   00000000'EF   02   FB  024D  1942           CALLS   #2,$GS_SETMAP                   ; MAP UBA
                 00   DD  0254  1943 20$:      PUSHL   #0                              ; IDCMPR DATA
                 57   DD  0256  1944           PUSHL   R7                              ; IDCDAR (SECTOR 0)
            7E   56   CE  0258  1945           MNEGL   R6,-(SP)                        ; IDCCNT (BYTE COUNT)
                 00   DD  025B  1946           PUSHL   #0                              ; IDCBAR DATA
                 0A   DD  025D  1947           PUSHL   #WRITE                          ; DEFINE COMMAND
   00000000'EF   05   FB  025F  1948           CALLS   #5,START_CMD                    ; EXECUTE WRITE DATA CMD
   00000000'EF   00   FB  0266  1949           CALLS   #0,STATUS_CHK                   ; CHECK FOR ANY ERRORS
            1D   50   E8  026D  1950           BLBS    R0,30$                          ; BRANCH IF NONE
                          0270  1951           $DS_ERRHARD_S     ,L$UNIT,MSG20B_WRITE,-
                          0270  1952                             PRINT_REG,R1
                          0286       ;;; TEST 31, SUBTEST 2, ERROR 1
                          028D  1953 30$:      $DS_CKLOOP        20$
                          0295  1954           $DS_ENDSUB
                          02A0  1955           $DS_BGNSUB
                          02AB  1956           STARS
                          02AB       ;;***********************************************************************
                          02AB  1957 ;++
                          02AB  1958 ; EXECUTE A READ DATA COMMAND AND VERIFY THE DATA IS CORRECT
                          02AB  1959 ; USING MAXIMUM WORD COUNT.
                          02AB  1960 ;--
                          02AB  1961
                 56   DD  02AB  1962           PUSHL   R6                              ; GET BYTE COUNT
        00000000'EF   7F  02AD  1963           PUSHAQ  PHY_RD_BUF                      ; GET ADRS OF READ BUFFER
   00000000'EF   02   FB  02B3  1964           CALLS   #2,$GS_SETMAP                   ; MAP UBA
                 00   DD  02BA  1965 20$:      PUSHL   #0                              ; IDCMPR DATA
                 57   DD  02BC  1966           PUSHL   R7                              ; IDCDAR = SECTOR 0
            7E   56   CE  02BE  1967           MNEGL   R6,-(SP)                        ; IDCCNT (BYTE COUNT)
                 00   DD  02C1  1968           PUSHL   #0                              ; IDCBAR DATA
                 0C   DD  02C3  1969           PUSHL   #READ                           ; DEFINE COMMAND
   00000000'EF   05   FB  02C5  1970           CALLS   #5,START_CMD                    ; EXECUTE READ DATA CMD
   00000000'EF   00   FB  02CC  1971           CALLS   #0,STATUS_CHK                   ; CHECK FOR ANY ERRORS
            1D   50   E8  02D3  1972           BLBS    R0,30$                          ; BRANCH IF NONE
                          02D6  1973           $DS_ERRHARD_S     ,L$UNIT,MSG21B_READ,-
                          02D6  1974                             PRINT_REG,R1
                          02EC       ;;; TEST 31, SUBTEST 3, ERROR 1
                          02F3  1975 30$:      $DS_CKLOOP        20$
        00000000'EF   DD  02FB  1976           PUSHL   PHY_WRT_BUF                     ; GET GOOD DATA
        00000000'EF   DD  0301  1977           PUSHL   PHY_RD_BUF                      ; GET DATA JUST READ
        00000000'EF   DD  0307  1978           PUSHL   WORD_COUNT                      ; GET WORD COUNT
   00000000'EF   03   FB  030D  1979           CALLS   #3,CMPDATA                      ; CHECK THE DATA
            1B   50   E8  0314  1980           BLBS    R0,50$                          ; BRANCH IF DATA OK
                          0317  1981           $DS_ERRHARD_S     ,L$UNIT,MSG68_DAT_MAX,-
                          0317  1982                             PRINT_DATA_ERR
                          032B       ;;; TEST 31, SUBTEST 3, ERROR 2
```

```
                                 0332 50$:     $DS_CKLOOP      20$
                                 033A          $DS_ENDSUB
                                 0345          $DS_BGNSUB
                                 0350          STARS
                                 0350     ;;**************************************************************
                                 0350     ;++
                                 0350     ; EXECUTE A WRITE CHECK DATA COMMAND AND CHECK THAT NO ERROR IS SET.
                                 0350     ; USE MAXIMUM WORD COUNT.
                                 0350     ;--
                                 0350
                    56     DD    0350          PUSHL   R6                          ; GET BYTE COUNT
           00000000'EF     7F    0352          PUSHAQ  PHY_WRT_BUF                 ; GET BUFFER ADDRESS
     00000000'EF    02     FB    0358          CALLS   #2,$GS_SETMAP               ; MAP UBA
                    00     DD    035F 20$:     PUSHL   #0                          ; IDCMPR DATA
                    57     DD    0361          PUSHL   R7                          ; IDCDAR = SECTOR 0
             7E     56     CE    0363          MNEGL   R6,-(SP)                    ; IDCCNT (BYTE COUNT)
                    00     DD    0366          PUSHL   #0                          ; IDCBAR DATA
                    02     DD    0368          PUSHL   #WRTCHK                     ; DEFINE COMMAND
     00000000'EF    05     FB    036A          CALLS   #5,START_CMD                ; EXECUTE WRITE CHECK CMD
     00000000'EF    00     FB    0371          CALLS   #0,STATUS_CHK               ; CHECK FOR ANY ERRORS
                 1D 50     E8    0378          BLBS    R0,30$                      ; BRANCH IF NONE
                                 037B          $DS_ERRHARD_S   ,L$UNIT,MSG16B_WRTCHK,-
                                 037B                          PRINT_REG,R1
                                 0391     ;;; TEST 31, SUBTEST 4, ERROR 1
                                 0398 30$:     $DS_CKLOOP      20$
                                 03A0          $DS_ENDSUB
                                 03AB          $DS_RELBUF_S    R9,VIR_WRT_BUF      ; RELEASE BUFFERS ALLOCATED
                                 03BC          $DS_ENDTEST
                                 03C7          $DS_ENDMOD
                                 0004          .END
```

```
$$$              = 0000015C R     08
$$E              = 00000001
$$M              = 00000005
$$N              = 00000001
$$S              = FFFFFFFF
$ENV             = 00000003  G
$ER              = FFFFFFFF
$GS_SETMAP         ********    X  07
$MO              = 00000001  G
$S$              = 0000015C R     08
$ST              = 00000000
$TN              = 0000001F
ALL              = 00000002  G
BASE_018           00000000 R     03
BASE_019           00000000 R     05
BASE_020           00000000 R     06
BASE_021           00000000 R     07
BASE_022           00000000 R     09
BASE_023           00000000 R     0A
BASE_024           00000000 R     0B
BASE_025           00000000 R     0C
BASE_026           00000000 R     0D
BASE_027           00000000 R     0E
BASE_028           00000000 R     0F
BASE_029           00000000 R     10
BASE_030           00000000 R     11
BASE_031           00000000 R     12
BIT...           = 00000028
BIT0             = 00000001
BIT1             = 00000002
BIT10            = 00000400
BIT11            = 00000800
BIT12            = 00001000
BIT13            = 00002000
BIT14            = 00004000
BIT15            = 00008000
BIT16            = 00010000
BIT17            = 00020000
BIT18            = 00040000
BIT19            = 00080000
BIT2             = 00000004
BIT20            = 00100000
BIT21            = 00200000
BIT22            = 00400000
BIT23            = 00800000
BIT24            = 01000000
BIT25            = 02000000
BIT26            = 04000000
BIT27            = 08000000
BIT28            = 10000000
BIT29            = 20000000
BIT3             = 00000008
BIT30            = 40000000
BIT31            = 80000000
BIT4             = 00000010
BIT5             = 00000020
BIT6             = 00000040
BIT7             = 00000080
BIT8             = 00000100
BIT9             = 00000200
BUFFER1            ********      X    0D
BUFFER2            ********      X    0F
CEP_FUNCTIONAL   = 00000000
CEP_REPAIR       = 00000001
CHANGE_DAR$A       ********      X    0C
CHC$_ABORT       = 00000002
CHC$_CLEAR       = 00000006
CHC$_CLRDFT      = 00000009
CHC$_DSINT       = 00000005
CHC$_ENINT       = 00000004
CHC$_INITA       = 00000000
CHC$_INITB       = 00000001
CHC$_PURGE       = 00000003
CHC$_SELF_TEST   = 0000000B
CHC$_SETDFT      = 00000008
CHC$_STATUS      = 00000007
CHC$_STOP        = 0000000A
CHI$M_CHNINT     = 00000001
CHI$M_DEVINT     = 00000002
CHI$M_IPL        = 0000007C
CHI$S_IPL        = 00000005
CHI$V_CHNINT     = 00000000
CHI$V_DEVINT     = 00000001
CHI$V_IPL        = 00000002
CHM$_FORWARD     = 00000000
CHM$_INVALIDATE  = 00000001
CHM$_MAP         = 00000002
CHM$_MFWDN       = 00000002
CHM$_MFWDNO      = 0000000A
CHM$_MFWDV       = 00000003
CHM$_MFWDVO      = 0000000B
CHM$_MREVN       = 00000006
CHM$_MREVNO      = 0000000E
CHM$_MREVV       = 00000007
CHM$_MREVVO      = 0000000F
CHM$_NFWDN       = 00000000
CHM$_NREVN       = 00000004
CHM$_OFFSET      = 00000008
CHM$_REVERSE     = 00000004
CHS$M_BADBDP     = 20000000
CHS$M_BADPAR     = 40000000
CHS$M_BUSIC      = 00800000
CHS$M_BUSINIT    = 00400000
CHS$M_BUSNXM     = 08000000
CHS$M_BUSPDN     = 01000000
CHS$M_CHNDPE     = 00000040
CHS$M_CHNERR     = 00000002
CHS$M_CHNMPE     = 00000080
CHS$M_CHPFOT     = 00000100
CHS$M_DEVBUS     = 00000010
CHS$M_DEVERR     = 00000004
CHS$M_DEVTO      = 00000020
CHS$M_ERRANY     = 0000000F
CHS$M_MBAATN     = 00100000
CHS$M_MBACPE     = 00200000
CHS$M_MBADTB     = 00040000
CHS$M_MBADTC     = 00080000
CHS$M_MBAEXC     = 00010000
CHS$M_MBANED     = 00020000
CHS$M_MBAWCKLWR= 02000000
CHS$M_MBAWCKUPR= 04000000
CHS$M_PGMERR     = 00000008
CHS$M_PGMHDE     = 00000800
CHS$M_SYSERR     = 00000001
CHS$M_SYSMEM     = 00000200
CHS$M_SYSSBI     = 00000400
CHS$M_UIE        = 10000000
CHS$V_BADBDP     = 0000001D
CHS$V_BADPAR     = 0000001E
CHS$V_BUSIC      = 00000017
CHS$V_BUSINIT    = 00000016
CHS$V_BUSNXM     = 0000001B
CHS$V_BUSPDN     = 00000018
CHS$V_CHNDPE     = 00000006
CHS$V_CHNERR     = 00000001
CHS$V_CHNMPE     = 00000007
CHS$V_CHPFOT     = 00000008
CHS$V_DEVBUS     = 00000004
CHS$V_DEVERR     = 00000002
CHS$V_DEVTO      = 00000005
CHS$V_MBAATN     = 00000014
CHS$V_MBACPE     = 00000015
CHS$V_MBADTB     = 00000012
CHS$V_MBADTC     = 00000013
CHS$V_MBAEXC     = 00000010
CHS$V_MBANED     = 00000011
CHS$V_MBAWCKLWR= 00000019
CHS$V_MBAWCKUPR= 0000001A
CHS$V_PGMERR     = 00000003
CHS$V_PGMHDE     = 0000000B
CHS$V_SYSERR     = 00000000
CHS$V_SYSMEM     = 00000009
CHS$V_SYSSBI     = 0000000A
CHS$V_UIE        = 0000001C
CMPDATA            ********      X    0C
CONFIG             ********      X    03
CSR$M_CRDY       = 00000080
CSR$M_DCK        = 00000800
CSR$M_DRDY       = 00000001
CSR$M_ECSO       = 00100000
CSR$M_ERR        = 00008000
CSR$M_HNF        = 00001000
CSR$M_NXM        = 00002000
CSR$M_OPI        = 00000400
CSR_BITS           ********      X    07
CURR_CYLN          ********      X    03
CURR_CYL_HD        ********      X    07
DAR_MNEM           ********      X    0C
```

```
DATA_018         00000024 R    03    DS$K_BLSS_M      = 0000000C  G          MSG23_CS_INC      ********   X    07
DATA_019         0000001C R    05    DS$K_BNEQU_B     = 0000000B  G          MSG30_CRDY        ********   X    0A
DATA_020         00000014 R    06    DS$K_BNEQU_M     = 00000017  G          MSG35_OPI_TO      ********   X    07
DATA_021         00000018 R    07    DS$K_BNEQ_B      = 0000000A  G          MSG36_OPI_RLY     ********   X    07
DATA_022         00000024 R    09    DS$K_BNEQ_M      = 00000016  G          MSG37_OPI_LAT     ********   X    07
DATA_023         00000024 R    0A    DS$K_BNERROR     = 00000027  G          MSG52_RD_HNF      ********   X    07
DATA_024         00000020 R    0B    DS$K_BVC_M       = 00000019  G          MSG53_WRT_NXM     ********   X    09
DATA_025         0000001C R    0C    DS$K_BVS_M       = 00000018  G          MSG54_DA_UPD      ********   X    0C
DATA_026         0000001C R    0D    DS$PRINTB          ********   X    12   MSG55_DAT_CP1     ********   X    0C
DATA_027         00000018 R    0E    DS$RELBUF          ********   X    12   MSG57_WCK_FRC     ********   X    0D
DATA_028         00000024 R    0F    DS$WAITMS          ********   X    07   MSG58_PAR_WRT     ********   X    0F
DATA_029         00000024 R    10    E.CS               ********   X    07   MSG59_PAR_RD      ********   X    0F
DATA_030         0000001C R    11    ENV$M_DOMAIN     = 00000002             MSG60_WRT_PRD     ********   X    0F
DATA_031         00000020 R    12    ENV$M_LEVEL      = 00000001             MSG61_DAT_PAR     ********   X    0F
DEFAULT        = 00000000  G         ENV$M_SUPER      = 000003FC             MSG62_DC_PAR      ********   X    0F
DRIVE_TBL        ********   X  03    ENV$S_DOMAIN     = 00000001             MSG63_PRD_TMW     ********   X    0F
DS$BGNSUB        ********   X  07    ENV$S_LEVEL      = 00000001             MSG64_WCK_PAR     ********   X    0F
DS$BRANCH        ********   X  12    ENV$S_SUPER      = 00000008             MSG65_DAT_CP2     ********   X    10
DS$BREAK         ********   X  03    ENV$V_DOMAIN     = 00000001             MSG67_DAT_ES      ********   X    11
DS$CKLOOP        ********   X  03    ENV$V_LEVEL      = 00000000             MSG68_DAT_MAX     ********   X    12
DS$ENDSUB        ********   X  07    ENV$V_SUPER      = 00000002             MSG_018           00000002 R     03
DS$ERRHARD       ********   X  03    ENV$_CPU         = 00000000             MSG_019           00000002 R     05
DS$ESCAPE        ********   X  07    ENV$_FUNCTIONAL= 00000000               MSG_020           00000002 R     06
DS$GETBUF        ********   X  12    ENV$_REPAIR      = 00000001             MSG_021           00000002 R     07
DS$K_BBC       = 0000001D  G         ENV$_SUPER       = 00000001             MSG_022           00000002 R     09
DS$K_BBCC      = 00000021  G         ENV$_SYSTEM      = 00000001             MSG_023           00000002 R     0A
DS$K_BBCCI     = 00000023  G         ERR_MSG            ********   X    03   MSG_024           00000002 R     0B
DS$K_BBCS      = 0000001F  G         GETSTAT          = 00000004             MSG_025           00000002 R     0C
DS$K_BBS       = 0000001C  G         GET_LUN$A          ********   X    03   MSG_026           00000002 R     0D
DS$K_BBSC      = 00000020  G         GET_POSITION       ********   X    03   MSG_027           00000002 R     0E
DS$K_BBSS      = 0000001E  G         GET_RANDOM         ********   X    05   MSG_028           00000002 R     0F
DS$K_BBSSI     = 00000022  G         HEAD_0           = 00000000             MSG_029           00000002 R     10
DS$K_BCC_M     = 0000001B  G         HEAD_1           = 00000001             MSG_030           00000002 R     11
DS$K_BCS_M     = 0000001A  G         IDCCSR           = 00000000             MSG_031           00000002 R     12
DS$K_BEQLU_B   = 00000005  G         IDCGETBUF          ********   X    12   NOCHK$M_CERR    = 00000002
DS$K_BEQLU_M   = 00000011  G         IDCMPR           = 00000010             NOCHK$M_CRDY    = 00000001
DS$K_BEQL_B    = 00000004  G         L$UNIT             ********   X    03   OFST_AVGTIM     = 00000080
DS$K_BEQL_M    = 00000010  G         LUN0             = 00000000             OFST_CSRTYP     = 00000058
DS$K_BERROR    = 00000026  G         LUN1             = 00000001             OFST_DEVMEM     = 0000001C
DS$K_BGEQU_B   = 00000007  G         LUN2             = 00000002             OFST_DEVNAM     = 00000020
DS$K_BGEQU_M   = 00000013  G         LUN3             = 00000003             OFST_DRIV       = 00000008
DS$K_BGEQ_B    = 00000006  G         MANUAL           = 00000001  G          OFST_DRVERR     = 0000004C
DS$K_BGEQ_M    = 00000012  G         MS108_SECTNZ       ********   X    0A   OFST_DRVRST     = 00000050
DS$K_BGTRU_B   = 00000009  G         MS109_BADCNT       ********   X    0A   OFST_DRVTYP     = 00000000
DS$K_BGTRU_M   = 00000015  G         MSG14_SYSINIT      ********   X    03   OFST_FMTLEN     = 00000030
DS$K_BGTR_B    = 00000008  G         MSG16A_WRTCHK      ********   X    10   OFST_GTSCMD     = 00000034
DS$K_BGTR_M    = 00000014  G         MSG16B_WRTCHK      ********   X    12   OFST_HDBITS     = 00000044
DS$K_BLBC      = 00000025  G         MSG16_WRTCHK       ********   X    0E   OFST_HDRDNC     = 00000054
DS$K_BLBS      = 00000024  G         MSG17_GETSTAT      ********   X    0A   OFST_HDRTMP     = 00000048
DS$K_BLEQU_B   = 00000003  G         MSG19_READH        ********   X    03   OFST_HIGRPM     = 00000074
DS$K_BLEQU_M   = 0000000F  G         MSG20A_WRITE       ********   X    10   OFST_HMBSEC     = 0000005C
DS$K_BLEQ_B    = 00000002  G         MSG20B_WRITE       ********   X    12   OFST_IGBCYL     = 00000018
DS$K_BLEQ_M    = 0000000E  G         MSG20_WRITE        ********   X    0C   OFST_LOWRPM     = 00000070
DS$K_BLSSU_B   = 00000001  G         MSG21A_READ        ********   X    10   OFST_LSTCYL     = 00000008
DS$K_BLSSU_M   = 0000000D  G         MSG21B_READ        ********   X    12   OFST_LSTSEC     = 00000010
DS$K_BLSS_B    = 00000000  G         MSG21_READ         ********   X    0C   OFST_LSTTRK     = 0000000C
```

```
OFST_MAXTIM     = 0000007C            T27_S1        00000088 RG   0E    TEST_026        00000040 RG   0D
OFST_MPBITS     = 00000040            T27_S1_X      00000106 R    0E    TEST_026_X      00000257 RG   0D
OFST_NMLSTS     = 00000038            T27_S2        00000111 RG   0E    TEST_027        0000003C RG   0E
OFST_ONETIM     = 00000078            T27_S2_X      0000016C R    0E    TEST_027_X      00000285 RG   0E
OFST_OP1STS     = 00000060            T27_S3        00000177 RG   0E    TEST_028        00000048 RG   0F
OFST_PHYU       = 00000004            T27_S3_X      00000211 R    0E    TEST_028_X      000003FF RG   0F
OFST_RANCYL     = 0000006C            T27_S4        0000021C RG   0E    TEST_029        000000E8 RG   10
OFST_SECLEN     = 00000028            T27_S4_X      00000277 R    0E    TEST_029_X      000003DE RG   10
OFST_SEKSTS     = 0000003C            T28_S1        0000007E RG   0F    TEST_030        00000040 RG   11
OFST_SKHD       = 0000000C            T28_S1_X      000000FC R    0F    TEST_030_X      000002F5 RG   11
OFST_STSDNC     = 00000064            T28_S2        00000107 RG   0F    TEST_031        000000E4 RG   12
OFST_TRKLEN     = 0000002C            T28_S2_X      000001CC R    0F    TEST_031_X      000003BF RG   12
OFST_TSTCYL     = 00000014            T28_S3        000001D7 RG   0F    VIR_RD_BUF      ********  X   12
OFST_UNIT       = 00000004            T28_S3_X      00000283 R    0F    VIR_WRT_BUF     ********  X   12
OFST_WTPSTS     = 00000068            T28_S4        0000028E RG   0F    WORD_COUNT      ********  X   0C
P$UNIT            ********  X   07    T28_S4_X      0000038B R    0F    WORK            ********  X   12
PHY_RD_BUF        ********  X   12    T28_S5        00000396 RG   0F    WRITE         = 0000000A
PHY_WRT_BUF       ********  X   12    T28_S5_X      000003F1 R    0F    WRITE_BUFFER    ********  X   09
POSITION_HDS      ********  X   03    T29_S1        0000012D RG   10    WRTCHK        = 00000002
PRINT_DATA_ERR    ********  X   0C    T29_S1_X      000001AC R    10
PRINT_REG         ********  X   03    T29_S2        000001B7 RG   10
R80$T           = 00000002            T29_S2_X      00000284 R    10
RD_IDCREGS        ********  X   07    T29_S3        0000028F RG   10
READ            = 0000000C            T29_S3_X      0000036A R    10
READ_BUFFER       ********  X   07    T29_S4        00000375 RG   10
REG_ERRPT         ********  X   07    T29_S4_X      000003D0 R    10
RL02$T          = 00000001            T30_S1        00000076 RG   11
SEP_FUNCTIONAL  = 00000002            T30_S1_X      000000F4 R    11
SEP_REPAIR      = 00000003            T30_S2        000000FF RG   11
SIZ...          = 00000001            T30_S2_X      000001CA R    11
START_CMD         ********  X   07    T30_S3        000001D5 RG   11
STATUS_CHK        ********  X   07    T30_S3_X      000002E7 R    11
SUBSYS_INIT       ********  X   03    T31_S1        000001B0 RG   12
T.CS              ********  X   07    T31_S1_X      0000022F R    12
T.DA              ********  X   0C    T31_S2        0000023A RG   12
T.MP              ********  X   0A    T31_S2_X      00000295 R    12
T21_S1            00000058 RG   07    T31_S3        000002A0 RG   12
T21_S1_X          0000010F R    07    T31_S3_X      0000033A R    12
T21_S2            0000011A RG   07    T31_S4        00000345 RG   12
T21_S2_X          000002E1 R    07    T31_S4_X      000003A0 R    12
T22_S1            00000076 RG   09    TEST_018      000000E8 RG   03
T22_S1_X          00000102 R    09    TEST_018_X    00000241 RG   03
T22_S2            0000010D RG   09    TEST_019      00000040 RG   05
T22_S2_X          00000217 R    09    TEST_019_X    00000135 RG   05
T25_S1            00000117 RG   0C    TEST_020      00000038 RG   06
T25_S1_X          00000196 R    0C    TEST_020_X    000000F8 RG   06
T25_S2            000001A1 RG   0C    TEST_021      0000003C RG   07
T25_S2_X          0000027B R    0C    TEST_021_X    000002EF RG   07
T25_S3            00000286 RG   0C    TEST_022      00000048 RG   09
T25_S3_X          00000361 R    0C    TEST_022_X    00000225 RG   09
T26_S1            00000085 RG   0D    TEST_023      00000048 RG   0A
T26_S1_X          00000103 R    0D    TEST_023_X    00000224 RG   0A
T26_S2            0000010E RG   0D    TEST_024      00000044 RG   0B
T26_S2_X          00000170 R    0D    TEST_024_X    000000E4 RG   0B
T26_S3            0000017B RG   0D    TEST_025      000000E0 RG   0C
T26_S3_X          00000249 R    0D    TEST_025_X    0000036F RG   0C
```

```
                              +----------------+
                              ! Psect synopsis !
                              +----------------+

PSECT name                Allocation           PSECT No.  Attributes
----------                ----------           ---------  ----------
.  ABS  .                 00000000  (    0.)   00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
HEADER                    00000000  (    0.)   01 (  1.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC PAGE
$ABS$                     00000000  (    0.)   02 (  2.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
TEST_018                  00000249  (  585.)   03 (  3.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
DISPATCH                  00000150  (  336.)   04 (  4.)  NOPIC   USR   CON   REL   LCL NOSHR NOEXE   RD   NOWRT NOVEC LONG
TEST_019                  0000013D  (  317.)   05 (  5.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_020                  00000100  (  256.)   06 (  6.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_021                  000002F7  (  759.)   07 (  7.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
ARGLIST                   00000168  (  360.)   08 (  8.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_022                  0000022D  (  557.)   09 (  9.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_023                  0000022C  (  556.)   0A ( 10.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_024                  000000EC  (  236.)   0B ( 11.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_025                  00000377  (  887.)   0C ( 12.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_026                  0000025F  (  607.)   0D ( 13.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_027                  0000028D  (  653.)   0E ( 14.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_028                  00000407  ( 1031.)   0F ( 15.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_029                  000003E6  (  998.)   10 ( 16.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_030                  000002FD  (  765.)   11 ( 17.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_031                  000003C7  (  967.)   12 ( 18.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
$TSTCNT                   00000004  (    4.)   13 ( 19.)  NOPIC   USR   OVR   REL   LCL NOSHR NOEXE   RD   NOWRT NOVEC LONG
```

```
                                        +-------------------------+
                                        ! Symbol Cross Reference !
                                        +-------------------------+

SYMBOL           VALUE           DEFINITION      REFERENCES...
------           -----           ----------      -------------
$$$              =0000015C-R     1985   (27)     1009   (17)     1032   (17)     1137   (19)     1158   (19)     1179   (19)
                                                 1208   (19)     1306   (21)     1327   (21)     1367   (21)     1400   (21)
                                                 1445   (21)     1545   (23)     1566   (23)     1602   (23)     1637   (23)
                                                 1721   (25)     1742   (25)     1779   (25)     1913   (27)     1934   (27)
                                                 1955   (27)     1985   (27)     385    (8)      414    (8)      538    (10)
                                                 562    (10)     834    (15)     855    (15)     894    (15)     988    (17)
$$E              =00000001       1985   (27)     1008   (17)     1031   (17)     1067   (17)     1157   (19)     1178   (19)
                                                 1207   (19)     1228   (19)     1326   (21)     1366   (21)     1399   (21)
                                                 1444   (21)     1465   (21)     1565   (23)     1601   (23)     1636   (23)
                                                 1657   (23)     1741   (25)     1778   (25)     1825   (25)     1933   (27)
                                                 1954   (27)     1984   (27)     2006   (27)     413    (8)      479    (8)
                                                 561    (10)     604    (10)     854    (15)     893    (15)     928    (15)
$$M              =00000005       2004   (27)     1004   (17)     1029   (17)     1059   (17)     1065   (17)     1123   (19)
                                                 1146   (19)     1153   (19)     1176   (19)     118    (4)      1197   (19)
                                                 1205   (19)     1226   (19)     1295   (21)     1315   (21)     1322   (21)
                                                 1350   (21)     1364   (21)     1389   (21)     1397   (21)     140    (4)
                                                 1426   (21)     1434   (21)     1442   (21)     146    (4)      1463   (21)
                                                 1531   (23)     155    (4)      1554   (23)     1561   (23)     1589   (23)
                                                 1599   (23)     1620   (23)     1626   (23)     1634   (23)     1655   (23)
                                                 166    (4)      1710   (25)     1730   (25)     1737   (25)     174    (4)
                                                 1766   (25)     1775   (25)     1805   (25)     1814   (25)     1822   (25)
                                                 1885   (27)     1922   (27)     1929   (27)     1952   (27)     1974   (27)
                                                 1982   (27)     2004   (27)     216    (5)      231    (5)      247    (5)
                                                 260    (5)      299    (6)      313    (6)      333    (6)      380    (8)
                                                 394    (8)      400    (8)      409    (8)      430    (8)      448    (8)
                                                 455    (8)      467    (8)      473    (8)      478    (8)      529    (10)
                                                 547    (10)     554    (10)     574    (10)     589    (10)     599    (10)
                                                 641    (12)     653    (12)     672    (12)     681    (12)     693    (12)
                                                 706    (12)     742    (13)     755    (13)     762    (13)     823    (15)
                                                 843    (15)     850    (15)     881    (15)     891    (15)     912    (15)
                                                 918    (15)     926    (15)     976    (17)     997    (17)
$$N              =00000001       2004   (27)     1004   (17)     1029   (17)     1059   (17)     1065   (17)     1146   (19)
                                                 1153   (19)     1176   (19)     1197   (19)     1205   (19)     1226   (19)
                                                 1315   (21)     1322   (21)     1350   (21)     1364   (21)     1389   (21)
                                                 1397   (21)     140    (4)      1426   (21)     1434   (21)     1442   (21)
                                                 146    (4)      1463   (21)     155    (4)      1554   (23)     1561   (23)
                                                 1589   (23)     1599   (23)     1620   (23)     1626   (23)     1634   (23)
                                                 1655   (23)     166    (4)      1730   (25)     1737   (25)     174    (4)
                                                 1766   (25)     1775   (25)     1805   (25)     1814   (25)     1822   (25)
                                                #-1907  (27)     1922   (27)     1929   (27)     1952   (27)     1974   (27)
                                                 1982   (27)     2004   (27)     231    (5)      247    (5)      260    (5)
                                                 313    (6)      333    (6)      394    (8)      400    (8)      409    (8)
                                                 430    (8)      448    (8)      455    (8)      467    (8)      473    (8)
                                                 478    (8)      547    (10)     554    (10)     574    (10)     589    (10)
                                                 599    (10)     63     (2)      653    (12)     672    (12)     681    (12)
                                                 693    (12)     706    (12)     755    (13)     762    (13)     843    (15)
                                                 850    (15)     881    (15)     891    (15)     912    (15)     918    (15)
                                                 926    (15)     997    (17)
$$S              =FFFFFFFF       2008   (27)     1009   (17)     1032   (17)     1069   (17)     1123   (19)     1137   (19)
```

```
                                                1158   (19)      1179   (19)      118    (4)       1208   (19)      1230   (19)
                                                1295   (21)      1306   (21)      1327   (21)      1367   (21)      1400   (21)
                                                1445   (21)      1467   (21)      1531   (23)      1545   (23)      1566   (23)
                                                1602   (23)      1637   (23)      1659   (23)      1710   (25)      1721   (25)
                                                1742   (25)      1779   (25)      181    (4)       1827   (25)      1885   (27)
                                                1913   (27)      1934   (27)      1955   (27)      1985   (27)      216    (5)
                                                265    (5)       299    (6)       338    (6)       380    (8)       385    (8)
                                                414    (8)       481    (8)       529    (10)      538    (10)      562    (10)
                                                606    (10)      641    (12)      71     (2)       711    (12)      742    (13)
                                                767    (13)      823    (15)      834    (15)      855    (15)      894    (15)
                                                930    (15)      976    (17)      988    (17)
$ENV              =00000003     61     (2)
$ER               =FFFFFFFF     2008   (27)    #-1004  (17)      1009   (17)     #-1029  (17)      1032   (17)     #-1059  (17)
                                               #-1065  (17)      1137   (19)     #-1146  (19)     #-1153  (19)      1158   (19)
                                               #-1176  (19)      1179   (19)     #-1197  (19)     #-1205  (19)      1208   (19)
                                               #-1226  (19)      1306   (21)     #-1315  (21)     #-1322  (21)      1327   (21)
                                               #-1350  (21)     #-1364  (21)      1367   (21)     #-1389  (21)     #-1397  (21)
                                               #-140   (4)       1400   (21)     #-1426  (21)     #-1434  (21)     #-1442  (21)
                                                1445   (21)     #-146   (4)      #-1463  (21)      1545   (23)     #-155   (4)
                                               #-1554  (23)     #-1561  (23)      1566   (23)     #-1589  (23)     #-1599  (23)
                                                1602   (23)     #-1620  (23)     #-1626  (23)     #-1634  (23)      1637   (23)
                                               #-1655  (23)     #-166   (4)       1721   (25)     #-1730  (25)     #-1737  (25)
                                               #-174   (4)       1742   (25)     #-1766  (25)     #-1775  (25)      1779   (25)
                                               #-1805  (25)     #-1814  (25)     #-1822  (25)     #-1903  (27)     #-1906  (27)
                                                1913   (27)     #-1922  (27)     #-1929  (27)      1934   (27)     #-1952  (27)
                                                1955   (27)     #-1974  (27)     #-1982  (27)      1985   (27)     #-2004  (27)
                                               #-231   (5)      #-247   (5)      #-260   (5)      #-313   (6)      #-333   (6)
                                                385    (8)      #-394   (8)      #-400   (8)      #-409   (8)       414    (8)
                                               #-430   (8)      #-448   (8)      #-455   (8)      #-467   (8)      #-473   (8)
                                               #-478   (8)       538    (10)     #-547   (10)     #-554   (10)      562    (10)
                                               #-574   (10)     #-589   (10)     #-599   (10)     #-653   (12)     #-672   (12)
                                               #-681   (12)     #-693   (12)     #-706   (12)     #-755   (13)     #-762   (13)
                                                834    (15)     #-843   (15)     #-850   (15)      855    (15)     #-881   (15)
                                               #-891   (15)      894    (15)     #-912   (15)     #-918   (15)     #-926   (15)
                                                988    (17)     #-997   (17)
$GS_SETMAP         00000000-XR                  1017   (17)      1048   (17)      1166   (19)      1187   (19)      1216   (19)
                                                1340   (21)      1354   (21)      1379   (21)      1416   (21)      1453   (21)
                                                1579   (23)      1610   (23)      1645   (23)      1755   (25)      1795   (25)
                                                1942   (27)      1964   (27)      1994   (27)      434    (8)       578    (10)
                                                868    (15)      902    (15)
$MO               =00000001     2009   (27)     2009   (27)
$S$               =0000015C-R   1985   (27)     1008   (17)      1009   (17)      1031   (17)      1032   (17)      1067   (17)
                                                1137   (19)      1157   (19)      1158   (19)      1178   (19)      1179   (19)
                                                1207   (19)      1208   (19)      1228   (19)      1306   (21)      1326   (21)
                                                1327   (21)      1366   (21)      1367   (21)      1399   (21)      1400   (21)
                                                1444   (21)      1445   (21)      1465   (21)      1545   (23)      1565   (23)
                                                1566   (23)      1601   (23)      1602   (23)      1636   (23)      1637   (23)
                                                1657   (23)      1721   (25)      1741   (25)      1742   (25)      1778   (25)
                                                1779   (25)      1825   (25)      1913   (27)      1933   (27)      1934   (27)
                                                1954   (27)      1955   (27)      1984   (27)      1985   (27)      2006   (27)
                                                385    (8)       413    (8)       414    (8)       479    (8)       538    (10)
                                                561    (10)      562    (10)      604    (10)      834    (15)      854    (15)
                                                855    (15)      893    (15)      894    (15)      928    (15)      988    (17)
$ST               =00000000     2008   (27)     1004   (17)      1005   (17)      1008   (17)      1009   (17)      1029   (17)
                                                1030   (17)      1031   (17)      1032   (17)      1059   (17)      1060   (17)
                                                1065   (17)      1066   (17)      1067   (17)      1068   (17)      1123   (19)
```

```
                                            1137  (19)    1146  (19)    1147  (19)    1153  (19)    1154  (19)
                                            1157  (19)    1158  (19)    1176  (19)    1177  (19)    1178  (19)
                                            1179  (19)    118   (4)     1197  (19)    1198  (19)    1205  (19)
                                            1206  (19)    1207  (19)    1208  (19)    1226  (19)    1227  (19)
                                            1228  (19)    1229  (19)    1295  (21)    1306  (21)    1315  (21)
                                            1316  (21)    1322  (21)    1323  (21)    1326  (21)    1327  (21)
                                            1350  (21)    1351  (21)    1364  (21)    1365  (21)    1366  (21)
                                            1367  (21)    1389  (21)    1390  (21)    1397  (21)    1398  (21)
                                            1399  (21)    140   (4)     1400  (21)    141   (4)     1426  (21)
                                            1427  (21)    1434  (21)    1435  (21)    1442  (21)    1443  (21)
                                            1444  (21)    1445  (21)    146   (4)     1463  (21)    1464  (21)
                                            1465  (21)    1466  (21)    147   (4)     1531  (23)    1545  (23)
                                            155   (4)     1554  (23)    1555  (23)    156   (4)     1561  (23)
                                            1562  (23)    1565  (23)    1566  (23)    1589  (23)    1590  (23)
                                            1599  (23)    1600  (23)    1601  (23)    1602  (23)    1620  (23)
                                            1621  (23)    1626  (23)    1627  (23)    1634  (23)    1635  (23)
                                            1636  (23)    1637  (23)    1655  (23)    1656  (23)    1657  (23)
                                            1658  (23)    166   (4)     167   (4)     1710  (25)    1721  (25)
                                            1730  (25)    1731  (25)    1737  (25)    1738  (25)    174   (4)
                                            1741  (25)    1742  (25)    175   (4)     1766  (25)    1767  (25)
                                            1775  (25)    1776  (25)    1778  (25)    1779  (25)    180   (4)
                                            1805  (25)    1806  (25)    1814  (25)    1815  (25)    1822  (25)
                                            1823  (25)    1825  (25)    1826  (25)    1885  (27)    1913  (27)
                                            1922  (27)    1923  (27)    1929  (27)    1930  (27)    1933  (27)
                                            1934  (27)    1952  (27)    1953  (27)    1954  (27)    1955  (27)
                                            1974  (27)    1975  (27)    1982  (27)    1983  (27)    1984  (27)
                                            1985  (27)    2004  (27)    2005  (27)    2006  (27)    2008  (27)
                                            216   (5)     231   (5)     232   (5)     247   (5)     248   (5)
                                            260   (5)     261   (5)     264   (5)     299   (6)     313   (6)
                                            314   (6)     333   (6)     334   (6)     337   (6)     380   (8)
                                            385   (8)     394   (8)     395   (8)     400   (8)     401   (8)
                                            409   (8)     410   (8)     413   (8)     414   (8)     430   (8)
                                            431   (8)     448   (8)     449   (8)     455   (8)     456   (8)
                                            467   (8)     468   (8)     473   (8)     474   (8)     478   (8)
                                            479   (8)     480   (8)     529   (10)    538   (10)    547   (10)
                                            548   (10)    554   (10)    555   (10)    561   (10)    562   (10)
                                            574   (10)    575   (10)    589   (10)    590   (10)    599   (10)
                                            600   (10)    604   (10)    605   (10)    641   (12)    653   (12)
                                            654   (12)    672   (12)    673   (12)    681   (12)    682   (12)
                                            693   (12)    694   (12)    706   (12)    707   (12)    710   (12)
                                            742   (13)    755   (13)    756   (13)    762   (13)    763   (13)
                                            766   (13)    823   (15)    834   (15)    843   (15)    844   (15)
                                            850   (15)    851   (15)    854   (15)    855   (15)    881   (15)
                                            882   (15)    891   (15)    892   (15)    893   (15)    894   (15)
                                            912   (15)    913   (15)    918   (15)    919   (15)    926   (15)
                                            927   (15)    928   (15)    929   (15)    976   (17)    988   (17)
                                            997   (17)    998   (17)
$TN              =0000001F     2009   (27)  1004  (17)    1006  (17)    1029  (17)    103   (4)     1059  (17)
                                            1065  (17)    1068  (17)    1069  (17)    1116  (19)    1123  (19)
                                            1126  (19)    1146  (19)    1153  (19)    1155  (19)    1176  (19)
                                            118   (4)     1197  (19)    1205  (19)    121   (4)     1226  (19)
                                            1229  (19)    1230  (19)    125   (4)     1288  (21)    1295  (21)
                                            1298  (21)    1315  (21)    1322  (21)    1324  (21)    1350  (21)
                                            1364  (21)    1389  (21)    1397  (21)    140   (4)     1426  (21)
                                            1434  (21)    1442  (21)    146   (4)     1463  (21)    1466  (21)
                                            1467  (21)    1516  (23)    1531  (23)    1534  (23)    1538  (23)
```

```
                                          155    (4)      1554   (23)     1561   (23)     1563   (23)     1589   (23)
                                          159    (4)      1599   (23)     1620   (23)     1626   (23)     1634   (23)
                                          1655   (23)     1658   (23)     1659   (23)     166    (4)      1703   (25)
                                          1710   (25)     1713   (25)     1730   (25)     1737   (25)     1739   (25)
                                          174    (4)      1766   (25)     1775   (25)     180    (4)      1805   (25)
                                          181    (4)      1814   (25)     1822   (25)     1826   (25)     1827   (25)
                                          1870   (27)     1885   (27)     1888   (27)     1892   (27)     1922   (27)
                                          1929   (27)     1931   (27)     1952   (27)     1974   (27)     1982   (27)
                                          2004   (27)     2008   (27)     2009   (27)     209    (5)      216    (5)
                                          219    (5)      223    (5)      231    (5)      247    (5)      260    (5)
                                          264    (5)      265    (5)      292    (6)      299    (6)      302    (6)
                                          313    (6)      333    (6)      337    (6)      338    (6)      373    (8)
                                          380    (8)      383    (8)      394    (8)      400    (8)      409    (8)
                                          411    (8)      430    (8)      448    (8)      450    (8)      455    (8)
                                          467    (8)      473    (8)      478    (8)      480    (8)      481    (8)
                                          522    (10)     529    (10)     532    (10)     547    (10)     554    (10)
                                          556    (10)     560    (10)     574    (10)     589    (10)     599    (10)
                                          605    (10)     606    (10)     634    (12)     641    (12)     644    (12)
                                          648    (12)     653    (12)     672    (12)     674    (12)     681    (12)
                                          683    (12)     693    (12)     695    (12)     706    (12)     708    (12)
                                          71     (2)      710    (12)     711    (12)     735    (13)     742    (13)
                                          745    (13)     749    (13)     755    (13)     762    (13)     766    (13)
                                          767    (13)     808    (15)     823    (15)     826    (15)     830    (15)
                                          843    (15)     850    (15)     852    (15)     881    (15)     891    (15)
                                          912    (15)     918    (15)     926    (15)     929    (15)     930    (15)
                                          969    (17)     976    (17)     979    (17)     997    (17)
ALL             =00000002   63     (2)    1123   (19)     118    (4)      1295   (21)     1531   (23)     1710   (25)
                                          1885   (27)     216    (5)      299    (6)      380    (8)      529    (10)
                                          641    (12)     742    (13)     823    (15)     976    (17)
BASE_018        00000000-R  71     (2)    118    (4)
BASE_019        00000000-R  181    (4)    216    (5)
BASE_020        00000000-R  265    (5)    299    (6)
BASE_021        00000000-R  338    (6)    380    (8)
BASE_022        00000000-R  481    (8)    529    (10)
BASE_023        00000000-R  606    (10)   641    (12)
BASE_024        00000000-R  711    (12)   742    (13)
BASE_025        00000000-R  767    (13)   823    (15)
BASE_026        00000000-R  930    (15)   976    (17)
BASE_027        00000000-R  1069   (17)   1123   (19)
BASE_028        00000000-R  1230   (19)   1295   (21)
BASE_029        00000000-R  1467   (21)   1531   (23)
BASE_030        00000000-R  1659   (23)   1710   (25)
BASE_031        00000000-R  1827   (25)   1885   (27)
BIT...          =00000028   1906   (27)   1903   (27)     1906   (27)     61     (2)      65     (2)      66     (2)
BIT0            =00000001   62     (2)
BIT1            =00000002   62     (2)
BIT10           =00000400   62     (2)
BIT11           =00000800   62     (2)
BIT12           =00001000   62     (2)
BIT13           =00002000   62     (2)
BIT14           =00004000   62     (2)
BIT15           =00008000   62     (2)
BIT16           =00010000   62     (2)
BIT17           =00020000   62     (2)
BIT18           =00040000   62     (2)
BIT19           =00080000   62     (2)
```

```
BIT2            =00000004       62      (2)
BIT20           =00100000       62      (2)
BIT21           =00200000       62      (2)
BIT22           =00400000       62      (2)
BIT23           =00800000       62      (2)
BIT24           =01000000       62      (2)
BIT25           =02000000       62      (2)
BIT26           =04000000       62      (2)
BIT27           =08000000       62      (2)
BIT28           =10000000       62      (2)
BIT29           =20000000       62      (2)
BIT3            =00000008       62      (2)
BIT30           =40000000       62      (2)
BIT31           =80000000       62      (2)
BIT4            =00000010       62      (2)
BIT5            =00000020       62      (2)
BIT6            =00000040       62      (2)
BIT7            =00000080       62      (2)
BIT8            =00000100       62      (2)
BIT9            =00000200       62      (2)
BUFFER1          00000000-XR                    #-1045   (17)      1436    (21)
BUFFER2          00000000-XR                     1437    (21)
CEP_FUNCTIONAL  =00000000       61      (2)
CEP_REPAIR      =00000001       61      (2)
CHANGE_DAR$A     00000000-XR                     1593    (23)      1769    (25)      1808    (25)      885     (15)
CHC$_ABORT      =00000002       65      (2)
CHC$_CLEAR      =00000006       65      (2)
CHC$_CLRDFT     =00000009       65      (2)
CHC$_DSINT      =00000005       65      (2)
CHC$_ENINT      =00000004       65      (2)
CHC$_INITA      =00000000       65      (2)
CHC$_INITB      =00000001       65      (2)
CHC$_PURGE      =00000003       65      (2)
CHC$_SELF_TEST  =0000000B       65      (2)
CHC$_SETDFT     =00000008       65      (2)
CHC$_STATUS     =00000007       65      (2)
CHC$_STOP       =0000000A       65      (2)
CHI$M_CHNINT    =00000001       65      (2)
CHI$M_DEVINT    =00000002       65      (2)
CHI$M_IPL       =0000007C       65      (2)
CHI$S_IPL       =00000005       65      (2)
CHI$V_CHNINT    =00000000       65      (2)
CHI$V_DEVINT    =00000001       65      (2)
CHI$V_IPL       =00000002       65      (2)
CHM$_FORWARD    =00000000       65      (2)      65      (2)
CHM$_INVALIDATE =00000001       65      (2)      65      (2)
CHM$_MAP        =00000002       65      (2)      65      (2)
CHM$_MFWDN      =00000002       65      (2)
CHM$_MFWDNO     =0000000A       65      (2)
CHM$_MFWDV      =00000003       65      (2)
CHM$_MFWDVO     =0000000B       65      (2)
CHM$_MREVN      =00000006       65      (2)
CHM$_MREVNO     =0000000E       65      (2)
CHM$_MREVV      =00000007       65      (2)
CHM$_MREVVO     =0000000F       65      (2)
CHM$_NFWDN      =00000000       65      (2)
```

```
CHM$_NREVN      =00000004    65       (2)
CHM$_OFFSET     =00000008    65       (2)       65       (2)
CHM$_REVERSE    =00000004    65       (2)       65       (2)
CHS$M_BADBDP    =20000000    65       (2)
CHS$M_BADPAR    =40000000    65       (2)
CHS$M_BUSIC     =00800000    65       (2)
CHS$M_BUSINIT   =00400000    65       (2)
CHS$M_BUSNXM    =08000000    65       (2)
CHS$M_BUSPDN    =01000000    65       (2)
CHS$M_CHNDPE    =00000040    65       (2)
CHS$M_CHNERR    =00000002    65       (2)       65       (2)
CHS$M_CHNMPE    =00000080    65       (2)
CHS$M_CHPFOT    =00000100    65       (2)
CHS$M_DEVBUS    =00000010    65       (2)
CHS$M_DEVERR    =00000004    65       (2)       65       (2)
CHS$M_DEVTO     =00000020    65       (2)
CHS$M_ERRANY    =0000000F    65       (2)
CHS$M_MBAATN    =00100000    65       (2)
CHS$M_MBACPE    =00200000    65       (2)
CHS$M_MBADTB    =00040000    65       (2)
CHS$M_MBADTC    =00080000    65       (2)
CHS$M_MBAEXC    =00010000    65       (2)
CHS$M_MBANED    =00020000    65       (2)
CHS$M_MBAWCKLWR=02000000     65       (2)
CHS$M_MBAWCKUPR=04000000     65       (2)
CHS$M_PGMERR    =00000008    65       (2)       65       (2)
CHS$M_PGMHDE    =00000800    65       (2)
CHS$M_SYSERR    =00000001    65       (2)       65       (2)
CHS$M_SYSMEM    =00000200    65       (2)
CHS$M_SYSSBI    =00000400    65       (2)
CHS$M_UIE       =10000000    65       (2)
CHS$V_BADBDP    =0000001D    65       (2)
CHS$V_BADPAR    =0000001E    65       (2)
CHS$V_BUSIC     =00000017    65       (2)
CHS$V_BUSINIT   =00000016    65       (2)
CHS$V_BUSNXM    =0000001B    65       (2)
CHS$V_BUSPDN    =00000018    65       (2)
CHS$V_CHNDPE    =00000006    65       (2)
CHS$V_CHNERR    =00000001    65       (2)
CHS$V_CHNMPE    =00000007    65       (2)
CHS$V_CHPFOT    =00000008    65       (2)
CHS$V_DEVBUS    =00000004    65       (2)
CHS$V_DEVERR    =00000002    65       (2)
CHS$V_DEVTO     =00000005    65       (2)
CHS$V_MBAATN    =00000014    65       (2)
CHS$V_MBACPE    =00000015    65       (2)
CHS$V_MBADTB    =00000012    65       (2)
CHS$V_MBADTC    =00000013    65       (2)
CHS$V_MBAEXC    =00000010    65       (2)
CHS$V_MBANED    =00000011    65       (2)
CHS$V_MBAWCKLWR=00000019     65       (2)
CHS$V_MBAWCKUPR=0000001A     65       (2)
CHS$V_PGMERR    =00000003    65       (2)
CHS$V_PGMHDE    =0000000B    65       (2)
CHS$V_SYSERR    =00000000    65       (2)
CHS$V_SYSMEM    =00000009    65       (2)
```

```
CHS$V_SYSSBI    =0000000A      65     (2)
CHS$V_UIE       =0000001C      65     (2)
CMPDATA          00000000-XR                 1202   (19)     1394   (21)     1431   (21)     1439   (21)     1631   (23)
                                             1819   (25)     1979   (27)      923   (15)
CONFIG           00000000-XR               #-1007   (17)   #-1055   (17)   #-1132   (19)   #-1156   (19)   #-126    (4)
                                           #-1305   (21)   #-1325   (21)   #-1539   (23)   #-1564   (23)   #-1715   (25)
                                           #-1740   (25)   #-1893   (27)   #-1932   (27)   #-224    (5)    #-249    (5)
                                           #-262    (5)    #-304    (6)    #-335    (6)    #-384    (8)    #-412    (8)
                                           #-451    (8)    #-533    (10)   #-557    (10)   #-585    (10)   #-649    (12)
                                           #-750    (13)   #-764    (13)   #-831    (15)   #-853    (15)   #-983    (17)
CSR$M_CRDY      =00000080                  #-1040   (17)   #-458    (8)    #-591    (10)   #-667    (12)   #-688    (12)
CSR$M_DCK       =00000800                  #-1040   (17)
CSR$M_DRDY      =00000001                  #-1041   (17)   #-458    (8)    #-592    (10)
CSR$M_ECSO      =00100000                  #-462    (8)
CSR$M_ERR       =00008000                  #-1040   (17)   #-457    (8)
CSR$M_HNF       =00001000                  #-457    (8)
CSR$M_NXM       =00002000                  #-591    (10)
CSR$M_OPI       =00000400                  #-443    (8)    #-457    (8)
CSR_BITS         00000000-XR               #-1065   (17)   #-467    (8)    #-599    (10)
CURR_CYLN        00000000-XR               #-148    (4)    #-157    (4)    #-558    (10)
CURR_CYL_HD      00000000-XR               #-402    (8)
DAR_MNEM         00000000-XR               #-1599   (23)   #-1626   (23)   #-1775   (25)   #-1814   (25)   #-891    (15)
                                           #-918    (15)
DATA_018         00000024-R    103    (4)     118   (4)
DATA_019         0000001C-R    209    (5)     216   (5)
DATA_020         00000014-R    292    (6)     299   (6)
DATA_021         00000018-R    373    (8)     380   (8)
DATA_022         00000024-R    522    (10)    529   (10)
DATA_023         00000024-R    634    (12)    641   (12)
DATA_024         00000020-R    735    (13)    742   (13)
DATA_025         0000001C-R    808    (15)    823   (15)
DATA_026         0000001C-R    969    (17)    976   (17)
DATA_027         00000018-R    1116   (19)   1123   (19)
DATA_028         00000024-R    1288   (21)   1295   (21)
DATA_029         00000024-R    1516   (23)   1531   (23)
DATA_030         0000001C-R    1703   (25)   1710   (25)
DATA_031         00000020-R    1870   (27)   1885   (27)
DEFAULT         =00000000      63     (2)    1123   (19)      118   (4)      1295   (21)     1531   (23)     1710   (25)
                                             1885   (27)      216   (5)       299   (6)      380   (8)       529   (10)
                                              641   (12)      742   (13)      823   (15)      976   (17)
DRIVE_TBL        00000000-XR               #-1127   (19)   #-122    (4)    #-1299   (21)   #-1535   (23)   #-1714   (25)
                                           #-1889   (27)   #-220    (5)    #-303    (6)    #-422    (8)    #-534    (10)
                                           #-645    (12)   #-746    (13)   #-827    (15)   #-980    (17)
DS$BGNSUB        00000000-XR                 1009   (17)     1032   (17)     1137   (19)     1158   (19)     1179   (19)
                                             1208   (19)     1306   (21)     1327   (21)     1367   (21)     1400   (21)
                                             1445   (21)     1545   (23)     1566   (23)     1602   (23)     1637   (23)
                                             1721   (25)     1742   (25)     1779   (25)     1913   (27)     1934   (27)
                                             1955   (27)     1985   (27)      385   (8)       414   (8)       538   (10)
                                              562   (10)      834   (15)      855   (15)      894   (15)      988   (17)
DS$BRANCH        00000000-XR                 1903   (27)     1906   (27)
DS$BREAK         00000000-XR                 1068   (17)     1229   (19)     1466   (21)     1658   (23)      180   (4)
                                             1826   (25)     2008   (27)      264   (5)       337   (6)       480   (8)
                                              605   (10)      710   (12)      766   (13)      929   (15)
DS$CKLOOP        00000000-XR                 1005   (17)     1030   (17)     1060   (17)     1066   (17)     1147   (19)
                                             1154   (19)     1177   (19)     1198   (19)     1206   (19)     1227   (19)
                                             1316   (21)     1323   (21)     1351   (21)     1365   (21)     1390   (21)
```

```
                                          1398    (21)    141     (4)     1427    (21)    1435    (21)    1443    (21)
                                          1464    (21)    147     (4)     1555    (23)    156     (4)     1562    (23)
                                          1590    (23)    1600    (23)    1621    (23)    1627    (23)    1635    (23)
                                          1656    (23)    167     (4)     1731    (25)    1738    (25)    175     (4)
                                          1767    (25)    1776    (25)    1806    (25)    1815    (25)    1823    (25)
                                          1923    (27)    1930    (27)    1953    (27)    1975    (27)    1983    (27)
                                          2005    (27)    232     (5)     248     (5)     261     (5)     314     (6)
                                          334     (6)     395     (8)     401     (8)     410     (8)     431     (8)
                                          449     (8)     456     (8)     468     (8)     474     (8)     548     (10)
                                          555     (10)    575     (10)    590     (10)    600     (10)    654     (12)
                                          673     (12)    682     (12)    694     (12)    707     (12)    756     (13)
                                          763     (13)    844     (15)    851     (15)    882     (15)    892     (15)
                                          913     (15)    919     (15)    927     (15)    998     (17)
DS$ENDSUB         00000000-XR             1008    (17)    1031    (17)    1067    (17)    1157    (19)    1178    (19)
                                          1207    (19)    1228    (19)    1326    (21)    1366    (21)    1399    (21)
                                          1444    (21)    1465    (21)    1565    (23)    1601    (23)    1636    (23)
                                          1657    (23)    1741    (25)    1778    (25)    1825    (25)    1933    (27)
                                          1954    (27)    1984    (27)    2006    (27)    413     (8)     479     (8)
                                          561     (10)    604     (10)    854     (15)    893     (15)    928     (15)
DS$ERRHARD        00000000-XR             1004    (17)    1029    (17)    1059    (17)    1065    (17)    1146    (19)
                                          1153    (19)    1176    (19)    1197    (19)    1205    (19)    1226    (19)
                                          1315    (21)    1322    (21)    1350    (21)    1364    (21)    1389    (21)
                                          1397    (21)    140     (4)     1426    (21)    1434    (21)    1442    (21)
                                          146     (4)     1463    (21)    155     (4)     1554    (23)    1561    (23)
                                          1589    (23)    1599    (23)    1620    (23)    1626    (23)    1634    (23)
                                          1655    (23)    166     (4)     1730    (25)    1737    (25)    174     (4)
                                          1766    (25)    1775    (25)    1805    (25)    1814    (25)    1822    (25)
                                          1922    (27)    1929    (27)    1952    (27)    1974    (27)    1982    (27)
                                          2004    (27)    231     (5)     247     (5)     260     (5)     313     (6)
                                          333     (6)     394     (8)     400     (8)     409     (8)     430     (8)
                                          448     (8)     455     (8)     467     (8)     473     (8)     478     (8)
                                          547     (10)    554     (10)    574     (10)    589     (10)    599     (10)
                                          653     (12)    672     (12)    681     (12)    693     (12)    706     (12)
                                          755     (13)    762     (13)    843     (15)    850     (15)    881     (15)
                                          891     (15)    912     (15)    918     (15)    926     (15)    997     (17)
DS$ESCAPE         00000000-XR             1006    (17)    1155    (19)    1324    (21)    1563    (23)    1739    (25)
                                          1931    (27)    411     (8)     450     (8)     556     (10)    674     (12)
                                          683     (12)    695     (12)    708     (12)    852     (15)
DS$GETBUF         00000000-XR             1902    (27)    1905    (27)
DS$K_BBC         =0000001D      1906    (27)
DS$K_BBCC        =00000021      1906    (27)
DS$K_BBCCI       =00000023      1906    (27)
DS$K_BBCS        =0000001F      1906    (27)
DS$K_BBS         =0000001C      1906    (27)
DS$K_BBSC        =00000020      1906    (27)
DS$K_BBSS        =0000001E      1906    (27)
DS$K_BBSSI       =00000022      1906    (27)
DS$K_BCC_M       =0000001B      1906    (27)
DS$K_BCS_M       =0000001A      1906    (27)
DS$K_BEQLU_B     =00000005      1906    (27)
DS$K_BEQLU_M     =00000011      1906    (27)
DS$K_BEQL_B      =00000004      1906    (27)
DS$K_BEQL_M      =00000010      1906    (27)
DS$K_BERROR      =00000026      1906    (27)   #-1903    (27)
DS$K_BGEQU_B     =00000007      1906    (27)
DS$K_BGEQU_M     =00000013      1906    (27)
```

```
DS$K_BGEQ_B      =00000006      1906   (27)
DS$K_BGEQ_M      =00000012      1906   (27)
DS$K_BGTRU_B     =00000009      1906   (27)
DS$K_BGTRU_M     =00000015      1906   (27)
DS$K_BGTR_B      =00000008      1906   (27)
DS$K_BGTR_M      =00000014      1906   (27)
DS$K_BLBC        =00000025      1906   (27)
DS$K_BLBS        =00000024      1906   (27)
DS$K_BLEQU_B     =00000003      1906   (27)
DS$K_BLEQU_M     =0000000F      1906   (27)
DS$K_BLEQ_B      =00000002      1906   (27)
DS$K_BLEQ_M      =0000000E      1906   (27)
DS$K_BLSSU_B     =00000001      1906   (27)
DS$K_BLSSU_M     =0000000D      1906   (27)
DS$K_BLSS_B      =00000000      1906   (27)
DS$K_BLSS_M      =0000000C      1906   (27)
DS$K_BNEQU_B     =0000000B      1906   (27)
DS$K_BNEQU_M     =00000017      1906   (27)
DS$K_BNEQ_B      =0000000A      1906   (27)
DS$K_BNEQ_M      =00000016      1906   (27)
DS$K_BNERROR     =00000027      1906   (27)   #-1906   (27)
DS$K_BVC_M       =00000019      1906   (27)
DS$K_BVS_M       =00000018      1906   (27)
DS$PRINTB         00000000-XR                   1907   (27)
DS$RELBUF         00000000-XR                   2007   (27)
DS$WAITMS         000000C0-XR                   442    (8)
E.CS              00000000-XR                  #-1042  (17)    #-1043  (17)    #-1044  (17)    #-1061  (17)    #-1065  (17)
                                               #-459   (8)     #-460   (8)     #-461   (8)     #-463   (8)     #-467   (8)
                                               #-592   (10)    #-593   (10)    #-594   (10)    #-595   (10)    #-599   (10)
ENV$M_DOMAIN     =00000002      61     (2)
ENV$M_LEVEL      =00000001      61     (2)
ENV$M_SUPER      =000003FC      61     (2)
ENV$S_DOMAIN     =00000001      61     (2)
ENV$S_LEVEL      =00000001      61     (2)
ENV$S_SUPER      =00000008      61     (2)
ENV$V_DOMAIN     =00000001      61     (2)      61     (2)
ENV$V_LEVEL      =00000000      61     (2)      61     (2)
ENV$V_SUPER      =00000002      61     (2)
ENV$_CPU         =00000000      61     (2)      61     (2)
ENV$_FUNCTIONAL  =00000000      61     (2)      61     (2)
ENV$_REPAIR      =00000001      61     (2)      61     (2)
ENV$_SUPER       =00000001      61     (2)
ENV$_SYSTEM      =00000001      61     (2)      61     (2)
ERR_MSG           00000000-XR                   1004   (17)     1153   (19)     1322   (21)     155    (4)      1561   (23)
                                                166    (4)      1737   (25)     174    (4)      1929   (27)     247    (5)
                                                260    (5)      333    (6)      409    (8)      554    (10)     762    (13)
                                                850    (15)
GETSTAT          =00000004                     #-658   (12)    #-665   (12)
GET_LUN$A         00000000-XR                   1125   (19)     120    (4)      1297   (21)     1533   (23)     1712   (25)
                                                1887   (27)     218    (5)      301    (6)      382    (8)      531    (10)
                                                643    (12)     744    (13)     825    (15)     978    (17)
GET_POSITION      00000000-XR                   143    (4)      397    (8)
GET_RANDOM        00000000-XR                   233    (5)      315    (6)
HEAD_0           =00000000                      104    (4)      106    (4)      107    (4)      109    (4)      110    (4)
                                                112    (4)      113    (4)      115    (4)      1517   (23)     1519   (23)
                                                1520   (23)     1522   (23)     1523   (23)     1525   (23)     1526   (23)
```

```
                                       1528   (23)     1871   (27)     1873   (27)     1874   (27)     1876   (27)
                                       1877   (27)     1879   (27)     1880   (27)     1882   (27)     809    (15)
                                       811    (15)     812    (15)     814    (15)     815    (15)     817    (15)
                                       818    (15)     820    (15)
HEAD_1          =00000001              105    (4)      108    (4)      111    (4)      114    (4)      1518   (23)
                                       1521   (23)     1524   (23)     1527   (23)     1872   (27)     1875   (27)
                                       1878   (27)     1881   (27)     810    (15)     813    (15)     816    (15)
                                       819    (15)
IDCCSR          =00000000             #-443   (8)     #-667   (12)    #-687   (12)    #-688   (12)
IDCGETBUF        00000000-XR           1907   (27)
IDCMPR          =00000010             #-675   (12)    #-696   (12)
L$UNIT           00000000-XR          #-1004  (17)    #-1029  (17)    #-1059  (17)    #-1065  (17)    #-1146  (19)
                                      #-1153  (19)    #-1176  (19)    #-1197  (19)    #-1205  (19)    #-1226  (19)
                                      #-1315  (21)    #-1322  (21)    #-1350  (21)    #-1364  (21)    #-1389  (21)
                                      #-1397  (21)    #-140   (4)     #-1426  (21)    #-1434  (21)    #-1442  (21)
                                      #-146   (4)     #-1463  (21)    #-155   (4)     #-1554  (23)    #-1561  (23)
                                      #-1589  (23)    #-1599  (23)    #-1620  (23)    #-1626  (23)    #-1634  (23)
                                      #-1655  (23)    #-166   (4)     #-1730  (25)    #-1737  (25)    #-174   (4)
                                      #-1766  (25)    #-1775  (25)    #-1805  (25)    #-1814  (25)    #-1822  (25)
                                      #-1922  (27)    #-1929  (27)    #-1952  (27)    #-1974  (27)    #-1982  (27)
                                      #-2004  (27)    #-231   (5)     #-247   (5)     #-260   (5)     #-313   (6)
                                      #-333   (6)     #-394   (8)     #-400   (8)     #-409   (8)     #-430   (8)
                                      #-448   (8)     #-455   (8)     #-467   (8)     #-473   (8)     #-478   (8)
                                      #-547   (10)    #-554   (10)    #-574   (10)    #-589   (10)    #-599   (10)
                                      #-653   (12)    #-672   (12)    #-681   (12)    #-693   (12)    #-706   (12)
                                      #-755   (13)    #-762   (13)    #-843   (15)    #-850   (15)    #-881   (15)
                                      #-891   (15)    #-912   (15)    #-918   (15)    #-926   (15)    #-997   (17)
LUN0            =00000000              104    (4)      105    (4)      106    (4)      1117   (19)     1289   (21)
                                       1517   (23)     1518   (23)     1519   (23)     1704   (25)     1871   (27)
                                       1872   (27)     1873   (27)     210    (5)      293    (6)      374    (8)
                                       523    (10)     635    (12)     736    (13)     809    (15)     810    (15)
                                       811    (15)     970    (17)
LUN1            =00000001              107    (4)      108    (4)      109    (4)      1118   (19)     1290   (21)
                                       1520   (23)     1521   (23)     1522   (23)     1705   (25)     1874   (27)
                                       1875   (27)     1876   (27)     211    (5)      294    (6)      375    (8)
                                       524    (10)     636    (12)     737    (13)     812    (15)     813    (15)
                                       814    (15)     971    (17)
LUN2            =00000002              110    (4)      111    (4)      1119   (19)     112    (4)      1291   (21)
                                       1523   (23)     1524   (23)     1525   (23)     1706   (25)     1877   (27)
                                       1878   (27)     1879   (27)     212    (5)      295    (6)      376    (8)
                                       525    (10)     637    (12)     738    (13)     815    (15)     816    (15)
                                       817    (15)     972    (17)
LUN3            =00000003              1120   (19)     113    (4)      114    (4)      115    (4)      1292   (21)
                                       1526   (23)     1527   (23)     1528   (23)     1707   (25)     1880   (27)
                                       1881   (27)     1882   (27)     213    (5)      296    (6)      377    (8)
                                       526    (10)     638    (12)     739    (13)     818    (15)     819    (15)
                                       820    (15)     973    (17)
MANUAL          =00000001     63      (2)
MS108_SECTNZ     00000000-XR           681    (12)
MS109_BADCNT     00000000-XR           706    (12)
MSG14_SYSINIT    00000000-XR           1146   (19)     1315   (21)     140    (4)      1554   (23)     1730   (25)
                                       1922   (27)     231    (5)      313    (6)      394    (8)      430    (8)
                                       547    (10)     574    (10)     653    (12)     755    (13)     843    (15)
                                       997    (17)
MSG16A_WRTCHK    00000000-XR           1655   (23)
MSG16B_WRTCHK    00000000-XR           2004   (27)
```

```
MSG16_WRTCHK     00000000-XR                 1226    (19)
MSG17_GETSTAT    00000000-XR                 672     (12)     693     (12)
MSG19_READH      00000000-XR                 146     (4)      400     (8)
MSG20A_WRITE     00000000-XR                 1589    (23)     1599    (23)
MSG20B_WRITE     00000000-XR                 1952    (27)
MSG20_WRITE      00000000-XR                 1029    (17)     1176    (19)     1350    (21)     1766    (25)     1775    (25)
                                             881     (15)     891     (15)
MSG21A_READ      00000000-XR                 1620    (23)     1626    (23)
MSG21B_READ      00000000-XR                 1974    (27)
MSG21_READ       00000000-XR                 1197    (19)     1805    (25)     1814    (25)     912     (15)     918     (15)
MSG23_CS_INC     00000000-XR                #-1065   (17)    #-467    (8)     #-599    (10)
MSG30_CRDY       00000000-XR                #-672    (12)    #-693    (12)
MSG35_OPI_TO     00000000-XR                #-448    (8)
MSG36_OPI_RLY    00000000-XR                #-478    (8)
MSG37_OPI_LAT    00000000-XR                #-473    (8)
MSG52_RD_HNF     00000000-XR                 448     (8)      455     (8)      467     (8)      473     (8)      478     (8)
MSG53_WRT_NXM    00000000-XR                 589     (10)     599     (10)
MSG54_DA_UPD     00000000-XR                #-1599   (23)    #-1626   (23)    #-1775   (25)    #-1814   (25)    #-891    (15)
                                            #-918    (15)
MSG55_DAT_CP1    00000000-XR                 1205    (19)     926     (15)
MSG57_WCK_FRC    00000000-XR                 1059    (17)     1065    (17)
MSG58_PAR_WRT    00000000-XR                 1364    (21)
MSG59_PAR_RD     00000000-XR                 1426    (21)
MSG60_WRT_PRD    00000000-XR                 1389    (21)
MSG61_DAT_PAR    00000000-XR                 1397    (21)
MSG62_DC_PAR     00000000-XR                 1434    (21)
MSG63_PRD_TMW    00000000-XR                 1442    (21)
MSG64_WCK_PAR    00000000-XR                 1463    (21)
MSG65_DAT_CP2    00000000-XR                 1634    (23)
MSG67_DAT_ES     00000000-XR                 1822    (25)
MSG68_DAT_MAX    00000000-XR                 1982    (27)
MSG_018          00000002-R    71      (2)   118     (4)
MSG_019          00000002-R    181     (4)   216     (5)
MSG_020          00000002-R    265     (5)   299     (6)
MSG_021          00000002-R    338     (6)   380     (8)
MSG_022          00000002-R    481     (8)   529     (10)
MSG_023          00000002-R    606     (10)  641     (12)
MSG_024          00000002-R    711     (12)  742     (13)
MSG_025          00000002-R    767     (13)  823     (15)
MSG_026          00000002-R    930     (15)  976     (17)
MSG_027          00000002-R    1069    (17)  1123    (19)
MSG_028          00000002-R    1230    (19)  1295    (21)
MSG_029          00000002-R    1467    (21)  1531    (23)
MSG_030          00000002-R    1659    (23)  1710    (25)
MSG_031          00000002-R    1827    (25)  1885    (27)
NOCHK$M_CERR    =00000002                   #-1055   (17)    #-451    (8)     #-585    (10)
NOCHK$M_CRDY    =00000001                   #-451    (8)
OFST_AVGTIM     =00000080      66      (2)
OFST_CSRTYP     =00000058      66      (2)  #-1043   (17)    #-461    (8)     #-594    (10)
OFST_DEVMEM     =0000001C      66      (2)
OFST_DEVNAM     =00000020      66      (2)
OFST_DRIV       =00000008                   #-123    (4)     #-1536   (23)    #-1890   (27)    #-828    (15)
OFST_DRVERR     =0000004C      66      (2)
OFST_DRVRST     =00000050      66      (2)
OFST_DRVTYP     =00000000      66      (2)  #-123    (4)     #-1536   (23)    #-1890   (27)    #-221    (5)     #-319    (6)
                                            #-646    (12)    #-747    (13)    #-828    (15)
```

```
OFST_FMTLEN     =00000030      66      (2)
OFST_GTSCMD     =00000034      66      (2)
OFST_HDBITS     =00000044      66      (2)
OFST_HDRDNC     =00000054      66      (2)
OFST_HDRTMP     =00000048      66      (2)
OFST_HIGRPM     =00000074      66      (2)
OFST_HMBSEC     =0000005C      66      (2)
OFST_IGBCYL     =00000018      66      (2)
OFST_LOWRPM     =00000070      66      (2)
OFST_LSTCYL     =00000008      66      (2)    #-135    (4)    #-227    (5)    #-308    (6)
OFST_LSTSEC     =00000010      66      (2)    #-1718   (25)   #-709    (12)
OFST_LSTTRK     =0000000C      66      (2)    #-309    (6)    #-765    (13)
OFST_MAXTIM     =0000007C      66      (2)
OFST_MPBITS     =00000040      66      (2)
OFST_NMLSTS     =00000038      66      (2)
OFST_ONETIM     =00000078      66      (2)
OFST_OP1STS     =00000060      66      (2)
OFST_PHYU       =00000004      66      (2)
OFST_RANCYL     =0000006C      66      (2)
OFST_SECLEN     =00000028      66      (2)    #-1129   (19)   #-1301   (21)   #-1541   (23)   #-1717   (25)   #-423    (8)
                                              #-832    (15)   #-982    (17)
OFST_SEKSTS     =0000003C      66      (2)
OFST_SKHD       =0000000C                     #-150    (4)    #-1544   (23)   #-1556   (23)   #-161    (4)    #-169    (4)
                                              #-1898   (27)   #-1924   (27)   #-845    (15)   #-871    (15)
OFST_STSDNC     =00000064      66      (2)
OFST_TRKLEN     =0000002C      66      (2)    #-1895   (27)
OFST_TSTCYL     =00000014      66      (2)    #-1128   (19)   #-1300   (21)   #-1540   (23)   #-1716   (25)   #-1894   (27)
                                              #-425    (8)    #-535    (10)   #-833    (15)   #-981    (17)
OFST_UNIT       =00000004                     #-1125   (19)   #-120    (4)    #-1297   (21)   #-1533   (23)   #-1712   (25)
                                              #-1887   (27)   #-218    (5)    #-301    (6)    #-382    (8)    #-531    (10)
                                              #-643    (12)   #-744    (13)   #-825    (15)   #-978    (17)
OFST_WTPSTS     =00000068      66      (2)
P$UNIT           00000000-XR                  #-1044   (17)   #-460    (8)    #-593    (10)   #-659    (12)
PHY_RD_BUF       00000000-XR                    1905   (27)     1963   (27)   #-1977   (27)
PHY_WRT_BUF      00000000-XR                    1902   (27)     1911   (27)     1941   (27)   #-1976   (27)     1993   (27)
POSITION_HDS     00000000-XR                    1001   (17)     1150   (19)     1319   (21)     152    (4)      1558   (23)
                                                163    (4)      171    (4)      1734   (25)     1926   (27)     244    (5)
                                                257    (5)      330    (6)      406    (8)      551    (10)     759    (13)
                                                847    (15)
PRINT_DATA_ERR   00000000-XR                    1205   (19)     1397   (21)     1434   (21)     1442   (21)     1634   (23)
                                                1822   (25)     1982   (27)     926    (15)
PRINT_REG        00000000-XR                    1004   (17)     1029   (17)     1059   (17)     1146   (19)     1153   (19)
                                                1176   (19)     1197   (19)     1226   (19)     1315   (21)     1322   (21)
                                                1350   (21)     1364   (21)     1389   (21)     140    (4)      1426   (21)
                                                146    (4)      1463   (21)     155    (4)      1554   (23)     1561   (23)
                                                1589   (23)     1620   (23)     1655   (23)     166    (4)      1730   (25)
                                                1737   (25)     174    (4)      1766   (25)     1805   (25)     1922   (27)
                                                1929   (27)     1952   (27)     1974   (27)     2004   (27)     231    (5)
                                                247    (5)      260    (5)      313    (6)      333    (6)      394    (8)
                                                400    (8)      409    (8)      430    (8)      448    (8)      455    (8)
                                                473    (8)      478    (8)      547    (10)     554    (10)     574    (10)
                                                589    (10)     653    (12)     672    (12)     693    (12)     755    (13)
                                                762    (13)     843    (15)     850    (15)     881    (15)     912    (15)
                                                997    (17)
R80$T           =00000002                       106    (4)      109    (4)      112    (4)      115    (4)      1519   (23)
                                                1522   (23)     1525   (23)     1528   (23)     1873   (27)     1876   (27)
```

```
                                                 1879   (27)     1882   (27)   #-221    (5)    #-319    (6)    #-646   (12)
                                               #-747    (13)      811   (15)     814   (15)      817   (15)      820   (15)
RD_IDCREGS         00000000-XR                   446    (8)       670   (12)     679   (12)      691   (12)      704   (12)
READ              =0000000C                    #-1192   (19)   #-1384   (21)  #-1421   (21)   #-1615   (23)   #-1800   (25)
                                               #-1969   (27)   #-440    (8)   #-458     (8)   #-907    (15)
READ_BUFFER        00000000-XR                  1186    (19)   #-1200   (19)    1378   (21)   #-1392   (21)   #-1412   (21)
                                                1415    (21)   #-1429   (21)  #-1575   (23)     1609   (23)   #-1629   (23)
                                               #-1789   (25)     1794   (25)  #-1817   (25)      433    (8)   #-864    (15)
                                                 901    (15)   #-921    (15)
REG_ERRPT          00000000-XR                  1065    (17)     1599   (23)    1626   (23)     1775   (25)     1814   (25)
                                                 467    (8)       599   (10)     681   (12)      706   (12)      891   (15)
                                                 918    (15)
RL02$T            =00000001                      104    (4)       105    (4)     107    (4)      108    (4)      110    (4)
                                                 111    (4)       113    (4)     114    (4)     1517   (23)     1518   (23)
                                                1520    (23)     1521   (23)    1523   (23)     1524   (23)     1526   (23)
                                                1527    (23)     1871   (27)    1872   (27)     1874   (27)     1875   (27)
                                                1877    (27)     1878   (27)    1880   (27)     1881   (27)      809   (15)
                                                 810    (15)      812   (15)     813   (15)      815   (15)      816   (15)
                                                 818    (15)      819   (15)
SEP_FUNCTIONAL    =00000002        61     (2)
SEP_REPAIR        =00000003        61     (2)     61    (2)
SIZ...            =00000001        65     (2)     61    (2)        65    (2)
START_CMD          00000000-XR                  1025    (17)     1054   (17)    1172   (19)     1193   (19)     1222   (19)
                                                1346    (21)     1360   (21)    1385   (21)     1422   (21)     1459   (21)
                                                1585    (23)     1616   (23)    1651   (23)     1762   (25)     1801   (25)
                                                1948    (27)     1970   (27)    2000   (27)      441    (8)      584   (10)
                                                 666    (12)      877   (15)     908   (15)
STATUS_CHK         00000000-XR                  1026    (17)     1056   (17)    1173   (19)     1194   (19)     1223   (19)
                                                1347    (21)     1361   (21)    1386   (21)     1423   (21)     1460   (21)
                                                1586    (23)     1617   (23)    1652   (23)     1763   (25)     1802   (25)
                                                1949    (27)     1971   (27)    2001   (27)      452    (8)      586   (10)
                                                 878    (15)      909   (15)
SUBSYS_INIT        00000000-XR                  1143    (19)     1312   (21)     137    (4)     1551   (23)     1727   (25)
                                                1919    (27)      228    (5)     310    (6)      391    (8)      427    (8)
                                                 544    (10)      571   (10)     650   (12)      752   (13)      840   (15)
                                                 994    (17)
T.CS               00000000-XR                 #-1061   (17)   #-1065   (17)  #-462     (8)   #-463     (8)   #-467     (8)
                                               #-595    (10)   #-599    (10)
T.DA               00000000-XR                 #-1595   (23)   #-1599   (23)  #-1622   (23)   #-1626   (23)   #-1771   (25)
                                               #-1775   (25)   #-1810   (25)  #-1814   (25)   #-887    (15)   #-891    (15)
                                               #-914    (15)   #-918    (15)
T.MP               00000000-XR                 #-675    (12)      676   (12)  #-696    (12)      697   (12)
T21_S1             00000058-R      385    (8)
T21_S1_X           0000010F-R      413    (8)
T21_S2             0000011A-R      414    (8)
T21_S2_X           000002E1-R      479    (8)
T22_S1             00000076-R      538    (10)
T22_S1_X           00000102-R      561    (10)
T22_S2             0000010D-R      562    (10)
T22_S2_X           00000217-R      604    (10)
T25_S1             00000117-R      834    (15)
T25_S1_X           00000196-R      854    (15)
T25_S2             000001A1-R      855    (15)
T25_S2_X           0000027B-R      893    (15)
T25_S3             00000286-R      894    (15)
T25_S3_X           00000361-R      928    (15)
```

```
T26_S1           00000085-R   988    (17)
T26_S1_X         00000103-R   1008   (17)
T26_S2           0000010E-R   1009   (17)
T26_S2_X         00000170-R   1031   (17)
T26_S3           0000017B-R   1032   (17)
T26_S3_X         00000249-R   1067   (17)
T27_S1           00000088-R   1137   (19)
T27_S1_X         00000106-R   1157   (19)
T27_S2           00000111-R   1158   (19)
T27_S2_X         0000016C-R   1178   (19)
T27_S3           00000177-R   1179   (19)
T27_S3_X         00000211-R   1207   (19)
T27_S4           0000021C-R   1208   (19)
T27_S4_X         00000277-R   1228   (19)
T28_S1           0000007E-R   1306   (21)
T28_S1_X         000000FC-R   1326   (21)
T28_S2           00000107-R   1327   (21)
T28_S2_X         000001CC-R   1366   (21)
T28_S3           000001D7-R   1367   (21)
T28_S3_X         00000283-R   1399   (21)
T28_S4           0000028E-R   1400   (21)
T28_S4_X         0000038B-R   1444   (21)
T28_S5           00000396-R   1445   (21)
T28_S5_X         000003F1-R   1465   (21)
T29_S1           0000012D-R   1545   (23)
T29_S1_X         000001AC-R   1565   (23)
T29_S2           000001B7-R   1566   (23)
T29_S2_X         00000284-R   1601   (23)
T29_S3           0000028F-R   1602   (23)
T29_S3_X         0000036A-R   1636   (23)
T29_S4           00000375-R   1637   (23)
T29_S4_X         000003D0-R   1657   (23)
T30_S1           00000076-R   1721   (25)
T30_S1_X         000000F4-R   1741   (25)
T30_S2           000000FF-R   1742   (25)
T30_S2_X         000001CA-R   1778   (25)
T30_S3           000001D5-R   1779   (25)
T30_S3_X         000002E7-R   1825   (25)
T31_S1           000001B0-R   1913   (27)
T31_S1_X         0000022F-R   1933   (27)
T31_S2           0000023A-R   1934   (27)
T31_S2_X         00000295-R   1954   (27)
T31_S3           000002A0-R   1955   (27)
T31_S3_X         0000033A-R   1984   (27)
T31_S4           00000345-R   1985   (27)
T31_S4_X         000003A0-R   2006   (27)
TEST_018         000000E8-R   118    (4)       118    (4)
TEST_018_X       00000241-R   180    (4)     #-121    (4)     #-125    (4)     #-159    (4)
TEST_019         00000040-R   216    (5)       216    (5)
TEST_019_X       00000135-R   264    (5)     #-219    (5)     #-223    (5)
TEST_020         00000038-R   299    (6)       299    (6)
TEST_020_X       000000F8-R   337    (6)     #-302    (6)
TEST_021         0000003C-R   380    (8)       380    (8)
TEST_021_X       000002EF-R   480    (8)     #-383    (8)       411    (8)       450    (8)
TEST_022         00000048-R   529    (10)      529    (10)
TEST_022_X       00000225-R   605    (10)    #-532    (10)      556    (10)    #-560    (10)
```

```
TEST_023            00000048-R   641     (12)      641     (12)
TEST_023_X          00000224-R   710     (12)    #-644     (12)    #-648      (12)      674     (12)      683     (12)      695     (12)
                                                   708     (12)
TEST_024            00000044-R   742     (13)      742     (13)
TEST_024_X          000000E4-R   766     (13)    #-745     (13)    #-749      (13)
TEST_025            000000E0-R   823     (15)      823     (15)
TEST_025_X          0000036F-R   929     (15)    #-826     (15)    #-830      (15)      852     (15)
TEST_026            00000040-R   976     (17)      976     (17)
TEST_026_X          00000257-R   1068    (17)      1006    (17)    #-979      (17)
TEST_027            0000003C-R   1123    (19)      1123    (19)
TEST_027_X          00000285-R   1229    (19)    #-1126    (19)      1155     (19)
TEST_028            00000048-R   1295    (21)      1295    (21)
TEST_028_X          000003FF-R   1466    (21)    #-1298    (21)      1324     (21)
TEST_029            000000E8-R   1531    (23)      1531    (23)
TEST_029_X          000003DE-R   1658    (23)    #-1534    (23)    #-1538     (23)      1563    (23)
TEST_030            00000040-R   1710    (25)      1710    (25)
TEST_030_X          000002F5-R   1826    (25)    #-1713    (25)      1739     (25)
TEST_031            000000E4-R   1885    (27)      1885    (27)
TEST_031_X          000003BF-R   2008    (27)    #-1888    (27)    #-1892     (27)      1931    (27)
VIR_RD_BUF          00000000-XR                    1905    (27)
VIR_WRT_BUF         00000000-XR                    1902    (27)      2007     (27)
WORD_COUNT          00000000-XR                  #-1201    (19)    #-1393     (21)    #-1430    (21)    #-1438    (21)    #-1630    (23)
                                                 #-1818    (25)    #-1978     (27)    #-922     (15)
WORK                00000000-XR                  #-1910    (27)      1911     (27)
WRITE              =0000000A                     #-1024    (17)    #-1171     (19)    #-1345    (21)    #-1359    (21)    #-1584    (23)
                                                 #-1761    (25)    #-1947     (27)    #-583     (10)    #-592     (10)    #-876     (15)
WRITE_BUFFER        00000000-XR                    1016    (17)      1047     (17)    #-1135    (19)      1165    (19)    #-1199    (19)
                                                   1215    (19)    #-1336     (21)      1339    (21)      1353    (21)    #-1375    (21)
                                                 #-1391    (21)    #-1410     (21)    #-1411    (21)    #-1428    (21)      1452    (21)
                                                 #-1574    (23)      1578     (23)    #-1628    (23)      1644    (23)    #-1750    (25)
                                                   1754    (25)    #-1788     (25)    #-1816    (25)      577     (10)    #-863     (15)
                                                   867     (15)    #-920      (15)    #-986     (17)
WRTCHK             =00000002                     #-1041    (17)    #-1053     (17)    #-1221    (19)    #-1458    (21)    #-1650    (23)
                                                 #-1999    (27)
```

! Macros Cross Reference !

MACRO	SIZE	DEFINITION		REFERENCES...									
$D1_ABTEST	1	1908	(27)	1908	(27)								
$D1_BDATA	1	103	(4)	103	(4)	1116	(19)	1288	(21)	1516	(23)	1703	(25)
				1870	(27)	209	(5)	292	(6)	373	(8)	522	(10)
				634	(12)	735	(13)	808	(15)	969	(17)		
$D1_BSUBT	1	385	(8)	1009	(17)	1032	(17)	1137	(19)	1158	(19)	1179	(19)
				1208	(19)	1306	(21)	1327	(21)	1367	(21)	1400	(21)
				1445	(21)	1545	(23)	1566	(23)	1602	(23)	1637	(23)
				1721	(25)	1742	(25)	1779	(25)	1913	(27)	1934	(27)
				1955	(27)	1985	(27)	385	(8)	414	(8)	538	(10)
				562	(10)	834	(15)	855	(15)	894	(15)	988	(17)
$D1_BTEST	1	118	(4)	1123	(19)	118	(4)	1295	(21)	1531	(23)	1710	(25)
				1885	(27)	216	(5)	299	(6)	380	(8)	529	(10)
				641	(12)	742	(13)	823	(15)	976	(17)		
$D1_ERR	1	140	(4)	1004	(17)	1029	(17)	1059	(17)	1065	(17)	1146	(19)
				1153	(19)	1176	(19)	1197	(19)	1205	(19)	1226	(19)
				1315	(21)	1322	(21)	1350	(21)	1364	(21)	1389	(21)
				1397	(21)	140	(4)	1426	(21)	1434	(21)	1442	(21)
				146	(4)	1463	(21)	155	(4)	1554	(23)	1561	(23)
				1589	(23)	1599	(23)	1620	(23)	1626	(23)	1634	(23)
				1655	(23)	166	(4)	1730	(25)	1737	(25)	174	(4)
				1766	(25)	1775	(25)	1805	(25)	1814	(25)	1822	(25)
				1922	(27)	1929	(27)	1952	(27)	1974	(27)	1982	(27)
				2004	(27)	231	(5)	247	(5)	260	(5)	313	(6)
				333	(6)	394	(8)	400	(8)	409	(8)	430	(8)
				448	(8)	455	(8)	467	(8)	473	(8)	478	(8)
				547	(10)	554	(10)	574	(10)	589	(10)	599	(10)
				653	(12)	672	(12)	681	(12)	693	(12)	706	(12)
				755	(13)	762	(13)	843	(15)	850	(15)	881	(15)
				891	(15)	912	(15)	918	(15)	926	(15)	997	(17)
$D1_ERR_S	1	140	(4)	1004	(17)	1029	(17)	1059	(17)	1065	(17)	1146	(19)
				1153	(19)	1176	(19)	1197	(19)	1205	(19)	1226	(19)
				1315	(21)	1322	(21)	1350	(21)	1364	(21)	1389	(21)
				1397	(21)	140	(4)	1426	(21)	1434	(21)	1442	(21)
				146	(4)	1463	(21)	155	(4)	1554	(23)	1561	(23)
				1589	(23)	1599	(23)	1620	(23)	1626	(23)	1634	(23)
				1655	(23)	166	(4)	1730	(25)	1737	(25)	174	(4)
				1766	(25)	1775	(25)	1805	(25)	1814	(25)	1822	(25)
				1922	(27)	1929	(27)	1952	(27)	1974	(27)	1982	(27)
				2004	(27)	231	(5)	247	(5)	260	(5)	313	(6)
				333	(6)	394	(8)	400	(8)	409	(8)	430	(8)
				448	(8)	455	(8)	467	(8)	473	(8)	478	(8)
				547	(10)	554	(10)	574	(10)	589	(10)	599	(10)
				653	(12)	672	(12)	681	(12)	693	(12)	706	(12)
				755	(13)	762	(13)	843	(15)	850	(15)	881	(15)
				891	(15)	912	(15)	918	(15)	926	(15)	997	(17)
$D1_ESCTEST	1	411	(8)	1006	(17)	1155	(19)	1324	(21)	1563	(23)	1739	(25)
				1931	(27)	411	(8)	450	(8)	556	(10)	674	(12)
				683	(12)	695	(12)	708	(12)	852	(15)		
$D1_ESUBT	1	413	(8)	1008	(17)	1031	(17)	1067	(17)	1157	(19)	1178	(19)

```
                                             1207  (19)    1228  (19)    1326  (21)    1366  (21)    1399  (21)
                                             1444  (21)    1465  (21)    1565  (23)    1601  (23)    1636  (23)
                                             1657  (23)    1741  (25)    1778  (25)    1825  (25)    1933  (27)
                                             1954  (27)    1984  (27)    2006  (27)    413   (8)     479   (8)
                                             561   (10)    604   (10)    854   (15)    893   (15)    928   (15)
$D1_ETEST          1           180   (4)     1068  (17)    1229  (19)    1466  (21)    1658  (23)    180   (4)
                                             1826  (25)    2008  (27)    264   (5)     337   (6)     480   (8)
                                             605   (10)    710   (12)    766   (13)    929   (15)
$D1_EXTEST         1           121   (4)     1126  (19)    121   (4)     125   (4)     1298  (21)    1534  (23)
                                             1538  (23)    159   (4)     1713  (25)    1888  (27)    1892  (27)
                                             219   (5)     223   (5)     302   (6)     383   (8)     532   (10)
                                             560   (10)    644   (12)    648   (12)    745   (13)    749   (13)
                                             826   (15)    830   (15)    979   (17)
$D1_PRINT_S        1           1907  (27)    1907  (27)
$D1_SBTTL          2           71    (2)     1069  (17)    1230  (19)    1467  (21)    1659  (23)    181   (4)
                                             1827  (25)    265   (5)     338   (6)     481   (8)     606   (10)
                                             71    (2)     711   (12)    767   (13)    930   (15)
$D2_BDATA          1           103   (4)     103   (4)     1116  (19)    1288  (21)    1516  (23)    1703  (25)
                                             1870  (27)    209   (5)     292   (6)     373   (8)     522   (10)
                                             634   (12)    735   (13)    808   (15)    969   (17)
$D2_BTEST          1           118   (4)     1123  (19)    118   (4)     1295  (21)    1531  (23)    1710  (25)
                                             1885  (27)    216   (5)     299   (6)     380   (8)     529   (10)
                                             641   (12)    742   (13)    823   (15)    976   (17)
$D2_SBTTL          1           71    (2)     1069  (17)    1230  (19)    1467  (21)    1659  (23)    181   (4)
                                             1827  (25)    265   (5)     338   (6)     481   (8)     606   (10)
                                             71    (2)     711   (12)    767   (13)    930   (15)
$DEF               1           1906  (27)
$DEFINI            1           67    (2)     67    (2)     68    (2)     69    (2)
$DS_ABORT          1           1908  (27)    1908  (27)
$DS_BERROR         1           1903  (27)    1903  (27)
$DS_BGNDATA        1           103   (4)     103   (4)     1116  (19)    1288  (21)    1516  (23)    1703  (25)
                                             1870  (27)    209   (5)     292   (6)     373   (8)     522   (10)
                                             634   (12)    735   (13)    808   (15)    969   (17)
$DS_BGNMOD         1           61    (2)     61    (2)
$DS_BGNSUB         1           385   (8)     1009  (17)    1032  (17)    1137  (19)    1158  (19)    1179  (19)
                                             1208  (19)    1306  (21)    1327  (21)    1367  (21)    1400  (21)
                                             1445  (21)    1545  (23)    1566  (23)    1602  (23)    1637  (23)
                                             1721  (25)    1742  (25)    1779  (25)    1913  (27)    1934  (27)
                                             1955  (27)    1985  (27)    385   (8)     414   (8)     538   (10)
                                             562   (10)    834   (15)    855   (15)    894   (15)    988   (17)
$DS_BGNTEST        1           118   (4)     1123  (19)    118   (4)     1295  (21)    1531  (23)    1710  (25)
                                             1885  (27)    216   (5)     299   (6)     380   (8)     529   (10)
                                             641   (12)    742   (13)    823   (15)    976   (17)
$DS_BITDEF         2           62    (2)     62    (2)
$DS_BNERROR        1           1906  (27)    1906  (27)
$DS_BREAK          1           180   (4)     1068  (17)    1229  (19)    1466  (21)    1658  (23)    180   (4)
                                             1826  (25)    2008  (27)    264   (5)     337   (6)     480   (8)
                                             605   (10)    710   (12)    766   (13)    929   (15)
$DS_CHCDEF         1           65    (2)     65    (2)
$DS_CHDEF          1           65    (2)     65    (2)
$DS_CHIDEF         1           65    (2)     65    (2)
$DS_CHMDEF         2           65    (2)     65    (2)
$DS_CHSDEF         3           65    (2)     65    (2)
$DS_CKLOOP         1           141   (4)     1005  (17)    1030  (17)    1060  (17)    1066  (17)    1147  (19)
                                             1154  (19)    1177  (19)    1198  (19)    1206  (19)    1227  (19)
                                             1316  (21)    1323  (21)    1351  (21)    1365  (21)    1390  (21)
```

```
                                    1398   (21)   141    (4)    1427   (21)   1435   (21)   1443   (21)
                                    1464   (21)   147    (4)    1555   (23)   156    (4)    1562   (23)
                                    1590   (23)   1600   (23)   1621   (23)   1627   (23)   1635   (23)
                                    1656   (23)   167    (4)    1731   (25)   1738   (25)   175    (4)
                                    1767   (25)   1776   (25)   1806   (25)   1815   (25)   1823   (25)
                                    1923   (27)   1930   (27)   1953   (27)   1975   (27)   1983   (27)
                                    2005   (27)   232    (5)    248    (5)    261    (5)    314    (6)
                                    334    (6)    395    (8)    401    (8)    410    (8)    431    (8)
                                    449    (8)    456    (8)    468    (8)    474    (8)    548    (10)
                                    555    (10)   575    (10)   590    (10)   600    (10)   654    (12)
                                    673    (12)   682    (12)   694    (12)   707    (12)   756    (13)
                                    763    (13)   844    (15)   851    (15)   882    (15)   892    (15)
                                    913    (15)   919    (15)   927    (15)   998    (17)
$DS_ENDDATA       1     116    (4)  1121   (19)   116    (4)    1293   (21)   1529   (23)   1708   (25)
                                    1883   (27)   214    (5)    297    (6)    378    (8)    527    (10)
                                    639    (12)   740    (13)   821    (15)   974    (17)
$DS_ENDMOD        1     2009   (27) 2009   (27)
$DS_ENDSUB        1     413    (8)  1008   (17)   1031   (17)   1067   (17)   1157   (19)   1178   (19)
                                    1207   (19)   1228   (19)   1326   (21)   1366   (21)   1399   (21)
                                    1444   (21)   1465   (21)   1565   (23)   1601   (23)   1636   (23)
                                    1657   (23)   1741   (25)   1778   (25)   1825   (25)   1933   (27)
                                    1954   (27)   1984   (27)   2006   (27)   413    (8)    479    (8)
                                    561    (10)   604    (10)   854    (15)   893    (15)   928    (15)
$DS_ENDTEST       1     180    (4)  1068   (17)   1229   (19)   1466   (21)   1658   (23)   180    (4)
                                    1826   (25)   2008   (27)   264    (5)    337    (6)    480    (8)
                                    605    (10)   710    (12)   766    (13)   929    (15)
$DS_ENVDEF        2     61     (2)  61     (2)
$DS_ERRHARD_S     1     139    (4)  1003   (17)   1028   (17)   1058   (17)   1063   (17)   1145   (19)
                                    1152   (19)   1175   (19)   1196   (19)   1204   (19)   1225   (19)
                                    1314   (21)   1321   (21)   1349   (21)   1363   (21)   1388   (21)
                                    139    (4)    1396   (21)   1425   (21)   1433   (21)   1441   (21)
                                    145    (4)    1462   (21)   154    (4)    1553   (23)   1560   (23)
                                    1588   (23)   1597   (23)   1619   (23)   1624   (23)   1633   (23)
                                    165    (4)    1654   (23)   1729   (25)   173    (4)    1736   (25)
                                    1765   (25)   1773   (25)   1804   (25)   1812   (25)   1821   (25)
                                    1921   (27)   1928   (27)   1951   (27)   1973   (27)   1981   (27)
                                    2003   (27)   230    (5)    246    (5)    259    (5)    312    (6)
                                    332    (6)    393    (8)    399    (8)    408    (8)    429    (8)
                                    447    (8)    454    (8)    465    (8)    472    (8)    477    (8)
                                    546    (10)   553    (10)   573    (10)   588    (10)   597    (10)
                                    652    (12)   671    (12)   680    (12)   692    (12)   705    (12)
                                    754    (13)   761    (13)   842    (15)   849    (15)   880    (15)
                                    889    (15)   911    (15)   916    (15)   925    (15)   996    (17)
$DS_ESCAPE        1     411    (8)  1006   (17)   1155   (19)   1324   (21)   1563   (23)   1739   (25)
                                    1931   (27)   411    (8)    450    (8)    556    (10)   674    (12)
                                    683    (12)   695    (12)   708    (12)   852    (15)
$DS_EXIT          1     121    (4)  1126   (19)   121    (4)    125    (4)    1298   (21)   1534   (23)
                                    1538   (23)   159    (4)    1713   (25)   1888   (27)   1892   (27)
                                    219    (5)    223    (5)    302    (6)    383    (8)    532    (10)
                                    560    (10)   644    (12)   648    (12)   745    (13)   749    (13)
                                    826    (15)   830    (15)   979    (17)
$DS_GETBUF_S      1     1901   (27) 1901   (27)   1904   (27)
$DS_PRINTB_S      1     1907   (27) 1907   (27)
$DS_QADEF         2     1903   (27) 1903   (27)   1906   (27)
$DS_RELBUF_S      1     2007   (27) 2007   (27)
$DS_SBTTL         1     71     (2)  1069   (17)   1230   (19)   1467   (21)   1659   (23)   181    (4)
```

```
                                            1827   (25)      265    (5)       338    (6)       481    (8)       606    (10)
                                            71     (2)       711    (12)      767    (13)      930    (15)
$DS_SECDEF          1          63     (2)   63     (2)
$DS_WAITMS_S        1          442    (8)   442    (8)
$ENRKA_SYMDEF       7          69     (2)   69     (2)
$EQU                1          1906   (27)  1903   (27)      1906   (27)      61     (2)       62     (2)       65     (2)
                                            66     (2)
$EQULS1             1          1906   (27)  1903   (27)      1906   (27)      61     (2)       65     (2)       66     (2)
$EQULST             1          61     (2)   1903   (27)      1906   (27)      61     (2)       65     (2)       66     (2)
$GBLINI             2          61     (2)   1903   (27)      1906   (27)      61     (2)       62     (2)       65     (2)
                                            66     (2)
$HDR_LAYOUTDEF      2          68     (2)   68     (2)
$IDC_REGDEF         10         67     (2)   67     (2)
$PUSHADR            1          140    (4)   1004   (17)      1029   (17)      1059   (17)      1065   (17)      1146   (19)
                                            1153   (19)      1176   (19)      1197   (19)      1205   (19)      1226   (19)
                                            1315   (21)      1322   (21)      1350   (21)      1364   (21)      1389   (21)
                                            1397   (21)      140    (4)       1426   (21)      1434   (21)      1442   (21)
                                            146    (4)       1463   (21)      155    (4)       1554   (23)      1561   (23)
                                            1589   (23)      1599   (23)      1620   (23)      1626   (23)      1634   (23)
                                            1655   (23)      166    (4)       1730   (25)      1737   (25)      174    (4)
                                            1766   (25)      1775   (25)      1805   (25)      1814   (25)      1822   (25)
                                            1902   (27)      1905   (27)      1922   (27)      1929   (27)      1952   (27)
                                            1974   (27)      1982   (27)      2004   (27)      2007   (27)      231    (5)
                                            247    (5)       260    (5)       313    (6)       333    (6)       394    (8)
                                            400    (8)       409    (8)       430    (8)       442    (8)       448    (8)
                                            455    (8)       467    (8)       473    (8)       478    (8)       547    (10)
                                            554    (10)      574    (10)      589    (10)      599    (10)      653    (12)
                                            672    (12)      681    (12)      693    (12)      706    (12)      755    (13)
                                            762    (13)      843    (15)      850    (15)      881    (15)      891    (15)
                                            912    (15)      918    (15)      926    (15)      997    (17)
$VIELD              1          61     (2)   61     (2)       65     (2)
$VIELD1             1          1906   (27)  61     (2)       65     (2)
BDEVFOUND           1          121    (4)   1126   (19)      121    (4)       1298   (21)      1534   (23)      1713   (25)
                                            1888   (27)      219    (5)       302    (6)       383    (8)       532    (10)
                                            644    (12)      745    (13)      826    (15)      979    (17)
DRV_TBL             3          66     (2)   66     (2)
ENRKADEF            1          64     (2)   64     (2)
GET_LUN             1          120    (4)   1125   (19)      120    (4)       1297   (21)      1533   (23)      1712   (25)
                                            1887   (27)      218    (5)       301    (6)       382    (8)       531    (10)
                                            643    (12)      744    (13)      825    (15)      978    (17)
STARS               1          127    (4)   1010   (17)      1033   (17)      1138   (19)      1159   (19)      1180   (19)
                                            1209   (19)      127    (4)       1307   (21)      1328   (21)      1368   (21)
                                            1401   (21)      1446   (21)      1546   (23)      1567   (23)      1603   (23)
                                            1638   (23)      1722   (25)      1743   (25)      1780   (25)      1914   (27)
                                            1935   (27)      1956   (27)      1986   (27)      237    (5)       250    (5)
                                            323    (6)       386    (8)       415    (8)       539    (10)      563    (10)
                                            835    (15)      856    (15)      895    (15)      989    (17)

                                       +--------------------------+
                                       ! Performance indicators !
                                       +--------------------------+

Phase                     Page faults    CPU Time        Elapsed Time
-----                     -----------    --------        ------------
Initialization                     25    00:00:00.12     00:00:00.67
Command processing                864    00:00:00.80     00:00:02.38
```

```
Pass 1                          2967      00:00:49.23      00:01:27.43
Symbol table sort                  6      00:00:01.15      00:00:01.78
Pass 2                           656      00:00:11.48      00:00:29.42
Symbol table output               39      00:00:00.31      00:00:00.42
Psect synopsis output              8      00:00:00.08      00:00:00.11
Cross-reference output           387      00:00:06.63      00:00:13.82
Assembler run totals            4956      00:01:09.80      00:02:16.10

The working set limit was 1024 pages.
438206 bytes (856 pages) of virtual memory were used to buffer the intermediate code.
There were 40 pages of symbol table space allocated to hold 686 non-local and 195 local symbols.
2010 source lines were read in Pass 1, producing 88 object records in Pass 2.
118 pages of virtual memory were used to define 53 macros.

                                      +----------------------------+
                                      ! Macro library statistics !
                                      +----------------------------+

Macro library name                     Macros defined
------------------                     --------------
$1$DUA9:[BOWSER.TAPE]ENRKA.MLB;16                   8
SYS$COMMON:[SYSLIB]LIB.MLB;3                        0
$1$DUA9:[DIAGLIB.LIBRARY.BLISS]DIAG.MLB;1          45
SYS$COMMON:[SYSLIB]STARLET.MLB;3                    6
TOTALS (all libraries)                             59

1022 GETS were required to define 59 macros.

There were no errors, warnings or information messages.

MACRO/LIST/CROSS ENRKA2
```

Table of contents

```
    0000          .TITLE  ENRKA3  IDC DISK FUNCTIONAL DIAGNOSTIC - SECTION 3
    0000          .IDENT  /1.1/
    0000          .LIST   MEB
    0000          .NLIST  CND
00000000          .PSECT  HEADER, PAGE, NOWRT
    0000
    0000   ;++
    0000   ; COPYRIGHT (C) 1982, 1986
    0000   ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS 01754
    0000   ;
    0000   ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON  A
    0000   ; SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLU-
    0000   ; SION OF THE ABOVE COPYRIGHT NOTICE.  THIS  SOFTWARE,  OR  ANY
    0000   ; OTHER  COPIES  THEREOF, MAY NOT BE PROVIDED OR OTHERWISE MADE
    0000   ; AVAILABLE TO ANY OTHER PERSON EXCEPT FOR USE ON  SUCH  SYSTEM
    0000   ; AND  TO  ONE WHO AGREES TO THESE LICENSE TERMS.  TITLE TO AND
    0000   ; OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN DEC.
    0000   ;
    0000   ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT
    0000   ; NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL
    0000   ; EQUIPMENT CORPORATION.
    0000   ;
    0000   ; DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF
    0000   ; ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
    0000   ;
    0000   ;++
    0000   ; FACILITY: VAX-11 DIAGNOSTIC LIBRARY
    0000   ;
    0000   ; ABSTRACT: THIS SECTION CONTAINS IDC SUBSYSTEM TESTS 32 THRU END
    0000   ;
    0000   ; ENVIRONMENT: VAX-11 DIAGNOSTIC SUPERVISOR
    0000   ;
    0000   ;--
    0000   ;++
    0000   ; AUTHOR: JOHN L. WALSH
    0000   ;
    0000   ; CREATION DATE:  MARCH 1982
    0000   ;
    0000   ; MODIFIED BY:
    0000   ;       V1.1 NO CHANGES TO THIS MODULE
    0000   ;
    0000   ;--
```

```
                0000          .SBTTL  DECLARATIONS
                0000
                0000    ;++
                0000    ; INCLUDE FILES
                0000    ;--
                0000          .LIBRARY /DISK$DIAG:[DIAGLIB.LIBRARY.BLISS]DIAG.MLB/
                0000          .LIBRARY /SYS$LIBRARY:LIB.MLB/
                0000          .LIBRARY /DISK$DIAG:[BOWSER.TAPE]ENRKA.MLB/
                0000
                0000    ;++
                0000    ; EQUATED SYMBOLS
                0000    ;--
                0000
                0000    ;
                0000    ; CREATED SYMBOLS FOR THE SUPERVISOR/PROGRAM INTERFACE
                0000    ;
                0000
                0000          $DS_BGNMOD      <SEP_REPAIR>,32
                0000    ;;;   Macro-32 Diagnostic macro library Version 8.2 "DIAG.MLB (233)"
                0000          ENRKADEF
                0000    ;;;   ENRKA MACRO LIBRARY VERSION     V1.1
                0000          $DS_DSADEF
                0000          $DS_BITDEF
                0000          $DS_CHDEF
                0000          $DS_PARDEF
                0000          $DS_SECDEF      MANUAL,ALL
                0000          $IDC_REGDEF
                0000          .SAVE   LOCAL_BLOCK
                0000          .PSECT  $ABS$,ABS
00000000  FE70                .=0
            00000000          .RESTORE
                0000          $HDR_LAYOUTDEF
                0000          .SAVE   LOCAL_BLOCK
                0000          .PSECT  $ABS$,ABS
00000000  FE70                .=0
            00000000          .RESTORE
                0000          $ENRKA_SYMDEF
                0000          .SAVE   LOCAL_BLOCK
                0000          .PSECT  $ABS$,ABS
00000000  FE70                .=0
            00000000          .RESTORE
                0000          DRV_TBL
                0000          .NLIST  MEB
                0000          $DS_SBTTL <READ DATA WITHOUT HEADER COMPARE TEST>,ALIGN=LONG
                0000          .SBTTL  TEST 32:  READ DATA WITHOUT HEADER COMPARE TEST
                0000          $D2_SBTTL       032,<READ DATA WITHOUT HEADER COMPARE TEST>,<LONG>
            00000000                  .PSECT  TEST_032, LONG, NOWRT
                0028
                0028
```

```
                                0028
                                0028  ;++
                                0028  ;
                                0028  ; TEST DESCRIPTION:
                                0028  ;
                                0028  ; THIS TEST WILL VERIFY THAT DATA CAN BE RECOVERED USING A READ WITHOUT
                                0028  ; HEADER COMPARE FUNCTION.  FIRST A WRITE DATA COMMAND IS DONE TO SECTOR
                                0028  ; THREE WITH A RANDOM DATA PATTERN. NEXT DO A WRITE DATA COMMAND TO
                                0028  ; SECTOR ONE. WHEN THIS FUNCTION COMPLETES THE HEADS WILL BE ONE SECTOR AWAY
                                0028  ; FROM SECTOR THREE. EXECUTE A READ WITH NO HEADER COMPARE FUNCTION
                                0028  ; WITH THE DISK ADDRESS REGISTER POINTING TO AN UNSELECTED TRACK ADDRESS.
                                0028  ; AND VERIFY THE THE DATA COME FROM SECTOR THREE ERROR FREE.
                                0028  ;
                                0028  ; TEST STEPS:
                                0028  ;
                                0028  ; A.  SUBTEST 1
                                0028  ;
                                0028  ;         1.  INITIALIZE SUBSYSTEM
                                0028  ;         2.  POSITION DRIVE TO TEST CYLINDER
                                0028  ;         3.  IF ERROR THEN ESCAPE TEST
                                0028  ;
                                0028  ; B.  SUBTEST 2
                                0028  ;
                                0028  ;         1.  INITIALIZE SUBSYSTEM
                                0028  ;         2.  EXECUTE WRITE DATA COMMAND TO SECTOR 3 USE RANDOM DATA.
                                0028  ;         3.  CHECK FOR ANY ERRORS
                                0028  ;
                                0028  ; C.  SUBTEST 3
                                0028  ;
                                0028  ;         1.  INITIALIZE SUBSYSTEM
                                0028  ;         2.  WRITE DATA TO SECTOR 1
                                0028  ;         3.  CHECK FOR ANY ERRORS
                                0028  ;         4.  EXECUTE READ WITH NO HEADER COMPARE FUNCTION
                                0028  ;         5.  CHECK FOR ANY ERRORS
                                0028  ;         6.  VERIFY IDCDAR UPDATED CORRECTLY
                                0028  ;         7.  VERIFY DATA READ IS CORRECT
                                0028  ;         8.  REPORT ANY DESCREPENCIES
                                0028  ;
                                0028  ; DEBUG:
                                0028  ;
                                0028  ;         SECTOR FORMAT CONTROL
                                0028  ;
                                0028  ;--
                                0028
                                0028            $DS_BGNDATA
          00000000 00000001     0028            .LONG    1,LUN0
          00000001 00000001     0030            .LONG    1,LUN1
          00000002 00000001     0038            .LONG    1,LUN2
          00000003 00000001     0040            .LONG    1,LUN3
                                0048            $DS_ENDDATA
                                004C
                                004C            $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
                                004E            GET_LUN -                                  ; GET LUN INFO
                                004E              UNIT= OFST_UNIT(AP)                      ; LOGICAL UNIT
                  04 AC   DD    004E            PUSHL    OFST_UNIT(AP)
      00000000'EF    01   FB    0051            CALLS    #1,GET_LUN$A
```

```
                                  0058    130         BDEVFOUND       5$                          ; IF DEV AVAILABLE GOTO 5$
          06 50     00       E2   0058                BBSS    #0,R0,5$
             50     01       D0   005C                        MOVL    #1,R0           ; NOT AN ERROR
                  02C9       31   005F                        BRW     TEST_032_X      ; EXIT TEST 32
   54     00000000'EF        D0   0062    131 5$:     MOVL    DRIVE_TBL,R4                        ; GET DRIVE INFO TABLE
          55     14 A4       D0   0069    132         MOVL    OFST_TSTCYL(R4),R5                  ; GET CYLINDER TO USE
          56     28 A4       D0   006D    133         MOVL    OFST_SECLEN(R4),R6                  ; GET BYTES PER SECTOR
          57     10 A4       D0   0071    134         MOVL    OFST_LSTSEC(R4),R7                  ; GET LAST SECTOR
                    59       D4   0075    135         CLRL    R9                                  ; INIT REGISTER
   59     10     10 55       F0   0077    136         INSV    R5,#16,#16,R9                       ; LOAD DESTINATION CYLINDER
          00000000'EF        D4   007C    137         CLRL    CONFIG                              ; CLEAR ANY PROGRAM FLAGS
                                  0082    138         $DS_BGNSUB
                                  008D    139         STARS
                                  008D            ;;**************************************************************
                                  008D    140     ;++
                                  008D    141     ; POSITION DRIVE TO CYLINDER 0 AND VERIFY THE SEEK
                                  008D    142     ;--
                                  008D    143
   00000000'EF      00       FB   008D    144 10$:    CALLS   #0,SUBSYS_INIT                      ; INITIALIZE SUBSYSTEM
             1D 50           E8   0094    145         BLBS    R0,20$                              ; BRANCH IF NO ERRORS
                                  0097    146         $DS_ERRHARD_S      ,L$UNIT,MSG14_SYSINIT,-
                                  0097    147                            PRINT_REG,R1
                                  00AD            ;;; TEST 32, SUBTEST 1, ERROR 1
                                  00B4    148 20$:    $DS_CKLOOP      10$
                    00       DD   00BC    149 35$:    PUSHL   #0                                  ; PUSH HEAD TO USE
                    55       DD   00BE    150         PUSHL   R5                                  ; PUSH DESTINATION CYLN
   00000000'EF      02       FB   00C0    151         CALLS   #2,POSITION_HDS                     ; EXECUTE THE SEEK
             1D 50           E8   00C7    152         BLBS    R0,40$                              ; BRANCH IF NO ERRORS
                                  00CA    153         $DS_ERRHARD_S      ,L$UNIT,@ERR_MSG,-
                                  00CA    154                            PRINT_REG,R1
                                  00E0            ;;; TEST 32, SUBTEST 1, ERROR 2
                                  00E7    155 40$:    $DS_CKLOOP      35$
                                  00EF    156         $DS_ESCAPE      TEST                        ; IF ERROR EXIT TEST
          00000000'EF        D4   00FA    157         CLRL    CONFIG                              ; CLEAR ANY PROGRAM FLAGS
                                  0100    158 50$:    $DS_ENDSUB
                                  010B    159         $DS_BGNSUB
                                  0116    160         STARS
                                  0116            ;;**************************************************************
                                  0116    161     ;++
                                  0116    162     ; DO A WRITE DATA COMMAND TO SECTOR 3
                                  0116    163     ;--
                                  0116    164
                    53       D4   0116    165         CLRL    R3                                  ; INIT INDEX
 12B9B0A1 8F      00000000'EF    7A   0118    166 5$:     EMUL    SEED,#314159265,-                   ; CALCULATE RANDOM NUMBER
         5A      1B0CB17F 8F          0123    167                 #453816703,R10                      ; PUT IN R10 AND R11
    5A   5A      7FFFFFFF 8F     7B   0129    168             EDIV    #2147483647,R10,-                   ; PUT MODULO BACK IN SEED
                 00000000'EF          0131    169                 R10,SEED
00000000'FF43    00000000'EF     D0   0136    170             MOVL    SEED,@WRITE_BUFFER[R3]              ; LOAD WRITE BUFFER
          00000000'FF43          D4   0142    171             CLRL    @READ_BUFFER[R3]                    ; CLEAR READ BUFFER
       C7 53     00000080 8F     F2   0149    172             AOBLSS  #128,R3,5$                          ; DO 256 WORDS
                          56     DD   0151    173             PUSHL   R6                                  ; GET BYTE COUNT
                 00000000'EF     7F   0153    174             PUSHAQ  WRITE_BUFFER                        ; GET BUFFER ADDRESS
   00000000'EF      02       FB   0159    175         CALLS   #2,$GS_SETMAP                       ; MAP UBA
             59     03       90   0160    176 20$:    MOVB    #3,R9                               ; SET DESTINATION TO SECTOR 3
                    00       DD   0163    177         PUSHL   #0                                  ; IDCMPR DATA
                    59       DD   0165    178         PUSHL   R9                                  ; IDCDAR (SECTOR 0)
```

```
               7E   56   CE  0167   179          MNEGL   R6,-(SP)                  ; IDCCNT (BYTE COUNT)
                    00   DD  016A   180          PUSHL   #0                        ; IDCBAR DATA
                    0A   DD  016C   181          PUSHL   #WRITE                    ; DEFINE COMMAND
         00000000'EF   05   FB  016E   182          CALLS   #5,START_CMD              ; EXECUTE WRITE DATA CMD
         00000000'EF   00   FB  0175   183          CALLS   #0,STATUS_CHK             ; CHECK FOR ANY ERRORS
               1D   50   E8  017C   184          BLBS    R0,30$                    ; BRANCH IF NONE
                             017F   185          $DS_ERRHARD_S    ,L$UNIT,MSG20_WRITE,-
                             017F   186                           PRINT_REG,R1
                             0195       ;;; TEST 32, SUBTEST 2, ERROR 1
                             019C   187 30$:     $DS_CKLOOP       20$
                             01A4   188          $DS_ENDSUB
                             01AF   189          $DS_BGNSUB
                             01BA   190          STARS
                             01BA       ;;*************************************************************
                             01BA   191 ;++
                             01BA   192 ; DO A WRITE DATA FUNCTION TO SECTOR 1.  WHEN THE
                             01BA   193 ; WRITE FUNCTION IS COMPLETE HEADS WILL BE ONE SECTOR AWAY
                             01BA   194 ; FROM SECTOR 3. THEN PERFORM THE READ DATA WITHOUT
                             01BA   195 ; HEADER COMPARE, THIS SHOULD READ SECTOR THREE. CHECK
                             01BA   196 ; THAT THE DATA READ BACK WAS FROM SECTOR THREE.
                             01BA   197 ;--
                             01BA   198
                    56   DD  01BA   199          PUSHL   R6                        ; GET BYTE COUNT
           00000000'EF   7F  01BC   200          PUSHAQ  READ_BUFFER               ; GET ADRS OF READ BUFFER
         00000000'EF   02   FB  01C2   201          CALLS   #2,$GS_SETMAP             ; MAP UBA
          55   59   02   C3  01C9   202          SUBL3   #2,R9,R5                  ; SET DESTINATION FOR SECTOR 1
     59   08   08   01   F0  01CD   203          INSV    #1,#8,#8,R9               ; CHOOSE AN UNSELECTED TRACK
               5A   0E   9A  01D2   204          MOVZBL  #RD_NHC,R10               ; LOAD READ NO HEADER COMPARE COMM -
cont> AND
       5A   00000000'EF   C8  01D5   205          BISL2   P$UNIT,R10                ; OR IN DRIVE SELECT BITS
          5B   59   01   C1  01DC   206          ADDL3   #1,R9,R11                 ; SET EXPECTED DAR DATA
                    00   DD  01E0   207 20$:     PUSHL   #0                        ; IDCMPR DATA
                    55   DD  01E2   208          PUSHL   R5                        ; IDCDAR (LAST SECTOR - 1)
               7E   56   CE  01E4   209          MNEGL   R6,-(SP)                  ; IDCCNT (BYTE COUNT)
                    00   DD  01E7   210          PUSHL   #0                        ; IDCBAR DATA
                    0A   DD  01E9   211          PUSHL   #WRITE                    ; DEFINE COMMAND
         00000000'EF   05   FB  01EB   212          CALLS   #5,START_CMD              ; EXECUTE WRITE DATA CMD
                    55   D4  01F2   213          CLRL    R5                        ; CLEAR WAIT COUNT
       62   00000080 8F   D3  01F4   214 25$:     BITL    #CSR$M_CRDY,IDCCSR(R2)    ; IF READY SET
                    3E   12  01FB   215          BNEQ    30$                       ; THEN GOTO 30$
    EF 55   00010000 8F   F2  01FD   216          AOBLSS  #^X10000,R5,25$           ; TRY FOR READ AGAIN
           00000000'EF   16  0205   217          JSB     RD_IDCREGS                ; SNAPSHOT REGISTERS
                             020B   218          $DS_ERRHARD_S    ,L$UNIT,MSG20_WRITE,-
                             020B   219                           PRINT_REG,R1
                             0221       ;;; TEST 32, SUBTEST 3, ERROR 1
                             0228   220          $DS_CKLOOP       20$
                             0230   221          $DS_ESCAPE       TEST             ; LEAVE TEST IF ERROR
               10 A2   D4  023B   222 30$:     CLRL    IDCMPR(R2)                ; CLEAR MULTI PURPOSE REG
               04 A2   D4  023E   223          CLRL    IDCBAR(R2)                ; LOAD BUS ADDRESS REG
          08 A2   56   CE  0241   224          MNEGL   R6,IDCCNT(R2)             ; LOAD BYTE COUNT
  62   5A   000F0080 8F   C9  0245   225          BISL3   #^XF0080,R10,IDCCSR(R2)   ; LOAD FOR DECODE OF DAR REG
          0C A2   59   D0  024D   226          MOVL    R9,IDCDAR(R2)             ; LOAD DESIRED DISK ADDRESS
               62   5A   D0  0251   227          MOVL    R10,IDCCSR(R2)            ; EXECUTE READ NO HDR CMP CMD
         00000000'EF   0E   D0  0254   228          MOVL    #RD_NHC,CURR_CMD          ; SETUP CURRENT COMMAND
         00000000'EF   00   FB  025B   229          CALLS   #0,STATUS_CHK             ; CHECK FOR ANY ERRORS
               1D   50   E8  0262   230          BLBS    R0,50$                    ; BRANCH IF NONE
                             0265   231          $DS_ERRHARD_S    ,L$UNIT,MSG22_READNHC,-
```

```
                                0265    232                     PRINT_REG,R1
                                027B        ;;; TEST 32, SUBTEST 3, ERROR 2
                                0282    233 50$:    $DS_CKLOOP  30$
                      5B    DD  028A    234         PUSHL   R11                     ; GET EXPECTED DAR DATA
          00000000'EF  01   FB  028C    235         CALLS   #1,CHANGE_DAR$A         ; CONVERT TO EXPECTED READ BACK VA -
cont> LUE
                  5B  51    D0  0293    236         MOVL    R1,R11                  ; GET NEW EXPECTED DAR VALUE
          00000000'EF  5B   D1  0296    237         CMPL    R11,T.DA                ; DID IDCDAR UPDATA OK ?
                      2F    13  029D    238         BEQL    60$                     ; BRANCH IF IT DID
                                029F    239         $DS_ERRHARD_S   ,L$UNIT,MSG22_READNHC,-
                                029F    240                     REG_ERRPT,#MSG54_DA_UPD,-
                                029F    241                     R11,T.DA,DAR_MNEM
                                02C7        ;;; TEST 32, SUBTEST 3, ERROR 3
                                02CE    242 60$:    $DS_CKLOOP  20$
              00000000'EF   D4  02D7    243         CLRL    T.DA                    ; CLR DA FOR ERR RPT
              00000000'EF   DD  02DD    244         PUSHL   WRITE_BUFFER            ; GET GOOD DATA
              00000000'EF   DD  02E3    245         PUSHL   READ_BUFFER             ; GET DATA JUST READ
              00000000'EF   DD  02E9    246         PUSHL   WORD_COUNT              ; GET WORD COUNT
          00000000'EF  03   FB  02EF    247         CALLS   #3,CMPDATA              ; CHECK THE DATA
                   1B 50    E8  02F6    248         BLBS    R0,70$                  ; BRANCH IF DATA OK
                                02F9    249         $DS_ERRHARD_S   ,L$UNIT,MSG69_DAT_NHC,-
                                02F9    250                     PRINT_DATA_ERR
                                030D        ;;; TEST 32, SUBTEST 3, ERROR 4
                                0314    251 70$:    $DS_CKLOOP  20$
                                031D    252         $DS_ENDSUB
                                0328    253         $DS_ENDTEST
                                0333    254         $DS_SBTTL <OVERLAP SEEK TEST>,ALIGN=LONG
                                0333                .SBTTL  TEST 33:  OVERLAP SEEK TEST
                                0333                $D2_SBTTL       033,<OVERLAP SEEK TEST>,<LONG>
                            00000000                        .PSECT  TEST_033, LONG, NOWRT
                                0014
```

```
                                 0014   256
                                 0014   257 ;++
                                 0014   258 ;
                                 0014   259 ; TEST DESCRIPTION:
                                 0014   260 ;
                                 0014   261 ; THIS TEST CHECKS THAT OVERLAPPED SEEK OPERATIONS CAN BE
                                 0014   262 ; PERFORMED BY THE IDC. THIS TEST IS ONLY RUN WHEN AT LEAST
                                 0014   263 ; TWO DRIVES ARE SELECTED FOR TESTING.
                                 0014   264 ;
                                 0014   265 ; TEST STEPS:
                                 0014   266 ;
                                 0014   267 ;         1.  INITIALIZE SUBSYSTEM
                                 0014   268 ;         2.  POSITION DRIVES TO CYLINDER 0
                                 0014   269 ;         3.  POSITION SECONDARY DRIVE TO IT MIDPOINT
                                 0014   270 ;             AND DO NOT WAIT FOR DRIVE READY.
                                 0014   271 ;         4.  SEEK DRIVE UNDER TEST TO ITS LAST CYLINDER
                                 0014   272 ;         5.  VERIFY THAT DRIVE UNDER TEST COMPLETED ITS
                                 0014   273 ;             SEEK WITHOUT ERRORS (INTERFERENCE FROM OTHER DRIVE).
                                 0014   274 ;
                                 0014   275 ; DEBUG:
                                 0014   276 ;
                                 0014   277 ;--
                                 0014   278
                                 0014   279         $DS_BGNDATA
          00000000 00000001      0014   280         .LONG   1,LUN0
          00000001 00000001      001C   281         .LONG   1,LUN1
          00000002 00000001      0024   282         .LONG   1,LUN2
          00000003 00000001      002C   283         .LONG   1,LUN3
                                 0034   284         $DS_ENDDATA
                                 0038   285
                                 0038   286         $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
                                 003A   287         GET_LUN -                               ; GET LUN INFO
                                 003A   288           UNIT= OFST_UNIT(AP)                   ; LOGICAL UNIT
                04 AC     DD     003A               PUSHL   OFST_UNIT(AP)
        00000000'EF    01 FB     003D               CALLS   #1,GET_LUN$A
                                 0044   289         BDEVFOUND       5$                      ; IF DEV AVAILABLE GOTO 5$
             06 50     00 E2     0044               BBSS    #0,R0,5$
                50     01 D0     0048                       MOVL    #1,R0           ; NOT AN ERROR
                0161      31     004B                       BRW     TEST_033_X      ; EXIT TEST 33
     54     00000000'EF   D0     004E   290 5$:     MOVL    DRIVE_TBL,R4                    ; GET DRIVE INFO TABLE
            00000000'EF   D4     0055   291         CLRL    CONFIG                          ; CLEAR ANY PROGRAM FLAGS
             57     28 A4 D0     005B   292         MOVL    OFST_SECLEN(R4),R7              ; GET BYTES PER SECTOR
             58     08 A4 D0     005F   293         MOVL    OFST_LSTCYL(R4),R8              ; GET TEST CYLINDER
                                 0063   294         GET_SECLUN -                            ; FIND ONE OTHER DISK TO USE
                                 0063   295           EXUNIT=       OFST_UNIT(AP)           ; BUT NOT THIS ONE
                04 AC     DD     0063               PUSHL   OFST_UNIT(AP)
        00000000'EF    01 FB     0066               CALLS   #1,GET_SECLUN$A
                09 50     E8     006D   296         BLBS    R0,10$                          ; IF ANOTHER DEVICE IS AVAILABLE G -
cont> OTO 10$
                50     01 D0     0070   297         MOVL    #1,R0                           ; ELSE INIT R0
                                 0073   298         $DS_EXIT        TEST                    ; EXIT THIS TEST
     55     00000000'EF   D0     0079   299 10$:    MOVL    S_DRIVE_TBL,R5                  ; GET ADDRESS OF DRIVE TABLE
      55     08 A5     02 C7     0080   300         DIVL3   #2,OFST_LSTCYL(R5),R5           ; FIND MID POINT OF DISK
                                 0085   301         STARS
                                 0085       ;;*****************************************************************
                                 0085   302 ;++
                                 0085   303 ; POSITION BOTH DRIVES TO CYLINDER ZERO AND VERIFY THE SEEK
```

```
                                 0085   ;--
                                 0085
         00000000'EF   00   FB   0085   15$:    CALLS   #0,SUBSYS_INIT                  ; INITIALIZE SUBSYSTEM
                 1D 50      E8   008C           BLBS    R0,20$                          ; BRANCH IF NO ERRORS
                                 008F           $DS_ERRHARD_S     ,L$UNIT,MSG14_SYSINIT,-
                                 008F                             PRINT_REG,R1
                                 00A5   ;;; TEST 33, SUBTEST 0, ERROR 1
                                 00AC   20$:    $DS_CKLOOP        15$
                      00    DD   00B4   35$:    PUSHL   #0                              ; PUSH HEAD TO USE
                      00    DD   00B6           PUSHL   #0                              ; PUSH DESTINATION CYLN
         00000000'EF   02   FB   00B8           CALLS   #2,POSITION_HDS                 ; EXECUTE THE SEEK
                 1D 50      E8   00BF           BLBS    R0,40$                          ; BRANCH IF NO ERRORS
                                 00C2           $DS_ERRHARD_S     ,L$UNIT,@ERR_MSG,-
                                 00C2                             PRINT_REG,R1
                                 00D8   ;;; TEST 33, SUBTEST 0, ERROR 2
                                 00DF   40$:    $DS_CKLOOP        35$
                                 00E7           $DS_ESCAPE        TEST                  ; IF ERROR EXIT TEST
            00000000'EF     D4   00F2           CLRL    CONFIG                          ; CLEAR ANY PROGRAM FLAGS
  00000000'EF  00000086 8F  C9   00F8           BISL3   #<CSR$M_CRDY!SEEK>,S_P$UNIT,-   ; SETUP DRIVE 2 FOR SEEK
                      62         0103                   IDCCSR(R2)
                   0C A2    D4   0104           CLRL    IDCDAR(R2)                      ; TO CYLINDER 0
          62   00000080 8F  CA   0107           BICL2   #CSR$M_CRDY,IDCCSR(R2)          ; START THE SEEK
                                 010E           $DS_WAITMS_S      #50                   ; WAIT FOR SEEK TO COMPLETE
                                 0119           STARS
                                 0119   ;;*******************************************************************
                                 0119   ;++
                                 0119   ; START SEEK ON SECOND DRIVE TO ITS MIDPOINT.
                                 0119   ; THEN START SEEK ON UNIT UNDER TEST TO ITS TEST
                                 0119   ; CYLINDER.  CHECK STATUS AT COMPLETION OF FUNCTION
                                 0119   ;--
                                 0119
                      59    D4   0119           CLRL    R9                              ; RESET WORK REGISTER
        59    10    10    55 F0  011B           INSV    R5,#16,#16,R9                   ; SET UP DAR FOR DRIVE 2
  00000000'EF  00000086 8F  C9   0120           BISL3   #<CSR$M_CRDY!SEEK>,S_P$UNIT,-   ; SETUP DRIVE 2 FOR SEEK
                      62         012B                   IDCCSR(R2)
                 0C A2 59   D0   012C           MOVL    R9,IDCDAR(R2)                   ; TO MID CYLINDER
          62   00000080 8F  CA   0130           BICL2   #CSR$M_CRDY,IDCCSR(R2)          ; START THE SEEK
                      53    D4   0137           CLRL    R3                              ; RESET COUNTER
          62   00000080 8F  D3   0139   45$:    BITL    #CSR$M_CRDY,IDCCSR(R2)          ; IF CRDY SET
                      2C    12   0140           BNEQ    50$                             ; THEN GOTO 50$
       EF 53   0000FFFF 8F  F2   0142           AOBLSS  #^XFFFF,R3,45$                  ; TRY AGAIN
               00000000'EF  16   014A           JSB     RD_IDCREGS                      ; SNAPSHOT REGISTERS
                                 0150           $DS_ERRHARD_S     ,L$UNIT,MSG30A_CRDY,-
                                 0150                             PRINT_REG
                                 0164   ;;; TEST 33, SUBTEST 0, ERROR 3
                                 016B           $DS_ABORT         TEST                  ; LEAVE THIS TEST
                      00    DD   016E   50$:    PUSHL   #0                              ; USE HEAD 0
                      58    DD   0170           PUSHL   R8                              ; DESTINATION CYLINDER
         00000000'EF   02   FB   0172           CALLS   #2,POSITION_HDS                 ; START SEEK
                 1D 50      E8   0179           BLBS    R0,60$                          ; IF NO ERROR GOTO 60$
                                 017C           $DS_ERRHARD_S     ,L$UNIT,MSG18B_SEEK,-
                                 017C                             PRINT_REG,R1
                                 0192   ;;; TEST 33, SUBTEST 0, ERROR 4
                                 0199   60$:    $DS_CKLOOP        50$
                                 01A1           $DS_WAITMS_S      #8                    ; WAIT FOR OTHER DRIVE TO FINISH
                                 01AC           $DS_ENDTEST
```

```
        01B7   355            $DS_SBTTL <DMA BOUNDRY TEST>,ALIGN=LONG
        01B7                  .SBTTL  TEST 34:  DMA BOUNDRY TEST
        01B7                  $D2_SBTTL       034,<DMA BOUNDRY TEST>,<LONG>
    00000000                          .PSECT  TEST_034, LONG, NOWRT
        0014
```

```
                              0014
                              0014      ;++
                              0014      ;
                              0014      ; TEST DESCRIPTION:
                              0014      ;
                              0014      ; TEST THAT THE IDC IS CAPABLE OF HANDLING ODD BYTE
                              0014      ; COUNTS, ODD ADDRESS WHILE PROCESSING PARTIAL SECTOR
                              0014      ; SECTOR TRANSFERS THAT CROSS PAGE BOUNDRIES.
                              0014      ;
                              0014      ; TEST STEPS:
                              0014      ;
                              0014      ; A.   SUBTEST 1
                              0014      ;
                              0014      ;           1.  INITIALIZE SUBSYSTEM
                              0014      ;           2.  EXECUTE SEEK TO TEST CYLINDER
                              0014      ;           3.  IF ERROR THEN ESCAPE TEST
                              0014      ;
                              0014      ; B.   SUBTEST 2
                              0014      ;
                              0014      ;           1.  INITIALIZE SUBSYSTEM
                              0014      ;           2.  EXECUTE WRITE DATA COMMAND
                              0014      ;           3.  CHECK FOR ANY ERRORS
                              0014      ;
                              0014      ; C.   SUBTEST 3
                              0014      ;
                              0014      ;           1.  INITIALIZE SUBSYSTEM
                              0014      ;           2.  EXECUTE READ DATA COMMAND
                              0014      ;           3.  CHECK FOR ANY ERRORS
                              0014      ;           4.  VERIFY DATA IS CORRECT - REPORT ANY DESCREPENCIES
                              0014      ;
                              0014      ; DEBUG:
                              0014      ;
                              0014      ; IDC MODULE
                              0014      ;
                              0014      ;--
                              0014
                              0014              $DS_BGNDATA
         00000000 00000001    0014              .LONG    1,LUN0
         00000001 00000001    001C              .LONG    1,LUN1
         00000002 00000001    0024              .LONG    1,LUN2
         00000003 00000001    002C              .LONG    1,LUN3
                              0034              $DS_ENDDATA
                              0038
                              0038              $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
                              003A              GET_LUN -                                   ; GET LUN INFO
                              003A                UNIT= OFST_UNIT(AP)                       ; LOGICAL UNIT
               04 AC    DD    003A              PUSHL   OFST_UNIT(AP)
     00000000'EF   01   FB    003D              CALLS   #1,GET_LUN$A
                              0044              BDEVFOUND        5$                         ; IF DEV AVAILABLE GOTO 5$
            06 50  00   E2    0044              BBSS    #0,R0,5$
               50  01   D0    0048                      MOVL    #1,R0          ; NOT AN ERROR
                 022B   31    004B                      BRW     TEST_034_X     ; EXIT TEST 34
      54   00000000'EF  D0    004E      5$:     MOVL    DRIVE_TBL,R4                        ; GET DRIVE INFO TABLE
           00000000'EF  D4    0055      6$:     CLRL    CONFIG                              ; CLEAR ANY PROGRAM FLAGS
               53  02   D0    005B              MOVL    #2,R3                               ; SET UP FOR DOUBLE SECTOR TRANSFER
               01  64   91    005E              CMPB    OFST_DRVTYP(R4),#RL02$T             ; IF RL02 TYPE DRIVE
```

```
                        03   13  0061   408          BEQL    7$                              ; THEN GOTO 7$
                  53    01   D0  0063   409          MOVL    #1,R3                           ; ELSE SINGLE SECTOR TRANSFER FOR  -
cont> RL02
                                 0066   410
      57    0F    28 A4   53   7A  0066   411 7$:    EMUL    R3,OFST_SECLEN(R4),#15,R7       ; CALC START BYTE COUNT
            58    14 A4   D0  006C   412             MOVL    OFST_TSTCYL(R4),R8              ; GET TEST CYLINDER
                  5A    0F   D0  0070   413          MOVL    #15,R10                         ; SETUP STARTING BUS ADDRESS REG
                                 0073   414          $DS_BGNSUB
                                 007E   415          STARS
                                 007E        ;;*****************************************************************
                                 007E   416 ;++
                                 007E   417 ; POSITION DRIVE TO TEST CYLINDER AND VERIFY THE SEEK
                                 007E   418 ;--
                                 007E   419
       00000000'EF    00   FB  007E   420 10$:    CALLS   #0,SUBSYS_INIT                  ; INITIALIZE SUBSYSTEM
                 1D 50    E8  0085   421          BLBS    R0,20$                          ; BRANCH IF NO ERRORS
                                 0088   422          $DS_ERRHARD_S   ,L$UNIT,MSG14_SYSINIT,-
                                 0088   423                          PRINT_REG,R1
                                 009E        ;;; TEST 34, SUBTEST 1, ERROR 1
                                 00A5   424 20$:    $DS_CKLOOP      10$
                        00   DD  00AD   425 35$:    PUSHL   #0                              ; PUSH HEAD TO USE
                        58   DD  00AF   426          PUSHL   R8                              ; PUSH DESTINATION CYLN
       00000000'EF    02   FB  00B1   427          CALLS   #2,POSITION_HDS                 ; EXECUTE THE SEEK
                 1D 50    E8  00B8   428          BLBS    R0,40$                          ; BRANCH IF NO ERRORS
                                 00BB   429          $DS_ERRHARD_S   ,L$UNIT,@ERR_MSG,-
                                 00BB   430                          PRINT_REG,R1
                                 00D1        ;;; TEST 34, SUBTEST 1, ERROR 2
                                 00D8   431 40$:    $DS_CKLOOP      35$
                                 00E0   432          $DS_ESCAPE      TEST                    ; IF ERROR EXIT TEST
          00000000'EF    D4  00EB   433          CLRL    CONFIG                          ; CLEAR ANY PROGRAM FLAGS
                                 00F1   434 50$:    $DS_ENDSUB
                                 00FC   435          $DS_BGNSUB
                                 0107   436          STARS
                                 0107        ;;*****************************************************************
                                 0107   437 ;++
                                 0107   438 ; DO A WRITE DATA COMMAND USING SPECIFIED STARTING
                                 0107   439 ; ADDRESS AND BUTE COUNT
                                 0107   440 ;--
                                 0107   441
                        53   D4  0107   442 1$:     CLRL    R3                              ; INIT INDEX
  11111111 11111111 8F   7D  0109   443 5$:     MOVQ    #^X1111111111111111,-
          00000000'FF43        0113   444                  @WRITE_BUFFER[R3]              ; LOAD WRITE BUFFER WITH DATA
          00000000'FF43   7C  0119   445          CLRQ    @READ_BUFFER[R3]               ; CLEAR READ BUFFER
  E1 53    00000080 8F   F2  0120   446          AOBLSS  #128,R3,5$                      ; DO 512 WORDS
           00000400 8F   DD  0128   447          PUSHL   #1024                           ; GET BYTE COUNT
           00000000'EF   7F  012E   448          PUSHAQ  WRITE_BUFFER                    ; GET BUFFER ADDRESS
       00000000'EF    02   FB  0134   449          CALLS   #2,$GS_SETMAP                   ; MAP FOR 2 SECTORS
                        59   D4  013B   450          CLRL    R9                              ; CLEAR REG
     59    10    10     58   F0  013D   451          INSV    R8,#16,#16,R9                   ; SET DESTINATION CYLINDER
                        00   DD  0142   452 25$:    PUSHL   #0                              ; IDCMPR DATA
                        59   DD  0144   453          PUSHL   R9                              ; IDCDAR DATA
                  7E    57   CE  0146   454          MNEGL   R7,-(SP)                        ; IDCCNT (BYTE COUNT)
                        5A   DD  0149   455          PUSHL   R10                             ; IDCBAR DATA
                        0A   DD  014B   456          PUSHL   #WRITE                          ; DEFINE COMMAND
       00000000'EF    05   FB  014D   457          CALLS   #5,START_CMD                    ; EXECUTE WRITE DATA CMD
       00000000'EF    00   FB  0154   458          CALLS   #0,STATUS_CHK                   ; CHECK FOR ANY ERRORS
                 1D 50    E8  015B   459          BLBS    R0,30$                          ; BRANCH IF NONE
```

```
                                015E   460            $DS_ERRHARD_S     ,L$UNIT,MSG20_WRT_BND,-
                                015E   461                              PRINT_REG,R1
                                0174        ;;; TEST 34, SUBTEST 2, ERROR 1
                                017B   462 30$:       $DS_CKLOOP        25$
                                0183   463            STARS
                                0183        ;;***********************************************************************
                                0183   464 ;++
                                0183   465 ; EXECUTE A READ DATA COMMAND FOR AND VERIFY SECTOR FILL
                                0183   466 ;--
                                0183   467
          00000400 8F     DD    0183   468            PUSHL     #1024                          ; GET BYTE COUNT
          00000000'EF     7F    0189   469            PUSHAQ    READ_BUFFER                    ; GET ADRS OF READ BUFFER
      00000000'EF    02   FB    018F   470            CALLS     #2,$GS_SETMAP                  ; MAP UBA DATA PATH #1
                     00   DD    0196   471 40$:       PUSHL     #0                             ; IDCMPR DATA
                     59   DD    0198   472            PUSHL     R9                             ; IDCDAR = SECTOR 0
         7E   00000400 8F CE    019A   473            MNEGL     #1024,-(SP)                    ; IDCCNT (BYTE COUNT)
                     00   DD    01A1   474            PUSHL     #0                             ; IDCBAR DATA
                     0C   DD    01A3   475            PUSHL     #READ                          ; DEFINE COMMAND
      00000000'EF    05   FB    01A5   476            CALLS     #5,START_CMD                   ; EXECUTE READ DATA CMD
      00000000'EF    00   FB    01AC   477            CALLS     #0,STATUS_CHK                  ; CHECK FOR ANY ERRORS
                  1D 50   E8    01B3   478            BLBS      R0,50$                         ; BRANCH IF NONE
                                01B6   479            $DS_ERRHARD_S     ,L$UNIT,MSG21_READ,-
                                01B6   480                              PRINT_REG,R1
                                01CC        ;;; TEST 34, SUBTEST 2, ERROR 2
                                01D3   481 50$:       $DS_CKLOOP        40$
                                01DB   482            ;++
                                01DB   483            ; MAKE WRITE BUFFER LOOD LIKE EXPECTED
                                01DB   484            ; READ BUFFER.
                                01DB   485            ;--
                                01DB   486
             083F 8F      BB    01DB   487            PUSHR     #^M<R0,R1,R2,R3,R4,R5,R11>     ; SAVE REGISTERS
   5B    57   00000200 8F C3    01DF   488            SUBL3     #512,R7,R11                    ; GET NUMBER OF BYTES OVER 512
         56   00000000'EF D0    01E7   489            MOVL      WRITE_BUFFER,R6                ; GET DATA ADDRESS
         59   00000200 8F D0    01EE   490            MOVL      #512,R9                        ; AND OFFSET VALUE
 00     00000000'FF  5B   2C    01F5   491            MOVC5     R11,@WRITE_BUFFER,#0,#512,(R6)[R9] ; LOAD BUFFER
       6649     0200 8F         01FD
             083F 8F      BA    0202   492            POPR      #^M<R0,R1,R2,R3,R4,R5,R11>     ; RESTORE REGISTERS
00000000'EF   00000180 8F D0    0206   493            MOVL      #384,WORD_COUNT                ; SAVE WORD COUNT FOR RL02
                01   64   91    0211   494            CMPB      OFST_DRVTYP(R4),#RL02$T        ; IF DRIVE IS AN RL02
                     0B   13    0214   495            BEQL      55$                            ; THEN GOTO 55$
00000000'EF   00000200 8F D0    0216   496            MOVL      #512,WORD_COUNT                ; ELSE TEST 512 WORDS FOR R80
          00000000'EF     DD    0221   497 55$:       PUSHL     WRITE_BUFFER                   ; GET GOOD DATA
          00000000'EF     DD    0227   498            PUSHL     READ_BUFFER                    ; GET DATA JUST READ
          00000000'EF     DD    022D   499            PUSHL     WORD_COUNT                     ; GET WORD COUNT
      00000000'EF    03   FB    0233   500            CALLS     #3,CMPDATA                     ; CHECK THE DATA
                  1B 50   E8    023A   501            BLBS      R0,60$                         ; BRANCH IF DATA OK
                                023D   502            $DS_ERRHARD_S     ,L$UNIT,MSG55_DAT_BND,-
                                023D   503                              PRINT_DATA_ERR
                                0251        ;;; TEST 34, SUBTEST 2, ERROR 3
                                0258   504 60$:       $DS_CKLOOP        40$
                     57   D7    0261   505            DECL      R7                             ; REDUCE BYTE COUNT
   FE9C 5A    FFFF 8F  01 3D    0263   506            ACBW      #1,#-1,R10,1$                  ; AND STARTING ADDRESS
                                026B   507            $DS_ENDSUB
                                0276   508            $DS_ENDTEST
                                0281   509            $DS_SBTTL <WRITE FORMAT FUNCTION TEST - R80>,ALIGN=LONG
                                0281                  .SBTTL  TEST 35:  WRITE FORMAT FUNCTION TEST - R80
```

```
        0281                $D2_SBTTL        035,<WRITE FORMAT FUNCTION TEST - R80>,<LONG>
    00000000                         .PSECT  TEST_035, LONG, NOWRT
        0024
```

```
                              0024  ;++
                              0024  ;
                              0024  ; TEST DESCRIPTION:
                              0024  ;
                              0024  ; THIS TEST WILL EXERCISE THE THE WRITE FORMAT FUNCTION.
                              0024  ; ALL CYLINDERS ON THE TEST CYLINDER ARE FORMATTED AND VERIFY
                              0024  ; BY READING ALL HEADERS BACK SEQUENTIALLY.
                              0024  ;
                              0024  ; TEST STEPS:
                              0024  ;
                              0024  ; A.  SUBTEST 1
                              0024  ;
                              0024  ;          1. INITIALIZE SUBSYSTEM
                              0024  ;          2.  EXECUTE SEEK TO TEST CYLINDER
                              0024  ;          3.  IF ERROR THEN ESCAPE TEST
                              0024  ;
                              0024  ; B.  SUBTEST 2
                              0024  ;
                              0024  ;          1. INITIALIZE SUBSYSTEM
                              0024  ;          2. BUILD HEADERS FOR TRACK
                              0024  ;          3. FORMAT TRACK
                              0024  ;          4. CHECK FOR ERRORS
                              0024  ;          5. VERIFY CORRECTNESS OF HEADERS
                              0024  ;
                              0024  ; DEBUG:
                              0024  ;
                              0024  ;
                              0024  ;--
                              0024
                              0024
                              0024             $DS_BGNDATA
         00000000 00000001    0024             .LONG    1,LUN0
         00000001 00000001    002C             .LONG    1,LUN1
         00000002 00000001    0034             .LONG    1,LUN2
         00000003 00000001    003C             .LONG    1,LUN3
                              0044             $DS_ENDDATA
                              0048
                              0048             $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
                              004A             GET_LUN -                                  ; GET LUN INFO
                              004A               UNIT= OFST_UNIT(AP)                      ; LOGICAL UNIT
                04 AC    DD   004A             PUSHL    OFST_UNIT(AP)
      00000000'EF   01   FB   004D             CALLS    #1,GET_LUN$A
                              0054             BDEVFOUND         5$                       ; IF DEV AVAILABLE GOTO 5$
           06 50    00   E2   0054             BBSS     #0,R0,5$
              50    01   D0   0058                      MOVL     #1,R0         ; NOT AN ERROR
                  01B0   31   005B                      BRW      TEST_035_X    ; EXIT TEST 35
                              005E  5$:        $DS_EXIT          TEST                     ; ****TEMP INSTRUCTION****
    54    00000000'EF    D0   0064             MOVL     DRIVE_TBL,R4                      ; GET DRIVE INFO TABLE
              02    64   91   006B             CMPB     OFST_DRVTYP(R4),#R80$T            ; IF DRIVE TYPE = R80
                    06   13   006E             BEQL     6$                                ; THEN GOTO 6$
                              0070             $DS_EXIT          TEST                     ; ELSE EXIT THIS TEST
         55     14 A4    D0   0076  6$:        MOVL     OFST_TSTCYL(R4),R5                ; GET TEST CYLINDER
                    57   D4   007A             CLRL     R7                                ; INIT REGISTER
   57    10     10  55   F0   007C             INSV     R5,#16,#16,R7                     ; SET UP DESTINATION CYLN
   57    08     08  0D   F0   0081             INSV     #13,#8,#8,R7                      ; AND DESTINATION TRACK = 12
   56    00000080 8F     D0   0086             MOVL     #<4*32>,R6                        ; BYTE COUNT FOR FORMAT COMMAND
```

```
             00000000'EF    D4  008D              CLRL    CONFIG                          ; CLEAR ANY PROGRAM FLAGS
                      53    D4  0093              CLRL    R3                              ; CLEAR INDEX
                                0095              $DS_BGNSUB
                                00A0              STARS
                                00A0      ;;*******************************************************************
                                00A0      ;++
                                00A0      ; POSITION DRIVE TO TEST CYLINDER AND VERIFY THE SEEK
                                00A0      ;--
                                00A0
         00000000'EF    00  FB  00A0      10$:    CALLS   #0,SUBSYS_INIT                  ; INITIALIZE SUBSYSTEM
                   1D 50    E8  00A7              BLBS    R0,20$                          ; BRANCH IF NO ERRORS
                                00AA              $DS_ERRHARD_S    ,L$UNIT,MSG14_SYSINIT,-
                                00AA                               PRINT_REG,R1
                                00C0      ;;; TEST 35, SUBTEST 1, ERROR 1
                                00C7      20$:    $DS_CKLOOP       10$
                      0D    DD  00CF      35$:    PUSHL   #13                             ; PUSH HEAD TO USE
                      55    DD  00D1              PUSHL   R5                              ; PUSH DESTINATION CYLN
         00000000'EF    02  FB  00D3              CALLS   #2,POSITION_HDS                 ; EXECUTE THE SEEK
                   1D 50    E8  00DA              BLBS    R0,40$                          ; BRANCH IF NO ERRORS
                                00DD              $DS_ERRHARD_S    ,L$UNIT,@ERR_MSG,-
                                00DD                               PRINT_REG,R1
                                00F3      ;;; TEST 35, SUBTEST 1, ERROR 2
                                00FA      40$:    $DS_CKLOOP       35$
                                0102              $DS_ESCAPE       TEST                   ; EXIT TEST IF ERROR
             00000000'EF    D4  010D              CLRL    CONFIG                          ; CLEAR ANY PROGRAM FLAGS
                                0113      50$:    $DS_ENDSUB
                                011E              $DS_BGNSUB
                                0129              STARS
                                0129      ;;*******************************************************************
                                0129      ;++
                                0129      ; BUILD HEADERS AND EXECUTE WRITE FORMAT FUNCTION
                                0129      ;--
                                0129
                      00    DD  0129              PUSHL   #0                              ; NO SKIP SECTORS PLEASE
                      0D    DD  012B              PUSHL   #13                             ; BUILD HDRS FOR TRACK 13
                   14 A4    DD  012D              PUSHL   OFST_TSTCYL(R4)                 ; CYLINDER = TEST CYLINDER
         00000000'EF    03  FB  0130              CALLS   #3,BUILD_HDRS                   ; GO BUILD HEADERS
                      56    DD  0137              PUSHL   R6                              ; GET BYTE COUNT
             00000000'EF    7F  0139              PUSHAQ  WRITE_BUFFER                    ; GET BUFFER ADDRESS
         00000000'EF    02  FB  013F              CALLS   #2,$GS_SETMAP                   ; MAP DATA PATH
                      00    DD  0146      20$:    PUSHL   #0                              ; IDCMPR DATA
                      57    DD  0148              PUSHL   R7                              ; IDCDAR (SECTOR 0)
                   7E 56    CE  014A              MNEGL   R6,-(SP)                        ; IDCCNT (BYTE COUNT)
                      00    DD  014D              PUSHL   #0                              ; IDCBAR DATA
             20000000 8F    DD  014F              PUSHL   #<CSR$M_FMT!NO_OP>              ; DEFINE COMMAND = WRITE FORMAT
         00000000'EF    05  FB  0155              CALLS   #5,START_CMD                    ; EXECUTE WRITE DATA CMD
         00000000'EF    00  FB  015C              CALLS   #0,STATUS_CHK                   ; CHECK FOR ANY ERRORS
                   1D 50    E8  0163              BLBS    R0,30$                          ; BRANCH IF NONE
                                0166              $DS_ERRHARD_S    ,L$UNIT,MSG20A_FORMAT,-
                                0166                               PRINT_REG,R1
                                017C      ;;; TEST 35, SUBTEST 2, ERROR 1
                                0183      30$:    $DS_CKLOOP       20$
                                018B              STARS
                                018B      ;;*******************************************************************
                                018B      ;++
                                018B      ; VERIFY ALL HEADERS ARE CORRECT BY WRITING ALL
```

```
                          018B   612 ; SECTORS ON THE TRACK.
                          018B   613 ;--
                          018B   614
                 58   7C  018B   615          CLRQ    R8                           ; RESET COUNTER AND DAR
     59   10  10 55   F0  018D   616          INSV    R5,#16,#16,R9                ; LOAD CYLINDER
     59   08  08 0D   F0  0192   617          INSV    #13,#8,#8,R9                 ; LOAD TRACK
     59   08  00 58   F0  0197   618 35$:     INSV    R8,#0,#8,R9                  ; LOAD DESIRED SECTOR
              28 A4   DD  019C   619          PUSHL   OFST_SECLEN(R4)              ; PUSH BYTE COUNT
        00000000'EF   7F  019F   620          PUSHAQ  WRITE_BUFFER                 ; AND ADDRESS OF MEMORY BUFFER
     00000000'EF  02  FB  01A5   621          CALLS   #2,$GS_SETMAP                ; SET UP MAPPING REGISTERS
                 00   DD  01AC   622          PUSHL   #0                           ; IDCMPR DATA
                 59   DD  01AE   623          PUSHL   R9                           ; IDCDAR
          7E  28 A4   CE  01B0   624          MNEGL   OFST_SECLEN(R4),-(SP)        ; IDCCNT (BYTE COUNT)
                 00   DD  01B4   625          PUSHL   #0                           ; IDCBAR
                 0A   DD  01B6   626          PUSHL   #WRITE                       ; SET UP WRITE FUNCTION
     00000000'EF  05  FB  01B8   627          CALLS   #5,START_CMD                 ; DO WRITE FUNCTION
     00000000'EF  00  FB  01BF   628          CALLS   #0,STATUS_CHK                ; CHECK FOR SUCCESSFUL COMPLETION
                          01C6   629          $DS_BNERROR     40$                  ; IF OK GOTO 40$
                          01D6   630          $DS_ERRHARD_S   ,L$UNIT,MSG20C_WRITE,-
                          01D6   631                          PRINT_REG,R1
                          01EC       ;;; TEST 35, SUBTEST 2, ERROR 2
                          01F3   632 40$:     $DS_CKLOOP      20$
          97 58   20  F2  01FC   633          AOBLSS  #32,R8,35$                   ; DO ALL SECTORS
                          0200   634          $DS_ENDSUB
                          020B   635          $DS_ENDTEST
                          0216   636          $DS_SBTTL <MANUAL SKIP SECTOR TEST - R80>,ALIGN=LONG
                          0216                .SBTTL  TEST 36:  MANUAL SKIP SECTOR TEST - R80
                          0216                $D2_SBTTL       036,<MANUAL SKIP SECTOR TEST - R80>,<LONG>
                      00000000                        .PSECT  TEST_036, LONG, NOWRT
                          0020
```

```
                              0020  ;++
                              0020  ;
                              0020  ; TEST DESCRIPTION:
                              0020  ;
                              0020  ; THIS TEST WILL VERIFY THE OPERATION OF A MANUAL SKIP
                              0020  ; SECTOR RECOVERY OPERATION.
                              0020  ;
                              0020  ; TEST STEPS:
                              0020  ;
                              0020  ; A.  SUBTEST 1
                              0020  ;
                              0020  ;           1. INITIALIZE SUBSYSTEM
                              0020  ;           2.  EXECUTE SEEK TO TEST CYLINDER
                              0020  ;           3.  IF ERROR THEN ESCAPE TEST
                              0020  ;
                              0020  ; B.  SUBTEST 2
                              0020  ;
                              0020  ;           1. INITIALIZE SUBSYSTEM
                              0020  ;           2. BUILD HEADERS FOR TRACK
                              0020  ;           3. FORMAT TRACK WITH SECTOR 0 MARKED AS A SKIP SECTOR
                              0020  ;           4. CHECK FOR ERRORS
                              0020  ;
                              0020  ; B.  SUBTEST 3
                              0020  ;           1. INIT SUBSYSTEM
                              0020  ;           2. WRITE SECTORS 0 AND 1 WITH SKIP SECTORS TURNED OFF
                              0020  ;              DATA = ^X3333 FOR SECTOR 0 AND ^XCCCC FOR SECTOR 1
                              0020  ;           3. READ DATA (SECTOR 0) WITH ASSI ASSERTED AND EXPECT SSE TO SET
                              0020  ;           4. RESTART READ BY SETTING SSEI AND CLEARING CRDY
                              0020  ;           5. VERIFY OPERATION COMPLETED ERROR FREE
                              0020  ;           6. CHECK DATA READ BACK WAS SECTOR 1 DATA (^XCCCC)
                              0020  ;
                              0020  ; C.  SUBTEST 4
                              0020  ;           1. INIT SUBSYSTEM
                              0020  ;           2. WRITE CHECK SECTOR 0 AND EXPECT SSE TO ASSERT
                              0020  ;           3. RESTART OPERTION BY SETTING SSEI AND CLEARING CRDY
                              0020  ;           4. VERIFY OPERATION COMPLETED ERROR FREE
                              0020  ;
                              0020  ; DEBUG:
                              0020  ;
                              0020  ;
                              0020  ;--
                              0020
                              0020
                              0020          $DS_BGNDATA
       00000000 00000001      0020          .LONG    1,LUN0
       00000001 00000001      0028          .LONG    1,LUN1
       00000002 00000001      0030          .LONG    1,LUN2
       00000003 00000001      0038          .LONG    1,LUN3
                              0040          $DS_ENDDATA
                              0044
                              0044          $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
                              0046          GET_LUN -                               ; GET LUN INFO
                              0046            UNIT= OFST_UNIT(AP)                   ; LOGICAL UNIT
               04 AC    DD    0046          PUSHL    OFST_UNIT(AP)
   00000000'EF    01    FB    0049          CALLS    #1,GET_LUN$A
                              0050          BDEVFOUND        5$                     ; IF DEV AVAILABLE GOTO 5$
```

```
          06 50    00    E2  0050                BBSS    #0,R0,5$
             50    01    D0  0054                        MOVL    #1,R0           ; NOT AN ERROR
                 049F    31  0057                        BRW     TEST_036_X      ; EXIT TEST 36
                             005A       5$:      $DS_EXIT        TEST                      ; ****TEMP INSTRUCTION****
      54   00000000'EF   D0  0060                MOVL    DRIVE_TBL,R4                      ; GET DRIVE INFO TABLE
             02    64    91  0067                CMPB    OFST_DRVTYP(R4),#R80$T            ; IF DRIVE TYPE = R80
                   06    13  006A                BEQL    6$                                ; THEN GOTO 6$
                             006C                $DS_EXIT        TEST                      ; ELSE EXIT THIS TEST
          55    14 A4    D0  0072       6$:      MOVL    OFST_TSTCYL(R4),R5                ; GET TEST CYLINDER
                   57    D4  0076                CLRL    R7                                ; INIT REGISTER
    57    10    10 55    F0  0078                INSV    R5,#16,#16,R7                     ; SET UP DESTINATION CYLN
    57    08    08 0D    F0  007D                INSV    #13,#8,#8,R7                      ; AND DESTINATION TRACK = 12
      56   00000080 8F   D0  0082                MOVL    #<4*32>,R6                        ; BYTE COUNT FOR FORMAT COMMAND
           00000000'EF   D4  0089                CLRL    CONFIG                            ; CLEAR ANY PROGRAM FLAGS
                   53    D4  008F                CLRL    R3                                ; CLEAR INDEX
                             0091                $DS_BGNSUB
                             009C                STARS
                             009C           ;;*************************************************************************
                             009C           ;++
                             009C           ; POSITION DRIVE TO TEST CYLINDER AND VERIFY THE SEEK
                             009C           ;--
                             009C
  00000000'EF      00    FB  009C       10$:     CALLS   #0,SUBSYS_INIT                    ; INITIALIZE SUBSYSTEM
             1D 50       E8  00A3                BLBS    R0,20$                            ; BRANCH IF NO ERRORS
                             00A6                $DS_ERRHARD_S   ,L$UNIT,MSG14_SYSINIT,-
                             00A6                                PRINT_REG,R1
                             00BC           ;;; TEST 36, SUBTEST 1, ERROR 1
                             00C3       20$:     $DS_CKLOOP      10$
                   0D    DD  00CB       35$:     PUSHL   #13                               ; PUSH HEAD TO USE
                   55    DD  00CD                PUSHL   R5                                ; PUSH DESTINATION CYLN
  00000000'EF      02    FB  00CF                CALLS   #2,POSITION_HDS                   ; EXECUTE THE SEEK
             1D 50       E8  00D6                BLBS    R0,40$                            ; BRANCH IF NO ERRORS
                             00D9                $DS_ERRHARD_S   ,L$UNIT,@ERR_MSG,-
                             00D9                                PRINT_REG,R1
                             00EF           ;;; TEST 36, SUBTEST 1, ERROR 2
                             00F6       40$:     $DS_CKLOOP      35$
                             00FE                $DS_ESCAPE      TEST                      ; EXIT TEST IF ERROR
           00000000'EF   D4  0109                CLRL    CONFIG                            ; CLEAR ANY PROGRAM FLAGS
                             010F       50$:     $DS_ENDSUB
                             011A                $DS_BGNSUB
                             0125                STARS
                             0125           ;;*************************************************************************
                             0125           ;++
                             0125           ; BUILD HEADERS AND EXECUTE WRITE FORMAT FUNCTION
                             0125           ;--
                             0125
                   00    DD  0125                PUSHL   #0                                ; NO SKIP SECTORS PLEASE
                   0D    DD  0127                PUSHL   #13                               ; BUILD HDRS FOR TRACK 13
                14 A4    DD  0129                PUSHL   OFST_TSTCYL(R4)                   ; CYLINDER = TEST CYLINDER
  00000000'EF      03    FB  012C                CALLS   #3,BUILD_HDRS                     ; GO BUILD HEADERS
                   56    DD  0133                PUSHL   R6                                ; GET BYTE COUNT
           00000000'EF   7F  0135                PUSHAQ  WRITE_BUFFER                      ; GET BUFFER ADDRESS
  00000000'EF      02    FB  013B                CALLS   #2,$GS_SETMAP                     ; MAP UBA
                   00    DD  0142       20$:     PUSHL   #0                                ; IDCMPR DATA
                   57    DD  0144                PUSHL   R7                                ; IDCDAR (SECTOR 0)
             7E    56    CE  0146                MNEGL   R6,-(SP)                          ; IDCCNT (BYTE COUNT)
```

```
                   00  DD  0149   741            PUSHL   #0                           ; IDCBAR DATA
          20000000 8F  DD  014B   742            PUSHL   #<CSR$M_FMT!NO_OP>           ; DEFINE COMMAND = WRITE FORMAT
       00000000'EF 05  FB  0151   743            CALLS   #5,START_CMD                 ; EXECUTE WRITE DATA CMD
       00000000'EF 00  FB  0158   744            CALLS   #0,STATUS_CHK                ; CHECK FOR ANY ERRORS
                1D 50  E8  015F   745            BLBS    R0,30$                       ; BRANCH IF NONE
                           0162   746            $DS_ERRHARD_S    ,L$UNIT,MSG20A_FORMAT,-
                           0162   747                             PRINT_REG,R1
                           0178       ;;; TEST 36, SUBTEST 2, ERROR 1
                           017F   748 30$:       $DS_CKLOOP       20$
                           0187   749            $DS_ENDSUB
                           0192   750            $DS_BGNSUB
                           019D   751            STARS
                           019D       ;;***********************************************************************
                           019D   752 ;++
                           019D   753 ; WRITE SECTOR 0 WITH ^X3333 AND SECTOR 1 WITH ^XCCCC.
                           019D   754 ; READ SECTOR 0 WITH AUTOMATIC SKIP SECTORS TURNED OFF.
                           019D   755 ; TEST THAT SKIP SECTOR ERROR ASSERTS. RESTART READ
                           019D   756 ; OPERATION VERIFY THAT NO ERRORS OCCUR AND THAT THE
                           019D   757 ; DATA READ BACK ORIGINATED FROM SECTOR 1.
                           019D   758 ;--
                           019D   759
                   53  D4  019D   760            CLRL    R3                           ; RESET INDEX COUNTER
 33333333 33333333 8F  7D  019F   761 5$:        MOVQ    #^X3333333333333333,-        ; LOAD DATA FOR FIRST SECTOR
       00000000'FF43       01A9   762                    @WRITE_BUFFER[R3]            ;
   E8 53   00000040 8F  F2  01AF   763           AOBLSS  #64,R3,5$                    ; DO ALL 256 WORDS
 CCCCCCCC CCCCCCCC 8F  7D  01B7   764 6$:        MOVQ    #^XCCCCCCCCCCCCCCCC,-        ; LOAD DATA FOR 2ND SECTOR
       00000000'FF43       01C1   765                    @WRITE_BUFFER[R3]            ;
   E8 53   00000080 8F  F2  01C7   766           AOBLSS  #128,R3,6$                   ; DO ALL 256 WORDS
      56    28 A4  02  C5  01CF   767            MULL3   #2,OFST_SECLEN(R4),R6        ; CALC 2 SECTOR BYTE COUNT
                   56  DD  01D4   768            PUSHL   R6                           ; BYTE COUNT
         00000000'EF   7F  01D6   769            PUSHAQ  WRITE_BUFFER                 ; MEMORY DATA BUFFER
       00000000'EF 02  FB  01DC   770            CALLS   #2,$GS_SETMAP                ; MAP DATA PATH
                   00  DD  01E3   771 20$:       PUSHL   #0                           ; IDCMPR DATA
                   57  DD  01E5   772            PUSHL   R7                           ; IDCDAR DATA
              7E   56  CE  01E7   773            MNEGL   R6,-(SP)                     ; LOAD BYTE COUNT
                   00  DD  01EA   774            PUSHL   #0                           ; IDCBAR DATA
          0840000A 8F  DD  01EC   775            PUSHL   #<CSR$M_ASSI!CSR$M_SSEI!WRITE>  ; WRITE CMD WITH ALL SKIP SECTORS  -
cont> SHUT OFF
       00000000'EF 05  FB  01F2   776            CALLS   #5,START_CMD                 ; START WRITE FUNCTION
       00000000'EF 00  FB  01F9   777            CALLS   #0,STATUS_CHK                ; CHECK FOR ANY ERRORS
                1D 50  E8  0200   778            BLBS    R0,30$                       ; BRANCH IF NONE
                           0203   779            $DS_ERRHARD_S    ,L$UNIT,MS113_UWRTERR,-
                           0203   780                             PRINT_REG,R1
                           0219       ;;; TEST 36, SUBTEST 3, ERROR 1
                           0220   781 30$:       $DS_CKLOOP       20$
              56   02  C6  0228   782            DIVL2   #2,R6                        ; SET BYTE COUNT TO 1 SECTOR
      58    0880008D 8F  D0  022B   783          MOVL    #<CSR$M_ASSI!CSR$M_SSE!CSR$M_CRDY!- ; SET EXPECTED CSR
                           0232   784                    READ!CSR$M_DRDY>,R8          ;
      58    00000000'EF  C8  0232   785          BISL2   P$UNIT,R8                    ; OR IN DRIVE SELECT BITS
                   56  DD  0239   786            PUSHL   R6                           ; BYTE COUNT
         00000000'EF   7F  023B   787            PUSHAQ  READ_BUFFER                  ; MEMORY DATA BUFFER
       00000000'EF 02  FB  0241   788            CALLS   #2,$GS_SETMAP                ; MAP DATA PATH
                   00  DD  0248   789 35$:       PUSHL   #0                           ; IDCMPR DATA
                   57  DD  024A   790            PUSHL   R7                           ; IDCDAR DATA
              7E   56  CE  024C   791            MNEGL   R6,-(SP)                     ; LOAD BYTE COUNT
                   00  DD  024F   792            PUSHL   #0                           ; IDCBAR DATA
          0800000C 8F  DD  0251   793            PUSHL   #<CSR$M_ASSI!READ>           ; READ CMD WITH SKIP SECTORS ENABLED
```

```
          00000000'EF   05   FB  0257    794            CALLS   #5,START_CMD                     ; START READ FUNCTION
          00000000'EF   12   D0  025E    795            MOVL    #<NOCHK$M_CERR!NOCHK$M_DSTA>,CONFIG ; WAIVER CONTROLLER ERROR CHECKS
          00000000'EF   00   FB  0265    796            CALLS   #0,STATUS_CHK                    ; CHECK FOR ANY ERRORS
                    1D 50    E8  026C    797            BLBS    R0,40$                           ; BRANCH IF NONE
                                 026F    798            $DS_ERRHARD_S    ,L$UNIT,MS114_URDERR,-
                                 026F    799                             PRINT_REG,R1
                                 0285        ;;; TEST 36, SUBTEST 3, ERROR 2
                                 028C    800 40$:       $DS_CKLOOP       35$
          00000000'EF   58   D1  0294    801            CMPL    R8,T.CS                          ; IF CSR = EXPECTED VALUE
                        29   13  029B    802            BEQL    50$                              ; THEN GOTO 50$
                                 029D    803            $DS_ERRHARD_S    ,L$UNIT,MS114_URDERR,-
                                 029D    804                             REG_ERRPT,#MSG23_CS_INC,-
                                 029D    805                             R8,T.CS
                                 02BF        ;;; TEST 36, SUBTEST 3, ERROR 3
                                 02C6    806 50$:       $DS_CKLOOP       35$
          59   0840000C 8F   D0  02CF    807            MOVL    #<CSR$M_ASSI!CSR$M_SSEI!READ>,R9 ; SET UP FOR READ RESTART
          59   00000000'EF   C8  02D6    808            BISL2   P$UNIT,R9                        ; OR IN UNIT SELECT BITS
          58   0840008D 8F   D0  02DD    809            MOVL    #<CSR$M_ASSI!CSR$M_SSEI!CSR$M_CRDY!- ; SET EXPECTED CSR
                                 02E4    810                    READ!CSR$M_DRDY>,R8              ;
          58   00000000'EF   C8  02E4    811            BISL2   P$UNIT,R8                        ; OR IN UNIT BITS
                    62   59  D0  02EB    812            MOVL    R9,IDCCSR(R2)                    ; RESTART READ OPERATION
          00000000'EF   02   D0  02EE    813            MOVL    #NOCHK$M_CERR,CONFIG             ; WAIVER CONTROLLER ERROR CHECKS
          00000000'EF   00   FB  02F5    814            CALLS   #0,STATUS_CHK                    ; CHECK STATUS
                    1D 50    E8  02FC    815            BLBS    R0,60$                           ; IF NO ERROR GOTO 60$
                                 02FF    816            $DS_ERRHARD_S    ,L$UNIT,MS115_RDRST,-
                                 02FF    817                             PRINT_REG,R1
                                 0315        ;;; TEST 36, SUBTEST 3, ERROR 4
                                 031C    818 60$:       $DS_CKLOOP       35$
          00000000'EF   58   D1  0325    819            CMPL    R8,T.CS                          ; IF CSR = EXPECTED VALUE
                        29   13  032C    820            BEQL    70$                              ; THEN GOTO 70$
                                 032E    821            $DS_ERRHARD_S    ,L$UNIT,MS115_RDRST,-
                                 032E    822                             REG_ERRPT,#MSG23_CS_INC,-
                                 032E    823                             R8,T.CS
                                 0350        ;;; TEST 36, SUBTEST 3, ERROR 5
                                 0357    824 70$:       $DS_CKLOOP       35$
               00000000'EF   DD  0360    825            PUSHL   SKIP_BUFFER                      ; GET GOOD DATA
               00000000'EF   DD  0366    826            PUSHL   READ_BUFFER                      ; GET DATA JUST READ
               00000000'EF   DD  036C    827            PUSHL   WORD_COUNT                       ; NUMBER OF WORDS TO COMPARE
          00000000'EF   03   FB  0372    828            CALLS   #3,CMPDATA                       ; CHECK THE DATA
                    1B 50    E8  0379    829            BLBS    R0,80$                           ; BRANCH IF DATA OK
                                 037C    830            $DS_ERRHARD_S    ,L$UNIT,MS116_SKPCMP,-
                                 037C    831                             PRINT_DATA_ERR
                                 0390        ;;; TEST 36, SUBTEST 3, ERROR 6
                                 0397    832 80$:       $DS_CKLOOP       35$
                                 03A0    833            $DS_ENDSUB
                                 03AB    834            $DS_BGNSUB
                                 03B6    835            STARS
                                 03B6        ;;********************************************************************
                                 03B6    836 ;++
                                 03B6    837 ; WRITE CHECK SECTOR 0 AND EXPECT SKIP SECTOR ERROR TO ASSERT.
                                 03B6    838 ; RESTART THE OPERATION AND EXPECT NORMAL COMPLETION.
                                 03B6    839 ;--
                                 03B6    840
          58   08800083 8F   D0  03B6    841            MOVL    #<CSR$M_ASSI!CSR$M_SSE!CSR$M_CRDY!- ; SET EXPECTED CSR
                                 03BD    842                    WRTCHK!CSR$M_DRDY>,R8            ;
          58   00000000'EF   C8  03BD    843            BISL2   P$UNIT,R8                        ; OR IN DRIVE SELECT BITS
```

```
                  56   DD  03C4           PUSHL   R6                                  ; BYTE COUNT
         00000000'EF   7F  03C6           PUSHAQ  SKIP_BUFFER                         ; MEMORY DATA BUFFER
   00000000'EF    02   FB  03CC           CALLS   #2,$GS_SETMAP                       ; MAP DATA PATH
                  00   DD  03D3      20$: PUSHL   #0                                  ; IDCMPR DATA
                  57   DD  03D5           PUSHL   R7                                  ; IDCDAR DATA
            7E    56   CE  03D7           MNEGL   R6,-(SP)                            ; LOAD BYTE COUNT
                  00   DD  03DA           PUSHL   #0                                  ; IDCBAR DATA
         08000002 8F   DD  03DC           PUSHL   #<CSR$M_ASSI!WRTCHK>                ; WRTCHK CMD WITH SKIP SECTORS ENA -
cont> BLED
   00000000'EF    05   FB  03E2           CALLS   #5,START_CMD                        ; START WRITE CHECK FUNCTION
   00000000'EF    12   D0  03E9           MOVL    #<NOCHK$M_CERR!NOCHK$M_DSTA>,CONFIG ; WAIVER CONTROLLER ERROR CHECKS
   00000000'EF    00   FB  03F0           CALLS   #0,STATUS_CHK                       ; CHECK FOR ANY ERRORS
             1D 50     E8  03F7           BLBS    R0,40$                              ; BRANCH IF NONE
                           03FA           $DS_ERRHARD_S    ,L$UNIT,MS114A_UWCKERR,-
                           03FA                            PRINT_REG,R1
                           0410      ;;; TEST 36, SUBTEST 4, ERROR 1
                           0417      40$: $DS_CKLOOP       20$
   00000000'EF    58   D1  041F           CMPL    R8,T.CS                             ; IF CSR = EXPECTED VALUE
                  29   13  0426           BEQL    50$                                 ; THEN GOTO 50$
                           0428           $DS_ERRHARD_S    ,L$UNIT,MS114A_UWCKERR,-
                           0428                            REG_ERRPT,#MSG23_CS_INC,-
                           0428                            R8,T.CS
                           044A      ;;; TEST 36, SUBTEST 4, ERROR 2
                           0451      50$: $DS_CKLOOP       20$
   59     08400002 8F  D0  045A           MOVL    #<CSR$M_ASSI!CSR$M_SSEI!WRTCHK>,R9 ; SET UP FOR WRTCHK RESTART
   59     00000000'EF  C8  0461           BISL2   P$UNIT,R9                          ; OR IN UNIT SELECT BITS
   58     08400083 8F  D0  0468           MOVL    #<CSR$M_ASSI!CSR$M_SSEI!CSR$M_CRDY!- ; SET EXPECTED CSR
                           046F                   WRTCHK!CSR$M_DRDY>,R8               ;
   58     00000000'EF  C8  046F           BISL2   P$UNIT,R8                           ; OR IN UNIT BITS
            62    59   D0  0476           MOVL    R9,IDCCSR(R2)                       ; RESTART WRTCHK OPERATION
   00000000'EF    02   D0  0479           MOVL    #NOCHK$M_CERR,CONFIG                ; WAIVER CONTROLLER ERROR CHECKS
   00000000'EF    00   FB  0480           CALLS   #0,STATUS_CHK                       ; CHECK STATUS
             1D 50     E8  0487           BLBS    R0,60$                              ; IF NO ERROR GOTO 60$
                           048A           $DS_ERRHARD_S    ,L$UNIT,MS115A_WKRST,-
                           048A                            PRINT_REG,R1
                           04A0      ;;; TEST 36, SUBTEST 4, ERROR 3
                           04A7      60$: $DS_CKLOOP       20$
   00000000'EF    58   D1  04B0           CMPL    R8,T.CS                             ; IF CSR = EXPECTED VALUE
                  29   13  04B7           BEQL    70$                                 ; THEN GOTO 70$
                           04B9           $DS_ERRHARD_S    ,L$UNIT,MS115A_WKRST,-
                           04B9                            REG_ERRPT,#MSG23_CS_INC,-
                           04B9                            R8,T.CS
                           04DB      ;;; TEST 36, SUBTEST 4, ERROR 4
                           04E2      70$: $DS_CKLOOP       20$
                           04EB           $DS_ENDSUB
                           04F6           $DS_ENDTEST
                           0501           $DS_SBTTL <AUTOMATIC SKIP SECTOR TEST - R80>,ALIGN=LONG
                           0501           .SBTTL  TEST 37:  AUTOMATIC SKIP SECTOR TEST - R80
                           0501           $D2_SBTTL        037,<AUTOMATIC SKIP SECTOR TEST - R80>,<LONG>
                   00000000                      .PSECT  TEST_037, LONG, NOWRT
                       0024
```

```
                              0024     ;++
                              0024     ;
                              0024     ; TEST DESCRIPTION:
                              0024     ;
                              0024     ; THIS TEST WILL VERIFY THE AUTOMATIC SKIP SECTOR FUNCTION
                              0024     ; OF THE IDC BY FORMATTING A TRACK WITH SECTOR 0 AS A SKIP
                              0024     ; SECTOR. PERFORM WRITE, READ, AND WRITECHECK OPERATIONS
                              0024     ; TO SECTOR 0 AND CHECK FOR ANY ERRORS. ALSO VERIFY THAT
                              0024     ; AT COMMAND COMPLETION THE IDCDAR IS POINTING TO SECTOR 2.
                              0024     ;
                              0024     ; TEST STEPS:
                              0024     ;
                              0024     ; A.  SUBTEST 1
                              0024     ;
                              0024     ;          1. INITIALIZE SUBSYSTEM
                              0024     ;          2.  EXECUTE SEEK TO TEST CYLINDER
                              0024     ;          3.  IF ERROR THEN ESCAPE TEST
                              0024     ;
                              0024     ; B.  SUBTEST 2
                              0024     ;
                              0024     ;          1. INITIALIZE SUBSYSTEM
                              0024     ;          2. BUILD HEADERS FOR TRACK
                              0024     ;          3. FORMAT TRACK
                              0024     ;          4. CHECK FOR ERRORS
                              0024     ;
                              0024     ; DEBUG:
                              0024     ;
                              0024     ;
                              0024     ;--
                              0024
                              0024
                              0024              $DS_BGNDATA
     00000000 00000001        0024              .LONG   1,LUN0
     00000001 00000001        002C              .LONG   1,LUN1
     00000002 00000001        0034              .LONG   1,LUN2
     00000003 00000001        003C              .LONG   1,LUN3
                              0044              $DS_ENDDATA
                              0048
                              0048              $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
                              004A              GET_LUN -                           ; GET LUN INFO
                              004A                UNIT= OFST_UNIT(AP)               ; LOGICAL UNIT
             04 AC      DD    004A              PUSHL   OFST_UNIT(AP)
  00000000'EF     01    FB    004D              CALLS   #1,GET_LUN$A
                              0054              BDEVFOUND       5$                  ; IF DEV AVAILABLE GOTO 5$
        06 50     00    E2    0054              BBSS    #0,R0,5$
           50     01    D0    0058                      MOVL    #1,R0       ; NOT AN ERROR
              04FB      31    005B                      BRW     TEST_037_X  ; EXIT TEST 37
                              005E     5$:      $DS_EXIT        TEST                ; ****TEMP INSTRUCTION****
  54    00000000'EF     D0    0064              MOVL    DRIVE_TBL,R4                ; GET DRIVE INFO TABLE
           02     64    91    006B              CMPB    OFST_DRVTYP(R4),#R80$T      ; IF DRIVE TYPE = R80
                  06    13    006E              BEQL    6$                          ; THEN GOTO 6$
                              0070              $DS_EXIT        TEST                ; ELSE EXIT THIS TEST
        55     14 A4    D0    0076     6$:      MOVL    OFST_TSTCYL(R4),R5          ; GET TEST CYLINDER
                  57    D4    007A              CLRL    R7                          ; INIT REGISTER
 57   10    10    55    F0    007C              INSV    R5,#16,#16,R7               ; SET UP DESTINATION CYLN
 57   08    08    0D    F0    0081              INSV    #13,#8,#8,R7                ; AND DESTINATION TRACK = 12
```

```
        56   00000080 8F    D0  0086                MOVL    #<4*32>,R6                      ; BYTE COUNT FOR FORMAT COMMAND
             00000000'EF    D4  008D                CLRL    CONFIG                          ; CLEAR ANY PROGRAM FLAGS
                      53    D4  0093                CLRL    R3                              ; CLEAR INDEX
                                0095                $DS_BGNSUB
                                00A0                STARS
                                00A0        ;;****************************************************************************
                                00A0        ;++
                                00A0        ; POSITION DRIVE TO TEST CYLINDER AND VERIFY THE SEEK
                                00A0        ;--
                                00A0
   00000000'EF        00    FB  00A0        10$:    CALLS   #0,SUBSYS_INIT                  ; INITIALIZE SUBSYSTEM
                   1D 50    E8  00A7                BLBS    R0,20$                          ; BRANCH IF NO ERRORS
                                00AA                $DS_ERRHARD_S       ,L$UNIT,MSG14_SYSINIT,-
                                00AA                                    PRINT_REG,R1
                                00C0        ;;; TEST 37, SUBTEST 1, ERROR 1
                                00C7        20$:    $DS_CKLOOP          10$
                      0D    DD  00CF        35$:    PUSHL   #13                             ; PUSH HEAD TO USE
                      55    DD  00D1                PUSHL   R5                              ; PUSH DESTINATION CYLN
   00000000'EF        02    FB  00D3                CALLS   #2,POSITION_HDS                 ; EXECUTE THE SEEK
                   1D 50    E8  00DA                BLBS    R0,40$                          ; BRANCH IF NO ERRORS
                                00DD                $DS_ERRHARD_S       ,L$UNIT,@ERR_MSG,-
                                00DD                                    PRINT_REG,R1
                                00F3        ;;; TEST 37, SUBTEST 1, ERROR 2
                                00FA        40$:    $DS_CKLOOP          35$
                                0102                $DS_ESCAPE          TEST                ; EXIT TEST IF ERROR
             00000000'EF    D4  010D                CLRL    CONFIG                          ; CLEAR ANY PROGRAM FLAGS
                                0113        50$:    $DS_ENDSUB
                                011E                $DS_BGNSUB
                                0129                STARS
                                0129        ;;****************************************************************************
                                0129        ;++
                                0129        ; BUILD HEADERS AND EXECUTE WRITE FORMAT FUNCTION
                                0129        ;--
                                0129
                      00    DD  0129                PUSHL   #0                              ; NO SKIP SECTORS PLEASE
                      0D    DD  012B                PUSHL   #13                             ; BUILD HDRS FOR TRACK 13
                   14 A4    DD  012D                PUSHL   OFST_TSTCYL(R4)                 ; CYLINDER = TEST CYLINDER
   00000000'EF        03    FB  0130                CALLS   #3,BUILD_HDRS                   ; GO BUILD HEADERS
                      56    DD  0137                PUSHL   R6                              ; GET BYTE COUNT
             00000000'EF    7F  0139                PUSHAQ  WRITE_BUFFER                    ; GET BUFFER ADDRESS
   00000000'EF        02    FB  013F                CALLS   #2,$GS_SETMAP                   ; MAP UBA
                      00    DD  0146        20$:    PUSHL   #0                              ; IDCMPR DATA
                      57    DD  0148                PUSHL   R7                              ; IDCDAR (SECTOR 0)
                   7E 56    CE  014A                MNEGL   R6,-(SP)                        ; IDCCNT (BYTE COUNT)
                      00    DD  014D                PUSHL   #0                              ; IDCBAR DATA
             20000000 8F    DD  014F                PUSHL   #<CSR$M_FMT!NO_OP>              ; DEFINE COMMAND = WRITE FORMAT
   00000000'EF        05    FB  0155                CALLS   #5,START_CMD                    ; EXECUTE WRITE DATA CMD
   00000000'EF        00    FB  015C                CALLS   #0,STATUS_CHK                   ; CHECK FOR ANY ERRORS
                   1D 50    E8  0163                BLBS    R0,30$                          ; BRANCH IF NONE
                                0166                $DS_ERRHARD_S       ,L$UNIT,MSG20A_FORMAT,-
                                0166                                    PRINT_REG,R1
                                017C        ;;; TEST 37, SUBTEST 2, ERROR 1
                                0183        30$:    $DS_CKLOOP          20$
                                018B                $DS_ENDSUB
                                0196                $DS_BGNSUB
                                01A1                STARS
```

```
                                01A1        ;;********************************************************************
                                01A1   989  ;++
                                01A1   990  ; WRITE SECTOR 0 WITH ^X3333 AND SECTOR 1 WITH ^XCCCC.
                                01A1   991  ; USING THE AUTOMATIC SKIP SECTOR FEATURE OF THE IDC
                                01A1   992  ; EXECUTE A WRITE DATA COMMAND TO SECTOR 0. TEST THAT
                                01A1   993  ; NO ERRORS OCCURR AND THAT THE IDC DAR REGISTER HAS
                                01A1   994  ; BEEN UPDATED TO SECTOR 2 AT THE COMPLETION OF THE
                                01A1   995  ; FUNCTION.
                                01A1   996  ;--
                                01A1   997
                     53     D4  01A1   998            CLRL    R3                              ; RESET INDEX COUNTER
   33333333 33333333 8F     7D  01A3   999  5$:       MOVQ    #^X3333333333333333,-           ; LOAD DATA FOR FIRST SECTOR
         00000000'FF43          01AD  1000                    @WRITE_BUFFER[R3]               ;
    E8 53   00000040 8F     F2  01B3  1001            AOBLSS  #64,R3,5$                       ; DO ALL 256 WORDS
   CCCCCCCC CCCCCCCC 8F     7D  01BB  1002  6$:       MOVQ    #^XCCCCCCCCCCCCCCCC,-           ; LOAD DATA FOR 2ND SECTOR
         00000000'FF43          01C5  1003                    @WRITE_BUFFER[R3]               ;
    E8 53   00000080 8F     F2  01CB  1004            AOBLSS  #128,R3,6$                      ; DO ALL 256 WORDS
      56    28 A4    02     C5  01D3  1005            MULL3   #2,OFST_SECLEN(R4),R6           ; CALC 2 SECTOR BYTE COUNT
                     56     DD  01D8  1006            PUSHL   R6                              ; BYTE COUNT
           00000000'EF      7F  01DA  1007            PUSHAQ  WRITE_BUFFER                    ; MEMORY DATA BUFFER
     00000000'EF     02     FB  01E0  1008            CALLS   #2,$GS_SETMAP                   ; MAP DATA PATH
                     00     DD  01E7  1009  20$:      PUSHL   #0                              ; IDCMPR DATA
                     57     DD  01E9  1010            PUSHL   R7                              ; IDCDAR DATA
              7E     56     CE  01EB  1011            MNEGL   R6,-(SP)                        ; LOAD BYTE COUNT
                     00     DD  01EE  1012            PUSHL   #0                              ; IDCBAR DATA
           0840000A 8F      DD  01F0  1013            PUSHL   #<CSR$M_ASSI!CSR$M_SSEI!WRITE>  ; WRITE CMD WITH ALL SKIP SECTORS  -
cont> SHUT OFF
     00000000'EF     05     FB  01F6  1014            CALLS   #5,START_CMD                    ; START WRITE FUNCTION
     00000000'EF     00     FB  01FD  1015            CALLS   #0,STATUS_CHK                   ; CHECK FOR ANY ERRORS
                1D   50     E8  0204  1016            BLBS    R0,30$                          ; BRANCH IF NONE
                                0207  1017            $DS_ERRHARD_S     ,L$UNIT,MS113_UWRTERR,-
                                0207  1018                              PRINT_REG,R1
                                021D        ;;; TEST 37, SUBTEST 3, ERROR 1
                                0224  1019  30$:      $DS_CKLOOP        20$
                     56  02     C6  022C  1020            DIVL2   #2,R6                           ; SET BYTE COUNT TO 1 SECTOR
        58     0000008B 8F      D0  022F  1021            MOVL    #<CSR$M_CRDY!WRITE!CSR$M_DRDY>,R8 ; SET EXPECTED IDCCSR REG
          5A     57      02     C1  0236  1022            ADDL3   #2,R7,R10                       ; SET EXPECTED IDCDAR (SEC + 2)
        58    00000000'EF       C8  023A  1023            BISL2   P$UNIT,R8                       ; OR IN DRIVE SELECT BITS
                     56     DD  0241  1024            PUSHL   R6                              ; BYTE COUNT
           00000000'EF      7F  0243  1025            PUSHAQ  SKIP_BUFFER                     ; MEMORY DATA BUFFER
     00000000'EF     02     FB  0249  1026            CALLS   #2,$GS_SETMAP                   ; MAP DATA PATH
                     00     DD  0250  1027  35$:      PUSHL   #0                              ; IDCMPR DATA
                     57     DD  0252  1028            PUSHL   R7                              ; IDCDAR DATA
              7E     56     CE  0254  1029            MNEGL   R6,-(SP)                        ; LOAD BYTE COUNT
                     00     DD  0257  1030            PUSHL   #0                              ; IDCBAR DATA
                     0A     DD  0259  1031            PUSHL   #WRITE                          ; WRITE CMD WITH AUTO SKIP
     00000000'EF     05     FB  025B  1032            CALLS   #5,START_CMD                    ; START READ FUNCTION
     00000000'EF     02     D0  0262  1033            MOVL    #NOCHK$M_CERR,CONFIG            ; WAIVER CONTROLLER ERROR CHECKS
     00000000'EF     00     FB  0269  1034            CALLS   #0,STATUS_CHK                   ; CHECK FOR ANY ERRORS
                1D   50     E8  0270  1035            BLBS    R0,40$                          ; BRANCH IF NONE
                                0273  1036            $DS_ERRHARD_S     ,L$UNIT,MS117_AUTOWRT,-
                                0273  1037                              PRINT_REG,R1
                                0289        ;;; TEST 37, SUBTEST 3, ERROR 2
                                0290  1038  40$:      $DS_CKLOOP        35$
     00000000'EF     58     D1  0298  1039            CMPL    R8,T.CS                         ; IF CSR = EXPECTED VALUE
                     29     13  029F  1040            BEQL    50$                             ; THEN GOTO 50$
                                02A1  1041            $DS_ERRHARD_S     ,L$UNIT,MS117_AUTOWRT,-
```

```
                          02A1  1042                         REG_ERRPT,#MSG23_CS_INC,-
                          02A1  1043                         R8,T.CS
                          02C3       ;;; TEST 37, SUBTEST 3, ERROR 3
                          02CA  1044 50$:    $DS_CKLOOP      35$
              5A    DD    02D2  1045         PUSHL   R10                             ; GET EXPECTED DAR DATA
    00000000'EF  01   FB    02D4  1046         CALLS   #1,CHANGE_DAR$A                 ; CONVERT TO EXPECTED READ BACK VA -
cont> LUE
        5A    51    D0    02DB  1047         MOVL    R1,R10                          ; GET NEW EXPECTED DAR VALUE
    00000000'EF  5A   D1    02DE  1048         CMPL    R10,T.DA                        ; IF DAR = EXPECTED VALUE
              2F    13    02E5  1049         BEQL    60$                             ; THEN GOTO 60$
                          02E7  1050         $DS_ERRHARD_S   ,L$UNIT,MS117_AUTOWRT,-
                          02E7  1051                         REG_ERRPT,#MSG28_DA_INC,-
                          02E7  1052                         R10,T.DA,DAR_MNEM
                          030F       ;;; TEST 37, SUBTEST 3, ERROR 4
                          0316  1053 60$:    $DS_CKLOOP      35$
                          031F  1054         $DS_ENDSUB
                          032A  1055         $DS_BGNSUB
                          0335  1056         STARS
                          0335       ;;*******************************************************************
                          0335  1057 ;++
                          0335  1058 ; READ SECTOR 0 AND EXPECT NORMAL COMPLETION.
                          0335  1059 ; CHECK THAT IDCDAR HAS UPDATED TO SECTOR 2.
                          0335  1060 ;--
                          0335  1061
58    0000008D 8F   D0    0335  1062         MOVL    #<CSR$M_CRDY!READ!CSR$M_DRDY>,R8 ; SET EXPECTED CSR
58    00000000'EF   C8    033C  1063         BISL2   P$UNIT,R8                       ; OR IN DRIVE SELECT BITS
              56    DD    0343  1064         PUSHL   R6                              ; BYTE COUNT
      00000000'EF   7F    0345  1065         PUSHAQ  READ_BUFFER                     ; MEMORY DATA BUFFER
    00000000'EF  02   FB    034B  1066         CALLS   #2,$GS_SETMAP                   ; MAP DATA PATH
              00    DD    0352  1067 20$:    PUSHL   #0                              ; IDCMPR DATA
              57    DD    0354  1068         PUSHL   R7                              ; IDCDAR DATA
        7E    56    CE    0356  1069         MNEGL   R6,-(SP)                        ; LOAD BYTE COUNT
              00    DD    0359  1070         PUSHL   #0                              ; IDCBAR DATA
              0C    DD    035B  1071         PUSHL   #READ                           ; READ CMD WITH AUTO SKIP
    00000000'EF  05   FB    035D  1072         CALLS   #5,START_CMD                    ; START WRITE CHECK FUNCTION
    00000000'EF  02   D0    0364  1073         MOVL    #NOCHK$M_CERR,CONFIG            ; WAIVER CONTROLLER ERROR CHECKS
    00000000'EF  00   FB    036B  1074         CALLS   #0,STATUS_CHK                   ; CHECK FOR ANY ERRORS
           1D 50    E8    0372  1075         BLBS    R0,40$                          ; BRANCH IF NONE
                          0375  1076         $DS_ERRHARD_S   ,L$UNIT,MS117_AUTORD,-
                          0375  1077                         PRINT_REG,R1
                          038B       ;;; TEST 37, SUBTEST 4, ERROR 1
                          0392  1078 40$:    $DS_CKLOOP      20$
    00000000'EF  58   D1    039A  1079         CMPL    R8,T.CS                         ; IF CSR = EXPECTED VALUE
              29    13    03A1  1080         BEQL    50$                             ; THEN GOTO 50$
                          03A3  1081         $DS_ERRHARD_S   ,L$UNIT,MS117_AUTORD,-
                          03A3  1082                         REG_ERRPT,#MSG23_CS_INC,-
                          03A3  1083                         R8,T.CS
                          03C5       ;;; TEST 37, SUBTEST 4, ERROR 2
                          03CC  1084 50$:    $DS_CKLOOP      20$
    00000000'EF  5A   D1    03D4  1085         CMPL    R10,T.DA                        ; IF DAR = EXPECTED VALUE
              2F    13    03DB  1086         BEQL    70$                             ; THEN GOTO 70$
                          03DD  1087         $DS_ERRHARD_S   ,L$UNIT,MS117_AUTORD,-
                          03DD  1088                         REG_ERRPT,#MSG28_DA_INC,-
                          03DD  1089                         R10,T.DA,DAR_MNEM
                          0405       ;;; TEST 37, SUBTEST 4, ERROR 3
                          040C  1090 70$:    $DS_CKLOOP      20$
      00000000'EF   DD    0415  1091         PUSHL   WORD_COUNT                      ; GET NUMBER OF WORDS TO COMPARE
```

```
          00000000'EF   DD  041B  1092          PUSHL   SKIP_BUFFER                    ; GET GOOD DATA
          00000000'EF   DD  0421  1093          PUSHL   READ_BUFFER                    ; GET DATA TO VERIFY
 00000000'EF     03     FB  0427  1094          CALLS   #3,CMPDATA                     ; COMPARE THE DATA
              1B 50     E8  042E  1095          BLBS    R0,80$                         ; BRANCH IF NO ERRORS
                            0431  1096          $DS_ERRHARD_S    ,L$UNIT,MS116A_AUTOCMP,-
                            0431  1097                           PRINT_DATA_ERR
                            0445       ;;; TEST 37, SUBTEST 4, ERROR 4
                            044C  1098 80$:     $DS_CKLOOP       20$
                            0455  1099          $DS_ENDSUB
                            0460  1100          $DS_BGNSUB
                            046B  1101          STARS
                            046B       ;;*******************************************************************
                            046B  1102 ;++
                            046B  1103 ; WRITE CHECK SECTOR 0 AND EXPECT NORMAL COMPLETION.
                            046B  1104 ; CHECK THAT IDCDAR HAS UPDATED TO SECTOR 2.
                            046B  1105 ;--
                            046B  1106
 58    00000083 8F      D0  046B  1107          MOVL    #<CSR$M_CRDY!WRTCHK!CSR$M_DRDY>,R8 ; SET EXPECTED CSR
 58    00000000'EF      C8  0472  1108          BISL2   P$UNIT,R8                      ; OR IN DRIVE SELECT BITS
                 56     DD  0479  1109          PUSHL   R6                             ; BYTE COUNT
       00000000'EF      7F  047B  1110          PUSHAQ  SKIP_BUFFER                    ; MEMORY DATA BUFFER
 00000000'EF     02     FB  0481  1111          CALLS   #2,$GS_SETMAP                  ; MAP DATA PATH
                 00     DD  0488  1112 20$:     PUSHL   #0                             ; IDCMPR DATA
                 57     DD  048A  1113          PUSHL   R7                             ; IDCDAR DATA
              7E 56     CE  048C  1114          MNEGL   R6,-(SP)                       ; LOAD BYTE COUNT
                 00     DD  048F  1115          PUSHL   #0                             ; IDCBAR DATA
                 0C     DD  0491  1116          PUSHL   #READ                          ; READ CMD WITH AUTO SKIP
 00000000'EF     05     FB  0493  1117          CALLS   #5,START_CMD                   ; START WRITE CHECK FUNCTION
 00000000'EF     02     D0  049A  1118          MOVL    #NOCHK$M_CERR,CONFIG           ; WAIVER CONTROLLER ERROR CHECKS
 00000000'EF     00     FB  04A1  1119          CALLS   #0,STATUS_CHK                  ; CHECK FOR ANY ERRORS
              1D 50     E8  04A8  1120          BLBS    R0,40$                         ; BRANCH IF NONE
                            04AB  1121          $DS_ERRHARD_S    ,L$UNIT,MS117_AUTOWCK,-
                            04AB  1122                           PRINT_REG,R1
                            04C1       ;;; TEST 37, SUBTEST 5, ERROR 1
                            04C8  1123 40$:     $DS_CKLOOP       20$
 00000000'EF     58     D1  04D0  1124          CMPL    R8,T.CS                        ; IF CSR = EXPECTED VALUE
                 29     13  04D7  1125          BEQL    50$                            ; THEN GOTO 50$
                            04D9  1126          $DS_ERRHARD_S    ,L$UNIT,MS117_AUTOWCK,-
                            04D9  1127                           REG_ERRPT,#MSG23_CS_INC,-
                            04D9  1128                           R8,T.CS
                            04FB       ;;; TEST 37, SUBTEST 5, ERROR 2
                            0502  1129 50$:     $DS_CKLOOP       20$
 00000000'EF     5A     D1  050A  1130          CMPL    R10,T.DA                       ; IF DAR = EXPECTED VALUE
                 2F     13  0511  1131          BEQL    70$                            ; THEN GOTO 70$
                            0513  1132          $DS_ERRHARD_S    ,L$UNIT,MS117_AUTOWCK,-
                            0513  1133                           REG_ERRPT,#MSG28_DA_INC,-
                            0513  1134                           R10,T.DA,DAR_MNEM
                            053B       ;;; TEST 37, SUBTEST 5, ERROR 3
                            0542  1135 70$:     $DS_CKLOOP       20$
                            054B  1136          $DS_ENDSUB
                            0556  1137          $DS_ENDTEST
                            0561  1138          $DS_SBTTL <ADDRESS RECALL TEST - RL02>,ALIGN=LONG
                            0561                .SBTTL  TEST 38:  ADDRESS RECALL TEST - RL02
                            0561                $D2_SBTTL        038,<ADDRESS RECALL TEST - RL02>,<LONG>
                    00000000                            .PSECT  TEST_038, LONG, NOWRT
                            0020
```

```
                                0020  1140 ;++
                                0020  1141 ;
                                0020  1142 ; TEST DESCRIPTION:
                                0020  1143 ;
                                0020  1144 ; THIS TEST WILL EXECUTE 512 PSEUDO RANDOM SEEKS.  THE SEEKS ARE PSEUDO
                                0020  1145 ; RANDOM BECAUSE EACH TIME THE TEST IS RUN IT WILL ALWAYS USE THE
                                0020  1146 ; SAME 512 SEEK DESTINATIONS.  ALL SEEKS ARE DONE WITH NO INTERVEANING
                                0020  1147 ; READ HEADERS.  THIS REQUIRES THE IDC TO KEEP TRACK OF THE CURRENT
                                0020  1148 ; CYLINDER THE HEADS ARE POSITIONED ON BEFORE CALCULATING THE DIFFERENCE
                                0020  1149 ; WORD FOR THE NEXT SEEK.  THIS IS THE ONLY TEST THAT USES THIS MODE
                                0020  1150 ; OF OPERATION.
                                0020  1151 ;
                                0020  1152 ; TEST STEPS:
                                0020  1153 ;
                                0020  1154 ;          1.  INITIALIZE SUBSYSTEM
                                0020  1155 ;          2.  CALCULATE RANDOM ADDRESS
                                0020  1156 ;          3.  EXECUTE SEEK TO DESTINATION
                                0020  1157 ;          4.  RETURN TO STEP 2 UNTIL 512 SEEKS HAVE BEEN DONE
                                0020  1158 ;          5.  READ HEADER TO GET CURRENT POSITION OF HEADS
                                0020  1159 ;          6.  IF DRIVE NO POSITION ON CYLINER XYZ REPORT ERROR
                                0020  1160 ;
                                0020  1161 ; DEBUG:
                                0020  1162 ;
                                0020  1163 ;         IDC MODULE
                                0020  1164 ;
                                0020  1165 ; ASSUMPTIONS:
                                0020  1166 ;
                                0020  1167 ;         RL02 SEEK LOGIC IS WORKING PROPERLY.
                                0020  1168 ;         ANY SEEK ERROR WILL CAUSE TEST TO ABORT.
                                0020  1169 ;
                                0020  1170 ;--
                                0020  1171
                                0020  1172
                                0020  1173            $DS_BGNDATA
          00000000 00000001     0020  1174            .LONG    1,LUN0
          00000001 00000001     0028  1175            .LONG    1,LUN1
          00000002 00000001     0030  1176            .LONG    1,LUN2
          00000003 00000001     0038  1177            .LONG    1,LUN3
                                0040  1178            $DS_ENDDATA
                                0044  1179
                                0044  1180            $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
                                0046  1181            GET_LUN -                              ; GET LUN INFO
                                0046  1182              UNIT= OFST_UNIT(AP)                  ; LOGICAL UNIT
              04 AC     DD      0046                  PUSHL    OFST_UNIT(AP)
      00000000'EF  01   FB      0049                  CALLS    #1,GET_LUN$A
                                0050  1183            BDEVFOUND       5$                     ; IF DEV AVAILABLE GOTO 5$
          06 50  00     E2      0050                  BBSS     #0,R0,5$
             50  01     D0      0054                           MOVL     #1,R0        ; NOT AN ERROR
               013E     31      0057                           BRW      TEST_038_X   ; EXIT TEST 38
   54    00000000'EF    D0      005A  1184 5$:        MOVL     DRIVE_TBL,R4           ; GET DRIVE INFO TABLE
             01  64     91      0061  1185            CMPB     OFST_DRVTYP(R4),#RL02$T ; IF DRIVE TYPE = RL02
                 06     13      0064  1186            BEQL     6$                     ; THEN GOTO 6$
                                0066  1187            $DS_EXIT          TEST          ; ELSE EXIT THIS TEST
          00000000'EF   D4      006C  1188 6$:        CLRL     CONFIG                 ; CLEAR PROGRAM FLAGS
   55     08 A4  01     C1      0072  1189            ADDL3    #1,OFST_LSTCYL(R4),R5  ; CALC UPPER CYLN LIMIT
                 53     D4      0077  1190            CLRL     R3                     ; RESET SEEK COUNTER
```

```
                        56   D4  0079  1191         CLRL    R6                              ; CLEAR PSEUDO RANDOM SEED
                        5B   D4  007B  1192         CLRL    R11                             ; ZERO DESTINATION REGISTER
         00000000'EF    00   FB  007D  1193 10$:    CALLS   #0,SUBSYS_INIT                  ; INIT SUBSYSTEM
                     1D 50   E8  0084  1194         BLBS    R0,20$                          ; BRANCH IF NO ERRORS
                                 0087  1195         $DS_ERRHARD_S   ,L$UNIT,MSG14_SYSINIT,-
                                 0087  1196                         PRINT_REG,R1
                                 009D       ;;; TEST 38, SUBTEST 0, ERROR 1
                                 00A4  1197 20$:    $DS_CKLOOP      10$
         12B9B0A1 8F    56   7A  00AC  1198 25$:    EMUL    R6,#314159265,-                 ; CALC RANDOM NUMBER
         57    1B0CB17F 8F       00B3  1199                 #453816703,R7                   ; PUT NUMBER IN R7 AND R8
   56   58   57    7FFFFFFF 8F   7B  00B9  1200         EDIV    #2147483647,R7,R8,R6            ; USE REMAINDER AS NEW SEED
                   57   56   D0  00C2  1201         MOVL    R6,R7                           ; SAVE NEW RANDOM NUMBER
                        58   D4  00C5  1202         CLRL    R8                              ; RESET DIVIDEND
         5A   59   57   55   7B  00C7  1203         EDIV    R5,R7,R9,R10                    ; PUT NUMBER IN LEGAL CYLN RANGE
         5B   10   10   5A   F0  00CC  1204         INSV    R10,#16,#16,R11                 ; LOAD CYLINDER
         5B   01   08   5A   F0  00D1  1205         INSV    R10,#8,#1,R11                   ; USE LSB OF CYLN AS TRACK
                        00   DD  00D6  1206         PUSHL   #0                              ; IDCMAP DATA
                        5B   DD  00D8  1207         PUSHL   R11                             ; IDCDAR DATA
                        00   DD  00DA  1208         PUSHL   #0                              ; IDCBCR DATA
                        00   DD  00DC  1209         PUSHL   #0                              ; IDCBAR DATA
                        06   DD  00DE  1210         PUSHL   #SEEK                           ; COMMAND = SEEK
         00000000'EF    05   FB  00E0  1211         CALLS   #5,START_CMD                    ; EXECUTE SEEK
         00000000'EF    00   FB  00E7  1212         CALLS   #0,STATUS_CHK                   ; CHECK FOR ERRORS
                     1D 50   E8  00EE  1213         BLBS    R0,30$                          ; BRANCH IF NONE
                                 00F1  1214         $DS_ERRHARD_S   ,L$UNIT,MS118_SEEKIDC,-
                                 00F1  1215                         PRINT_REG,R1
                                 0107       ;;; TEST 38, SUBTEST 0, ERROR 2
                                 010E  1216 30$:    $DS_CKLOOP      25$
                                 0116  1217         $DS_ESCAPE      TEST                    ; IF ERROR ESCAPE TEST
      83 53    00000200 8F   F2  0121  1218         AOBLSS  #512,R3,25$                     ; LOOP UNITL 512 SEEKS DONE
            5A    6C A4      D0  0129  1219         MOVL    OFST_RANCYL(R4),R10             ; GET EXPECTED FINAL DESTINATION
                        00   DD  012D  1220 35$:    PUSHL   #0                              ; SET NO FLAGS
         00000000'EF    01   FB  012F  1221         CALLS   #1,GET_POSITION                 ; GET CURRNET DRIVE POSITION
                     1D 50   E8  0136  1222         BLBS    R0,40$                          ; BRANCH IF NO ERRORS
                                 0139  1223         $DS_ERRHARD_S   ,L$UNIT,MSG19_READH,-
                                 0139  1224                         PRINT_REG,R1
                                 014F       ;;; TEST 38, SUBTEST 0, ERROR 3
                                 0156  1225 40$:    $DS_CKLOOP      35$
         5A    00000000'EF   D1  015E  1226         CMPL    CURR_CYLN,R10                   ; IF DESTINATION = EXPECTED
                        25   13  0165  1227         BEQL    50$                             ; THEN GOTO 50$
                                 0167  1228         $DS_ERRHARD_S   ,L$UNIT,MS119_IDCPOS,-
                                 0167  1229                         REG_ERRPT,#0,-
                                 0167  1230                         R10,CURR_CYLN
                                 0185       ;;; TEST 38, SUBTEST 0, ERROR 4
                                 018C  1231 50$:    $DS_CKLOOP      25$
                                 0195  1232         $DS_ENDTEST
                                 01A0  1233         $DS_SBTTL <SUBSYSTEM ACTIVITY TEST>,ALIGN=LONG
                                 01A0               .SBTTL  TEST 39:  SUBSYSTEM ACTIVITY TEST
                                 01A0               $D2_SBTTL       039,<SUBSYSTEM ACTIVITY TEST>,<LONG>
                             00000000                       .PSECT  TEST_039, LONG, NOWRT
                                 001C
```

```
                                  001C
                                  001C       ;++
                                  001C       ;
                                  001C       ; TEST DESCRIPTION:
                                  001C       ;
                                  001C       ; THIS TEST IS INTENDED TO EXERCISE THE IDC SUBSYSTEM USING
                                  001C       ; RANDOM DISK ADDRESSES FOR SEEKS, WRITES, WRITECHECKS, AND
                                  001C       ; READS.  IT IS INTENDED AS A TOOL FOR MAXIMIZING FUNCTIONS
                                  001C       ; IN A STANDALONE ENVIRONMENT.
                                  001C       ;
                                  001C       ; TEST STEPS:
                                  001C       ;
                                  001C       ;            1.  INITIALIZE SUBSYSTEM
                                  001C       ;            2.  EXECUTE READ HEADER/SEEK/READ HEADER AND
                                  001C       ;                CHECK FOR ERRORS
                                  001C       ;            3.  EXECUTE WRITE AND CHECK FOR ERRORS
                                  001C       ;            4.  EXECUTE WRITE CHECK AND CHECK FOR ERRORS
                                  001C       ;            5.  EXECUTE READ AND CHECK FOR ERRORS
                                  001C       ;
                                  001C       ; DEBUG:
                                  001C       ;
                                  001C       ;            NONE
                                  001C       ;
                                  001C       ;--
                                  001C
                                  001C               $DS_BGNDATA
           00000000 00000001      001C               .LONG    1,LUN0
           00000001 00000001      0024               .LONG    1,LUN1
           00000002 00000001      002C               .LONG    1,LUN2
           00000003 00000001      0034               .LONG    1,LUN3
                                  003C               $DS_ENDDATA
                                  0040
                                  0040               $DS_BGNTEST <DEFAULT,ALL>,ALIGN=LONG
                        53    7C  0042               CLRQ     R3                                   ; CLEAR INDEX
      6DB6DB6D B6DB6DB6 8F    7D  0044  5$:          MOVQ     #^X6DB6DB6DB6DB6DB6,-                ; LOAD WORST CASE PATTERN
               00000000'FF44      004E                        @WRITE_BUFFER[R4]                    ;
   E8 54   00000040 8F    F2      0054               AOBLSS   #64,R4,5$                            ; DO 256 WORDS
                                  005C               GET_LUN -                                 ; GET LUN INFO
                                  005C                 UNIT= OFST_UNIT(AP)                     ; LOGICAL UNIT
                   04 AC  DD      005C               PUSHL    OFST_UNIT(AP)
      00000000'EF    01   FB      005F               CALLS    #1,GET_LUN$A
           06 50     00   E2      0066               BBSS     #0,R0,1$
                                  006A               $DS_EXIT          TEST                    ; EXIT IF NO UNIT
      54   00000000'EF    D0      0070  1$:          MOVL     DRIVE_TBL,R4                         ; GET DRIVE INFO TABLE
      00000000'EF    00   FB      0077               CALLS    #0,RD_BAD_BLK                        ; LOG IN BAD BLOCK FILE TO MEM
                13 50     E8      007E               BLBS     R0,8$                                ; IF READ OK GOTO 8$
                                  0081               $DS_PRINTF_S      FMT_BADBLK,#DEV_NAME    ; PRINT WARNING
      55    08 A4    03   C3      0094  8$:          SUBL3    #3,OFST_LSTCYL(R4),R5                ; SET UPPER CYLINDER LIMIT
                                  0099                                                         ; TO PROTECT SKIP AND/OR BAD SECTO -
cont> R FILE
      56    0C A4    01   C1      0099               ADDL3    #1,OFST_LSTTRK(R4),R6                ; SET UPPER TRACK LIMIT
      57    10 A4    01   C1      009E               ADDL3    #1,OFST_LSTSEC(R4),R7                ; SET UPPER SECTOR LIMIT
      00000000'EF    00   FB      00A3  7$:          CALLS    #0,SUBSYS_INIT                       ; INIT SUBSYSTEM
                1D 50     E8      00AA               BLBS     R0,6$                                ; BRANCH IF NO ERRORS
                                  00AD               $DS_ERRHARD_S      ,L$UNIT,MSG14_SYSINIT,-
                                  00AD                                  PRINT_REG,R1
                                  00C3       ;;; TEST 39, SUBTEST 0, ERROR 1
```

```
                                 00CA  1288 6$:     $DS_CKLOOP      7$
                                 00D2  1289 10$:
                     52   D4     00D2  1290 15$:    CLRL    R2                              ; CLEAR DIVIDEND FOR EDIV
                                 00D4  1291         $READEF_S -                             ; READ EVENT FLAGS
                                 00D4  1292           EFN=          #0,-                    ; EVENT GROUP #0
                                 00D4  1293           STATE=        CURR_FLAGS              ; PUT FLAGS HERE
         00000000'EF   16        00E3  1294 17$:    JSB     GET_RANDOM                      ; GET RANDOM NUMBER
   5B    58    51  57  7B        00E9  1295         EDIV    R7,R1,R8,R11                    ; CALC RANDOM SECTOR
   5A    58    51  56  7B        00EE  1296         EDIV    R6,R1,R8,R10                    ; CALC RANDOM TRACK
   59    58    51  55  7B        00F3  1297         EDIV    R5,R1,R8,R9                     ; CALC RANDOM CYLINDER
                     5B   DD     00F8  1298         PUSHL   R11                             ; GET SECTOR
                     5A   DD     00FA  1299         PUSHL   R10                             ; AND TRACK
                     59   DD     00FC  1300         PUSHL   R9                              ; AND CYLINDER
      00000000'EF    03   FB     00FE  1301         CALLS   #3,CHECK_BADBLK                 ; SEE IF LISTED IN BAD BLOCK FILE
                  CA 50   E8     0105  1302         BLBS    R0,15$                          ; IF IT IS GOTO 15$
              02     64   91     0108  1303         CMPB    OFST_DRVTYP(R4),#R80$T          ; IF DRIVE NOT AN R80
                     0C   12     010B  1304         BNEQ    20$                             ; THEN GOTO 20$
   04 00000000'EF    14   E0     010D  1305         BBS     #DEL$V_PROT,EVENT_FLAGS,20$     ; IF WRITE ANYWHERE THEN GOTO 20$
           59     14 A4   D0     0115  1306         MOVL    OFST_TSTCYL(R4),R9              ; ELSE USE FE CYLINDER
   5B    08    08  5A     F0     0119  1307 20$:    INSV    R10,#8,#8,R11                   ; AND LOAD IN DESTINATION REGISTER
   5B    10    10  59     F0     011E  1308         INSV    R9,#16,#16,R11                  ; LOAD DESTINATION CYLINDER
    52     00000000'EF    D0     0123  1309         MOVL    BASE_ADDR,R2                    ; RESTORE BASE ADDRESS TO R2
                     53   D5     012A  1310         TSTL    R3                              ; IF NOT FIRST TIME HERE
                     08   12     012C  1311         BNEQ    30$                             ; THEN GOTO 30$
   00 00000000'EF    0A   E5     012E  1312         BBCC    #DEL$V_READH,CURR_FLAGS,30$     ; ENABLE READ HEADERS THIS TIME ONLY
                     5A   DD     0136  1313 30$:    PUSHL   R10                             ; GET TRACK
                     59   DD     0138  1314         PUSHL   R9                              ; GET CYLINDER
      00000000'EF    02   FB     013A  1315         CALLS   #2,POSITION_HDS                 ; POSITION HEADS TO DESTINATION
                  2C 50   E8     0141  1316         BLBS    R0,40$                          ; BRANCH IF NO ERRORS
   07 00000000'EF    0A   E0     0144  1317         BBS     #DEL$V_READH,CURR_FLAGS,35$     ; IF READ HDRS NOT WIAVERED GOTO 35$
      00000000'EF    07   CA     014C  1318         BICL2   #PRT$M_ALL,RGLIST               ; CLEAR INVALID OUTPUTS
                                 0153  1319 35$:    $DS_ERRHARD_S     ,L$UNIT,@ERR_MSG,-
                                 0153  1320                           PRINT_REG,R1
                                 0169       ;;; TEST 39, SUBTEST 0, ERROR 2
                                 0170  1321 40$:    $DS_CKLOOP      30$
                  28 A4   DD     0178  1322 50$:    PUSHL   OFST_SECLEN(R4)                 ; BYTE COUNT
         00000000'EF      7F     017B  1323         PUSHAQ  WRITE_BUFFER                    ; MEMORY DATA BUFFER
      00000000'EF    02   FB     0181  1324         CALLS   #2,$GS_SETMAP                   ; MAP DATA PATH
   4D 00000000'EF    0B   E0     0188  1325         BBS     #DEL$V_WRIT,CURR_FLAGS,80$      ; IF WRITES DISABLED GOTO 80$
                     00   DD     0190  1326 60$:    PUSHL   #0                              ; IDCMPR DATA
                     5B   DD     0192  1327         PUSHL   R11                             ; IDCDAR DATA
           7E     28 A4   CE     0194  1328         MNEGL   OFST_SECLEN(R4),-(SP)           ; IDCCNT (BYTE COUNT)
                     00   DD     0198  1329         PUSHL   #0                              ; IDCBAR DATA
                     0A   DD     019A  1330         PUSHL   #WRITE                          ; DEFINE COMMAND = WRITE DATA
      00000000'EF    05   FB     019C  1331         CALLS   #5,START_CMD                    ; EXECUTE WRITE DATA CMD
00000000'EF    00000080 8F   D0  01A3  1332         MOVL    #NOCHK$M_ATTN,CONFIG            ; ****TEMP INSTRUCTION****
      00000000'EF    00   FB     01AE  1333         CALLS   #0,STATUS_CHK                   ; CHECK FOR ANY ERRORS
                  1D 50   E8     01B5  1334         BLBS    R0,70$                          ; BRANCH IF NONE
                                 01B8  1335         $DS_ERRHARD_S     ,L$UNIT,MSG20_WRITE,-
                                 01B8  1336                           PRINT_REG,R1
                                 01CE       ;;; TEST 39, SUBTEST 0, ERROR 3
                                 01D5  1337 70$:    $DS_CKLOOP      60$
   55 00000000'EF    0B   E0     01DD  1338 80$:    BBS     #DEL$V_WRIT,CURR_FLAGS,100$     ; IF WRITES DIABLED GOTO 100$
   4D 00000000'EF    0C   E0     01E5  1339         BBS     #DEL$V_WRCK,CURR_FLAGS,100$     ; IF WRITE CHECKS DISABLED GOTO 100$
                     00   DD     01ED  1340 85$:    PUSHL   #0                              ; IDCMPR DATA
                     5B   DD     01EF  1341         PUSHL   R11                             ; IDCDAR DATA
```

```
           7E    28 A4    CE   01F1   1342            MNEGL   OFST_SECLEN(R4),-(SP)           ; IDCCNT (BYTE COUNT)
                    00    DD   01F5   1343            PUSHL   #0                              ; IDCBAR DATA
                    02    DD   01F7   1344            PUSHL   #WRTCHK                         ; DEFINE COMMAND = WRITECHECK DATA
        00000000'EF 05    FB   01F9   1345            CALLS   #5,START_CMD                    ; EXECUTE WRITECHECK DATA CMD
 00000000'EF   00000080 8F  D0   0200   1346            MOVL    #NOCHK$M_ATTN,CONFIG            ; ****TEMP INSTRUCTION****
        00000000'EF 00    FB   020B   1347            CALLS   #0,STATUS_CHK                   ; CHECK FOR ANY ERRORS
                 1D 50    E8   0212   1348            BLBS    R0,90$                          ; BRANCH IF NONE
                               0215   1349            $DS_ERRHARD_S   ,L$UNIT,MSG16_WRTCHK,-
                               0215   1350                            PRINT_REG,R1
                               022B        ;;; TEST 39, SUBTEST 0, ERROR 4
                               0232   1351 90$:       $DS_CKLOOP      85$
     64 00000000'EF 0D    E0   023A   1352 100$:      BBS     #DEL$V_READ,CURR_FLAGS,120$     ; IF READS DISABLED GOTO 120$
                 28 A4    DD   0242   1353            PUSHL   OFST_SECLEN(R4)                 ; BYTE COUNT
            00000000'EF   7F   0245   1354            PUSHAQ  WRITE_BUFFER                    ; MEMORY DATA BUFFER
        00000000'EF 02    FB   024B   1355            CALLS   #2,$GS_SETMAP                   ; MAP DATA PATH
                    00    DD   0252   1356 105$:      PUSHL   #0                              ; IDCMPR DATA
                    5B    DD   0254   1357            PUSHL   R11                             ; IDCDAR DATA
           7E    28 A4    CE   0256   1358            MNEGL   OFST_SECLEN(R4),-(SP)           ; IDCCNT (BYTE COUNT)
                    00    DD   025A   1359            PUSHL   #0                              ; IDCBAR DATA
                    0C    DD   025C   1360            PUSHL   #READ                           ; DEFINE COMMAND = READ DATA
        00000000'EF 05    FB   025E   1361            CALLS   #5,START_CMD                    ; EXECUTE READ DATA CMD
 00000000'EF   00000080 8F  D0   0265   1362            MOVL    #NOCHK$M_ATTN,CONFIG            ; ****TEMP INSTRUCTION****
        00000000'EF 00    FB   0270   1363            CALLS   #0,STATUS_CHK                   ; CHECK FOR ANY ERRORS
                 24 50    E8   0277   1364            BLBS    R0,110$                         ; BRANCH IF NONE
                               027A   1365            $DS_ERRHARD_S   ,L$UNIT,MSG21_READ,-
                               027A   1366                            PRINT_REG,R1
                               0290        ;;; TEST 39, SUBTEST 0, ERROR 5
        00000000'EF 00    FB   0297   1367            CALLS   #0,CHECK_ECC                    ; CHECK IF CORRECTABLE ERROR
                               029E   1368                                                    ; AND TRY TO CORRECT DATA
                               029E   1369 110$:      $DS_CKLOOP      105$
   FDF3 53   01   00000400 8F  F1   02A6   1370 120$:      ACBL    #1024,#1,R3,7$                  ; TEST NEXT ITERATION
                               02B0   1371            $DS_ENDTEST
                               02BB   1372            $DS_SBTTL <DRIVE INTERFACE TEST>,ALIGN=LONG
                               02BB                   .SBTTL  TEST 40:  DRIVE INTERFACE TEST
                               02BB                   $D2_SBTTL       040,<DRIVE INTERFACE TEST>,<LONG>
                           00000000                         .PSECT  TEST_040, LONG, NOWRT
                               0018
```

```
                                0018
                                0018  ;++
                                0018  ;
                                0018  ; TEST DESCRIPTION:
                                0018  ;
                                0018  ; THIS TEST WILL ASK THE OPERATOR TO UNLOAD THE CURRENT DRIVE UNDER
                                0018  ; TEST.  THEN WHEN THE HEADS ARE UNLOADED TO OPEN THE COVER (IF RL02)
                                0018  ; AND SET THE WRITE LOCK SWITCH.  A GET STATUS COMMAND IS EXECUTED
                                0018  ; AND PROPER DRIVE STATUS IS VERIFIED.
                                0018  ; NEXT ASK THE OPERATOR TO CLOSE THE COVER (RL02) AND RESET THE WRITE
                                0018  ; LOCK SWITCH ON THE DRIVE UNDER TEST.  THEN EXECUTE A GET STATUS COMMAND
                                0018  ; AND CHECK FOR PROPER DRIVE STATUS.
                                0018  ;
                                0018  ; TEST STEPS:
                                0018  ;
                                0018  ;           1.   INITIALIZE SUBSYSTEM
                                0018  ;           2.   ASK OPERATOR TO UNLOAD DRIVE
                                0018  ;           3.   WAIT FOR OPERATOR TO HIT RETURN
                                0018  ;           4.   EXECUTE GET STATUS FUNCTION
                                0018  ;           5.   VERIFY DRIVE STATUS
                                0018  ;           6.   ASK OPERATOR TO CLOSE COVER AND RESET WRITE LOCK
                                0018  ;           7.   WAIT FOR OPERATOR TO HIT RETURN
                                0018  ;           8.   EXECUTE GET STATUS FUNCTION
                                0018  ;           9.   VERIFY DRIVE STATUS
                                0018  ;
                                0018  ; DEBUG:
                                0018  ;
                                0018  ;           DRIVE LOGIC MODULE
                                0018  ;           COVER LATCH (RL02)
                                0018  ;           SWITCH PANEL
                                0018  ;
                                0018  ;--
                                0018
                                0018
                                0018              $DS_BGNDATA
         00000000 00000001      0018              .LONG    1,LUN0
         00000001 00000001      0020              .LONG    1,LUN1
         00000002 00000001      0028              .LONG    1,LUN2
         00000003 00000001      0030              .LONG    1,LUN3
                                0038              $DS_ENDDATA
                                003C
                                003C              $DS_BGNTEST <MANUAL,ALL>,ALIGN=LONG
                                003E              GET_LUN -                                  ; GET LUN INFO
                                003E                UNIT= OFST_UNIT(AP)                      ; LOGICAL UNIT
                 04 AC    DD    003E              PUSHL    OFST_UNIT(AP)
        00000000'EF  01   FB    0041              CALLS    #1,GET_LUN$A
                                0048              BDEVFOUND          5$                      ; IF DEV AVAILABLE GOTO 5$
              06 50  00   E2    0048              BBSS     #0,R0,5$
                 50  01   D0    004C                       MOVL     #1,R0        ; NOT AN ERROR
               0203       31    004F                       BRW      TEST_040_X   ; EXIT TEST 40
                                0052  5$:         $DS_BOPER          6$                      ; IF OPERATOR PRESENT GOTO 6$
                                005A              $DS_EXIT           TEST                    ; ELSE EXIT TEST
     54   00000000'EF     D0    0060  6$:         MOVL     DRIVE_TBL,R4                      ; GET DRIVE INFO TABLE
            55    60 A4   D0    0067              MOVL     OFST_OP1STS(R4),R5                ; GET EXPECTED DRIVE STATUS
            56    34 A4   D0    006B              MOVL     OFST_GTSCMD(R4),R6                ; GET BITS FOR GET STATUS COMMAND
            57    40 A4   D0    006F              MOVL     OFST_MPBITS(R4),R7                ; GET MPR BIT MNEMONICS
```

```
          58    64 A4   D0   0073        MOVL    OFST_STSDNC(R4),R8                 ; GET DRIVE STATUS DONT CARE BITS
                  0C    D0   0077 10$:   MOVL    #<NOCHK$M_DRDY!NOCHK$M_DERR>,-; SET PROGRAM FLAGS
          00000000'EF        0079                CONFIG
       00000000'EF   00 FB   007E        CALLS   #0,SUBSYS_INIT                     ; INIT SUBSYSTEM
               1D 50    E8   0085        BLBS    R0,20$                             ; BRANCH IF NO ERRORS
                             0088        $DS_ERRHARD_S    ,L$UNIT,MSG14_SYSINIT,-
                             0088                         PRINT_REG,R1
                             009E  ;;; TEST 40, SUBTEST 0, ERROR 1
                             00A5 20$:   $DS_CKLOOP       10$
                             00AD        $DS_PRINTF_S     OPR1_UNLOAD,#DEV_NAME
                             00C0        $DS_ASKSTR_S     OPR_RETURN,CHAR_BUFFER,-
                             00C0                         #1,,CHAR_BUFFER
                  56    DD   00DF 25$:   PUSHL   R6                                 ; IDCMPR DATA GET STATUS
                  00    DD   00E1        PUSHL   #0                                 ; IDCDAR DATA
                  00    DD   00E3        PUSHL   #0                                 ; IDCCNT DATA
                  00    DD   00E5        PUSHL   #0                                 ; IDCBAR DATA
                  04    DD   00E7        PUSHL   #GETSTAT                           ; DEFINE COMMAND
       00000000'EF   05 FB   00E9        CALLS   #5,START_CMD                       ; EXECUTE GET STATUS COMMAND
                  0C    D0   00F0        MOVL    #<NOCHK$M_DRDY!NOCHK$M_DERR>,-; SET PROGRAM FLAGS
          00000000'EF        00F2                CONFIG
       00000000'EF   00 FB   00F7        CALLS   #0,STATUS_CHK                      ; CHECK FOR ANY ERRORS
               1D 50    E8   00FE        BLBS    R0,30$                             ; BRANCH IF NONE
                             0101        $DS_ERRHARD_S    ,L$UNIT,MSG17_GETSTAT,-
                             0101                         PRINT_REG,R1
                             0117  ;;; TEST 40, SUBTEST 0, ERROR 2
                             011E 30$:   $DS_CKLOOP       25$
       00000000'EF   58 CA   0126        BICL2   R8,T.MP                            ; CLEAR DONT CARE BITS
       00000000'EF   55 D1   012D        CMPL    R5,T.MP                            ; EXPECTED = ACTUAL ?
                  27    13   0134        BEQL    40$                                ; BRANCH IF YES
                             0136        $DS_ERRHARD_S    ,L$UNIT,MSG73_GS_UND,-
                             0136                         REG_ERRPT,#0,-
                             0136                         R5,T.MP,R7
                             0156  ;;; TEST 40, SUBTEST 0, ERROR 3
                             015D 40$:   $DS_CKLOOP       25$
    1B 00000000'EF   00 E1   0166        BBC     #CSR$V_DRDY,T.CS,50$               ; DRIVE READY SHOULD BE CLEAR
                             016E        $DS_ERRHARD_S    ,L$UNIT,MSG74_DRDY_NC,-
                             016E                         PRINT_REG
                             0182  ;;; TEST 40, SUBTEST 0, ERROR 4
                             0189 50$:   $DS_CKLOOP       25$
                             0192        $DS_PRINTF_S     OPR2_CLOSE,#DEV_NAME
                             01A5        $DS_ASKSTR_S     OPR_RETURN,CHAR_BUFFER,-
                             01A5                         #1,,CHAR_BUFFER
                  56    DD   01C4 55$:   PUSHL   R6                                 ; IDCMPR = GET STATUS COMMAND
                  00    DD   01C6        PUSHL   #0                                 ; IDCDAR DATA
                  00    DD   01C8        PUSHL   #0                                 ; IDCCNT DATA
                  00    DD   01CA        PUSHL   #0                                 ; IDCBAR DATA
                  04    DD   01CC        PUSHL   #GETSTAT                           ; DEFINE COMMAND
       00000000'EF   05 FB   01CE        CALLS   #5,START_CMD                       ; EXECUTE GET STATUS COMMAND
                  0C    D0   01D5        MOVL    #<NOCHK$M_DRDY!NOCHK$M_DERR>,-; SET PROGRAM FLAGS
          00000000'EF        01D7                CONFIG
       00000000'EF   00 FB   01DC        CALLS   #0,STATUS_CHK                      ; CHECK FOR ANY ERRORS
               1D 50    E8   01E3        BLBS    R0,60$                             ; BRANCH IF NONE
                             01E6        $DS_ERRHARD_S    ,L$UNIT,MSG17_GETSTAT,-
                             01E6                         PRINT_REG,R1
                             01FC  ;;; TEST 40, SUBTEST 0, ERROR 5
                             0203 60$:   $DS_CKLOOP       55$
```

```
             55    00002020 8F   CA  020B  1476         BICL2   #<MPR_02$M_WL!MPR_02$M_CO>,R5    ; CLEAR WRT LOCK AND COVER OPEN
             00000000'EF    58   CA  0212  1477         BICL2   R8,T.MP                          ; CLEAR DONT CARE BITS
             00000000'EF    55   D1  0219  1478         CMPL    R5,T.MP                          ; EXPECTED = ACTUAL ?
                            27   13  0220  1479         BEQL    70$                              ; BRANCH IF YES
                                     0222  1480         $DS_ERRHARD_S      ,L$UNIT,MSG73_GS_UND,-
                                     0222  1481                            REG_ERRPT,#0,-
                                     0222  1482                            R5,T.MP,R7
                                     0242       ;;; TEST 40, SUBTEST 0, ERROR 6
                                     0249  1483 70$:    $DS_CKLOOP         55$
                                     0252  1484         $DS_ENDTEST
                                     025D  1485         $DS_SBTTL <HEAD LOADING TEST - RL02>,ALIGN=LONG
                                     025D               .SBTTL  TEST 41:  HEAD LOADING TEST - RL02
                                     025D               $D2_SBTTL          041,<HEAD LOADING TEST - RL02>,<LONG>
                                 00000000                       .PSECT  TEST_041, LONG, NOWRT
                                     001C
```

```
                              001C
                              001C  ;++
                              001C  ;
                              001C  ; TEST DESCRIPTION:
                              001C  ;
                              001C  ; THIS TEST WILL INSTRUCT THE OPERATOR TO LOAD THE HEADS ON THE UNIT
                              001C  ; CURRENTLY UNDER TEST.  THE OCCURANCE OF ALL DRIVE STATES WILL BE
                              001C  ; CHECK AND ALSO THE DRIVE STATUS WITHIN EACH STATE WILL BE CHECKED.
                              001C  ; ONCE THE HEADS ARE LOADED DRIVE READY BIT IS CHECKED IN THE CONTROL
                              001C  ; AND STATUS REGISTER.  THEN A DRIVE RESET IS DONE AND VOLUME CHECK
                              001C  ; BIT IN THE DRIVE STATUS WORD IS CHECKED FOR THE RESET STATE.
                              001C  ;
                              001C  ; TEST STEPS:
                              001C  ;
                              001C  ;           1.  INIT SUBSYSTEM
                              001C  ;           2.  TELL OPERATOR TO LOAD HEADS ON UUT
                              001C  ;           3.  DONT WAIT FOR ANY RETURN FROM OPERATOR
                              001C  ;           4.  WAIT FOR DRIVE STATE 1 (SPIN UP)
                              001C  ;           5.  TEST DRIVE STATUS IN SPIN UP STATE
                              001C  ;           6.  WAIT FOR DRIVE STATE 2 (BRUSH CYCLE)
                              001C  ;           7.  TEST DRIVE STATUS IN STATE 2
                              001C  ;           8.  WAIT FOR BRUSH CYCLE TO START (BH BIT RESET)
                              001C  ;           9   WAIT FOR BRUSH CYCLE TO COMPLETE (BH BIT SET)
                              001C  ;           10. CHECK STATE 3 DRIVE STATUS (LOAD HEADS)
                              001C  ;           11. WAIT FOR DRIVE STATE 4 (SEEK)
                              001C  ;           12. CHECK DRIVE STATE 4 STATUS
                              001C  ;           13. WAIT FOR DRIVE STATE 5 (LOCK ON TRACK)
                              001C  ;           14. CHECK STATE 5 DRIVE STATUS
                              001C  ;           15. CHECK DRIVE READY BIT IN IDCCSR IS SET
                              001C  ;           16. EXECUTE DRIVE RESET
                              001C  ;           17. CHECK THAT VOLUME CHECK BIT IN DRIVE STATUS IS CLEAR
                              001C  ;
                              001C  ; DEBUG:
                              001C  ;
                              001C  ;           DRIVE LOGIC MODULE
                              001C  ;           BRUSH MOTOR
                              001C  ;           BRUSH SWITCH
                              001C  ;           AC SERVO MODULE
                              001C  ;
                              001C  ;--
                              001C
                              001C
                              001C              $DS_BGNDATA
         00000000 00000001    001C              .LONG   1,LUN0
         00000001 00000001    0024              .LONG   1,LUN1
         00000002 00000001    002C              .LONG   1,LUN2
         00000003 00000001    0034              .LONG   1,LUN3
                              003C              $DS_ENDDATA
                              0040
                              0040              $DS_BGNTEST <MANUAL,ALL>,ALIGN=LONG
                              0042              GET_LUN -                                        ; GET LUN INFO
                              0042                UNIT= OFST_UNIT(AP)                            ; LOGICAL UNIT
             04 AC      DD    0042              PUSHL   OFST_UNIT(AP)
  00000000'EF    01     FB    0045              CALLS   #1,GET_LUN$A
                              004C              BDEVFOUND       5$                               ; IF DEV AVAILABLE GOTO 5$
        06 50    00     E2    004C              BBSS    #0,R0,5$
```

```
                 50    01  D0  0050                       MOVL    #1,R0                  ; NOT AN ERROR
                     07E8  31  0053                       BRW     TEST_041_X             ; EXIT TEST 41
          54  00000000'EF  D0  0056  5$:     MOVL    DRIVE_TBL,R4                          ; GET DRIVE INFO TABLE
                 01    64  91  005D          CMPB    OFST_DRVTYP(R4),#RL02$T               ; IF SELECTED DRIVE IS AN RL02
                       06  13  0060          BEQL    6$                                    ; THEN GOTO 6$
                               0062          $DS_EXIT            TEST                      ; ELSE EXIT TEST
                               0068  6$:     $DS_BOPER           7$                        ; IF OPERATOR PRESENT GOTO 7$
                               0070          $DS_EXIT            TEST                      ; ELSE EXIT TEST
              56    34 A4  D0  0076  7$:     MOVL    OFST_GTSCMD(R4),R6                    ; GET GET STATUS COMMAND BITS
              57    40 A4  D0  007A          MOVL    OFST_MPBITS(R4),R7                    ; GET MPR BIT MNEMONICS
                       0C  D0  007E  10$:    MOVL    #<NOCHK$M_DRDY!NOCHK$M_DERR>,-; SET PROGRAM FLAGS
              00000000'EF      0080                  CONFIG
      00000000'EF      00  FB  0085          CALLS   #0,SUBSYS_INIT                        ; INIT SUBSYSTEM
                    1D 50  E8  008C          BLBS    R0,20$                                ; BRANCH IF NO ERRORS
                               008F          $DS_ERRHARD_S       ,L$UNIT,MSG14_SYSINIT,-
                               008F                              PRINT_REG,R1
                               00A5  ;;; TEST 41, SUBTEST 0, ERROR 1
                               00AC  20$:    $DS_CKLOOP          10$
                               00B4          $DS_PRINTF_S        OPR3_START,#DEV_NAME
00    00989680 8F    FFFFFFC4 8F  7A  00C7   EMUL    #-60,#<10*1000*1000>,#0,DELTA_TIME ; SET WAIT TIME = 60 SECS
              00000000'EF      00D3
                               00D8          $SETIMR_S           #32,DELTA_TIME,,          ; START TIMER
                               00E9          STARS
                               00E9  ;;*********************************************************************
                               00E9  ;++
                               00E9  ; LOOP HERE WAITING FOR DRIVE TO PROGRESS FROM THE LOAD CARTRIGE
                               00E9  ; STATE TO THE SPIN UP STATE.
                               00E9  ;--
                               00E9
   00000000'EF    0089 8F  3C  00E9          MOVZWL  #<MPR_02$M_DT!MPR_02$M_BH!-           ; SET EXPECTED MPR REG
                               00F2                  MPR_02$M_STA>,E.MP                    ;
                       56  DD  00F2  30$:    PUSHL   R6                                    ; IDCMPR DATA=GET STATUS
                       00  DD  00F4          PUSHL   #0                                    ; IDCDAR DATA
                       00  DD  00F6          PUSHL   #0                                    ; IDCCNT DATA
                       00  DD  00F8          PUSHL   #0                                    ; IDCBAR DATA
                       04  DD  00FA          PUSHL   #GETSTAT                              ; DEFINE COMMAND
      00000000'EF      05  FB  00FC          CALLS   #5,START_CMD                          ; EXECUTE GET STATUS CMD
                       0C  D0  0103          MOVL    #<NOCHK$M_DRDY!NOCHK$M_DERR>,-; SET PROGRAM FLAGS
              00000000'EF      0105                  CONFIG
      00000000'EF      00  FB  010A          CALLS   #0,STATUS_CHK                         ; CHECK FOR ANY ERRORS
                    2B 50  E8  0111          BLBS    R0,40$                                ; BRANCH IF NONE
                               0114          $DS_ERRHARD_S       ,L$UNIT,MSG17_GETSTAT,-
                               0114                              PRINT_REG,R1
                               012A  ;;; TEST 41, SUBTEST 0, ERROR 2
                               0131          $DS_CKLOOP          30$
                               0139          $DS_EXIT            TEST
   55    00000000'EF    03    00  EF  013F  40$:    EXTZV   #0,#3,T.MP,R5              ; GET DRIVE STATUS BITS
                    55    01  91  0148          CMPB    #MPR_02$M_STA,R5                   ; HAS DRIVE STATE 1 REACHED ?
                          03  12  014B          BNEQ    45$                                ; BRANCH IF NOT
                        02F1  31  014D          BRW     130$                               ; ELSE GOTO 130$
                                  0150  45$:    $READEF_S           #32,TIME_OUT           ; GET EVENT FLAGS
        8B 00000000'EF    00  E1  015F          BBC     #0,TIME_OUT,30$                    ; BRANCH IF NO TIME OUT OCCURED
                                  0167          $DS_ERRHARD_S       ,L$UNIT,MSG75_NO_ST1,-
                                  0167                              PRINT_REG
                                  017B  ;;; TEST 41, SUBTEST 0, ERROR 3
                                  0182          $SCANTIM_S                                 ; CANCEL TIMER REQUEST
```

```
                                  018B  1589          $DS_CKLOOP      10$
                                  0194  1590          $DS_EXIT        TEST
                                  019A  1591 50$:             STARS
                                  019A       ;;*********************************************************************
                                  019A  1592 ;++
                                  019A  1593 ; CHECK DRIVE STATUS FOR THE SPIN UP STATE
                                  019A  1594 ;--
                                  019A  1595
    00000000'EF  00000000'EF  D1  019A  1596          CMPL    E.MP,T.MP                          ; DRIVE STATUS OK SO FAR ?
                          43  13  01A5  1597          BEQL    60$                                ; BRANCH IF IT IS
                                  01A7  1598          $DS_ERRHARD_S   ,L$UNIT,MSG76_ST1_BS,-
                                  01A7  1599                          REG_ERRPT,#0,-
                                  01A7  1600                          E.MP,T.MP,R7
                                  01CB       ;;; TEST 41, SUBTEST 0, ERROR 4
                                  01D2  1601          $CANTIM_S                                  ; CANCEL TIMER REQUEST
                                  01DB  1602          $DS_CKLOOP      10$
                                  01E4  1603          $DS_EXIT        TEST
                                  01EA  1604          STARS
                                  01EA       ;;*********************************************************************
                                  01EA  1605 ;++
                                  01EA  1606 ; LOOP HERE WAITING FOR DRIVE TO PROGRESS FROM SPINUP STATE TO
                                  01EA  1607 ; THE BRUSH CYCLE STATE
                                  01EA  1608 ;--
                                  01EA  1609
       00000000'EF   008A 8F  3C  01EA  1610 60$:     MOVZWL  #<MPR_02$M_DT!MPR_02$M_BH!-        ; SET EXPECTED MPR FOR STATE 2
                                  01F3  1611                  MPR_02$M_STB>,E.MP                 ;
                          56  DD  01F3  1612 65$:     PUSHL   R6                                 ; IDCMPR DATA=GET STATUS
                          00  DD  01F5  1613          PUSHL   #0                                 ; IDCDAR DATA
                          00  DD  01F7  1614          PUSHL   #0                                 ; IDCCNT DATA
                          00  DD  01F9  1615          PUSHL   #0                                 ; IDCBAR DATA
                          04  DD  01FB  1616          PUSHL   #GETSTAT                           ; DEFINE COMMAND
           00000000'EF    05  FB  01FD  1617          CALLS   #5,START_CMD                       ; EXECUTE GET STATUS COMMAND
                          0C  D0  0204  1618          MOVL    #<NOCHK$M_DRDY!NOCHK$M_DERR>,-; SET PROGRAM FLAGS
                 00000000'EF      0206  1619                  CONFIG
           00000000'EF    00  FB  020B  1620          CALLS   #0,STATUS_CHK                      ; CHECK FOR ANY ERRORS
                       34 50  E8  0212  1621          BLBS    R0,70$                             ; BRANCH IF NONE
                                  0215  1622          $DS_ERRHARD_S   ,L$UNIT,MSG17_GETSTAT,-
                                  0215  1623                          PRINT_REG,R1
                                  022B       ;;; TEST 41, SUBTEST 0, ERROR 5
                                  0232  1624          $CANTIM_S                                  ; CANCEL TIMER REQUEST
                                  023B  1625          $DS_CKLOOP      65$
                                  0243  1626          $DS_EXIT        TEST
  55   00000000'EF    03  00  EF  0249  1627 70$:     EXTZV   #0,#3,T.MP,R5                      ; GET THE STATUS BITS
                  55  02  91  0252  1628          CMPB    #<MPR_02$M_STB>,R5                 ; STATE 2 REACHED ?
                      4A  13  0255  1629          BEQL    80$                                ; BRANCH IF YES
                                  0257  1630          $READEF_S       #32,TIME_OUT               ; GET EVENT FLAGS
       85 00000000'EF     00  E1  0266  1631          BBC     #0,TIME_OUT,65$                    ; BRANCH IF NO TIMEOUT OCCURED
                                  026E  1632          $DS_ERRHARD_S   ,L$UNIT,MSG77_NO_ST2,-
                                  026E  1633                          PRINT_REG
                                  0282       ;;; TEST 41, SUBTEST 0, ERROR 6
                                  0289  1634          $CANTIM_S                                  ; CANCEL TIMER REQUEST
                                  0292  1635          $DS_CKLOOP      10$
                                  029B  1636          $DS_EXIT        TEST
                                  02A1  1637 80$:     STARS
                                  02A1       ;;*********************************************************************
                                  02A1  1638 ;++
```

```
                                 02A1  ; CHECK DRIVE STATUS FOR BRUSH CYCLE STATE
                                 02A1  ;--
                                 02A1
 00000000'EF   00000000'EF   D1  02A1        CMPL    E.MP,T.MP                           ; DRIVE STATUS STILL OK
                        43   13  02AC        BEQL    90$                                 ; BRANCH IF YES
                                 02AE        $DS_ERRHARD_S    ,L$UNIT,MSG78_ST2_BS,-
                                 02AE                         REG_ERRPT,#0,-
                                 02AE                         E.MP,T.MP,R7
                                 02D2  ;;; TEST 41, SUBTEST 0, ERROR 7
                                 02D9        $CANTIM_S                                   ; CANCEL TIMER REQUEST
                                 02E2        $DS_CKLOOP       10$
                                 02EB        $DS_EXIT         TEST
                                 02F1        STARS
                                 02F1  ;;*****************************************************************
                                 02F1  ;++
                                 02F1  ; WAIT FOR BRUSH CYCLE TO START (BH BIT RESET)
                                 02F1  ;--
                                 02F1
                                 02F1  90$:
                        56   DD  02F1  95$:  PUSHL   R6                                  ; IDCMPR DATA=GET STATUS
                        00   DD  02F3        PUSHL   #0                                  ; IDCDAR DATA
                        00   DD  02F5        PUSHL   #0                                  ; IDCCNT DATA
                        00   DD  02F7        PUSHL   #0                                  ; IDCBAR DATA
                        04   DD  02F9        PUSHL   #GETSTAT                            ; DEFINE COMMAND
       00000000'EF      05   FB  02FB        CALLS   #5,START_CMD                        ; EXECUTE GET STATUS
                        0C   D0  0302        MOVL    #<NOCHK$M_DRDY!NOCHK$M_DERR>,-; SET PROGRAM FLAGS
                00000000'EF      0304                CONFIG
       00000000'EF      00   FB  0309        CALLS   #0,STATUS_CHK                       ; CHECK FOR ANY ERRORS
                     34 50   E8  0310        BLBS    R0,100$                             ; BRANCH IF NONE
                                 0313        $DS_ERRHARD_S    ,L$UNIT,MSG17_GETSTAT,-
                                 0313                         PRINT_REG,R1
                                 0329  ;;; TEST 41, SUBTEST 0, ERROR 8
                                 0330        $CANTIM_S                                   ; CANCEL TIMER REQUEST
                                 0339        $DS_CKLOOP       95$
                                 0341        $DS_EXIT         TEST
    4A 00000000'EF      03   E1  0347  100$: BBC     #MPR_02$V_BH,T.MP,110$              ; BRANCH IF BRUSH CYCLE STARTED
                                 034F        $READEF_S        #32,TIME_OUT               ; GET EVENT FLAGS
    8B 00000000'EF      00   E1  035E        BBC     #0,TIME_OUT,95$                     ; BRANCH IF NO TIMEOUT OCCURED
                                 0366        $DS_ERRHARD_S    ,L$UNIT,MSG79_BH_NC,-
                                 0366                         PRINT_REG
                                 037A  ;;; TEST 41, SUBTEST 0, ERROR 9
                                 0381        $CANTIM_S                                   ; CANCEL TIMER REQUEST
                                 038A        $DS_CKLOOP       10$
                                 0393        $DS_EXIT         TEST
                                 0399  110$: STARS
                                 0399  ;;*****************************************************************
                                 0399  ;++
                                 0399  ; WAIT FOR BRUSH CYCLE TO COMPLETE (BH BIT SET)
                                 0399  ;--
                                 0399
                        56   DD  0399  115$: PUSHL   R6                                  ; IDCMPR DATA=GET STATUS
                        00   DD  039B        PUSHL   #0                                  ; IDCDAR DATA
                        00   DD  039D        PUSHL   #0                                  ; IDCCNT DATA
                        00   DD  039F        PUSHL   #0                                  ; IDCBAR DATA
                        04   DD  03A1        PUSHL   #GETSTAT                            ; DEFINE COMMAND
       00000000'EF      05   FB  03A3        CALLS   #5,START_CMD                        ; EXECUTE GET STATUS CMD
```

```
                         0C      D0  03AA        MOVL     #<NOCHK$M_DERR!NOCHK$M_DRDY>,-  ; SET PROGRAM FLAGS
              00000000'EF            03AC                 CONFIG
         00000000'EF  00     FB  03B1        CALLS    #0,STATUS_CHK                   ; CHECK FOR ANY ERRORS
                    34 50    E8  03B8        BLBS     R0,120$                         ; BRANCH IF NONE
                                 03BB        $DS_ERRHARD_S    ,L$UNIT,MSG17_GETSTAT,-
                                 03BB                         PRINT_REG,R1
                                 03D1   ;;; TEST 41, SUBTEST 0, ERROR 10
                                 03D8        $CANTIM_S                                ; CANCEL TIMER REQUEST
                                 03E1        $DS_CKLOOP       115$
                                 03E9        $DS_EXIT         TEST
     4A 00000000'EF   03     E0  03EF   120$: BBS     #MPR_02$V_BH,T.MP,130$          ; BRANCH IF BRUSH CYCLE DONE
                                 03F7        $READEF_S        #32,TIME_OUT            ; GET EVENT FLAGS
     8B 00000000'EF   00     E1  0406        BBC      #0,TIME_OUT,115$                ; BRANCH IF NO TIMEOUT OCCURED
                                 040E        $DS_ERRHARD_S    ,L$UNIT,MSG80_BH_NF,-
                                 040E                         PRINT_REG
                                 0422   ;;; TEST 41, SUBTEST 0, ERROR 11
                                 0429        $CANTIM_S                                ; CANCEL TIMER REQUEST
                                 0432        $DS_CKLOOP       10$
                                 043B        $DS_EXIT         TEST
                                 0441        STARS
                                 0441   ;;*********************************************************************
                                 0441   ;++
                                 0441   ; CHECK DRIVE STATE 3 (LOAD HEADS) STATUS
                                 0441   ;--
                                 0441
                                 0441   130$: $CANTIM_S                               ; CANCEL TIMER REQUEST
00   00989680 8F   FFFFFFE2 8F   7A  044A        EMUL     #-30,#<10*1000*1000>,#0,DELTA_TIME ; SET WAIT TIME = 30 SECS
              00000000'EF            0456
                                 045B        $SETIMR_S        #32,DELTA_TIME,,        ; START TIMER
       00000000'EF   008B 8F     3C  046C        MOVZWL   #<MPR_02$M_DT!  -               ; SET EXPECTED MPR REG
                                 0475                 MPR_02$M_BH!MPR_02$M_STB!-      ;
                                 0475                 MPR_02$M_STA>,E.MP              ;
                         56      DD  0475   135$: PUSHL   R6                              ; IDCMPR DATA=GET STATUS
                         00      DD  0477        PUSHL    #0                              ; IDCDAR DATA
                         00      DD  0479        PUSHL    #0                              ; IDCCNT DATA
                         00      DD  047B        PUSHL    #0                              ; IDCBAR DATA
                         04      DD  047D        PUSHL    #GETSTAT                        ; DEFINE COMMAND
         00000000'EF  05     FB  047F        CALLS    #5,START_CMD                    ; EXECUTE GET STATUS CMD
                         0C      D0  0486        MOVL     #<NOCHK$M_DRDY!NOCHK$M_DERR>,-; SET PROGRAM FLAGS
              00000000'EF            0488                 CONFIG
         00000000'EF  00     FB  048D        CALLS    #0,STATUS_CHK                   ; CHECK FOR ANY ERRORS
                    34 50    E8  0494        BLBS     R0,140$                         ; BRANCH IF NONE
                                 0497        $DS_ERRHARD_S    ,L$UNIT,MSG17_GETSTAT,-
                                 0497                         PRINT_REG,R1
                                 04AD   ;;; TEST 41, SUBTEST 0, ERROR 12
                                 04B4        $CANTIM_S                                ; CANCEL TIMER REQUEST
                                 04BD        $DS_CKLOOP       135$
                                 04C5        $DS_EXIT         TEST
 55   00000000'EF   03   00  EF  04CB   140$: EXTZV   #0,#3,T.MP,R5                   ; GET THE STATUS BITS
                    55   03  91  04D4        CMPB     #3,R5                           ; STATE 3 REACHED ?
                    32       13  04D7        BEQL     145$                            ; BRANCH IF YES
                                 04D9        $READEF_S        #32,TIME_OUT            ; GET EVENT FLAGS
     85 00000000'EF   00     E1  04E8        BBC      #0,TIME_OUT,135$                ; BRANCH IF NO TIMEOUT OCCURED
                                 04F0        $DS_ERRHARD_S    ,L$UNIT,MSG77_NO_ST3,-
                                 04F0                         PRINT_REG
                                 0504   ;;; TEST 41, SUBTEST 0, ERROR 13
```

```
                                  050B  145$:    $CANTIM_S                                       ; CANCEL TIMER REQUEST
                                  0514           $DS_CKLOOP      10$
    00000000'EF  00000200 8F  CA  051D           BICL2   #MPR_02$M_VC,T.MP                       ; VC IS A DONT CARE AT THIS TIME
    00000000'EF  00000000'EF  D1  0528           CMPL    E.MP,T.MP                               ; CHECK STATE 3 STATUS
                          3A  13  0533           BEQL    150$                                    ; BRANCH IF OK
                                  0535           $DS_ERRHARD_S    ,L$UNIT,MSG81_ST3_BS,-
                                  0535                            REG_ERRPT,#0,-
                                  0535                            E.MP,T.MP,R7
                                  0559     ;;; TEST 41, SUBTEST 0, ERROR 14
                                  0560           $DS_CKLOOP      10$
                                  0569           $DS_EXIT        TEST
                                  056F           STARS
                                  056F     ;;*********************************************************************
                                  056F     ;++
                                  056F     ; WAIT FOR DRIVE TO PROGRESS FROM LOAD HEADS STATE TO THE
                                  056F     ; SEEK STATE (STATE 4)
                                  056F     ;--
                                  056F
               59    3A98 8F  3C  056F  150$:    MOVZWL  #15000,R9                               ; SET WAIT COUNT
                                  0574  153$:    $DS_WAITUS_S    #1                              ; WAIT 10 MICRO SECS
                          56  DD  057F  155$:    PUSHL   R6                                      ; IDCMPR DATA=GET STATUS
                          00  DD  0581           PUSHL   #0                                      ; IDCDAR DATA
                          00  DD  0583           PUSHL   #0                                      ; IDCCNT DATA
                          00  DD  0585           PUSHL   #0                                      ; IDCBAR DATA
                          04  DD  0587           PUSHL   #GETSTAT                                ; DEFINE COMMAND
           00000000'EF    05  FB  0589           CALLS   #5,START_CMD                            ; EXECUTE GET STATUS CMD
                          0C  D0  0590           MOVL    #<NOCHK$M_DRDY!NOCHK$M_DERR>,-; SET PROGRAM FLAGS
              00000000'EF         0592                   CONFIG
           00000000'EF    00  FB  0597           CALLS   #0,STATUS_CHK                           ; CHECK FOR ANY ERRORS
                       2B 50  E8  059E           BLBS    R0,160$                                 ; BRANCH IF NONE
                                  05A1           $DS_ERRHARD_S    ,L$UNIT,MSG17_GETSTAT,-
                                  05A1                            PRINT_REG,R1
                                  05B7     ;;; TEST 41, SUBTEST 0, ERROR 15
                                  05BE           $DS_CKLOOP      155$
                                  05C6           $DS_EXIT        TEST
   55   00000000'EF    03    00  EF  05CC  160$:    EXTZV   #0,#3,T.MP,R5                           ; GET STATE BITS
                      55    04  91  05D5           CMPB    #<MPR_02$M_STC>,R5                      ; REACHED STATE 4 YET ?
                            2D  13  05D8           BEQL    170$                                    ; BRANCH IF YES
                         97 59  F5  05DA           SOBGTR  R9,153$                                 ; TRY AGAIN
                                    05DD           $DS_ERRHARD_S    ,L$UNIT,MSG82_NO_ST4,-
                                    05DD                            PRINT_REG
                                    05F1     ;;; TEST 41, SUBTEST 0, ERROR 16
                                    05F8           $DS_CKLOOP      10$
                                    0601           $DS_EXIT        TEST
                                    0607           STARS
                                    0607     ;;*********************************************************************
                                    0607     ;++
                                    0607     ; CHECK THE DRIVE STATUS FOR THE SEEK STATE (STATE 4)
                                    0607     ;--
                                    0607
        00000000'EF    029C 8F  3C  0607  170$:    MOVZWL  #<MPR_02$M_DT!MPR_02$M_HO!-     ; SET EXPECTED MPR REG
                                    0610                   MPR_02$M_VC!MPR_02$M_BH!-       ;
                                    0610                   MPR_02$M_STC>,E.MP              ;
    00000000'EF    00000000'EF  D1  0610           CMPL    E.MP,T.MP                       ; STATE 4 STATUS OK ?
                            3A  13  061B           BEQL    175$                            ; BRANCH IF IT IS
                                    061D           $DS_ERRHARD_S    ,L$UNIT,MSG83_ST4_BS,-
```

```
                                  061D  1791                                  REG_ERRPT,#0,-
                                  061D  1792                                  E.MP,T.MP,R7
                                  0641        ;;; TEST 41, SUBTEST 0, ERROR 17
                                  0648  1793            $DS_CKLOOP            10$
                                  0651  1794            $DS_EXIT              TEST
          2A 00000000'EF    0E    E0  0657  1795 175$:  BBS     #CSR$V_DE,T.CS,180$                ; DRIVE ERROR SHOULD BE SET NOW
                                  065F  1796            $DS_ERRHARD_S    ,L$UNIT,MS83A_ST4_DE,-
                                  065F  1797                                  PRINT_REG
                                  0673        ;;; TEST 41, SUBTEST 0, ERROR 18
                                  067A  1798            $DS_CKLOOP            10$
                                  0683  1799            $DS_EXIT              TEST
                                  0689  1800            STARS
                                  0689        ;;*******************************************************************
                                  0689  1801  ;++
                                  0689  1802  ; WAIT FOR DRIVE TO PROGRESS FROM SEEK STATE TO THE LOCKED ON
                                  0689  1803  ; TRACK STATE (STATE 5)
                                  0689  1804  ;--
                                  0689  1805
               59     05DC 8F     3C  0689  1806 180$:  MOVZWL  #1500,R9                            ; SET WAIT COUNT FOR STATE 5
                                  068E  1807 183$:  $DS_WAITUS_S      #10                       ; WAIT 100 MICROSECS
                          56      DD  0699  1808 185$:  PUSHL   R6                                  ; IDCMPR DATA=GET STATUS
                          00      DD  069B  1809            PUSHL   #0                              ; IDCDAR
                          00      DD  069D  1810            PUSHL   #0                              ; IDCCNT DATA
                          00      DD  069F  1811            PUSHL   #0                              ; IDCBAR DATA
                          04      DD  06A1  1812            PUSHL   #GETSTAT                        ; DEFINE COMMAND
          00000000'EF     05      FB  06A3  1813            CALLS   #5,START_CMD                    ; EXECUTE GET STATUS CMD
                          0C      D0  06AA  1814            MOVL    #<NOCHK$M_DRDY!NOCHK$M_DERR>,-; SET PROGRAM FLAGS
                  00000000'EF         06AC  1815                    CONFIG
          00000000'EF     00      FB  06B1  1816            CALLS   #0,STATUS_CHK                   ; CHECK FOR ANY ERRORS
                       2B 50      E8  06B8  1817            BLBS    R0,190$                         ; BRANCH IF NONE
                                  06BB  1818            $DS_ERRHARD_S    ,L$UNIT,MSG17_GETSTAT,-
                                  06BB  1819                                  PRINT_REG,R1
                                  06D1        ;;; TEST 41, SUBTEST 0, ERROR 19
                                  06D8  1820            $DS_CKLOOP            185$
                                  06E0  1821            $DS_EXIT              TEST
                                  06E6  1822            STARS
                                  06E6        ;;*******************************************************************
                                  06E6  1823  ;++
                                  06E6  1824  ; CHECK DRIVE STATUS FOR LOCKED ON TRACK STATE (STATE 5)
                                  06E6  1825  ;--
                                  06E6  1826
   55     00000000'EF     03    00    EF  06E6  1827 190$:  EXTZV   #0,#3,T.MP,R5                      ; GET DRIVE STATE BITS
                     55    05      91  06EF  1828            CMPB    #<MPR_02$M_STC!MPR_02$M_STA>,R5 ; IS DRIVE IN STATE 5 ?
                           2D      13  06F2  1829            BEQL    200$                            ; BRANCH IF IT IS
                        97 59      F5  06F4  1830            SOBGTR  R9,183$                         ; DECR AND TRY AGAIN
                                  06F7  1831            $DS_ERRHARD_S     ,L$UNIT,MSG84_NO_ST5,-
                                  06F7  1832                                  PRINT_REG
                                  070B        ;;; TEST 41, SUBTEST 0, ERROR 20
                                  0712  1833            $DS_CKLOOP            10$
                                  071B  1834            $DS_EXIT              TEST
       00000000'EF    029D 8F     3C  0721  1835 200$:  MOVZWL  #<MPR_02$M_VC!MPR_02$M_DT!-         ; SET EXPECTED MPR REG
                                  072A  1836                    MPR_02$M_HO!MPR_02$M_BH!-         ;
                                  072A  1837                    MPR_02$M_STC!MPR_02$M_STA>,E.MP   ;
  00000000'EF    00000000'EF      D1  072A  1838            CMPL    E.MP,T.MP                       ; STATE 5 STATUS OK ?
                          3A      13  0735  1839            BEQL    210$                            ; BRANCH IF IT IS
                                  0737  1840            $DS_ERRHARD_S    ,L$UNIT,MSG85_ST5_BS,-
```

```
                              0737  1841                       REG_ERRPT,#0,-
                              0737  1842                       E.MP,T.MP,R7
                              075B       ;;; TEST 41, SUBTEST 0, ERROR 21
                              0762  1843         $DS_CKLOOP    185$
                              076B  1844         $DS_EXIT      TEST
                              0771  1845         STARS
                              0771       ;;*******************************************************************
                              0771  1846 ;++
                              0771  1847 ; CHECK THAT DRIVE READY (DRDY BIT) IN THE CONTROL AND STATUS REGISTER
                              0771  1848 ; IS SET NO THAT HEADS ARE LOADED.
                              0771  1849 ;--
                              0771  1850
                              0771  1851 210$:   $DS_WAITMS_S   #10                        ; WAIT 100 MS FOR DRDY TO SET
         00000000'EF    16    077C  1852         JSB     RD_IDCREGS                         ; UPDATE IDC REGISTERS
   1B 00000000'EF    00    E0    0782  1853         BBS     #CSR$V_DRDY,T.CS,220$              ; BRANCH IF DRIVE READY SET
                              078A  1854         $DS_ERRHARD_S   ,L$UNIT,MSG86_DRDY_NS,-
                              078A  1855                       PRINT_REG
                              079E       ;;; TEST 41, SUBTEST 0, ERROR 22
                              07A5  1856 220$:   $DS_CKLOOP    10$
                              07AE  1857         STARS
                              07AE       ;;*******************************************************************
                              07AE  1858 ;++
                              07AE  1859 ; EXECUTE A DRIVE RESET AND GET THE DRIVE STATUS WORD AND TEST
                              07AE  1860 ; THAT THE VOLUME CHECK BIT (VC) IS RESET.
                              07AE  1861 ;--
                              07AE  1862
 00000000'EF    009D 8F    3C    07AE  1863         MOVZWL  #<MPR_02$M_DT!MPR_02$M_HO!-        ; SET EXPECTED DRIVE STATUS
                              07B7  1864                 MPR_02$M_BH!MPR_02$M_STC!MPR_02$M_STA>,E.MP ;
              50 A4    DD    07B7  1865 225$:   PUSHL   OFST_DRVRST(R4)                    ; IDCMPR DATA=DRIVE RESET
                 00    DD    07BA  1866         PUSHL   #0                                 ; IDCDAR DATA
                 00    DD    07BC  1867         PUSHL   #0                                 ; IDCCNT DATA
                 00    DD    07BE  1868         PUSHL   #0                                 ; IDCBAR DATA
                 04    DD    07C0  1869         PUSHL   #GETSTAT                           ; DEFINE COMMAND
      00000000'EF    05    FB    07C2  1870         CALLS   #5,START_CMD                       ; EXECUTE DRIVE RESET
                 0C    D0    07C9  1871         MOVL    #<NOCHK$M_DRDY!NOCHK$M_DERR>,-; SET PROGRAM FLAGS
         00000000'EF          07CB  1872                 CONFIG
      00000000'EF    00    FB    07D0  1873         CALLS   #0,STATUS_CHK                      ; CHECK FOR ANY ERRORS
              1D 50    E8    07D7  1874         BLBS    R0,230$                            ; BRANCH IF NONE
                              07DA  1875         $DS_ERRHARD_S   ,L$UNIT,MSG17_GETSTAT,-
                              07DA  1876                       PRINT_REG,R1
                              07F0       ;;; TEST 41, SUBTEST 0, ERROR 23
                              07F7  1877 230$:   $DS_CKLOOP    225$
   2B 00000000'EF    09    E1    07FF  1878         BBC     #MPR_02$V_VC,T.MP,240$             ; BRANCH IF VOLUME CHECK IS 0
                              0807  1879         $DS_ERRHARD_S   ,L$UNIT,MSG87_VC_NC,-
                              0807  1880                       REG_ERRPT,#0,-
                              0807  1881                       E.MP,T.MP,R7
                              082B       ;;; TEST 41, SUBTEST 0, ERROR 24
                              0832  1882 240$:   $DS_CKLOOP    10$
                              083B  1883         $DS_ENDTEST
                              0846  1884         $DS_SBTTL <HEAD UNLOADING TEST - RL02>,ALIGN=LONG
                              0846               .SBTTL  TEST 42:  HEAD UNLOADING TEST - RL02
                              0846               $D2_SBTTL       042,<HEAD UNLOADING TEST - RL02>,<LONG>
                          00000000                       .PSECT  TEST_042, LONG, NOWRT
                              0020
```

```
                              0020
                              0020  ;++
                              0020  ;
                              0020  ; TEST DESCRITPION:
                              0020  ;
                              0020  ; THIS TEST WILL REQUEST THE OPERATOR TO UNLOAD THE DRIVE CURRENTLY
                              0020  ; UNDER TEST.  THEN WE WILL TEST THE DRIVE STATUS DURING THE SPIN
                              0020  ; DOWN AND HEAD UNLOADING STATES AND WAIT FOR DRIVE TO STOP IN THE
                              0020  ; LOAD CARTRIDGE STATE.
                              0020  ;
                              0020  ; TEST STEPS:
                              0020  ;
                              0020  ;        1.  INITIALIZE SUBSYSTEM
                              0020  ;        2.  TELL OPERATOR TO UNLOAD DRIVE UNDER TEST
                              0020  ;        3.  DO NOT WAIT FOR RETURN FROM OPERATOR
                              0020  ;        4.  WAIT FOR DRIVE TO PROGRESS FROM STATE 5 TO STATE 6
                              0020  ;        5.  TEST DRIVE STATUS IN STATE 6
                              0020  ;        6.  WAIT FOR DRIVE TO PROGRESS FROM STATE 6 TO STATE 7
                              0020  ;        7.  TEST DRIVE STATUS IN STATE 7
                              0020  ;        8.  WAIT FOR DRIVE TO PROGRESS FROM STATE 7 TO STATE 0
                              0020  ;
                              0020  ; DEBUG:
                              0020  ;
                              0020  ;        DRIVE LOGIC MODULE
                              0020  ;        AC SERVO
                              0020  ;
                              0020  ;--
                              0020
                              0020
                              0020          $DS_BGNDATA
      00000000 00000001       0020          .LONG   1,LUN0
      00000001 00000001       0028          .LONG   1,LUN1
      00000002 00000001       0030          .LONG   1,LUN2
      00000003 00000001       0038          .LONG   1,LUN3
                              0040          $DS_ENDDATA
                              0044
                              0044          $DS_BGNTEST <MANUAL,ALL>,ALIGN=LONG
                              0046          GET_LUN -                               ; GET LUN INFO
                              0046            UNIT= OFST_UNIT(AP)                   ; LOGICAL UNIT
              04 AC    DD     0046          PUSHL   OFST_UNIT(AP)
  00000000'EF    01    FB     0049          CALLS   #1,GET_LUN$A
                              0050          BDEVFOUND        5$                     ; IF DEV AVAILABLE GOTO 5$
        06 50    00    E2     0050          BBSS    #0,R0,5$
           50    01    D0     0054                  MOVL    #1,R0           ; NOT AN ERROR
             0337      31     0057                  BRW     TEST_042_X      ; EXIT TEST 42
54    00000000'EF      D0     005A    5$:   MOVL    DRIVE_TBL,R4            ; GET DIRVE INFO TABLE
           01    64    91     0061          CMPB    OFST_DRVTYP(R4),#RL02$T ; IF DRIVE IS AN RL02
                 06    13     0064          BEQL    6$                      ; THEN GOTO 6$
                              0066          $DS_EXIT         TEST           ; ELSE EXIT TEST
                              006C    6$:   $DS_BOPER        7$             ; IF OPERATOR PRESENT GOTO 7$
                              0074          $DS_EXIT         TEST           ; ELSE EXIT TEST
     56    34 A4       D0     007A    7$:   MOVL    OFST_GTSCMD(R4),R6      ; GET GET STATUS CMD BITS
     57    40 A4       D0     007E          MOVL    OFST_MPBITS(R4),R7      ; GET MPR REG BIT MNEMONICS
                 0C    D0     0082    10$:  MOVL    #<NOCHK$M_DRDY!NOCHK$M_DERR>,-
       00000000'EF            0084                  CONFIG                  ; SET PROGRAM FLAGS
  00000000'EF    00    FB     0089          CALLS   #0,SUBSYS_INIT          ; INIT SUBSYSTEM
```

```
                        1D 50    E8  0090  1937            BLBS    R0,20$                                ; BRANCH IF NO ERRORS
                                     0093  1938            $DS_ERRHARD_S   ,L$UNIT,MSG14_SYSINIT,-
                                     0093  1939                            PRINT_REG,R1
                                     00A9       ;;; TEST 42, SUBTEST 0, ERROR 1
                                     00B0  1940 20$:       $DS_CKLOOP      10$
                                     00B8  1941            $DS_PRINTF_S    OPR5_UNLOAD,#DEV_NAME     ; TELL OPR TO UNLOAD DRIVE
                                     00CB  1942            STARS
                                     00CB       ;;********************************************************************
                                     00CB  1943 ;++
                                     00CB  1944 ; WAIT HERE FOR DRIVE STATE 6 (UNLOAD HEADS) TO OCCUR
                                     00CB  1945 ;--
                                     00CB  1946
00     00989680 8F    FFFFFFC4 8F    7A  00CB  1947            EMUL    #-60,#<10*1000*1000>,#0,-         ; SET TIME CONSTANT
                     00000000'EF         00D7  1948                    DELTA_TIME                        ; FOR 60 SECONDS
                                     00DC  1949            $SETIMR_S       #32,DELTA_TIME,,              ; START TIMER
                           56    DD  00ED  1950 25$:       PUSHL   R6                                    ; IDCMPR DATA=GET STATUS
                           00    DD  00EF  1951            PUSHL   #0                                    ; IDCDAR DATA
                           00    DD  00F1  1952            PUSHL   #0                                    ; IDCCNT DATA
                           00    DD  00F3  1953            PUSHL   #0                                    ; IDCBAR DATA
                           04    DD  00F5  1954            PUSHL   #GETSTAT                              ; DEFINE COMMAND
              00000000'EF  05    FB  00F7  1955            CALLS   #5,START_CMD                          ; EXECUTE GET STATUS CMD
                           0C    D0  00FE  1956            MOVL    #<NOCHK$M_DRDY!NOCHK$M_DERR>,-; SET FLAGS
                  00000000'EF        0100  1957                    CONFIG
              00000000'EF  00    FB  0105  1958            CALLS   #0,STATUS_CHK                         ; CHECK FOR ANY ERRORS
                        2B 50    E8  010C  1959            BLBS    R0,30$                                ; BRANCH IF NONE
                                     010F  1960            $DS_ERRHARD_S   ,L$UNIT,MSG17_GETSTAT,-
                                     010F  1961                            PRINT_REG,R1
                                     0125       ;;; TEST 42, SUBTEST 0, ERROR 2
                                     012C  1962            $DS_CKLOOP      25$
                                     0134  1963            $DS_EXIT        TEST
    55     00000000'EF     03    00    EF  013A  1964 30$:       EXTZV   #0,#3,T.MP,R5                         ; R5 <-- STATE BITS
                           55    06    91  0143  1965            CMPB    #<MPR_02$M_STC!MPR_02$M_STB>,R5 ; REACHED STATE 6 ?
                           41    13  0146  1966            BEQL    40$                                   ; BRANCH IF YES
                                     0148  1967            $READEF_S       #32,TIME_OUT                  ; GET EVENT FLAGS
          8E 00000000'EF   00    E1  0157  1968            BBC     #0,TIME_OUT,25$                       ; BRANCH IF NOT TIMED OUT
                                     015F  1969            $DS_ERRHARD_S   ,L$UNIT,MSG88_NO_ST6,-
                                     015F  1970                            PRINT_REG
                                     0173       ;;; TEST 42, SUBTEST 0, ERROR 3
                                     017A  1971            $DS_CKLOOP      10$
                                     0183  1972            $DS_EXIT        TEST
                                     0189  1973 40$:       $CANTIM_S                                     ; STOP TIMER
                                     0192  1974            STARS
                                     0192       ;;********************************************************************
                                     0192  1975 ;++
                                     0192  1976 ; CHECK DRIVE STATUS FOR THE UNLOAD HEADS STATE
                                     0192  1977 ;--
                                     0192  1978
        00000000'EF    009E 8F   3C  0192  1979            MOVZWL  #<MPR_02$M_DT!MPR_02$M_HO!-       ; SET EXPECTED DRIVE STATUS
                                     019B  1980                    MPR_02$M_BH!MPR_02$M_STC!-        ;
                                     019B  1981                    MPR_02$M_STB>,E.MP                ;
     00000000'EF    00000000'EF  D1  019B  1982            CMPL    E.MP,T.MP                             ; EXPECTED = ACTUAL ?
                           3A    13  01A6  1983            BEQL    50$                                   ; BRANCH IF IT IS
                                     01A8  1984            $DS_ERRHARD_S   ,L$UNIT,MSG89_ST6_BS,-
                                     01A8  1985                            REG_ERRPT,#0,-
                                     01A8  1986                            E.MP,T.MP,R7
                                     01CC       ;;; TEST 42, SUBTEST 0, ERROR 4
```

```
                               01D3               $DS_CKLOOP      10$
                               01DC               $DS_EXIT        TEST
                               01E2               STARS
                               01E2       ;;*****************************************************************
                               01E2       ;++
                               01E2       ; WAIT HERE FOR DRIVE STATE 7 (SPIN DOWN) TO OCCUR
                               01E2       ;--
                               01E2
              54    1E    D0   01E2  50$:     MOVL    #30,R4                          ; SET WAIT COUNT FOR STATE 7
                               01E5  55$:     $DS_WAITMS_S    #1                      ; WAIT A MILLI SEC
                    56    DD   01F0           PUSHL   R6                              ; IDCMPR DATA=GET STATUS
                    00    DD   01F2           PUSHL   #0                              ; IDCDAR DATA
                    00    DD   01F4           PUSHL   #0                              ; IDCCNT DATA
                    00    DD   01F6           PUSHL   #0                              ; IDCBAR DATA
                    04    DD   01F8           PUSHL   #GETSTAT                        ; DEFINE COMMAND
       00000000'EF  05    FB   01FA           CALLS   #5,START_CMD                    ; EXECUTE DRIVE STATUS
                    0C    D0   0201           MOVL    #<NOCHK$M_DRDY!NOCHK$M_DERR>,-; SET FLAGS FOR SUBROUTINE
          00000000'EF          0203                   CONFIG
       00000000'EF  00    FB   0208           CALLS   #0,STATUS_CHK                   ; CHECK FOR ANY ERRORS
              2B 50       E8   020F           BLBS    R0,60$                          ; BRANCH IF NONE
                               0212           $DS_ERRHARD_S   ,L$UNIT,MSG17_GETSTAT,-
                               0212                           PRINT_REG,R1
                               0228       ;;; TEST 42, SUBTEST 0, ERROR 5
                               022F           $DS_CKLOOP      55$
                               0237           $DS_EXIT        TEST
  55   00000000'EF    03    00    EF   023D  60$:     EXTZV   #0,#3,T.MP,R5                   ; R5 <-- STATE BITS
              55    07    91   0246           CMPB    #<MPR_02$M_STC!MPR_02$M_STB!-   ; STATE 7 REACHED
                               0249                   MPR_02$M_STA>,R5                ;
                    2D    13   0249           BEQL    70$                             ; BRANCH IF YES
              97 54       F5   024B           SOBGTR  R4,55$                          ; TRY AGAIN UNTIL TIMING OUT
                               024E           $DS_ERRHARD_S   ,L$UNIT,MSG90_NO_ST7,-
                               024E                           PRINT_REG
                               0262       ;;; TEST 42, SUBTEST 0, ERROR 6
                               0269           $DS_CKLOOP      10$
                               0272           $DS_EXIT        TEST
                               0278           STARS
                               0278       ;;*****************************************************************
                               0278       ;++
                               0278       ; CHECK DRIVE STATUS FOR THEE SPIN DOWN STATE
                               0278       ;--
                               0278
     00000000'EF    008F 8F   3C   0278  70$:     MOVZWL  #<MPR_02$M_DT!MPR_02$M_BH!-     ; SET EXPECTED DRIVE STATUS
                               0281                   MPR_02$M_STC!MPR_02$M_STB!-     ;
                               0281                   MPR_02$M_STA>,E.MP              ;
 00000000'EF    00000000'EF   D1   0281           CMPL    E.MP,T.MP                       ; EXPECTED = ACTUAL ?
                    3A    13   028C           BEQL    80$                             ; BRANCH IF YES
                               028E           $DS_ERRHARD_S   ,L$UNIT,MSG91_ST7_BS,-
                               028E                           REG_ERRPT,#0,-
                               028E                           E.MP,T.MP,R7
                               02B2       ;;; TEST 42, SUBTEST 0, ERROR 7
                               02B9           $DS_CKLOOP      10$
                               02C2           $DS_EXIT        TEST
                               02C8           STARS
                               02C8       ;;*****************************************************************
                               02C8       ;++
                               02C8       ; WAIT HERE FOR DRIVE STATE 0 (LOAD CARTRIDGE) TO OCCUR
```

```
                                   02C8  2037 ;--
                                   02C8  2038
00    00989680 8F   FFFFFFE2 8F  7A 02C8  2039 80$:   EMUL    #-30,#<10*1000*1000>,#0,-        ; SET TIMER FOR 30 SECONDS
                    00000000'EF        02D4  2040                 DELTA_TIME
                                   02D9  2041         $SETIMR_S         #32,DELTA_TIME,,       ; START TIMER
                         56    DD  02EA  2042 85$:    PUSHL   R6                               ; IDCMPR DATA=GET STATUS
                         00    DD  02EC  2043         PUSHL   #0                               ; IDCDAR
                         00    DD  02EE  2044         PUSHL   #0                               ; IDCCNT DATA
                         00    DD  02F0  2045         PUSHL   #0                               ; IDCBAR DATA
                         04    DD  02F2  2046         PUSHL   #GETSTAT                         ; DEFINE CMD
              00000000'EF  05  FB  02F4  2047         CALLS   #5,START_CMD                     ; EXECUTE GET STATUS CMD
                         0C    D0  02FB  2048         MOVL    #<NOCHK$M_DRDY!NOCHK$M_DERR>,-; SET FLAGS FOR SUBROUTINE
                    00000000'EF    02FD  2049                 CONFIG
              00000000'EF  00  FB  0302  2050         CALLS   #0,STATUS_CHK                    ; CHECK FOR ANY ERRORS
                      2B 50    E8  0309  2051         BLBS    R0,90$                           ; BRANCH IF NONE
                                   030C  2052         $DS_ERRHARD_S     ,L$UNIT,MSG17_GETSTAT,-
                                   030C  2053                           PRINT_REG,R1
                                   0322       ;;; TEST 42, SUBTEST 0, ERROR 8
                                   0329  2054         $DS_CKLOOP        85$
                                   0331  2055         $DS_EXIT          TEST
   55   00000000'EF    03    00    EF  0337  2056 90$:    EXTZV   #0,#3,T.MP,R5                    ; R5 <-- STATE BITS
                         55    D5  0340  2057         TSTL    R5                               ; STATE ZERO REACHED ?
                         41    13  0342  2058         BEQL    100$                             ; BRANCH IF YES
                                   0344  2059         $READEF_S         #32,TIME_OUT           ; GET EVENT FLAGS
        8F 00000000'EF     00    E1  0353  2060         BBC     #0,TIME_OUT,85$                  ; BRANCH IF NOT TIMED OUT
                                   035B  2061         $DS_ERRHARD_S     ,L$UNIT,MSG92_NO_STO,-
                                   035B  2062                           PRINT_REG
                                   036F       ;;; TEST 42, SUBTEST 0, ERROR 9
                                   0376  2063         $DS_CKLOOP        10$
                                   037F  2064         $DS_EXIT          TEST
                                   0385  2065 100$:   $CANTIM_S                                ; STOP TIMER
                                   038E  2066         $DS_ENDTEST
                                   0399  2067         $DS_SBTTL <DRIVE SELECT TEST>,ALIGN=LONG
                                   0399               .SBTTL  TEST 43:  DRIVE SELECT TEST
                                   0399               $D2_SBTTL         043,<DRIVE SELECT TEST>,<LONG>
                           00000000                           .PSECT    TEST_043, LONG, NOWRT
                                   0014
```

```
                             0014
                             0014      ;++
                             0014      ;
                             0014      ; TEST DESCRIPTION:
                             0014      ;
                             0014      ; THIS TEST WILL ASK THE OPERATOR TO REMOVE ALL DRIVE PLUGS FROM THE
                             0014      ; RL02 DRIVES.  TELL OPERATOR TO INSERT DRIVE PLUG #0 IN UUT AND VERIFY
                             0014      ; THAT GETTING A DRIVE STATUS FROM THAT UNIT IS POSSIBLE WITHOUT ERROR.
                             0014      ; THEN DO A GET STATUS COMMAND FOR EACH OF THE REMAINING DRIVE PLUGS AND
                             0014      ; VERIFY THAT OPERATION INCOMPLETE ERROR IS SET IN THE IDCCSR REGISTER.
                             0014      ;
                             0014      ; TEST STEPS:
                             0014      ;
                             0014      ;          1.  INITIALIZE SUBSYSTEM
                             0014      ;          2.  TELL OPERATOR TO REMOVE ALL DRIVE PLUGS
                             0014      ;          3.  TELL OPERATOR TO INSERT PLUG IN UUT
                             0014      ;          4.  EXECUTE GET STATUS CMD - CHECK FOR ANY ERRORS
                             0014      ;          5.  EXECUTE GET STATUS CMD FOR REMAING ADDRESS PLUGS
                             0014      ;          6.  VERIFY THE OPI ERROR SET IN IDCCSR REGISTER
                             0014      ;          7.  RETURN TO STEP 3 UNTIL ALL PLUGS HAVE BEEN TESTED
                             0014      ;
                             0014      ; DEBUG:
                             0014      ;
                             0014      ;          DRIVE SELECT LOGIC
                             0014      ;
                             0014      ;--
                             0014
                             0014
                             0014                $DS_BGNDATA
       00000000 00000001     0014                .LONG    1,LUN0
       00000001 00000001     001C                .LONG    1,LUN1
       00000002 00000001     0024                .LONG    1,LUN2
       00000003 00000001     002C                .LONG    1,LUN3
                             0034                $DS_ENDDATA
                             0038
                             0038                $DS_BGNTEST <MANUAL,ALL>,ALIGN=LONG
                             003A                GET_LUN -                                      ; GET LUN INFO
                             003A                  UNIT= OFST_UNIT(AP)                          ; LOGICAL UNIT
              04 AC   DD     003A                PUSHL    OFST_UNIT(AP)
   00000000'EF   01   FB     003D                CALLS    #1,GET_LUN$A
                             0044                BDEVFOUND         5$                           ; IF DEV AVAILABLE GOTO 5$
        06 50    00   E2     0044                BBSS     #0,R0,5$
           50    01   D0     0048                         MOVL     #1,R0          ; NOT AN ERROR
               01FD   31     004B                         BRW      TEST_043_X     ; EXIT TEST 43
                             004E      5$:       $DS_BOPER         6$                           ; IF OPERATOR PRESENT GOTO 6$
                             0056                $DS_EXIT          TEST                         ; ELSE EXIT TEST
53     00000000'EF    D0     005C      6$:       MOVL     DRIVE_TBL,R3                          ; GET DRIVE INFO TABLE
      56    34 A3     D0     0063                MOVL     OFST_GTSCMD(R3),R6                    ; PICKUPPGET STATUS CMD BITS
                 54   D4     0067                CLRL     R4                                    ; R4 <-- UNIT PLUG #0
                 0C   D0     0069      10$:      MOVL     #<NOCHK$M_DRDY!NOCHK$M_DERR>,-; SET FLAGS FOR SUBROUTINE
        00000000'EF          006B                         CONFIG
   00000000'EF   00   FB     0070                CALLS    #0,SUBSYS_INIT                        ; INIT SUBSYSTEM
           1D 50      E8     0077                BLBS     R0,20$                                ; BRANCH IF NO ERRORS
                             007A                $DS_ERRHARD_S      ,L$UNIT,MSG14_SYSINIT,-
                             007A                                   PRINT_REG,R1
                             0090      ;;; TEST 43, SUBTEST 0, ERROR 1
```

```
                              0097  2119 20$:      $DS_CKLOOP      10$
                              009F  2120           $DS_PRINTF_S    OPR6_REMOVE                 ; TELL OPR REMOVE ALL UNIT PLUGS
                              00AC  2121           $DS_ASKSTR_S    OPR_RETURN,CHAR_BUFFER,-
                              00AC  2122                           #1,,CHAR_BUFFER
                              00CB  2123 25$:      $DS_PRINTF_S    OPR7_INSERT,R4,#DEV_NAME; TELL OPR INSERT UNIT PLUG # X
                              00E0  2124           $DS_ASKSTR_S    OPR_RETURN,CHAR_BUFFER,-
                              00E0  2125                           #1,,CHAR_BUFFER
         10 A2     56    D0   00FF  2126 27$:      MOVL    R6,IDCMPR(R2)                       ; IDCMPR = GET STATUS BITS
      58    54     08    9C   0103  2127           ROTL    #8,R4,R8                            ; POSITION DRIVE NUMBER
            58     04    88   0107  2128           BISB2   #GETSTAT,R8                         ; OR IN GET STATUS CMD
   00000000'EF     04    9A   010A  2129           MOVZBL  #GETSTAT,CURR_CMD                   ; SAVE COMMAND
   62    000F0080  8F    D0   0111  2130           MOVL    #^XF0080,IDCCSR(R2)                 ; RESET ATTENTIONS
            62     58    D0   0118  2131           MOVL    R8,IDCCSR(R2)                       ; EXECUTE GET STATUS CMD
   00000000'EF     04    D0   011B  2132           MOVL    #NOCHK$M_DRDY,CONFIG                ; SET FLAG FOR SUBROUTINE
   00000000'EF     00    FB   0122  2133           CALLS   #0,STATUS_CHK                       ; CHECK FOR ANY ERRORS
            1D     50    E8   0129  2134           BLBS    R0,30$                              ; BRANCH IF NONE
                              012C  2135           $DS_ERRHARD_S    ,L$UNIT,MSG93_RELV_UP,-
                              012C  2136                           PRINT_REG,R1
                              0142       ;;; TEST 43, SUBTEST 0, ERROR 2
                              0149  2137 30$:      $DS_CKLOOP      27$
      55    54     01    C1   0151  2138           ADDL3   #1,R4,R5                            ; R5 <-- CURRENT UNIT # + 1
            55     04    D1   0155  2139 35$:      CMPL    #4,R5                               ; IS DRIVE # = 4
                   02    12   0158  2140           BNEQ    37$                                 ; BRANCH IF NOT
                   55    D4   015A  2141           CLRL    R5                                  ; MAKE CURRENT DRIVE # = 0
            54     55    D1   015C  2142 37$:      CMPL    R5,R4                               ; IS THIS THE PLUG IN THERE NOW?
                   03    12   015F  2143           BNEQ    39$                                 ; BRANCH IF NOT
                 00B2    31   0161  2144           BRW     60$                                 ; ELSE BRANCH TO 60$
   00000000'EF     04    9A   0164  2145 39$:      MOVZBL  #GETSTAT,CURR_CMD                   ; SAVE CURRENT COMMAND
         10 A2     56    D0   016B  2146           MOVL    R6,IDCMPR(R2)                       ; IDCMPR = GET STATUS BITS
      58    55     08    9C   016F  2147           ROTL    #8,R5,R8                            ; SET UP DRIVE NUMBER
            58     04    88   0173  2148           BISB2   #GETSTAT,R8                         ; OR IN GET STATUS CODE
62    58   000F0000  8F  C9   0176  2149           BISL3   #^XF0000,R8,IDCCSR(R2)              ; EXECUTE THE GET STATUS CMD
   00000000'EF     0E    D0   017E  2150           MOVL    #<NOCHK$M_CERR!NOCHK$M_DRDY!-
                              0185  2151                   NOCHK$M_DERR>,CONFIG                ; SET FLAGS FOR SUBROUTINE
   00000000'EF     00    FB   0185  2152           CALLS   #0,STATUS_CHK                       ; GO WAIT FOR RDY TO SET
            1D     50    E8   018C  2153           BLBS    R0,40$                              ; BRANCH IF IT DID
                              018F  2154           $DS_ERRHARD_S    ,L$UNIT,MSG97_GS_NXD,-
                              018F  2155                           PRINT_REG,R1
                              01A5       ;;; TEST 43, SUBTEST 0, ERROR 3
                              01AC  2156 40$:      $DS_CKLOOP      39$
      59    55     08    9C   01B4  2157           ROTL    #8,R5,R9                            ; POSITION UNIT NUMBER
 00000000'EF     8484  8F    3C   01B8  2158           MOVZWL  #<CSR$M_ERR!CSR$M_OPI!-             ; SET EXPECTED IDCCSR REG
                              01C1  2159                   CSR$M_CRDY!CSR$M_F1>,E.CS           ;
   00000000'EF     59    C8   01C1  2160           BISL2   R9,E.CS                             ; OR IN THE UNIT NUMBER
00000000'EF   00000000'EF D1  01C8  2161           CMPL    E.CS,T.CS                           ; EXPECTED = ACTUAL
                   33    13   01D3  2162           BEQL    50$                                 ; BRANCH IF OK
                              01D5  2163           $DS_ERRHARD_S    ,L$UNIT,MSG97_GS_NXD,-
                              01D5  2164                           REG_ERRPT,#MSG23_CS_INC,-
                              01D5  2165                           E.CS,T.CS,#CSR_BITS
                              0201       ;;; TEST 43, SUBTEST 0, ERROR 4
                              0208  2166 50$:      $DS_CKLOOP      39$
                   55    D6   0211  2167           INCL    R5                                  ; BUMP R5 TO NEXT UNIT
                 FF3F    31   0213  2168           BRW     35$                                 ; GET NEXT DRIVE NUMBER TO TEST
     FEAF 54       01    03  3D   0216  2169 60$:      ACBW    #3,#1,R4,25$                        ; GET TEST NEXT UNIT PLUG
                              021C  2170           $DS_PRINTF_S    OPR8_SETUP                  ; TELL OPR PUT BACK ORIGIONAL PLUG
                              0229  2171           $DS_ASKSTR_S    OPR_RETURN,CHAR_BUFFER,-
```

```
         0229  2172                          #1,,CHAR_BUFFER
         0248  2173        $DS_ENDTEST
         0253  2174        $DS_SBTTL <WRITE PROTECTION TEST>,ALIGN=LONG
         0253              .SBTTL  TEST 44:  WRITE PROTECTION TEST
         0253              $D2_SBTTL         044,<WRITE PROTECTION TEST>,<LONG>
     00000000                      .PSECT  TEST_044, LONG, NOWRT
         0018
```

```
                           0018
                           0018  ;++
                           0018  ;
                           0018  ; TEST DESCRIPTION:
                           0018  ;
                           0018  ; THIS TEST WILL WRITE A PATTERN IN SECTOR 0 THEN ASK THE OPERATOR
                           0018  ; TO SET THE WRITE LOCK SWITCH ON THE UUT.  EXECUTE ANOTHER WRITE
                           0018  ; DATA CMD TO THE SAME SECTOR AND VERIFY THAT WRITE GATE ERROR AND
                           0018  ; DRIVE ERROR ARE SET.  EXECUTE A DRIVE RESET AND VERIFY THAT ERRORS
                           0018  ; ARE RESET THEN EXECUTE A READ DATA CMD AND VERIFY THAT THE DATA
                           0018  ; WAS NOT DISTURBED WHILE THE WRITE LOCK SWITCH WAS SET.
                           0018  ;
                           0018  ; TEST STEPS:
                           0018  ;
                           0018  ; A.  SUBTEST 1
                           0018  ;
                           0018  ;           1.  INITIALIZE SUBSYSTEM
                           0018  ;           2.  POSITION DRIVE TO TEST CYLINDER
                           0018  ;           3.  IF ERROR THEN ESCAPE TEST
                           0018  ;
                           0018  ; B.  SUBTEST 2
                           0018  ;
                           0018  ;           1.  INITIALIZE SUBSYSTEM
                           0018  ;           2.  EXECUTE WRITE DATA COMMAND
                           0018  ;           3.  CHECK FOR ANY ERRORS
                           0018  ;
                           0018  ; C.  SUBTEST 3
                           0018  ;
                           0018  ;           1.  INITIALIZE SUBSYSTEM
                           0018  ;           2.  TELL OPERATOR TO SET WRITE LOCK SWITCH ON UUT
                           0018  ;           3.  EXECUTE WRITE DATA CMD
                           0018  ;           4.  VERIFY THAT WRITE GATE AND DRIVE ERROR ARE SET
                           0018  ;           5.  EXECUTE DRIVE RESET FUNCTION
                           0018  ;           6.  VERIFY THAT WRITE GATE ERROR IS RESET
                           0018  ;           7.  INIT SUBSYSTEM
                           0018  ;           8.  EXEUCTE READ DATA CMD - CHECK FOR ERRORS
                           0018  ;           9.  VERIFY THAT DATA WAS NOT DISTURBED
                           0018  ;           10. TELL OPERATOR TO RESET WRITE LOCK SWITCH
                           0018  ;
                           0018  ; DEBUG:
                           0018  ;
                           0018  ;           SWITCH PANEL
                           0018  ;
                           0018  ;--
                           0018
                           0018
                           0018             $DS_BGNDATA
00000000 00000001          0018             .LONG   1,LUN0
00000001 00000001          0020             .LONG   1,LUN1
00000002 00000001          0028             .LONG   1,LUN2
00000003 00000001          0030             .LONG   1,LUN3
                           0038             $DS_ENDDATA
                           003C
                           003C             $DS_BGNTEST <MANUAL,ALL>,ALIGN=LONG
                           003E             GET_LUN -                            ; GET LUN INFO
                           003E               UNIT= OFST_UNIT(AP)                ; LOGICAL UNIT
```

```
              04 AC   DD  003E               PUSHL   OFST_UNIT(AP)
      00000000'EF  01 FB  0041               CALLS   #1,GET_LUN$A
                          0048  2232         BDEVFOUND       5$             ; IF DEV AVAILABLE GOTO 5$
           06 50   00 E2  0048               BBSS    #0,R0,5$
              50   01 D0  004C               MOVL    #1,R0           ; NOT AN ERROR
                 04CE 31  004F               BRW     TEST_044_X      ; EXIT TEST 44
                          0052  2233 5$:     $DS_BOPER       6$             ; IF OPERATOR PRESENT GOTO 6$
                          005A  2234         $DS_EXIT        TEST           ; ELSE EXIT TEST
    54   00000000'EF  D0  0060  2235 6$:     MOVL    DRIVE_TBL,R4           ; GET DRIVE INFO TABLE
           55   14 A4 D0  0067  2236         MOVL    OFST_TSTCYL(R4),R5     ; GET TEST CYLINDER TO USE
           56   28 A4 D0  006B  2237         MOVL    OFST_SECLEN(R4),R6     ; GET BYTES PER SECTOR
                   57 D4  006F  2238         CLRL    R7                     ; INIT REGISTER
    57   10   10   55 F0  0071  2239         INSV    R5,#16,#16,R7          ; LOAD DESTINATION ADDRESS
           58   68 A4 D0  0076  2240         MOVL    OFST_WTPSTS(R4),R8     ; GET EXPECTED WRT PROT STATUS
           59   64 A4 D0  007A  2241         MOVL    OFST_STSDNC(R4),R9     ; GET STATUS DONT CARE BITS
           5A   40 A4 D0  007E  2242         MOVL    OFST_MPBITS(R4),R10    ; GET MPR BIT MNEMONICS
          00000000'EF D4  0082  2243         CLRL    CONFIG                 ; CLEAR ANY PROGRAM FLAGS
                          0088  2244         $DS_PRINTF_S    OPR11_ORIGIN,#DEV_NAME  ; TELL OPR TO SET UP DRIVE
                          009B  2245         $DS_ASKSTR_S    OPR_RETURN,CHAR_BUFFER,-
                          009B  2246                         #1,,CHAR_BUFFER
                          00BA  2247         $DS_BGNSUB
                          00C5  2248         STARS
                          00C5       ;;*******************************************************************
                          00C5  2249 ;++
                          00C5  2250 ; POSITION DRIVE TO TEST CYLINDER AND VERIFY THE SEEK
                          00C5  2251 ;--
                          00C5  2252
      00000000'EF  00 FB  00C5  2253 10$:    CALLS   #0,SUBSYS_INIT         ; INITIALIZE SUBSYSTEM
              1D 50   E8  00CC  2254         BLBS    R0,20$                 ; BRANCH IF NO ERRORS
                          00CF  2255         $DS_ERRHARD_S   ,L$UNIT,MSG14_SYSINIT,-
                          00CF  2256                         PRINT_REG,R1
                          00E5       ;;; TEST 44, SUBTEST 1, ERROR 1
                          00EC  2257 20$:    $DS_CKLOOP      10$
                   00 DD  00F4  2258 35$:    PUSHL   #0                     ; PUSH HEAD TO USE
                   55 DD  00F6  2259         PUSHL   R5                     ; PUSH DESTINATION CYLN
      00000000'EF  02 FB  00F8  2260         CALLS   #2,POSITION_HDS        ; EXECUTE THE SEEK
              1D 50   E8  00FF  2261         BLBS    R0,40$                 ; BRANCH IF NO ERRORS
                          0102  2262         $DS_ERRHARD_S   ,L$UNIT,@ERR_MSG,-
                          0102  2263                         PRINT_REG,R1
                          0118       ;;; TEST 44, SUBTEST 1, ERROR 2
                          011F  2264 40$:    $DS_CKLOOP      35$
                          0127  2265         $DS_ESCAPE      TEST           ; IF ERROR EXIT TEST
          00000000'EF D4  0132  2266         CLRL    CONFIG                 ; CLEAR ANY PROGRAM FLAGS
                          0138  2267 50$:    $DS_ENDSUB
                          0143  2268         $DS_BGNSUB
                          014E  2269         STARS
                          014E       ;;*******************************************************************
                          014E  2270 ;++
                          014E  2271 ; DO A WRITE DATA COMMAND TO SECTOR 0 OF 128 WORDS
                          014E  2272 ;--
                          014E  2273
                   53 D4  014E  2274         CLRL    R3                     ; INIT INDEX
 AAAAAAAA AAAAAAAA 8F 7D  0150  2275 5$:     MOVQ    #^XAAAAAAAAAAAAAAAA,-  ; LOAD DATA INTO BUFFER
       00000000'FF43      015A  2276                 @WRITE_BUFFER[R3]
       00000000'FF43  7C  0160  2277         CLRQ    @READ_BUFFER[R3]       ; CLEAR READ BUFFER
  E1 53   00000040 8F F2  0167  2278         AOBLSS  #64,R3,5$              ; DO 256 WORDS
```

```
        00000000'EF    00    FB  016F   2279 10$:    CALLS   #0,SUBSYS_INIT                          ; INITIALIZE SUBSYSTEM
               1D 50         E8  0176   2280         BLBS    R0,20$                                  ; BRANCH IF NO ERRORS
                                 0179   2281         $DS_ERRHARD_S    ,L$UNIT,MSG14_SYSINIT,-
                                 0179   2282                          PRINT_REG,R1
                                 018F        ;;; TEST 44, SUBTEST 2, ERROR 1
                                 0196   2283 20$:    $DS_CKLOOP       10$
                       56    DD  019E   2284         PUSHL   R6                                      ; GET BYTE COUNT
             00000000'EF     7F  01A0   2285         PUSHAQ  WRITE_BUFFER                            ; GET BUFFER ADDRESS
        00000000'EF    02    FB  01A6   2286         CALLS   #2,$GS_SETMAP                           ; MAP UBA
                       00    DD  01AD   2287         PUSHL   #0                                      ; IDCMPR DATA
                       57    DD  01AF   2288         PUSHL   R7                                      ; IDCDAR (SECTOR 0)
                  7E   56    CE  01B1   2289         MNEGL   R6,-(SP)                                ; IDCCNT (BYTE COUNT)
                       00    DD  01B4   2290         PUSHL   #0                                      ; IDCBAR DATA
                       0A    DD  01B6   2291         PUSHL   #WRITE                                  ; DEFINE COMMAND
        00000000'EF    05    FB  01B8   2292         CALLS   #5,START_CMD                            ; EXECUTE WRITE DATA CMD
        00000000'EF    00    FB  01BF   2293         CALLS   #0,STATUS_CHK                           ; CHECK FOR ANY ERRORS
               1D 50         E8  01C6   2294         BLBS    R0,30$                                  ; BRANCH IF NONE
                                 01C9   2295         $DS_ERRHARD_S    ,L$UNIT,MSG20_WRITE,-
                                 01C9   2296                          PRINT_REG,R1
                                 01DF        ;;; TEST 44, SUBTEST 2, ERROR 2
                                 01E6   2297 30$:    $DS_CKLOOP       10$
                                 01EE   2298         $DS_ENDSUB
                                 01F9   2299         $DS_BGNSUB
                                 0204   2300         STARS
                                 0204        ;;*****************************************************************
                                 0204   2301 ;++
                                 0204   2302 ; TELL OPERATOR TO SET THE WRITE LOCK SWITCH ON THE CURRENT UUT.
                                 0204   2303 ; WHEN THAT HAS BEEN ACCOMPLISHED EXECUTE ANOTHER WRITE DATA COMMAND
                                 0204   2304 ; TO SECTOR 0 AND CHECK THAT WRITE GATE ERROR AND DRIVE ERROR ARE SET.
                                 0204   2305 ; THEN CLEAR THESE BITS WITH A DRIVE RESET.
                                 0204   2306 ;--
                                 0204   2307
                                 0204   2308         $DS_PRINTF_S     OPR12_WRTLOCK,#DEV_NAME ; TELL OPR TO SET WRITE LOCK
                                 0217   2309         $DS_ASKSTR_S     OPR_RETURN,CHAR_BUFFER,-
                                 0217   2310                          #1,,CHAR_BUFFER
                       53    D4  0236   2311 1$:     CLRL    R3                                      ; CLEAR INDEX
   55555555 55555555 8F      7D  0238   2312 5$:     MOVQ    #^X5555555555555555,-                   ; CHANGE DATA IN WRITE BUFFER
           00000000'FF43         0242   2313                 @WRITE_BUFFER[R3]
      E8 53   00000040 8F    F2  0248   2314         AOBLSS  #64,R3,5$                               ; DO 256 WORDS
        00000000'EF    00    FB  0250   2315 10$:    CALLS   #0,SUBSYS_INIT                          ; INIT SUBSYSTEM
               1D 50         E8  0257   2316         BLBS    R0,20$                                  ; BRANCH IF NO ERRORS
                                 025A   2317         $DS_ERRHARD_S    ,L$UNIT,MSG14_SYSINIT,-
                                 025A   2318                          PRINT_REG,R1
                                 0270        ;;; TEST 44, SUBTEST 3, ERROR 1
                                 0277   2319 20$:    $DS_CKLOOP       10$
                       56    DD  027F   2320         PUSHL   R6                                      ; GET BYTE COUNT
             00000000'EF     7F  0281   2321         PUSHAQ  WRITE_BUFFER                            ; GET ADRS OF WRITE BUFFER
        00000000'EF    02    FB  0287   2322         CALLS   #2,$GS_SETMAP                           ; MAP UBA
                       00    DD  028E   2323         PUSHL   #0                                      ; IDCMPR = WORD COUNT 128 WORDS
                       57    DD  0290   2324         PUSHL   R7                                      ; IDCDAR = SECTOR 0 HEAD 0
                  7E   56    CE  0292   2325         MNEGL   R6,-(SP)                                ; IDCCNT (BYTE COUNT)
                       00    DD  0295   2326         PUSHL   #0                                      ; IDCBAR DATA
                       0A    DD  0297   2327         PUSHL   #WRITE                                  ; DEFINE COMMAND
        00000000'EF    05    FB  0299   2328         CALLS   #5,START_CMD                            ; EXECUTE WRITE DATA CMD
        00000000'EF    08    D0  02A0   2329         MOVL    #NOCHK$M_DERR,CONFIG                    ; SET FLAGS FOR SUBROUTINE
        00000000'EF    00    FB  02A7   2330         CALLS   #0,STATUS_CHK                           ; CHECK FOR ANY ERRORS
```

```
                   1D 50    E8  02AE  2331         BLBS    R0,30$                          ; BRANCH IF NONE
                                02B1  2332         $DS_ERRHARD_S   ,L$UNIT,MSG98_WRT_WL,-
                                02B1  2333                         PRINT_REG,R1
                                02C7       ;;; TEST 44, SUBTEST 3, ERROR 2
                                02CE  2334 30$:    $DS_CKLOOP      10$
         00000000'EF   58   D0  02D7  2335         MOVL    R8,E.MP                         ; SET EXPECTED DRIVE STATUS
         00000000'EF   59   CA  02DE  2336         BICL2   R9,T.MP                         ; CLEAR DONT CARE BITS
  00000000'EF   00000000'EF D1  02E5  2337         CMPL    E.MP,T.MP                       ; EXPECTED = ACTUAL
                       2F   13  02F0  2338         BEQL    40$                             ; BRANCH IF YES
                                02F2  2339         $DS_ERRHARD_S   ,L$UNIT,MSG98_WRT_WL,-
                                02F2  2340                         REG_ERRPT,#MSG47_DS_INC,-
                                02F2  2341                         E.MP,T.MP,R10
                                031A       ;;; TEST 44, SUBTEST 3, ERROR 3
                                0321  2342 40$:    $DS_CKLOOP      10$
      00000000'EF   C08B 8F 3C  032A  2343         MOVZWL  #<CSR$M_ERR!CSR$M_DE!-          ; SET EXPECTED IDCCSR REG
                                0333  2344                 CSR$M_CRDY!CSR$M_F2!-           ;
                                0333  2345                 CSR$M_F0!CSR$M_DRDY>,E.CS       ;
  00000000'EF   00000000'EF C8  0333  2346         BISL2   P$UNIT,E.CS                     ; OR IN DRIVE NUMBER
        00000000'EF   58 A4 C8  033E  2347         BISL2   OFST_CSRTYP(R4),E.CS            ; OR IN DRIVE TYPE BIT
  00000000'EF   00000000'EF D1  0346  2348         CMPL    E.CS,T.CS                       ; ACTUAL = EXPECTED
                       33   13  0351  2349         BEQL    50$                             ; BRANCH IF YES
                                0353  2350         $DS_ERRHARD_S   ,L$UNIT,MSG98_WRT_WL,-
                                0353  2351                         REG_ERRPT,#MSG23_CS_INC,-
                                0353  2352                         E.CS,T.CS,#CSR_BITS
                                037F       ;;; TEST 44, SUBTEST 3, ERROR 4
                                0386  2353 50$:    $DS_CKLOOP      10$
               01      64   91  038F  2354         CMPB    OFST_DRVTYP(R4),#RL02$T         ; IF DRIVE IS AN RL02
                       32   13  0392  2355         BEQL    55$                             ; THEN GOTO 55$
                                0394  2356         $DS_PRINTF_S    OPR13A_RST_FLT,#DEV_NAME ; TELL OPR TO RESET FAULT
                                03A7  2357         $DS_ASKSTR_S    OPR_RETURN,CHAR_BUFFER,-
                                03A7  2358                         #1,,CHAR_BUFFER
                    50 A4   DD  03C6  2359 55$:    PUSHL   OFST_DRVRST(R4)                 ; IDCMPR DATA=DRIVE RESET
                       00   DD  03C9  2360         PUSHL   #0                              ; IDCDAR DATA
                       00   DD  03CB  2361         PUSHL   #0                              ; IDCCNT DATA
                       00   DD  03CD  2362         PUSHL   #0                              ; IDCBAR DATA
                       04   DD  03CF  2363         PUSHL   #GETSTAT                        ; DEFINE COMMNAD
         00000000'EF   05   FB  03D1  2364         CALLS   #5,START_CMD                    ; EXEUCTE DRIVE RESET
         00000000'EF   00   FB  03D8  2365         CALLS   #0,STATUS_CHK                   ; CHECK FOR ANY ERRORS
                    1D 50   E8  03DF  2366         BLBS    R0,60$                          ; BRANCH IF NONE
                                03E2  2367         $DS_ERRHARD_S   ,L$UNIT,MSG99_RST_WGE,-
                                03E2  2368                         PRINT_REG,R1
                                03F8       ;;; TEST 44, SUBTEST 3, ERROR 5
                                03FF  2369 60$:    $DS_CKLOOP      55$
                                0407  2370         STARS
                                0407       ;;*******************************************************************
                                0407  2371 ;++
                                0407  2372 ; DO A READ DATA COMMAND AND CHECK THAT DATA WAS NOT DISTURBED
                                0407  2373 ; WHILE THE WRITE LOCK SWITCH IS SET
                                0407  2374 ;--
                                0407  2375
                       53   D4  0407  2376         CLRL    R3                              ; SET INDEX INTO BUFFER
     AAAAAAAA AAAAAAAA 8F   7D  0409  2377 65$:    MOVQ    #^XAAAAAAAAAAAAAAAA,-           ; CHANGE WRT BUF BACK TO -
           00000000'FF43        0413  2378                 @WRITE_BUFFER[R3]               ; WHAT EXPECTED DATA IS
      E8 53   00000040 8F   F2  0419  2379         AOBLSS  #64,R3,65$                      ; LOAD 256 WORDS
         00000000'EF   00   FB  0421  2380 70$:    CALLS   #0,SUBSYS_INIT                  ; INIT SUBSYSTEM
                    1D 50   E8  0428  2381         BLBS    R0,80$                          ; BRANCH IF NO ERRORS
```

```
                          042B  2382            $DS_ERRHARD_S    ,L$UNIT,MSG14_SYSINIT,-
                          042B  2383                             PRINT_REG,R1
                          0441        ;;; TEST 44, SUBTEST 3, ERROR 6
                          0448  2384 80$:       $DS_CKLOOP       70$
              56     DD   0450  2385            PUSHL    R6                              ; GET BYTE COUNT
       00000000'EF   7F   0452  2386            PUSHAQ   READ_BUFFER                     ; GET BUFFER ADDRS
00000000'EF   02     FB   0458  2387            CALLS    #2,$GS_SETMAP                   ; MAP DATA PATH #1
              00     DD   045F  2388            PUSHL    #0                              ; IDCMPR DATA
              57     DD   0461  2389            PUSHL    R7                              ; IDCDAR = SECTOR 0 HEAD 0
          7E  56     CE   0463  2390            MNEGL    R6,-(SP)                        ; IDCCNT (BYTE COUNT)
              00     DD   0466  2391            PUSHL    #0                              ; IDCBAR DATA
              0C     DD   0468  2392            PUSHL    #READ                           ; DEFINE COMMAND
00000000'EF   05     FB   046A  2393            CALLS    #5,START_CMD                    ; EXECUTE READ DATA CMD
00000000'EF   00     FB   0471  2394            CALLS    #0,STATUS_CHK                   ; CHECK FOR ANY ERRORS
          1D  50     E8   0478  2395            BLBS     R0,90$                          ; BRANCH IF NONE
                          047B  2396            $DS_ERRHARD_S    ,L$UNIT,MSG21_READ,-
                          047B  2397                             PRINT_REG,R1
                          0491        ;;; TEST 44, SUBTEST 3, ERROR 7
                          0498  2398 90$:       $DS_CKLOOP       70$
       00000000'EF   DD   04A0  2399            PUSHL    WRITE_BUFFER                    ; GET GOOD DATA
       00000000'EF   DD   04A6  2400            PUSHL    READ_BUFFER                     ; GET UNKNOWN DATA
       00000000'EF   DD   04AC  2401            PUSHL    WORD_COUNT                      ; GET # OF WORDS TO COMPARE
00000000'EF   03     FB   04B2  2402            CALLS    #3,CMPDATA                      ; GO CHECK THE DATA
          1B  50     E8   04B9  2403            BLBS     R0,100$                         ; BRANCH IF NO ERRORS
                          04BC  2404            $DS_ERRHARD_S    ,L$UNIT,MS100_DAT_DST,-
                          04BC  2405                             PRINT_DATA_ERR
                          04D0        ;;; TEST 44, SUBTEST 3, ERROR 8
                          04D7  2406 100$:      $DS_CKLOOP       1$
                          04E0  2407            $DS_ENDSUB
                          04EB  2408            $DS_PRINTF_S     OPR13_RST_WL,#DEV_NAME  ; TELL OPR TO RESET WRT LOCK
                          04FE  2409            $DS_ASKSTR_S     OPR_RETURN,CHAR_BUFFER,-
                          04FE  2410                             #1,,CHAR_BUFFER
                          051D  2411            $DS_ENDTEST
                          0528  2412            $DS_ENDMOD
                          0004  2413            .END
```

```
$$$                              = 000000D8 R       05      BIT5                             = 00000020
$$E                              = 00000001                 BIT6                             = 00000040
$$M                              = 00000004                 BIT7                             = 00000080
$$N                              = 00000002                 BIT8                             = 00000100
$$S                              = FFFFFFFF                 BIT9                             = 00000200
$$T1                             = 00000001                 BUILD_HDRS                         ********   X   08
$ENV                             = 00000003  G              CEP_FUNCTIONAL                   = 00000000
$ER                              = FFFFFFFF                 CEP_REPAIR                       = 00000001
$GS_SETMAP                         ********   X     03      CHANGE_DAR$A                       ********   X   03
$MO                              = 00000001  G              CHAR_BUFFER                        ********   X   0D
$S$                              = 000000D8 R       05      CHC$_ABORT                       = 00000002
$ST                              = 00000000                 CHC$_CLEAR                       = 00000006
$TN                              = 0000002C                 CHC$_CLRDFT                      = 00000009
ALL                              = 00000002  G              CHC$_DSINT                       = 00000005
BASE_032                           00000000 R       03      CHC$_ENINT                       = 00000004
BASE_033                           00000000 R       06      CHC$_INITA                       = 00000000
BASE_034                           00000000 R       07      CHC$_INITB                       = 00000001
BASE_035                           00000000 R       08      CHC$_PURGE                       = 00000003
BASE_036                           00000000 R       09      CHC$_SELF_TEST                   = 0000000B
BASE_037                           00000000 R       0A      CHC$_SETDFT                      = 00000008
BASE_038                           00000000 R       0B      CHC$_STATUS                      = 00000007
BASE_039                           00000000 R       0C      CHC$_STOP                        = 0000000A
BASE_040                           00000000 R       0D      CHECK_BADBLK                       ********   X   0C
BASE_041                           00000000 R       0E      CHECK_ECC                          ********   X   0C
BASE_042                           00000000 R       0F      CHI$M_CHNINT                     = 00000001
BASE_043                           00000000 R       10      CHI$M_DEVINT                     = 00000002
BASE_044                           00000000 R       11      CHI$M_IPL                        = 0000007C
BASE_ADDR                          ********   X     0C      CHI$S_IPL                        = 00000005
BIT...                           = 00000028                 CHI$V_CHNINT                     = 00000000
BIT0                             = 00000001                 CHI$V_DEVINT                     = 00000001
BIT1                             = 00000002                 CHI$V_IPL                        = 00000002
BIT10                            = 00000400                 CHM$_FORWARD                     = 00000000
BIT11                            = 00000800                 CHM$_INVALIDATE                  = 00000001
BIT12                            = 00001000                 CHM$_MAP                         = 00000002
BIT13                            = 00002000                 CHM$_MFWDN                       = 00000002
BIT14                            = 00004000                 CHM$_MFWDNO                      = 0000000A
BIT15                            = 00008000                 CHM$_MFWDV                       = 00000003
BIT16                            = 00010000                 CHM$_MFWDVO                      = 0000000B
BIT17                            = 00020000                 CHM$_MREVN                       = 00000006
BIT18                            = 00040000                 CHM$_MREVNO                      = 0000000E
BIT19                            = 00080000                 CHM$_MREVV                       = 00000007
BIT2                             = 00000004                 CHM$_MREVVO                      = 0000000F
BIT20                            = 00100000                 CHM$_NFWDN                       = 00000000
BIT21                            = 00200000                 CHM$_NREVN                       = 00000004
BIT22                            = 00400000                 CHM$_OFFSET                      = 00000008
BIT23                            = 00800000                 CHM$_REVERSE                     = 00000004
BIT24                            = 01000000                 CHS$M_BADBDP                     = 20000000
BIT25                            = 02000000                 CHS$M_BADPAR                     = 40000000
BIT26                            = 04000000                 CHS$M_BUSIC                      = 00800000
BIT27                            = 08000000                 CHS$M_BUSINIT                    = 00400000
BIT28                            = 10000000                 CHS$M_BUSNXM                     = 08000000
BIT29                            = 20000000                 CHS$M_BUSPDN                     = 01000000
BIT3                             = 00000008                 CHS$M_CHNDPE                     = 00000040
BIT30                            = 40000000                 CHS$M_CHNERR                     = 00000002
BIT31                            = 80000000                 CHS$M_CHNMPE                     = 00000080
BIT4                             = 00000010                 CHS$M_CHPFOT                     = 00000100
```

```
CHS$M_DEVBUS              = 00000010              CSR$M_OPI                 = 00000400
CHS$M_DEVERR              = 00000004              CSR$M_SSE                 = 00800000
CHS$M_DEVTO               = 00000020              CSR$M_SSEI                = 00400000
CHS$M_ERRANY              = 0000000F              CSR$V_DE                  = 0000000E
CHS$M_MBAATN              = 00100000              CSR$V_DRDY                = 00000000
CHS$M_MBACPE              = 00200000              CSR_BITS                    ********   X   10
CHS$M_MBADTB              = 00040000              CURR_CMD                    ********   X   03
CHS$M_MBADTC              = 00080000              CURR_CYLN                   ********   X   0B
CHS$M_MBAEXC              = 00010000              CURR_FLAGS                  ********   X   0C
CHS$M_MBANED              = 00020000              DAR_MNEM                    ********   X   03
CHS$M_MBAWCKLWR           = 02000000              DATA_032                    00000028 R     03
CHS$M_MBAWCKUPR           = 04000000              DATA_033                    00000014 R     06
CHS$M_PGMERR              = 00000008              DATA_034                    00000014 R     07
CHS$M_PGMHDE              = 00000800              DATA_035                    00000024 R     08
CHS$M_SYSERR              = 00000001              DATA_036                    00000020 R     09
CHS$M_SYSMEM              = 00000200              DATA_037                    00000024 R     0A
CHS$M_SYSSBI              = 00000400              DATA_038                    00000020 R     0B
CHS$M_UIE                 = 10000000              DATA_039                    0000001C R     0C
CHS$V_BADBDP              = 0000001D              DATA_040                    00000018 R     0D
CHS$V_BADPAR              = 0000001E              DATA_041                    0000001C R     0E
CHS$V_BUSIC               = 00000017              DATA_042                    00000020 R     0F
CHS$V_BUSINIT             = 00000016              DATA_043                    00000014 R     10
CHS$V_BUSNXM              = 0000001B              DATA_044                    00000018 R     11
CHS$V_BUSPDN              = 00000018              DEFAULT                   = 00000000   G
CHS$V_CHNDPE              = 00000006              DEL$V_PROT                = 00000014
CHS$V_CHNERR              = 00000001              DEL$V_READ                = 0000000D
CHS$V_CHNMPE              = 00000007              DEL$V_READH               = 0000000A
CHS$V_CHPFOT              = 00000008              DEL$V_WRCK                = 0000000C
CHS$V_DEVBUS              = 00000004              DEL$V_WRIT                = 0000000B
CHS$V_DEVERR              = 00000002              DELTA_TIME                  ********   X   0E
CHS$V_DEVTO               = 00000005              DEV_NAME                    ********   X   0C
CHS$V_MBAATN              = 00000014              DRIVE_TBL                   ********   X   03
CHS$V_MBACPE              = 00000015              DS$ASKSTR                   ********   X   0D
CHS$V_MBADTB              = 00000012              DS$BGNSUB                   ********   X   03
CHS$V_MBADTC              = 00000013              DS$BRANCH                   ********   X   08
CHS$V_MBAEXC              = 00000010              DS$BREAK                    ********   X   03
CHS$V_MBANED              = 00000011              DS$CKLOOP                   ********   X   03
CHS$V_MBAWCKLWR           = 00000019              DS$ENDSUB                   ********   X   03
CHS$V_MBAWCKUPR           = 0000001A              DS$ERRHARD                  ********   X   03
CHS$V_PGMERR              = 00000003              DS$ESCAPE                   ********   X   03
CHS$V_PGMHDE              = 0000000B              DS$K_BBC                  = 0000001D   G
CHS$V_SYSERR              = 00000000              DS$K_BBCC                 = 00000021   G
CHS$V_SYSMEM              = 00000009              DS$K_BBCCI                = 00000023   G
CHS$V_SYSSBI              = 0000000A              DS$K_BBCS                 = 0000001F   G
CHS$V_UIE                 = 0000001C              DS$K_BBS                  = 0000001C   G
CMPDATA                     ********   X   03     DS$K_BBSC                 = 00000020   G
CONFIG                      ********   X   03     DS$K_BBSS                 = 0000001E   G
CSR$M_ASSI                = 08000000              DS$K_BBSSI                = 00000022   G
CSR$M_CRDY                = 00000080              DS$K_BCC_M                = 0000001B   G
CSR$M_DE                  = 00004000              DS$K_BCS_M                = 0000001A   G
CSR$M_DRDY                = 00000001              DS$K_BEQLU_B              = 00000005   G
CSR$M_ERR                 = 00008000              DS$K_BEQLU_M              = 00000011   G
CSR$M_F0                  = 00000002              DS$K_BEQL_B               = 00000004   G
CSR$M_F1                  = 00000004              DS$K_BEQL_M               = 00000010   G
CSR$M_F2                  = 00000008              DS$K_BERROR               = 00000026   G
CSR$M_FMT                 = 20000000              DS$K_BGEQU_B              = 00000007   G
```

```
DS$K_BGEQU_M                    = 00000013  G
DS$K_BGEQ_B                     = 00000006  G
DS$K_BGEQ_M                     = 00000012  G
DS$K_BGTRU_B                    = 00000009  G
DS$K_BGTRU_M                    = 00000015  G
DS$K_BGTR_B                     = 00000008  G
DS$K_BGTR_M                     = 00000014  G
DS$K_BLBC                       = 00000025  G
DS$K_BLBS                       = 00000024  G
DS$K_BLEQU_B                    = 00000003  G
DS$K_BLEQU_M                    = 0000000F  G
DS$K_BLEQ_B                     = 00000002  G
DS$K_BLEQ_M                     = 0000000E  G
DS$K_BLSSU_B                    = 00000001  G
DS$K_BLSSU_M                    = 0000000D  G
DS$K_BLSS_B                     = 00000000  G
DS$K_BLSS_M                     = 0000000C  G
DS$K_BNEQU_B                    = 0000000B  G
DS$K_BNEQU_M                    = 00000017  G
DS$K_BNEQ_B                     = 0000000A  G
DS$K_BNEQ_M                     = 00000016  G
DS$K_BNERROR                    = 00000027  G
DS$K_BVC_M                      = 00000019  G
DS$K_BVS_M                      = 00000018  G
DS$PRINTF                         ********  X    0C
DS$WAITMS                         ********  X    06
DS$WAITUS                         ********  X    0E
DSA$AL_APTMAIL                    0000FE00
DSA$AT_APTTXT                     0000FA00
DSA$GL_APTCOM                     0000FE04
DSA$GL_DEVLEN                     0000FE58
DSA$GL_ERRNO                      0000FE44
DSA$GL_EVENT                      0000FE48
DSA$GL_FLAGS                      0000FE00
DSA$GL_MSGTYP                     0000FE40
DSA$GL_PASSES                     0000FE08
DSA$GL_PASSNO                     0000FE54
DSA$GL_SECTNO                     0000FE10
DSA$GL_SID                        0000FE14
DSA$GL_SUBTNO                     0000FE4C
DSA$GL_TESTNO                     0000FE50
DSA$GL_UNITS                      0000FE0C
DSA$GQ_MSGPTR                     0000FE68
DSA$GT_DEVNAM                     0000FE5C
DSA$V_OPER                      = 0000000C
E.CS                              ********  X    10
E.MP                              ********  X    0E
ENV$M_DOMAIN                    = 00000002
ENV$M_LEVEL                     = 00000001
ENV$M_SUPER                     = 000003FC
ENV$S_DOMAIN                    = 00000001
ENV$S_LEVEL                     = 00000001
ENV$S_SUPER                     = 00000008
ENV$V_DOMAIN                    = 00000001
ENV$V_LEVEL                     = 00000000
ENV$V_SUPER                     = 00000002
ENV$_CPU                        = 00000000
ENV$_FUNCTIONAL                 = 00000000
ENV$_REPAIR                     = 00000001
ENV$_SUPER                      = 00000001
ENV$_SYSTEM                     = 00000001
ERR_MSG                           ********  X    03
EVENT_FLAGS                       ********  X    0C
FMT_BADBLK                        ********  X    0C
GETSTAT                         = 00000004
GET_LUN$A                         ********  X    03
GET_POSITION                      ********  X    0B
GET_RANDOM                        ********  X    0C
GET_SECLUN$A                      ********  X    06
IDCBAR                          = 00000004
IDCCNT                          = 00000008
IDCCSR                          = 00000000
IDCDAR                          = 0000000C
IDCMPR                          = 00000010
L$UNIT                            ********  X    03
LUN0                            = 00000000
LUN1                            = 00000001
LUN2                            = 00000002
LUN3                            = 00000003
MANUAL                          = 00000001  G
MPR_02$M_BH                     = 00000008
MPR_02$M_CO                     = 00000020
MPR_02$M_DT                     = 00000080
MPR_02$M_HO                     = 00000010
MPR_02$M_STA                    = 00000001
MPR_02$M_STB                    = 00000002
MPR_02$M_STC                    = 00000004
MPR_02$M_VC                     = 00000200
MPR_02$M_WL                     = 00002000
MPR_02$V_BH                     = 00000003
MPR_02$V_VC                     = 00000009
MS100_DAT_DST                     ********  X    11
MS113_UWRTERR                     ********  X    09
MS114A_UWCKERR                    ********  X    09
MS114_URDERR                      ********  X    09
MS115A_WKRST                      ********  X    09
MS115_RDRST                       ********  X    09
MS116A_AUTOCMP                    ********  X    0A
MS116_SKPCMP                      ********  X    09
MS117_AUTORD                      ********  X    0A
MS117_AUTOWCK                     ********  X    0A
MS117_AUTOWRT                     ********  X    0A
MS118_SEEKIDC                     ********  X    0B
MS119_IDCPOS                      ********  X    0B
MS83A_ST4_DE                      ********  X    0E
MSG14_SYSINIT                     ********  X    03
MSG16_WRTCHK                      ********  X    0C
MSG17_GETSTAT                     ********  X    0D
MSG18B_SEEK                       ********  X    06
MSG19_READH                       ********  X    0B
MSG20A_FORMAT                     ********  X    08
MSG20C_WRITE                      ********  X    08
```

```
MSG20_WRITE                   ********   X   03    OFST_CSRTYP                   = 00000058
MSG20_WRT_BND                 ********   X   07    OFST_DEVMEM                   = 0000001C
MSG21_READ                    ********   X   07    OFST_DEVNAM                   = 00000020
MSG22_READNHC                 ********   X   03    OFST_DRVERR                   = 0000004C
MSG23_CS_INC                  ********   X   09    OFST_DRVRST                   = 00000050
MSG28_DA_INC                  ********   X   0A    OFST_DRVTYP                   = 00000000
MSG30A_CRDY                   ********   X   06    OFST_FMTLEN                   = 00000030
MSG47_DS_INC                  ********   X   11    OFST_GTSCMD                   = 00000034
MSG54_DA_UPD                  ********   X   03    OFST_HDBITS                   = 00000044
MSG55_DAT_BND                 ********   X   07    OFST_HDRDNC                   = 00000054
MSG69_DAT_NHC                 ********   X   03    OFST_HDRTMP                   = 00000048
MSG73_GS_UND                  ********   X   0D    OFST_HIGRPM                   = 00000074
MSG74_DRDY_NC                 ********   X   0D    OFST_HMBSEC                   = 0000005C
MSG75_NO_ST1                  ********   X   0E    OFST_IGBCYL                   = 00000018
MSG76_ST1_BS                  ********   X   0E    OFST_LOWRPM                   = 00000070
MSG77_NO_ST2                  ********   X   0E    OFST_LSTCYL                   = 00000008
MSG77_NO_ST3                  ********   X   0E    OFST_LSTSEC                   = 00000010
MSG78_ST2_BS                  ********   X   0E    OFST_LSTTRK                   = 0000000C
MSG79_BH_NC                   ********   X   0E    OFST_MAXTIM                   = 0000007C
MSG80_BH_NF                   ********   X   0E    OFST_MPBITS                   = 00000040
MSG81_ST3_BS                  ********   X   0E    OFST_NMLSTS                   = 00000038
MSG82_NO_ST4                  ********   X   0E    OFST_ONETIM                   = 00000078
MSG83_ST4_BS                  ********   X   0E    OFST_OP1STS                   = 00000060
MSG84_NO_ST5                  ********   X   0E    OFST_PHYU                     = 00000004
MSG85_ST5_BS                  ********   X   0E    OFST_RANCYL                   = 0000006C
MSG86_DRDY_NS                 ********   X   0E    OFST_SECLEN                   = 00000028
MSG87_VC_NC                   ********   X   0E    OFST_SEKSTS                   = 0000003C
MSG88_NO_ST6                  ********   X   0F    OFST_STSDNC                   = 00000064
MSG89_ST6_BS                  ********   X   0F    OFST_TRKLEN                   = 0000002C
MSG90_NO_ST7                  ********   X   0F    OFST_TSTCYL                   = 00000014
MSG91_ST7_BS                  ********   X   0F    OFST_UNIT                     = 00000004
MSG92_NO_ST0                  ********   X   0F    OFST_WTPSTS                   = 00000068
MSG93_RELV_UP                 ********   X   10    OPR11_ORIGIN                    ********   X   11
MSG97_GS_NXD                  ********   X   10    OPR12_WRTLOCK                   ********   X   11
MSG98_WRT_WL                  ********   X   11    OPR13A_RST_FLT                  ********   X   11
MSG99_RST_WGE                 ********   X   11    OPR13_RST_WL                    ********   X   11
MSG_032                       00000002 R     03    OPR1_UNLOAD                     ********   X   0D
MSG_033                       00000002 R     06    OPR2_CLOSE                      ********   X   0D
MSG_034                       00000002 R     07    OPR3_START                      ********   X   0E
MSG_035                       00000002 R     08    OPR5_UNLOAD                     ********   X   0F
MSG_036                       00000002 R     09    OPR6_REMOVE                     ********   X   10
MSG_037                       00000002 R     0A    OPR7_INSERT                     ********   X   10
MSG_038                       00000002 R     0B    OPR8_SETUP                      ********   X   10
MSG_039                       00000002 R     0C    OPR_RETURN                      ********   X   0D
MSG_040                       00000002 R     0D    P$UNIT                          ********   X   03
MSG_041                       00000002 R     0E    PAR$M_ATDEF                   = 00000010
MSG_042                       00000002 R     0F    PAR$M_ATHI                    = 00000008
MSG_043                       00000002 R     10    PAR$M_ATLO                    = 00000004
MSG_044                       00000002 R     11    PAR$M_NODEF                   = 00000002
NOCHK$M_ATTN                = 00000080             PAR$V_ATDEF                   = 00000004
NOCHK$M_CERR                = 00000002             PAR$V_ATHI                    = 00000003
NOCHK$M_DERR                = 00000008             PAR$V_ATLO                    = 00000002
NOCHK$M_DRDY                = 00000004             PAR$V_NODEF                   = 00000001
NOCHK$M_DSTA                = 00000010             PAR$_BIN                      = 00000002
NO_OP                       = 00000000             PAR$_DEC                      = 0000000A
OFST_AVGTIM                 = 00000080             PAR$_HEX                      = 00000010
```

```
PAR$_NO                         = 00000000
PAR$_OCT                        = 00000008
PAR$_YES                        = 00000001
POSITION_HDS                      ********  X    03
PRINT_DATA_ERR                    ********  X    03
PRINT_REG                         ********  X    03
PRT$M_ALL                       = 00000007
R80$T                           = 00000002
RD_BAD_BLK                        ********  X    0C
RD_IDCREGS                        ********  X    03
RD_NHC                          = 0000000E
READ                            = 0000000C
READ_BUFFER                       ********  X    03
REG_ERRPT                         ********  X    03
RGLIST                            ********  X    0C
RL02$T                          = 00000001
SEED                              ********  X    03
SEEK                            = 00000006
SEP_FUNCTIONAL                  = 00000002
SEP_REPAIR                      = 00000003
SIZ...                          = 00000001
SKIP_BUFFER                       ********  X    09
START_CMD                         ********  X    03
STATUS_CHK                        ********  X    03
SUBSYS_INIT                       ********  X    03
SYS$CANTIM                        ******** GX    0E
SYS$READEF                        ******** GX    0C
SYS$SETIMR                        ******** GX    0E
S_DRIVE_TBL                       ********  X    06
S_P$UNIT                          ********  X    06
T.CS                              ********  X    09
T.DA                              ********  X    03
T.MP                              ********  X    0D
T32_S1                            00000082 RG    03
T32_S1_X                          00000100 R     03
T32_S2                            0000010B RG    03
T32_S2_X                          000001A4 R     03
T32_S3                            000001AF RG    03
T32_S3_X                          0000031D R     03
T34_S1                            00000073 RG    07
T34_S1_X                          000000F1 R     07
T34_S2                            000000FC RG    07
T34_S2_X                          0000026B R     07
T35_S1                            00000095 RG    08
T35_S1_X                          00000113 R     08
T35_S2                            0000011E RG    08
T35_S2_X                          00000200 R     08
T36_S1                            00000091 RG    09
T36_S1_X                          0000010F R     09
T36_S2                            0000011A RG    09
T36_S2_X                          00000187 R     09
T36_S3                            00000192 RG    09
T36_S3_X                          000003A0 R     09
T36_S4                            000003AB RG    09
T36_S4_X                          000004EB R     09
T37_S1                            00000095 RG    0A
T37_S1_X                          00000113 R     0A
T37_S2                            0000011E RG    0A
T37_S2_X                          0000018B R     0A
T37_S3                            00000196 RG    0A
T37_S3_X                          0000031F R     0A
T37_S4                            0000032A RG    0A
T37_S4_X                          00000455 R     0A
T37_S5                            00000460 RG    0A
T37_S5_X                          0000054B R     0A
T44_S1                            000000BA RG    11
T44_S1_X                          00000138 R     11
T44_S2                            00000143 RG    11
T44_S2_X                          000001EE R     11
T44_S3                            000001F9 RG    11
T44_S3_X                          000004E0 R     11
TEST_032                          0000004C RG    03
TEST_032_X                        0000032B RG    03
TEST_033                          00000038 RG    06
TEST_033_X                        000001AF RG    06
TEST_034                          00000038 RG    07
TEST_034_X                        00000279 RG    07
TEST_035                          00000048 RG    08
TEST_035_X                        0000020E RG    08
TEST_036                          00000044 RG    09
TEST_036_X                        000004F9 RG    09
TEST_037                          00000048 RG    0A
TEST_037_X                        00000559 RG    0A
TEST_038                          00000044 RG    0B
TEST_038_X                        00000198 RG    0B
TEST_039                          00000040 RG    0C
TEST_039_X                        000002B3 RG    0C
TEST_040                          0000003C RG    0D
TEST_040_X                        00000255 RG    0D
TEST_041                          00000040 RG    0E
TEST_041_X                        0000083E RG    0E
TEST_042                          00000044 RG    0F
TEST_042_X                        00000391 RG    0F
TEST_043                          00000038 RG    10
TEST_043_X                        0000024B RG    10
TEST_044                          0000003C RG    11
TEST_044_X                        00000520 RG    11
TIME_OUT                          ********  X    0E
WORD_COUNT                        ********  X    03
WRITE                           = 0000000A
WRITE_BUFFER                      ********  X    03
WRTCHK                          = 00000002
```

```
                                                 +----------------+
                                                 ! Psect synopsis !
                                                 +----------------+

PSECT name                       Allocation          PSECT No.  Attributes
----------                       ----------          ---------  ----------
.  ABS  .                        00000000  (     0.)  00 (   0.)  NOPIC    USR    CON    ABS    LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
HEADER                           00000000  (     0.)  01 (   1.)  NOPIC    USR    CON    REL    LCL NOSHR   EXE   RD   NOWRT NOVEC PAGE
$ABS$                            0000FE70  (65136.)  02 (   2.)  NOPIC    USR    CON    ABS    LCL NOSHR   EXE   RD     WRT NOVEC BYTE
TEST_032                         00000333  (   819.)  03 (   3.)  NOPIC    USR    CON    REL    LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
DISPATCH                         00000138  (   312.)  04 (   4.)  NOPIC    USR    CON    REL    LCL NOSHR NOEXE   RD   NOWRT NOVEC LONG
ARGLIST                          000000E4  (   228.)  05 (   5.)  NOPIC    USR    CON    REL    LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_033                         000001B7  (   439.)  06 (   6.)  NOPIC    USR    CON    REL    LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_034                         00000281  (   641.)  07 (   7.)  NOPIC    USR    CON    REL    LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_035                         00000216  (   534.)  08 (   8.)  NOPIC    USR    CON    REL    LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_036                         00000501  (  1281.)  09 (   9.)  NOPIC    USR    CON    REL    LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_037                         00000561  (  1377.)  0A (  10.)  NOPIC    USR    CON    REL    LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_038                         000001A0  (   416.)  0B (  11.)  NOPIC    USR    CON    REL    LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_039                         000002BB  (   699.)  0C (  12.)  NOPIC    USR    CON    REL    LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_040                         0000025D  (   605.)  0D (  13.)  NOPIC    USR    CON    REL    LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_041                         00000846  (  2118.)  0E (  14.)  NOPIC    USR    CON    REL    LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_042                         00000399  (   921.)  0F (  15.)  NOPIC    USR    CON    REL    LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_043                         00000253  (   595.)  10 (  16.)  NOPIC    USR    CON    REL    LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
TEST_044                         00000528  (  1320.)  11 (  17.)  NOPIC    USR    CON    REL    LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
$TSTCNT                          00000004  (     4.)  12 (  18.)  NOPIC    USR    OVR    REL    LCL NOSHR NOEXE   RD   NOWRT NOVEC LONG
```

```
                              +--------------------------+
                              ! Symbol Cross Reference !
                              +--------------------------+

SYMBOL              VALUE          DEFINITION      REFERENCES...
------              -----          ----------      -------------
$$$                 =000000D8-R    2299    (24)    1055    (11)    1100    (11)    138     (4)     159     (4)
                                                   189     (4)     2247    (24)    2268    (24)    2299    (24)
                                                   414     (8)     435     (8)     564     (9)     585     (9)
                                                   704     (10)    725     (10)    750     (10)    834     (10)
                                                   941     (11)    962     (11)    987     (11)
$$E                 =00000001      2299    (24)    1054    (11)    1099    (11)    1136    (11)    158     (4)
                                                   188     (4)     2267    (24)    2298    (24)    2407    (24)
                                                   252     (4)     434     (8)     507     (8)     584     (9)
                                                   634     (9)     724     (10)    749     (10)    833     (10)
                                                   883     (10)    961     (11)    986     (11)
$$M                 =00000004      2405    (24)    1018    (11)    1037    (11)    1043    (11)    1052    (11)
                                                   1077    (11)    1083    (11)    1089    (11)    1097    (11)
                                                   1122    (11)    1128    (11)    1134    (11)    1180    (12)
                                                   1196    (12)    1215    (12)    1224    (12)    1230    (12)
                                                   1267    (14)    127     (4)     1287    (14)    1320    (14)
                                                   1336    (14)    1350    (14)    1366    (14)    1415    (16)
                                                   1431    (16)    1447    (16)    1454    (16)    1458    (16)
                                                   147     (4)     1474    (16)    1482    (16)    1536    (18)
                                                   154     (4)     1553    (18)    1577    (18)    1587    (18)
                                                   1600    (18)    1623    (18)    1633    (18)    1646    (18)
                                                   1667    (18)    1675    (18)    1695    (18)    1703    (18)
                                                   1729    (18)    1739    (18)    1747    (18)    1769    (18)
                                                   1777    (18)    1792    (18)    1797    (18)    1819    (18)
                                                   1832    (18)    1842    (18)    1855    (18)    186     (4)
                                                   1876    (18)    1881    (18)    1922    (20)    1939    (20)
                                                   1961    (20)    1970    (20)    1986    (20)    2007    (20)
                                                   2016    (20)    2031    (20)    2053    (20)    2062    (20)
                                                   2104    (22)    2118    (22)    2136    (22)    2155    (22)
                                                   2165    (22)    219     (4)     2229    (24)    2256    (24)
                                                   2263    (24)    2282    (24)    2296    (24)    2318    (24)
                                                   232     (4)     2333    (24)    2341    (24)    2352    (24)
                                                   2368    (24)    2383    (24)    2397    (24)    2405    (24)
                                                   241     (4)     250     (4)     286     (6)     309     (6)
                                                   316     (6)     344     (6)     351     (6)     400     (8)
                                                   423     (8)     430     (8)     461     (8)     480     (8)
                                                   503     (8)     548     (9)     573     (9)     580     (9)
                                                   607     (9)     631     (9)     688     (10)    713     (10)
                                                   720     (10)    747     (10)    780     (10)    799     (10)
                                                   805     (10)    817     (10)    823     (10)    831     (10)
                                                   857     (10)    863     (10)    875     (10)    881     (10)
                                                   925     (11)    950     (11)    957     (11)    984     (11)
$$N                 =00000002      2408    (24)    1018    (11)    1037    (11)    1043    (11)    1052    (11)
                                                   1077    (11)    1083    (11)    1089    (11)    1097    (11)
                                                   1122    (11)    1128    (11)    1134    (11)    1196    (12)
                                                   1215    (12)    1224    (12)    1230    (12)    1279    (14)
                                                   1287    (14)    1320    (14)    1336    (14)    1350    (14)
                                                   1366    (14)    1431    (16)    1433    (16)    1447    (16)
                                                   1454    (16)    1458    (16)    1460    (16)    147     (4)
                                                   1474    (16)    1482    (16)    154     (4)     1553    (18)
```

```
                                                        1555    (18)      1577    (18)      1587    (18)      1600    (18)
                                                        1623    (18)      1633    (18)      1646    (18)      1667    (18)
                                                        1675    (18)      1695    (18)      1703    (18)      1729    (18)
                                                        1739    (18)      1747    (18)      1769    (18)      1777    (18)
                                                        1792    (18)      1797    (18)      1819    (18)      1832    (18)
                                                        1842    (18)      1855    (18)      186     (4)       1876    (18)
                                                        1881    (18)      1939    (20)      1941    (20)      1961    (20)
                                                        1970    (20)      1986    (20)      2007    (20)      2016    (20)
                                                        2031    (20)      2053    (20)      2062    (20)      2118    (22)
                                                      #-2120    (22)      2123    (22)      2136    (22)      2155    (22)
                                                        2165    (22)    #-2170    (22)      219     (4)       2244    (24)
                                                        2256    (24)      2263    (24)      2282    (24)      2296    (24)
                                                        2308    (24)      2318    (24)      232     (4)       2333    (24)
                                                        2341    (24)      2352    (24)      2356    (24)      2368    (24)
                                                        2383    (24)      2397    (24)      2405    (24)      2408    (24)
                                                        241     (4)       250     (4)       309     (6)       316     (6)
                                                        344     (6)       351     (6)       423     (8)       430     (8)
                                                        461     (8)       480     (8)       503     (8)       573     (9)
                                                        580     (9)       607     (9)       631     (9)       67      (2)
                                                        713     (10)      720     (10)      747     (10)      780     (10)
                                                        799     (10)      805     (10)      817     (10)      823     (10)
                                                        831     (10)      857     (10)      863     (10)      875     (10)
                                                        881     (10)      950     (11)      957     (11)      984     (11)
$$S                            =FFFFFFFF     2411   (24)  1055    (11)      1100    (11)      1138    (11)      1180    (12)
                                                        1233    (12)      1267    (14)      127     (4)       1372    (14)
                                                        138     (4)       1415    (16)      1485    (16)      1536    (18)
                                                        159     (4)       1884    (18)      189     (4)       1922    (20)
                                                        2067    (20)      2104    (22)      2174    (22)      2229    (24)
                                                        2247    (24)      2268    (24)      2299    (24)      254     (4)
                                                        286     (6)       355     (6)       400     (8)       414     (8)
                                                        435     (8)       509     (8)       548     (9)       564     (9)
                                                        585     (9)       636     (9)       688     (10)      704     (10)
                                                        725     (10)      73      (2)       750     (10)      834     (10)
                                                        885     (10)      925     (11)      941     (11)      962     (11)
                                                        987     (11)
$$T1                           =00000001     2065   (20)  1557    (18)      1588    (18)      1601    (18)      1624    (18)
                                                        1634    (18)      1647    (18)      1668    (18)      1676    (18)
                                                        1696    (18)      1704    (18)      1712    (18)      1714    (18)
                                                        1730    (18)      1740    (18)      1949    (20)      1973    (20)
                                                        2041    (20)      2065    (20)
$ENV                           =00000003     61     (2)
$ER                            =FFFFFFFF     2411   (24)  #-1018  (11)    #-1037    (11)    #-1043    (11)    #-1052    (11)
                                                        1055    (11)    #-1077    (11)    #-1083    (11)    #-1089    (11)
                                                      #-1097    (11)      1100    (11)    #-1122    (11)    #-1128    (11)
                                                      #-1134    (11)    #-1196    (12)    #-1215    (12)    #-1224    (12)
                                                      #-1230    (12)    #-1287    (14)    #-1320    (14)    #-1336    (14)
                                                      #-1350    (14)    #-1366    (14)      138     (4)     #-1431    (16)
                                                      #-1447    (16)    #-1454    (16)    #-1458    (16)    #-147     (4)
                                                      #-1474    (16)    #-1482    (16)    #-154     (4)     #-1553    (18)
                                                      #-1577    (18)    #-1587    (18)      159     (4)     #-1600    (18)
                                                      #-1623    (18)    #-1633    (18)    #-1646    (18)    #-1667    (18)
                                                      #-1675    (18)    #-1695    (18)    #-1703    (18)    #-1729    (18)
                                                      #-1739    (18)    #-1747    (18)    #-1769    (18)    #-1777    (18)
                                                      #-1792    (18)    #-1797    (18)    #-1819    (18)    #-1832    (18)
                                                      #-1842    (18)    #-1855    (18)    #-186     (4)     #-1876    (18)
                                                      #-1881    (18)      189     (4)     #-1939    (20)    #-1961    (20)
```

```
                                                     #-1970  (20)    #-1986  (20)    #-2007  (20)    #-2016  (20)
                                                     #-2031  (20)    #-2053  (20)    #-2062  (20)    #-2118  (22)
                                                     #-2136  (22)    #-2155  (22)    #-2165  (22)    #-219   (4)
                                                      2247   (24)    #-2256  (24)    #-2263  (24)     2268   (24)
                                                     #-2282  (24)    #-2296  (24)     2299   (24)    #-2318  (24)
                                                     #-232   (4)     #-2333  (24)    #-2341  (24)    #-2352  (24)
                                                     #-2368  (24)    #-2383  (24)    #-2397  (24)    #-2405  (24)
                                                     #-241   (4)     #-250   (4)     #-309   (6)     #-316   (6)
                                                     #-344   (6)     #-351   (6)      414    (8)     #-423   (8)
                                                     #-430   (8)      435    (8)     #-461   (8)     #-480   (8)
                                                     #-503   (8)      564    (9)     #-573   (9)     #-580   (9)
                                                      585    (9)     #-607   (9)     #-629   (9)     #-631   (9)
                                                      704    (10)    #-713   (10)    #-720   (10)     725    (10)
                                                     #-747   (10)     750    (10)    #-780   (10)    #-799   (10)
                                                     #-805   (10)    #-817   (10)    #-823   (10)    #-831   (10)
                                                      834    (10)    #-857   (10)    #-863   (10)    #-875   (10)
                                                     #-881   (10)     941    (11)    #-950   (11)    #-957   (11)
                                                      962    (11)    #-984   (11)     987    (11)
$GS_SETMAP                    00000000-XR             1008   (11)     1026   (11)     1066   (11)     1111   (11)
                                                      1324   (14)     1355   (14)     175    (4)      201    (4)
                                                      2286   (24)     2322   (24)     2387   (24)     449    (8)
                                                      470    (8)      597    (9)      621    (9)      737    (10)
                                                      770    (10)     788    (10)     846    (10)     974    (11)
$MO                           =00000001    2412  (24) 2412   (24)
$S$                           =000000D8-R  2299  (24) 1054   (11)     1055   (11)     1099   (11)     1100   (11)
                                                      1136   (11)     138    (4)      158    (4)      159    (4)
                                                      188    (4)      189    (4)      2247   (24)     2267   (24)
                                                      2268   (24)     2298   (24)     2299   (24)     2407   (24)
                                                      252    (4)      414    (8)      434    (8)      435    (8)
                                                      507    (8)      564    (9)      584    (9)      585    (9)
                                                      634    (9)      704    (10)     724    (10)     725    (10)
                                                      749    (10)     750    (10)     833    (10)     834    (10)
                                                      883    (10)     941    (11)     961    (11)     962    (11)
                                                      986    (11)     987    (11)
$ST                           =00000000    2411  (24) 1018   (11)     1019   (11)     1037   (11)     1038   (11)
                                                      1043   (11)     1044   (11)     1052   (11)     1053   (11)
                                                      1054   (11)     1055   (11)     1077   (11)     1078   (11)
                                                      1083   (11)     1084   (11)     1089   (11)     1090   (11)
                                                      1097   (11)     1098   (11)     1099   (11)     1100   (11)
                                                      1122   (11)     1123   (11)     1128   (11)     1129   (11)
                                                      1134   (11)     1135   (11)     1136   (11)     1137   (11)
                                                      1180   (12)     1196   (12)     1197   (12)     1215   (12)
                                                      1216   (12)     1224   (12)     1225   (12)     1230   (12)
                                                      1231   (12)     1232   (12)     1267   (14)     127    (4)
                                                      1287   (14)     1288   (14)     1320   (14)     1321   (14)
                                                      1336   (14)     1337   (14)     1350   (14)     1351   (14)
                                                      1366   (14)     1369   (14)     1371   (14)     138    (4)
                                                      1415   (16)     1431   (16)     1432   (16)     1447   (16)
                                                      1448   (16)     1454   (16)     1455   (16)     1458   (16)
                                                      1459   (16)     147    (4)      1474   (16)     1475   (16)
                                                      148    (4)      1482   (16)     1483   (16)     1484   (16)
                                                      1536   (18)     154    (4)      155    (4)      1553   (18)
                                                      1554   (18)     1577   (18)     1578   (18)     158    (4)
                                                      1587   (18)     1589   (18)     159    (4)      1600   (18)
                                                      1602   (18)     1623   (18)     1625   (18)     1633   (18)
                                                      1635   (18)     1646   (18)     1648   (18)     1667   (18)
```

```
                                                         1669    (18)       1675    (18)       1677    (18)       1695    (18)
                                                         1697    (18)       1703    (18)       1705    (18)       1729    (18)
                                                         1731    (18)       1739    (18)       1741    (18)       1747    (18)
                                                         1748    (18)       1769    (18)       1770    (18)       1777    (18)
                                                         1778    (18)       1792    (18)       1793    (18)       1797    (18)
                                                         1798    (18)       1819    (18)       1820    (18)       1832    (18)
                                                         1833    (18)       1842    (18)       1843    (18)       1855    (18)
                                                         1856    (18)       186     (4)        187     (4)        1876    (18)
                                                         1877    (18)       188     (4)        1881    (18)       1882    (18)
                                                         1883    (18)       189     (4)        1922    (20)       1939    (20)
                                                         1940    (20)       1961    (20)       1962    (20)       1970    (20)
                                                         1971    (20)       1986    (20)       1987    (20)       2007    (20)
                                                         2008    (20)       2016    (20)       2017    (20)       2031    (20)
                                                         2032    (20)       2053    (20)       2054    (20)       2062    (20)
                                                         2063    (20)       2066    (20)       2104    (22)       2118    (22)
                                                         2119    (22)       2136    (22)       2137    (22)       2155    (22)
                                                         2156    (22)       2165    (22)       2166    (22)       2173    (22)
                                                         219     (4)        220     (4)        2229    (24)       2247    (24)
                                                         2256    (24)       2257    (24)       2263    (24)       2264    (24)
                                                         2267    (24)       2268    (24)       2282    (24)       2283    (24)
                                                         2296    (24)       2297    (24)       2298    (24)       2299    (24)
                                                         2318    (24)       2319    (24)       232     (4)        233     (4)
                                                         2333    (24)       2334    (24)       2341    (24)       2342    (24)
                                                         2352    (24)       2353    (24)       2368    (24)       2369    (24)
                                                         2383    (24)       2384    (24)       2397    (24)       2398    (24)
                                                         2405    (24)       2406    (24)       2407    (24)       241     (4)
                                                         2411    (24)       242     (4)        250     (4)        251     (4)
                                                         252     (4)        253     (4)        286     (6)        309     (6)
                                                         310     (6)        316     (6)        317     (6)        344     (6)
                                                         351     (6)        352     (6)        354     (6)        400     (8)
                                                         414     (8)        423     (8)        424     (8)        430     (8)
                                                         431     (8)        434     (8)        435     (8)        461     (8)
                                                         462     (8)        480     (8)        481     (8)        503     (8)
                                                         504     (8)        507     (8)        508     (8)        548     (9)
                                                         564     (9)        573     (9)        574     (9)        580     (9)
                                                         581     (9)        584     (9)        585     (9)        607     (9)
                                                         608     (9)        631     (9)        632     (9)        634     (9)
                                                         635     (9)        688     (10)       704     (10)       713     (10)
                                                         714     (10)       720     (10)       721     (10)       724     (10)
                                                         725     (10)       747     (10)       748     (10)       749     (10)
                                                         750     (10)       780     (10)       781     (10)       799     (10)
                                                         800     (10)       805     (10)       806     (10)       817     (10)
                                                         818     (10)       823     (10)       824     (10)       831     (10)
                                                         832     (10)       833     (10)       834     (10)       857     (10)
                                                         858     (10)       863     (10)       864     (10)       875     (10)
                                                         876     (10)       881     (10)       882     (10)       883     (10)
                                                         884     (10)       925     (11)       941     (11)       950     (11)
                                                         951     (11)       957     (11)       958     (11)       961     (11)
                                                         962     (11)       984     (11)       985     (11)       986     (11)
                                                         987     (11)
$TN                            =0000002C     2412   (24) 1018    (11)       1037    (11)       1043    (11)       1052    (11)
                                                         1077    (11)       1083    (11)       1089    (11)       1097    (11)
                                                         1122    (11)       1128    (11)       1134    (11)       1137    (11)
                                                         1138    (11)       1173    (12)       1180    (12)       1183    (12)
                                                         1187    (12)       1196    (12)       120     (4)        1215    (12)
                                                         1217    (12)       1224    (12)       1230    (12)       1232    (12)
```

```
                                                            1233   (12)     1260   (14)     1267   (14)     127    (4)
                                                            1275   (14)     1287   (14)     130    (4)      1320   (14)
                                                            1336   (14)     1350   (14)     1366   (14)     1371   (14)
                                                            1372   (14)     1408   (16)     1415   (16)     1418   (16)
                                                            1420   (16)     1431   (16)     1447   (16)     1454   (16)
                                                            1458   (16)     147    (4)      1474   (16)     1482   (16)
                                                            1484   (16)     1485   (16)     1529   (18)     1536   (18)
                                                            1539   (18)     154    (4)      1543   (18)     1545   (18)
                                                            1553   (18)     156    (4)      1577   (18)     1579   (18)
                                                            1587   (18)     1590   (18)     1600   (18)     1603   (18)
                                                            1623   (18)     1626   (18)     1633   (18)     1636   (18)
                                                            1646   (18)     1649   (18)     1667   (18)     1670   (18)
                                                            1675   (18)     1678   (18)     1695   (18)     1698   (18)
                                                            1703   (18)     1706   (18)     1729   (18)     1732   (18)
                                                            1739   (18)     1747   (18)     1749   (18)     1769   (18)
                                                            1771   (18)     1777   (18)     1779   (18)     1792   (18)
                                                            1794   (18)     1797   (18)     1799   (18)     1819   (18)
                                                            1821   (18)     1832   (18)     1834   (18)     1842   (18)
                                                            1844   (18)     1855   (18)     186    (4)      1876   (18)
                                                            1881   (18)     1883   (18)     1884   (18)     1915   (20)
                                                            1922   (20)     1925   (20)     1929   (20)     1931   (20)
                                                            1939   (20)     1961   (20)     1963   (20)     1970   (20)
                                                            1972   (20)     1986   (20)     1988   (20)     2007   (20)
                                                            2009   (20)     2016   (20)     2018   (20)     2031   (20)
                                                            2033   (20)     2053   (20)     2055   (20)     2062   (20)
                                                            2064   (20)     2066   (20)     2067   (20)     2097   (22)
                                                            2104   (22)     2107   (22)     2109   (22)     2118   (22)
                                                            2136   (22)     2155   (22)     2165   (22)     2173   (22)
                                                            2174   (22)     219    (4)      221    (4)      2222   (24)
                                                            2229   (24)     2232   (24)     2234   (24)     2256   (24)
                                                            2263   (24)     2265   (24)     2282   (24)     2296   (24)
                                                            2318   (24)     232    (4)      2333   (24)     2341   (24)
                                                            2352   (24)     2368   (24)     2383   (24)     2397   (24)
                                                            2405   (24)     241    (4)      2411   (24)     2412   (24)
                                                            250    (4)      253    (4)      254    (4)      279    (6)
                                                            286    (6)      289    (6)      298    (6)      309    (6)
                                                            316    (6)      318    (6)      344    (6)      351    (6)
                                                            354    (6)      355    (6)      393    (8)      400    (8)
                                                            403    (8)      423    (8)      430    (8)      432    (8)
                                                            461    (8)      480    (8)      503    (8)      508    (8)
                                                            509    (8)      541    (9)      548    (9)      551    (9)
                                                            552    (9)      556    (9)      573    (9)      580    (9)
                                                            582    (9)      607    (9)      631    (9)      635    (9)
                                                            636    (9)      681    (10)     688    (10)     691    (10)
                                                            692    (10)     696    (10)     713    (10)     720    (10)
                                                            722    (10)     73     (2)      747    (10)     780    (10)
                                                            799    (10)     805    (10)     817    (10)     823    (10)
                                                            831    (10)     857    (10)     863    (10)     875    (10)
                                                            881    (10)     884    (10)     885    (10)     918    (11)
                                                            925    (11)     928    (11)     929    (11)     933    (11)
                                                            950    (11)     957    (11)     959    (11)     984    (11)
ALL                               =00000002     67     (2)  1180   (12)     1267   (14)     127    (4)      1415   (16)
                                                            1536   (18)     1922   (20)     2104   (22)     2229   (24)
                                                            286    (6)      400    (8)      548    (9)      688    (10)
                                                            925    (11)
BASE_032                          00000000-R    73     (2)  127    (4)
```

```
BASE_033                        00000000-R    254    (4)       286     (6)
BASE_034                        00000000-R    355    (6)       400     (8)
BASE_035                        00000000-R    509    (8)       548     (9)
BASE_036                        00000000-R    636    (9)       688     (10)
BASE_037                        00000000-R    885    (10)      925     (11)
BASE_038                        00000000-R    1138   (11)      1180    (12)
BASE_039                        00000000-R    1233   (12)      1267    (14)
BASE_040                        00000000-R    1372   (14)      1415    (16)
BASE_041                        00000000-R    1485   (16)      1536    (18)
BASE_042                        00000000-R    1884   (18)      1922    (20)
BASE_043                        00000000-R    2067   (20)      2104    (22)
BASE_044                        00000000-R    2174   (22)      2229    (24)
BASE_ADDR                       00000000-XR                   #-1309   (14)
BIT...                         =00000028      629    (9)       61      (2)       629     (9)       65      (2)       66      (2)
                                                               71      (2)
BIT0                           =00000001      64     (2)
BIT1                           =00000002      64     (2)
BIT10                          =00000400      64     (2)
BIT11                          =00000800      64     (2)
BIT12                          =00001000      64     (2)
BIT13                          =00002000      64     (2)
BIT14                          =00004000      64     (2)
BIT15                          =00008000      64     (2)
BIT16                          =00010000      64     (2)
BIT17                          =00020000      64     (2)
BIT18                          =00040000      64     (2)
BIT19                          =00080000      64     (2)
BIT2                           =00000004      64     (2)
BIT20                          =00100000      64     (2)
BIT21                          =00200000      64     (2)
BIT22                          =00400000      64     (2)
BIT23                          =00800000      64     (2)
BIT24                          =01000000      64     (2)
BIT25                          =02000000      64     (2)
BIT26                          =04000000      64     (2)
BIT27                          =08000000      64     (2)
BIT28                          =10000000      64     (2)
BIT29                          =20000000      64     (2)
BIT3                           =00000008      64     (2)
BIT30                          =40000000      64     (2)
BIT31                          =80000000      64     (2)
BIT4                           =00000010      64     (2)
BIT5                           =00000020      64     (2)
BIT6                           =00000040      64     (2)
BIT7                           =00000080      64     (2)
BIT8                           =00000100      64     (2)
BIT9                           =00000200      64     (2)
BUILD_HDRS                      00000000-XR                    594     (9)       734     (10)      971     (11)
CEP_FUNCTIONAL                 =00000000      61     (2)
CEP_REPAIR                     =00000001      61     (2)
CHANGE_DAR$A                    00000000-XR                    1046    (11)      235     (4)
CHAR_BUFFER                     00000000-XR                    1435    (16)      1462    (16)      2122    (22)      2125    (22)
                                                               2172    (22)      2246    (24)      2310    (24)      2358    (24)
                                                               2410    (24)
CHC$_ABORT                     =00000002      65     (2)
CHC$_CLEAR                     =00000006      65     (2)
```

```
CHC$_CLRDFT              =00000009      65      (2)
CHC$_DSINT               =00000005      65      (2)
CHC$_ENINT               =00000004      65      (2)
CHC$_INITA               =00000000      65      (2)
CHC$_INITB               =00000001      65      (2)
CHC$_PURGE               =00000003      65      (2)
CHC$_SELF_TEST           =0000000B      65      (2)
CHC$_SETDFT              =00000008      65      (2)
CHC$_STATUS              =00000007      65      (2)
CHC$_STOP                =0000000A      65      (2)
CHECK_BADBLK              00000000-XR                   1301    (14)
CHECK_ECC                 00000000-XR                   1367    (14)
CHI$M_CHNINT             =00000001      65      (2)
CHI$M_DEVINT             =00000002      65      (2)
CHI$M_IPL                =0000007C      65      (2)
CHI$S_IPL                =00000005      65      (2)
CHI$V_CHNINT             =00000000      65      (2)
CHI$V_DEVINT             =00000001      65      (2)
CHI$V_IPL                =00000002      65      (2)
CHM$_FORWARD             =00000000      65      (2)     65      (2)
CHM$_INVALIDATE          =00000001      65      (2)     65      (2)
CHM$_MAP                 =00000002      65      (2)     65      (2)
CHM$_MFWDN               =00000002      65      (2)
CHM$_MFWDNO              =0000000A      65      (2)
CHM$_MFWDV               =00000003      65      (2)
CHM$_MFWDVO              =0000000B      65      (2)
CHM$_MREVN               =00000006      65      (2)
CHM$_MREVNO              =0000000E      65      (2)
CHM$_MREVV               =00000007      65      (2)
CHM$_MREVVO              =0000000F      65      (2)
CHM$_NFWDN               =00000000      65      (2)
CHM$_NREVN               =00000004      65      (2)
CHM$_OFFSET              =00000008      65      (2)     65      (2)
CHM$_REVERSE             =00000004      65      (2)     65      (2)
CHS$M_BADBDP             =20000000      65      (2)
CHS$M_BADPAR             =40000000      65      (2)
CHS$M_BUSIC              =00800000      65      (2)
CHS$M_BUSINIT            =00400000      65      (2)
CHS$M_BUSNXM             =08000000      65      (2)
CHS$M_BUSPDN             =01000000      65      (2)
CHS$M_CHNDPE             =00000040      65      (2)
CHS$M_CHNERR             =00000002      65      (2)     65      (2)
CHS$M_CHNMPE             =00000080      65      (2)
CHS$M_CHPFOT             =00000100      65      (2)
CHS$M_DEVBUS             =00000010      65      (2)
CHS$M_DEVERR             =00000004      65      (2)     65      (2)
CHS$M_DEVTO              =00000020      65      (2)
CHS$M_ERRANY             =0000000F      65      (2)
CHS$M_MBAATN             =00100000      65      (2)
CHS$M_MBACPE             =00200000      65      (2)
CHS$M_MBADTB             =00040000      65      (2)
CHS$M_MBADTC             =00080000      65      (2)
CHS$M_MBAEXC             =00010000      65      (2)
CHS$M_MBANED             =00020000      65      (2)
CHS$M_MBAWCKLWR          =02000000      65      (2)
CHS$M_MBAWCKUPR          =04000000      65      (2)
```

```
CHS$M_PGMERR                   =00000008      65       (2)       65      (2)
CHS$M_PGMHDE                   =00000800      65       (2)
CHS$M_SYSERR                   =00000001      65       (2)       65      (2)
CHS$M_SYSMEM                   =00000200      65       (2)
CHS$M_SYSSBI                   =00000400      65       (2)
CHS$M_UIE                      =10000000      65       (2)
CHS$V_BADBDP                   =0000001D      65       (2)
CHS$V_BADPAR                   =0000001E      65       (2)
CHS$V_BUSIC                    =00000017      65       (2)
CHS$V_BUSINIT                  =00000016      65       (2)
CHS$V_BUSNXM                   =0000001B      65       (2)
CHS$V_BUSPDN                   =00000018      65       (2)
CHS$V_CHNDPE                   =00000006      65       (2)
CHS$V_CHNERR                   =00000001      65       (2)
CHS$V_CHNMPE                   =00000007      65       (2)
CHS$V_CHPFOT                   =00000008      65       (2)
CHS$V_DEVBUS                   =00000004      65       (2)
CHS$V_DEVERR                   =00000002      65       (2)
CHS$V_DEVTO                    =00000005      65       (2)
CHS$V_MBAATN                   =00000014      65       (2)
CHS$V_MBACPE                   =00000015      65       (2)
CHS$V_MBADTB                   =00000012      65       (2)
CHS$V_MBADTC                   =00000013      65       (2)
CHS$V_MBAEXC                   =00000010      65       (2)
CHS$V_MBANED                   =00000011      65       (2)
CHS$V_MBAWCKLWR                =00000019      65       (2)
CHS$V_MBAWCKUPR                =0000001A      65       (2)
CHS$V_PGMERR                   =00000003      65       (2)
CHS$V_PGMHDE                   =0000000B      65       (2)
CHS$V_SYSERR                   =00000000      65       (2)
CHS$V_SYSMEM                   =00000009      65       (2)
CHS$V_SYSSBI                   =0000000A      65       (2)
CHS$V_UIE                      =0000001C      65       (2)
CMPDATA                         00000000-XR                    1094    (11)      2402   (24)       247    (4)        500    (8)
                                                               828     (10)
CONFIG                          00000000-XR                   #-1033   (11)     #-1073  (11)     #-1118  (11)     #-1188  (12)
                                                              #-1332   (14)     #-1346  (14)     #-1362  (14)     #-137   (4)
                                                              #-1427   (16)     #-1443  (16)     #-1470  (16)     #-1549  (18)
                                                              #-157    (4)      #-1573  (18)     #-1619  (18)     #-1663  (18)
                                                              #-1691   (18)     #-1725  (18)     #-1765  (18)     #-1815  (18)
                                                              #-1872   (18)     #-1935  (20)     #-1957  (20)     #-2003  (20)
                                                              #-2049   (20)     #-2114  (22)     #-2132  (22)     #-2151  (22)
                                                              #-2243   (24)     #-2266  (24)     #-2329  (24)     #-291   (6)
                                                              #-319    (6)      #-405   (8)      #-433   (8)      #-562   (9)
                                                              #-583    (9)      #-702   (10)     #-723   (10)     #-795   (10)
                                                              #-813    (10)     #-853   (10)     #-871   (10)     #-939   (11)
                                                              #-960    (11)
CSR$M_ASSI                     =08000000                      #-1013   (11)     #-775   (10)     #-783   (10)     #-793   (10)
                                                              #-807    (10)     #-809   (10)     #-841   (10)     #-851   (10)
                                                              #-865    (10)     #-867   (10)
CSR$M_CRDY                     =00000080                      #-1021   (11)     #-1062  (11)     #-1107  (11)     #-214   (4)
                                                              #-2159   (22)     #-2344  (24)     #-320   (6)      #-323   (6)
                                                              #-334    (6)      #-337   (6)      #-339   (6)      #-783   (10)
                                                              #-809    (10)     #-841   (10)     #-867   (10)
CSR$M_DE                       =00004000                      #-2343   (24)
CSR$M_DRDY                     =00000001                      #-1021   (11)     #-1062  (11)     #-1107  (11)     #-2345  (24)
```

```
                                                   #-784   (10)   #-810   (10)   #-842   (10)   #-868   (10)
CSR$M_ERR        =00008000                         #-2158  (22)   #-2343  (24)
CSR$M_F0         =00000002                         #-2345  (24)
CSR$M_F1         =00000004                         #-2159  (22)
CSR$M_F2         =00000008                         #-2344  (24)
CSR$M_FMT        =20000000                         #-602   (9)    #-742   (10)   #-979   (11)
CSR$M_OPI        =00000400                         #-2158  (22)
CSR$M_SSE        =00800000                         #-783   (10)   #-841   (10)
CSR$M_SSEI       =00400000                         #-1013  (11)   #-775   (10)   #-807   (10)   #-809   (10)
                                                   #-865   (10)   #-867   (10)
CSR$V_DE         =0000000E                         #-1795  (18)
CSR$V_DRDY       =00000000                         #-1456  (16)   #-1853  (18)
CSR_BITS          00000000-XR                      #-2165  (22)   #-2352  (24)
CURR_CMD          00000000-XR                      #-2129  (22)   #-2145  (22)   #-228   (4)
CURR_CYLN         00000000-XR                      #-1226  (12)   #-1230  (12)
CURR_FLAGS        00000000-XR                       1293   (14)    1312   (14)    1317   (14)    1325   (14)
                                                    1338   (14)    1339   (14)    1352   (14)
DAR_MNEM          00000000-XR                      #-1052  (11)   #-1089  (11)   #-1134  (11)   #-241   (4)
DATA_032          00000028-R   120    (4)           127    (4)
DATA_033          00000014-R   279    (6)           286    (6)
DATA_034          00000014-R   393    (8)           400    (8)
DATA_035          00000024-R   541    (9)           548    (9)
DATA_036          00000020-R   681    (10)          688    (10)
DATA_037          00000024-R   918    (11)          925    (11)
DATA_038          00000020-R   1173   (12)          1180   (12)
DATA_039          0000001C-R   1260   (14)          1267   (14)
DATA_040          00000018-R   1408   (16)          1415   (16)
DATA_041          0000001C-R   1529   (18)          1536   (18)
DATA_042          00000020-R   1915   (20)          1922   (20)
DATA_043          00000014-R   2097   (22)          2104   (22)
DATA_044          00000018-R   2222   (24)          2229   (24)
DEFAULT          =00000000     67     (2)           1180   (12)    1267   (14)    127    (4)     286    (6)
                                                    400    (8)     548    (9)     688    (10)    925    (11)
DEL$V_PROT       =00000014                         #-1305  (14)
DEL$V_READ       =0000000D                         #-1352  (14)
DEL$V_READH      =0000000A                         #-1312  (14)   #-1317  (14)
DEL$V_WRCK       =0000000C                         #-1339  (14)
DEL$V_WRIT       =0000000B                         #-1325  (14)   #-1338  (14)
DELTA_TIME        00000000-XR                      #-1556  (18)    1557   (18)   #-1713  (18)    1714   (18)
                                                   #-1948  (20)    1949   (20)   #-2040  (20)    2041   (20)
DEV_NAME          00000000-XR                      #-1279  (14)   #-1433  (16)   #-1460  (16)   #-1555  (18)
                                                   #-1941  (20)   #-2123  (22)   #-2244  (24)   #-2308  (24)
                                                   #-2356  (24)   #-2408  (24)
DRIVE_TBL         00000000-XR                      #-1184  (12)   #-1276  (14)   #-131   (4)    #-1421  (16)
                                                   #-1540  (18)   #-1926  (20)   #-2110  (22)   #-2235  (24)
                                                   #-290   (6)    #-404   (8)    #-553   (9)    #-693   (10)
                                                   #-930   (11)
DS$ASKSTR         00000000-XR                       1435   (16)    1462   (16)    2122   (22)    2125   (22)
                                                    2172   (22)    2246   (24)    2310   (24)    2358   (24)
                                                    2410   (24)
DS$BGNSUB         00000000-XR                       1055   (11)    1100   (11)    138    (4)     159    (4)
                                                    189    (4)     2247   (24)    2268   (24)    2299   (24)
                                                    414    (8)     435    (8)     564    (9)     585    (9)
                                                    704    (10)    725    (10)    750    (10)    834    (10)
                                                    941    (11)    962    (11)    987    (11)
DS$BRANCH         00000000-XR                       629    (9)
```

DS$BREAK	00000000-XR	1137	(11)	1232	(12)	1371	(14)	1484	(16)
		1883	(18)	2066	(20)	2173	(22)	2411	(24)
		253	(4)	354	(6)	508	(8)	635	(9)
		884	(10)						
DS$CKLOOP	00000000-XR	1019	(11)	1038	(11)	1044	(11)	1053	(11)
		1078	(11)	1084	(11)	1090	(11)	1098	(11)
		1123	(11)	1129	(11)	1135	(11)	1197	(12)
		1216	(12)	1225	(12)	1231	(12)	1288	(14)
		1321	(14)	1337	(14)	1351	(14)	1369	(14)
		1432	(16)	1448	(16)	1455	(16)	1459	(16)
		1475	(16)	148	(4)	1483	(16)	155	(4)
		1554	(18)	1578	(18)	1589	(18)	1602	(18)
		1625	(18)	1635	(18)	1648	(18)	1669	(18)
		1677	(18)	1697	(18)	1705	(18)	1731	(18)
		1741	(18)	1748	(18)	1770	(18)	1778	(18)
		1793	(18)	1798	(18)	1820	(18)	1833	(18)
		1843	(18)	1856	(18)	187	(4)	1877	(18)
		1882	(18)	1940	(20)	1962	(20)	1971	(20)
		1987	(20)	2008	(20)	2017	(20)	2032	(20)
		2054	(20)	2063	(20)	2119	(22)	2137	(22)
		2156	(22)	2166	(22)	220	(4)	2257	(24)
		2264	(24)	2283	(24)	2297	(24)	2319	(24)
		233	(4)	2334	(24)	2342	(24)	2353	(24)
		2369	(24)	2384	(24)	2398	(24)	2406	(24)
		242	(4)	251	(4)	310	(6)	317	(6)
		352	(6)	424	(8)	431	(8)	462	(8)
		481	(8)	504	(8)	574	(9)	581	(9)
		608	(9)	632	(9)	714	(10)	721	(10)
		748	(10)	781	(10)	800	(10)	806	(10)
		818	(10)	824	(10)	832	(10)	858	(10)
		864	(10)	876	(10)	882	(10)	951	(11)
		958	(11)	985	(11)				
DS$ENDSUB	00000000-XR	1054	(11)	1099	(11)	1136	(11)	158	(4)
		188	(4)	2267	(24)	2298	(24)	2407	(24)
		252	(4)	434	(8)	507	(8)	584	(9)
		634	(9)	724	(10)	749	(10)	833	(10)
		883	(10)	961	(11)	986	(11)		
DS$ERRHARD	00000000-XR	1018	(11)	1037	(11)	1043	(11)	1052	(11)
		1077	(11)	1083	(11)	1089	(11)	1097	(11)
		1122	(11)	1128	(11)	1134	(11)	1196	(12)
		1215	(12)	1224	(12)	1230	(12)	1287	(14)
		1320	(14)	1336	(14)	1350	(14)	1366	(14)
		1431	(16)	1447	(16)	1454	(16)	1458	(16)
		147	(4)	1474	(16)	1482	(16)	154	(4)
		1553	(18)	1577	(18)	1587	(18)	1600	(18)
		1623	(18)	1633	(18)	1646	(18)	1667	(18)
		1675	(18)	1695	(18)	1703	(18)	1729	(18)
		1739	(18)	1747	(18)	1769	(18)	1777	(18)
		1792	(18)	1797	(18)	1819	(18)	1832	(18)
		1842	(18)	1855	(18)	186	(4)	1876	(18)
		1881	(18)	1939	(20)	1961	(20)	1970	(20)
		1986	(20)	2007	(20)	2016	(20)	2031	(20)
		2053	(20)	2062	(20)	2118	(22)	2136	(22)
		2155	(22)	2165	(22)	219	(4)	2256	(24)
		2263	(24)	2282	(24)	2296	(24)	2318	(24)
		232	(4)	2333	(24)	2341	(24)	2352	(24)

```
                                                        2368    (24)    2383    (24)    2397    (24)    2405    (24)
                                                        241     (4)     250     (4)     309     (6)     316     (6)
                                                        344     (6)     351     (6)     423     (8)     430     (8)
                                                        461     (8)     480     (8)     503     (8)     573     (9)
                                                        580     (9)     607     (9)     631     (9)     713     (10)
                                                        720     (10)    747     (10)    780     (10)    799     (10)
                                                        805     (10)    817     (10)    823     (10)    831     (10)
                                                        857     (10)    863     (10)    875     (10)    881     (10)
                                                        950     (11)    957     (11)    984     (11)
DS$ESCAPE                       00000000-XR             1217    (12)    156     (4)     221     (4)     2265    (24)
                                                        318     (6)     432     (8)     582     (9)     722     (10)
                                                        959     (11)
DS$K_BBC                        =0000001D    629   (9)
DS$K_BBCC                       =00000021    629   (9)
DS$K_BBCCI                      =00000023    629   (9)
DS$K_BBCS                       =0000001F    629   (9)
DS$K_BBS                        =0000001C    629   (9)
DS$K_BBSC                       =00000020    629   (9)
DS$K_BBSS                       =0000001E    629   (9)
DS$K_BBSSI                      =00000022    629   (9)
DS$K_BCC_M                      =0000001B    629   (9)
DS$K_BCS_M                      =0000001A    629   (9)
DS$K_BEQLU_B                    =00000005    629   (9)
DS$K_BEQLU_M                    =00000011    629   (9)
DS$K_BEQL_B                     =00000004    629   (9)
DS$K_BEQL_M                     =00000010    629   (9)
DS$K_BERROR                     =00000026    629   (9)
DS$K_BGEQU_B                    =00000007    629   (9)
DS$K_BGEQU_M                    =00000013    629   (9)
DS$K_BGEQ_B                     =00000006    629   (9)
DS$K_BGEQ_M                     =00000012    629   (9)
DS$K_BGTRU_B                    =00000009    629   (9)
DS$K_BGTRU_M                    =00000015    629   (9)
DS$K_BGTR_B                     =00000008    629   (9)
DS$K_BGTR_M                     =00000014    629   (9)
DS$K_BLBC                       =00000025    629   (9)
DS$K_BLBS                       =00000024    629   (9)
DS$K_BLEQU_B                    =00000003    629   (9)
DS$K_BLEQU_M                    =0000000F    629   (9)
DS$K_BLEQ_B                     =00000002    629   (9)
DS$K_BLEQ_M                     =0000000E    629   (9)
DS$K_BLSSU_B                    =00000001    629   (9)
DS$K_BLSSU_M                    =0000000D    629   (9)
DS$K_BLSS_B                     =00000000    629   (9)
DS$K_BLSS_M                     =0000000C    629   (9)
DS$K_BNEQU_B                    =0000000B    629   (9)
DS$K_BNEQU_M                    =00000017    629   (9)
DS$K_BNEQ_B                     =0000000A    629   (9)
DS$K_BNEQ_M                     =00000016    629   (9)
DS$K_BNERROR                    =00000027    629   (9)    #-629    (9)
DS$K_BVC_M                      =00000019    629   (9)
DS$K_BVS_M                      =00000018    629   (9)
DS$PRINTF                       00000000-XR             1279    (14)    1433    (16)    1460    (16)    1555    (18)
                                                        1941    (20)    2120    (22)    2123    (22)    2170    (22)
                                                        2244    (24)    2308    (24)    2356    (24)    2408    (24)
DS$WAITMS                       00000000-XR             1851    (18)    1995    (20)    324     (6)     353     (6)
```

```
DS$WAITUS                       00000000-XR                    1757    (18)     1807   (18)
DSA$GL_FLAGS                    0000FE00                       1419    (16)     1544   (18)      1930   (20)      2108   (22)
                                                               2233    (24)
DSA$V_OPER                     =0000000C                      #-1419   (16)    #-1544  (18)     #-1930  (20)     #-2108  (22)
                                                              #-2233   (24)
E.CS                            00000000-XR                   #-2159   (22)    #-2160  (22)     #-2161  (22)     #-2165  (22)
                                                              #-2345   (24)    #-2346  (24)     #-2347  (24)     #-2348  (24)
                                                              #-2352   (24)
E.MP                            00000000-XR                   #-1565   (18)    #-1596  (18)     #-1600  (18)     #-1611  (18)
                                                              #-1642   (18)    #-1646  (18)     #-1717  (18)     #-1743  (18)
                                                              #-1747   (18)    #-1787  (18)     #-1788  (18)     #-1792  (18)
                                                              #-1837   (18)    #-1838  (18)     #-1842  (18)     #-1864  (18)
                                                              #-1881   (18)    #-1981  (20)     #-1982  (20)     #-1986  (20)
                                                              #-2026   (20)    #-2027  (20)     #-2031  (20)     #-2335  (24)
                                                              #-2337   (24)    #-2341  (24)
ENV$M_DOMAIN                   =00000002      61       (2)
ENV$M_LEVEL                    =00000001      61       (2)
ENV$M_SUPER                    =000003FC      61       (2)
ENV$S_DOMAIN                   =00000001      61       (2)
ENV$S_LEVEL                    =00000001      61       (2)
ENV$S_SUPER                    =00000008      61       (2)
ENV$V_DOMAIN                   =00000001      61       (2)       61      (2)
ENV$V_LEVEL                    =00000000      61       (2)       61      (2)
ENV$V_SUPER                    =00000002      61       (2)
ENV$_CPU                       =00000000      61       (2)       61      (2)
ENV$_FUNCTIONAL                =00000000      61       (2)       61      (2)
ENV$_REPAIR                    =00000001      61       (2)       61      (2)
ENV$_SUPER                     =00000001      61       (2)
ENV$_SYSTEM                    =00000001      61       (2)       61      (2)
ERR_MSG                         00000000-XR                    1320    (14)      154   (4)       2263   (24)       316   (6)
                                                               430     (8)       580   (9)        720   (10)       957   (11)
EVENT_FLAGS                     00000000-XR                    1305    (14)
FMT_BADBLK                      00000000-XR                    1279    (14)
GETSTAT                        =00000004                      #-1440   (16)    #-1467  (16)     #-1570  (18)     #-1616  (18)
                                                              #-1660   (18)    #-1688  (18)     #-1722  (18)     #-1762  (18)
                                                              #-1812   (18)    #-1869  (18)     #-1954  (20)     #-2000  (20)
                                                              #-2046   (20)    #-2128  (22)     #-2129  (22)     #-2145  (22)
                                                              #-2148   (22)    #-2363  (24)
GET_LUN$A                       00000000-XR                    1182    (12)     1273   (14)       129   (4)       1417   (16)
                                                               1538    (18)     1924   (20)      2106   (22)      2231   (24)
                                                               288     (6)       402   (8)        550   (9)        690   (10)
                                                               927     (11)
GET_POSITION                    00000000-XR                    1221    (12)
GET_RANDOM                      00000000-XR                    1294    (14)
GET_SECLUN$A                    00000000-XR                    295     (6)
IDCBAR                         =00000004                      #-223    (4)
IDCCNT                         =00000008                      #-224    (4)
IDCCSR                         =00000000                      #-2130   (22)    #-2131  (22)     #-214   (4)      #-2149  (22)
                                                              #-225    (4)     #-227   (4)      #-321   (6)      #-323   (6)
                                                              #-335    (6)     #-337   (6)      #-339   (6)      #-812   (10)
                                                              #-870    (10)
IDCDAR                         =0000000C                      #-226    (4)     #-322   (6)      #-336   (6)
IDCMPR                         =00000010                      #-2126   (22)    #-2146  (22)     #-222   (4)
L$UNIT                          00000000-XR                   #-1018   (11)    #-1037  (11)     #-1043  (11)     #-1052  (11)
                                                              #-1077   (11)    #-1083  (11)     #-1089  (11)     #-1097  (11)
                                                              #-1122   (11)    #-1128  (11)     #-1134  (11)     #-1196  (12)
```

```
                                                      #-1215     (12)   #-1224     (12)   #-1230     (12)   #-1287     (14)
                                                      #-1320     (14)   #-1336     (14)   #-1350     (14)   #-1366     (14)
                                                      #-1431     (16)   #-1447     (16)   #-1454     (16)   #-1458     (16)
                                                      #-147      (4)    #-1474     (16)   #-1482     (16)   #-154      (4)
                                                      #-1553     (18)   #-1577     (18)   #-1587     (18)   #-1600     (18)
                                                      #-1623     (18)   #-1633     (18)   #-1646     (18)   #-1667     (18)
                                                      #-1675     (18)   #-1695     (18)   #-1703     (18)   #-1729     (18)
                                                      #-1739     (18)   #-1747     (18)   #-1769     (18)   #-1777     (18)
                                                      #-1792     (18)   #-1797     (18)   #-1819     (18)   #-1832     (18)
                                                      #-1842     (18)   #-1855     (18)   #-186      (4)    #-1876     (18)
                                                      #-1881     (18)   #-1939     (20)   #-1961     (20)   #-1970     (20)
                                                      #-1986     (20)   #-2007     (20)   #-2016     (20)   #-2031     (20)
                                                      #-2053     (20)   #-2062     (20)   #-2118     (22)   #-2136     (22)
                                                      #-2155     (22)   #-2165     (22)   #-219      (4)    #-2256     (24)
                                                      #-2263     (24)   #-2282     (24)   #-2296     (24)   #-2318     (24)
                                                      #-232      (4)    #-2333     (24)   #-2341     (24)   #-2352     (24)
                                                      #-2368     (24)   #-2383     (24)   #-2397     (24)   #-2405     (24)
                                                      #-241      (4)    #-250      (4)    #-309      (6)    #-316      (6)
                                                      #-344      (6)    #-351      (6)    #-423      (8)    #-430      (8)
                                                      #-461      (8)    #-480      (8)    #-503      (8)    #-573      (9)
                                                      #-580      (9)    #-607      (9)    #-631      (9)    #-713      (10)
                                                      #-720      (10)   #-747      (10)   #-780      (10)   #-799      (10)
                                                      #-805      (10)   #-817      (10)   #-823      (10)   #-831      (10)
                                                      #-857      (10)   #-863      (10)   #-875      (10)   #-881      (10)
                                                      #-950      (11)   #-957      (11)   #-984      (11)
LUN0                          =00000000                 1174     (12)     121      (4)      1261     (14)     1409     (16)
                                                        1530     (18)     1916     (20)     2098     (22)     2223     (24)
                                                        280      (6)      394      (8)      542      (9)      682      (10)
                                                        919      (11)
LUN1                          =00000001                 1175     (12)     122      (4)      1262     (14)     1410     (16)
                                                        1531     (18)     1917     (20)     2099     (22)     2224     (24)
                                                        281      (6)      395      (8)      543      (9)      683      (10)
                                                        920      (11)
LUN2                          =00000002                 1176     (12)     123      (4)      1263     (14)     1411     (16)
                                                        1532     (18)     1918     (20)     2100     (22)     2225     (24)
                                                        282      (6)      396      (8)      544      (9)      684      (10)
                                                        921      (11)
LUN3                          =00000003                 1177     (12)     124      (4)      1264     (14)     1412     (16)
                                                        1533     (18)     1919     (20)     2101     (22)     2226     (24)
                                                        283      (6)      397      (8)      545      (9)      685      (10)
                                                        922      (11)
MANUAL                        =00000001     67     (2)  1415     (16)     1536     (18)     1922     (20)     2104     (22)
                                                        2229     (24)
MPR_02$M_BH                   =00000008               #-1564     (18)   #-1610     (18)   #-1716     (18)   #-1786     (18)
                                                      #-1836     (18)   #-1864     (18)   #-1980     (20)   #-2024     (20)
MPR_02$M_CO                   =00000020               #-1476     (16)
MPR_02$M_DT                   =00000080               #-1564     (18)   #-1610     (18)   #-1715     (18)   #-1785     (18)
                                                      #-1835     (18)   #-1863     (18)   #-1979     (20)   #-2024     (20)
MPR_02$M_HO                   =00000010               #-1785     (18)   #-1836     (18)   #-1863     (18)   #-1979     (20)
MPR_02$M_STA                  =00000001               #-1565     (18)   #-1581     (18)   #-1717     (18)   #-1828     (18)
                                                      #-1837     (18)   #-1864     (18)   #-2012     (20)   #-2026     (20)
MPR_02$M_STB                  =00000002               #-1611     (18)   #-1628     (18)   #-1716     (18)   #-1965     (20)
                                                      #-1981     (20)   #-2011     (20)   #-2025     (20)
MPR_02$M_STC                  =00000004               #-1773     (18)   #-1787     (18)   #-1828     (18)   #-1837     (18)
                                                      #-1864     (18)   #-1965     (20)   #-1980     (20)   #-2011     (20)
                                                      #-2025     (20)
```

```
MPR_02$M_VC                  =00000200              #-1742   (18)   #-1786   (18)   #-1835   (18)
MPR_02$M_WL                  =00002000              #-1476   (16)
MPR_02$V_BH                  =00000003              #-1671   (18)   #-1699   (18)
MPR_02$V_VC                  =00000009              #-1878   (18)
MS100_DAT_DST                 00000000-XR             2405   (24)
MS113_UWRTERR                 00000000-XR             1018   (11)     780    (10)
MS114A_UWCKERR                00000000-XR             857    (10)     863    (10)
MS114_URDERR                  00000000-XR             799    (10)     805    (10)
MS115A_WKRST                  00000000-XR             875    (10)     881    (10)
MS115_RDRST                   00000000-XR             817    (10)     823    (10)
MS116A_AUTOCMP                00000000-XR             1097   (11)
MS116_SKPCMP                  00000000-XR             831    (10)
MS117_AUTORD                  00000000-XR             1077   (11)     1083   (11)     1089   (11)
MS117_AUTOWCK                 00000000-XR             1122   (11)     1128   (11)     1134   (11)
MS117_AUTOWRT                 00000000-XR             1037   (11)     1043   (11)     1052   (11)
MS118_SEEKIDC                 00000000-XR             1215   (12)
MS119_IDCPOS                  00000000-XR             1230   (12)
MS83A_ST4_DE                  00000000-XR             1797   (18)
MSG14_SYSINIT                 00000000-XR             1196   (12)     1287   (14)     1431   (16)     147    (4)
                                                      1553   (18)     1939   (20)     2118   (22)     2256   (24)
                                                      2282   (24)     2318   (24)     2383   (24)     309    (6)
                                                      423    (8)      573    (9)      713    (10)     950    (11)
MSG16_WRTCHK                  00000000-XR             1350   (14)
MSG17_GETSTAT                 00000000-XR             1447   (16)     1474   (16)     1577   (18)     1623   (18)
                                                      1667   (18)     1695   (18)     1729   (18)     1769   (18)
                                                      1819   (18)     1876   (18)     1961   (20)     2007   (20)
                                                      2053   (20)
MSG18B_SEEK                   00000000-XR             351    (6)
MSG19_READH                   00000000-XR             1224   (12)
MSG20A_FORMAT                 00000000-XR             607    (9)      747    (10)     984    (11)
MSG20C_WRITE                  00000000-XR             631    (9)
MSG20_WRITE                   00000000-XR             1336   (14)     186    (4)      219    (4)      2296   (24)
MSG20_WRT_BND                 00000000-XR             461    (8)
MSG21_READ                    00000000-XR             1366   (14)     2397   (24)     480    (8)
MSG22_READNHC                 00000000-XR             232    (4)      241    (4)
MSG23_CS_INC                  00000000-XR           #-1043   (11)   #-1083   (11)   #-1128   (11)   #-2165   (22)
                                                    #-2352   (24)   #-805    (10)   #-823    (10)   #-863    (10)
                                                    #-881    (10)
MSG28_DA_INC                  00000000-XR           #-1052   (11)   #-1089   (11)   #-1134   (11)
MSG30A_CRDY                   00000000-XR             344    (6)
MSG47_DS_INC                  00000000-XR           #-2341   (24)
MSG54_DA_UPD                  00000000-XR           #-241    (4)
MSG55_DAT_BND                 00000000-XR             503    (8)
MSG69_DAT_NHC                 00000000-XR             250    (4)
MSG73_GS_UND                  00000000-XR             1454   (16)     1482   (16)
MSG74_DRDY_NC                 00000000-XR             1458   (16)
MSG75_NO_ST1                  00000000-XR             1587   (18)
MSG76_ST1_BS                  00000000-XR             1600   (18)
MSG77_NO_ST2                  00000000-XR             1633   (18)
MSG77_NO_ST3                  00000000-XR             1739   (18)
MSG78_ST2_BS                  00000000-XR             1646   (18)
MSG79_BH_NC                   00000000-XR             1675   (18)
MSG80_BH_NF                   00000000-XR             1703   (18)
MSG81_ST3_BS                  00000000-XR             1747   (18)
MSG82_NO_ST4                  00000000-XR             1777   (18)
MSG83_ST4_BS                  00000000-XR             1792   (18)
```

```
MSG84_NO_ST5                    00000000-XR                  1832   (18)
MSG85_ST5_BS                    00000000-XR                  1842   (18)
MSG86_DRDY_NS                   00000000-XR                  1855   (18)
MSG87_VC_NC                     00000000-XR                  1881   (18)
MSG88_NO_ST6                    00000000-XR                  1970   (20)
MSG89_ST6_BS                    00000000-XR                  1986   (20)
MSG90_NO_ST7                    00000000-XR                  2016   (20)
MSG91_ST7_BS                    00000000-XR                  2031   (20)
MSG92_NO_ST0                    00000000-XR                  2062   (20)
MSG93_RELV_UP                   00000000-XR                  2136   (22)
MSG97_GS_NXD                    00000000-XR                  2155   (22)     2165   (22)
MSG98_WRT_WL                    00000000-XR                  2333   (24)     2341   (24)      2352   (24)
MSG99_RST_WGE                   00000000-XR                  2368   (24)
MSG_032                         00000002-R   73     (2)      127    (4)
MSG_033                         00000002-R   254    (4)      286    (6)
MSG_034                         00000002-R   355    (6)      400    (8)
MSG_035                         00000002-R   509    (8)      548    (9)
MSG_036                         00000002-R   636    (9)      688    (10)
MSG_037                         00000002-R   885    (10)     925    (11)
MSG_038                         00000002-R   1138   (11)     1180   (12)
MSG_039                         00000002-R   1233   (12)     1267   (14)
MSG_040                         00000002-R   1372   (14)     1415   (16)
MSG_041                         00000002-R   1485   (16)     1536   (18)
MSG_042                         00000002-R   1884   (18)     1922   (20)
MSG_043                         00000002-R   2067   (20)     2104   (22)
MSG_044                         00000002-R   2174   (22)     2229   (24)
NOCHK$M_ATTN                   =00000080                   #-1332   (14)   #-1346   (14)    #-1362   (14)
NOCHK$M_CERR                   =00000002                   #-1033   (11)   #-1073   (11)    #-1118   (11)    #-2150   (22)
                                                           #-795    (10)   #-813    (10)    #-853    (10)    #-871    (10)
NOCHK$M_DERR                   =00000008                   #-1426   (16)   #-1442   (16)    #-1469   (16)    #-1548   (18)
                                                           #-1572   (18)   #-1618   (18)    #-1662   (18)    #-1690   (18)
                                                           #-1724   (18)   #-1764   (18)    #-1814   (18)    #-1871   (18)
                                                           #-1934   (20)   #-1956   (20)    #-2002   (20)    #-2048   (20)
                                                           #-2113   (22)   #-2151   (22)    #-2329   (24)
NOCHK$M_DRDY                   =00000004                   #-1426   (16)   #-1442   (16)    #-1469   (16)    #-1548   (18)
                                                           #-1572   (18)   #-1618   (18)    #-1662   (18)    #-1690   (18)
                                                           #-1724   (18)   #-1764   (18)    #-1814   (18)    #-1871   (18)
                                                           #-1934   (20)   #-1956   (20)    #-2002   (20)    #-2048   (20)
                                                           #-2113   (22)   #-2132   (22)    #-2150   (22)
NOCHK$M_DSTA                   =00000010                   #-795    (10)   #-853    (10)
NO_OP                          =00000000                   #-602    (9)    #-742    (10)    #-979    (11)
OFST_AVGTIM                    =00000080     71     (2)
OFST_CSRTYP                    =00000058     71     (2)    #-2347   (24)
OFST_DEVMEM                    =0000001C     71     (2)
OFST_DEVNAM                    =00000020     71     (2)
OFST_DRVERR                    =0000004C     71     (2)
OFST_DRVRST                    =00000050     71     (2)    #-1865   (18)   #-2359   (24)
OFST_DRVTYP                    =00000000     71     (2)    #-1185   (12)   #-1303   (14)    #-1541   (18)    #-1927   (20)
                                                           #-2354   (24)   #-407    (8)     #-494    (8)     #-554    (9)
                                                           #-694    (10)   #-931    (11)
OFST_FMTLEN                    =00000030     71     (2)
OFST_GTSCMD                    =00000034     71     (2)    #-1423   (16)   #-1546   (18)    #-1932   (20)    #-2111   (22)
OFST_HDBITS                    =00000044     71     (2)
OFST_HDRDNC                    =00000054     71     (2)
OFST_HDRTMP                    =00000048     71     (2)
OFST_HIGRPM                    =00000074     71     (2)
```

```
OFST_HMBSEC                       =0000005C      71      (2)
OFST_IGBCYL                       =00000018      71      (2)
OFST_LOWRPM                       =00000070      71      (2)
OFST_LSTCYL                       =00000008      71      (2)     #-1189    (12)    #-1280    (14)    #-293     (6)     #-300     (6)
OFST_LSTSEC                       =00000010      71      (2)     #-1283    (14)    #-134     (4)
OFST_LSTTRK                       =0000000C      71      (2)     #-1282    (14)
OFST_MAXTIM                       =0000007C      71      (2)
OFST_MPBITS                       =00000040      71      (2)     #-1424    (16)    #-1547    (18)    #-1933    (20)    #-2242    (24)
OFST_NMLSTS                       =00000038      71      (2)
OFST_ONETIM                       =00000078      71      (2)
OFST_OP1STS                       =00000060      71      (2)     #-1422    (16)
OFST_PHYU                         =00000004      71      (2)
OFST_RANCYL                       =0000006C      71      (2)     #-1219    (12)
OFST_SECLEN                       =00000028      71      (2)     #-1005    (11)    #-1322    (14)    #-1328    (14)    #-133     (4)
                                                                  #-1342    (14)    #-1353    (14)    #-1358    (14)    #-2237    (24)
                                                                  #-292     (6)     #-411     (8)     #-619     (9)     #-624     (9)
                                                                  #-767     (10)
OFST_SEKSTS                       =0000003C      71      (2)
OFST_STSDNC                       =00000064      71      (2)     #-1425    (16)    #-2241    (24)
OFST_TRKLEN                       =0000002C      71      (2)
OFST_TSTCYL                       =00000014      71      (2)     #-1306    (14)    #-132     (4)     #-2236    (24)    #-412     (8)
                                                                  #-557     (9)     #-593     (9)     #-697     (10)    #-733     (10)
                                                                  #-934     (11)    #-970     (11)
OFST_UNIT                         =00000004                       #-1182    (12)    #-1273    (14)    #-129     (4)     #-1417    (16)
                                                                  #-1538    (18)    #-1924    (20)    #-2106    (22)    #-2231    (24)
                                                                  #-288     (6)     #-295     (6)     #-402     (8)     #-550     (9)
                                                                  #-690     (10)    #-927     (11)
OFST_WTPSTS                       =00000068      71      (2)     #-2240    (24)
OPR11_ORIGIN                       00000000-XR                       2244   (24)
OPR12_WRTLOCK                      00000000-XR                       2308   (24)
OPR13A_RST_FLT                     00000000-XR                       2356   (24)
OPR13_RST_WL                       00000000-XR                       2408   (24)
OPR1_UNLOAD                        00000000-XR                       1433   (16)
OPR2_CLOSE                         00000000-XR                       1460   (16)
OPR3_START                         00000000-XR                       1555   (18)
OPR5_UNLOAD                        00000000-XR                       1941   (20)
OPR6_REMOVE                        00000000-XR                       2120   (22)
OPR7_INSERT                        00000000-XR                       2123   (22)
OPR8_SETUP                         00000000-XR                       2170   (22)
OPR_RETURN                         00000000-XR                       1435   (16)       1462  (16)       2122  (22)       2125  (22)
                                                                     2172   (22)       2246  (24)       2310  (24)       2358  (24)
                                                                     2410   (24)
P$UNIT                             00000000-XR                    #-1023    (11)    #-1063    (11)    #-1108    (11)    #-205     (4)
                                                                  #-2346    (24)    #-785     (10)    #-808     (10)    #-811     (10)
                                                                  #-843     (10)    #-866     (10)    #-869     (10)
PAR$M_ATDEF                       =00000010      66      (2)
PAR$M_ATHI                        =00000008      66      (2)
PAR$M_ATLO                        =00000004      66      (2)
PAR$M_NODEF                       =00000002      66      (2)
PAR$V_ATDEF                       =00000004      66      (2)
PAR$V_ATHI                        =00000003      66      (2)
PAR$V_ATLO                        =00000002      66      (2)
PAR$V_NODEF                       =00000001      66      (2)
PAR$_BIN                          =00000002      66      (2)
PAR$_DEC                          =0000000A      66      (2)
PAR$_HEX                          =00000010      66      (2)
```

```
PAR$_NO              =00000000    66   (2)
PAR$_OCT             =00000008    66   (2)
PAR$_YES             =00000001    66   (2)
POSITION_HDS          00000000-XR           1315    (14)    151     (4)     2260    (24)    313     (6)
                                            348     (6)     427     (8)     577     (9)     717     (10)
                                            954     (11)
PRINT_DATA_ERR        00000000-XR           1097    (11)    2405    (24)    250     (4)     503     (8)
                                            831     (10)
PRINT_REG             00000000-XR           1018    (11)    1037    (11)    1077    (11)    1122    (11)
                                            1196    (12)    1215    (12)    1224    (12)    1287    (14)
                                            1320    (14)    1336    (14)    1350    (14)    1366    (14)
                                            1431    (16)    1447    (16)    1458    (16)    147     (4)
                                            1474    (16)    154     (4)     1553    (18)    1577    (18)
                                            1587    (18)    1623    (18)    1633    (18)    1667    (18)
                                            1675    (18)    1695    (18)    1703    (18)    1729    (18)
                                            1739    (18)    1769    (18)    1777    (18)    1797    (18)
                                            1819    (18)    1832    (18)    1855    (18)    186     (4)
                                            1876    (18)    1939    (20)    1961    (20)    1970    (20)
                                            2007    (20)    2016    (20)    2053    (20)    2062    (20)
                                            2118    (22)    2136    (22)    2155    (22)    219     (4)
                                            2256    (24)    2263    (24)    2282    (24)    2296    (24)
                                            2318    (24)    232     (4)     2333    (24)    2368    (24)
                                            2383    (24)    2397    (24)    309     (6)     316     (6)
                                            344     (6)     351     (6)     423     (8)     430     (8)
                                            461     (8)     480     (8)     573     (9)     580     (9)
                                            607     (9)     631     (9)     713     (10)    720     (10)
                                            747     (10)    780     (10)    799     (10)    817     (10)
                                            857     (10)    875     (10)    950     (11)    957     (11)
                                            984     (11)
PRT$M_ALL            =00000007             #-1318   (14)
R80$T                =00000002             #-1303   (14)   #-554    (9)    #-694    (10)   #-931    (11)
RD_BAD_BLK            00000000-XR           1277    (14)
RD_IDCREGS            00000000-XR           1852    (18)    217     (4)     342     (6)
RD_NHC               =0000000E             #-204    (4)    #-228    (4)
READ                 =0000000C             #-1062   (11)   #-1071   (11)   #-1116   (11)   #-1360   (14)
                                           #-2392   (24)   #-475    (8)    #-784    (10)   #-793    (10)
                                           #-807    (10)   #-810    (10)
READ_BUFFER           00000000-XR           1065    (11)   #-1093   (11)   #-171    (4)     200     (4)
                                           #-2277   (24)    2386    (24)   #-2400   (24)   #-245    (4)
                                           #-445    (8)     469     (8)    #-498    (8)     787     (10)
                                           #-826    (10)
REG_ERRPT             00000000-XR           1043    (11)    1052    (11)    1083    (11)    1089    (11)
                                            1128    (11)    1134    (11)    1230    (12)    1454    (16)
                                            1482    (16)    1600    (18)    1646    (18)    1747    (18)
                                            1792    (18)    1842    (18)    1881    (18)    1986    (20)
                                            2031    (20)    2165    (22)    2341    (24)    2352    (24)
                                            241     (4)     805     (10)    823     (10)    863     (10)
                                            881     (10)
RGLIST                00000000-XR          #-1318   (14)
RL02$T               =00000001             #-1185   (12)   #-1541   (18)   #-1927   (20)   #-2354   (24)
                                           #-407    (8)    #-494    (8)
SEED                  00000000-XR          #-166    (4)    #-169    (4)    #-170    (4)
SEEK                 =00000006             #-1210   (12)   #-320    (6)    #-334    (6)
SEP_FUNCTIONAL       =00000002    61   (2)
SEP_REPAIR           =00000003    61   (2)  61      (2)
SIZ...               =00000001    66   (2)  61      (2)     65      (2)     66      (2)
```

```
SKIP_BUFFER       00000000-XR            1025   (11)  #-1092  (11)   1110  (11)  #-825   (10)
                                          845   (10)
START_CMD         00000000-XR            1014   (11)   1032   (11)   1072  (11)   1117   (11)
                                         1211   (12)   1331   (14)   1345  (14)   1361   (14)
                                         1441   (16)   1468   (16)   1571  (18)   1617   (18)
                                         1661   (18)   1689   (18)   1723  (18)   1763   (18)
                                         1813   (18)   182    (4)    1870  (18)   1955   (20)
                                         2001   (20)   2047   (20)   212   (4)    2292   (24)
                                         2328   (24)   2364   (24)   2393  (24)   457    (8)
                                         476    (8)    603    (9)    627   (9)    743    (10)
                                         776    (10)   794    (10)   852   (10)   980    (11)
STATUS_CHK        00000000-XR            1015   (11)   1034   (11)   1074  (11)   1119   (11)
                                         1212   (12)   1333   (14)   1347  (14)   1363   (14)
                                         1444   (16)   1471   (16)   1574  (18)   1620   (18)
                                         1664   (18)   1692   (18)   1726  (18)   1766   (18)
                                         1816   (18)   183    (4)    1873  (18)   1958   (20)
                                         2004   (20)   2050   (20)   2133  (22)   2152   (22)
                                         229    (4)    2293   (24)   2330  (24)   2365   (24)
                                         2394   (24)   458    (8)    477   (8)    604    (9)
                                         628    (9)    744    (10)   777   (10)   796    (10)
                                         814    (10)   854    (10)   872   (10)   981    (11)
SUBSYS_INIT       00000000-XR            1193   (12)   1284   (14)   1428  (16)   144    (4)
                                         1550   (18)   1936   (20)   2115  (22)   2253   (24)
                                         2279   (24)   2315   (24)   2380  (24)   306    (6)
                                         420    (8)    570    (9)    710   (10)   947    (11)
SYS$CANTIM        00000000-XR            1588   (18)   1601   (18)   1624  (18)   1634   (18)
                                         1647   (18)   1668   (18)   1676  (18)   1696   (18)
                                         1704   (18)   1712   (18)   1730  (18)   1740   (18)
                                         1973   (20)   2065   (20)
SYS$READEF        00000000-XR            1293   (14)   1584   (18)   1630  (18)   1672   (18)
                                         1700   (18)   1736   (18)   1967  (20)   2059   (20)
SYS$SETIMR        00000000-XR            1557   (18)   1714   (18)   1949  (20)   2041   (20)
S_DRIVE_TBL       00000000-XR           #-299   (6)
S_P$UNIT          00000000-XR           #-320   (6)   #-334   (6)
T.CS              00000000-XR           #-1039  (11)  #-1043  (11)  #-1079  (11)  #-1083  (11)
                                        #-1124  (11)  #-1128  (11)   1456   (16)   1795   (18)
                                         1853   (18)  #-2161  (22)  #-2165  (22)  #-2348  (24)
                                        #-2352  (24)  #-801   (10)  #-805   (10)  #-819   (10)
                                        #-823   (10)  #-859   (10)  #-863   (10)  #-877   (10)
                                        #-881   (10)
T.DA              00000000-XR           #-1048  (11)  #-1052  (11)  #-1085  (11)  #-1089  (11)
                                        #-1130  (11)  #-1134  (11)  #-237   (4)   #-241   (4)
                                        #-243   (4)
T.MP              00000000-XR           #-1449  (16)  #-1450  (16)  #-1454  (16)  #-1477  (16)
                                        #-1478  (16)  #-1482  (16)   1580   (18)  #-1596  (18)
                                        #-1600  (18)   1627   (18)  #-1642  (18)  #-1646  (18)
                                         1671   (18)   1699   (18)   1733   (18)  #-1742  (18)
                                        #-1743  (18)  #-1747  (18)   1772   (18)  #-1788  (18)
                                        #-1792  (18)   1827   (18)  #-1838  (18)  #-1842  (18)
                                         1878   (18)  #-1881  (18)   1964   (20)  #-1982  (20)
                                        #-1986  (20)   2010   (20)  #-2027  (20)  #-2031  (20)
                                         2056   (20)  #-2336  (24)  #-2337  (24)  #-2341  (24)
T32_S1            00000082-R     138    (4)
T32_S1_X          00000100-R     158    (4)
T32_S2            0000010B-R     159    (4)
T32_S2_X          000001A4-R     188    (4)
```

```
T32_S3                         000001AF-R   189    (4)
T32_S3_X                       0000031D-R   252    (4)
T34_S1                         00000073-R   414    (8)
T34_S1_X                       000000F1-R   434    (8)
T34_S2                         000000FC-R   435    (8)
T34_S2_X                       0000026B-R   507    (8)
T35_S1                         00000095-R   564    (9)
T35_S1_X                       00000113-R   584    (9)
T35_S2                         0000011E-R   585    (9)
T35_S2_X                       00000200-R   634    (9)
T36_S1                         00000091-R   704    (10)
T36_S1_X                       0000010F-R   724    (10)
T36_S2                         0000011A-R   725    (10)
T36_S2_X                       00000187-R   749    (10)
T36_S3                         00000192-R   750    (10)
T36_S3_X                       000003A0-R   833    (10)
T36_S4                         000003AB-R   834    (10)
T36_S4_X                       000004EB-R   883    (10)
T37_S1                         00000095-R   941    (11)
T37_S1_X                       00000113-R   961    (11)
T37_S2                         0000011E-R   962    (11)
T37_S2_X                       0000018B-R   986    (11)
T37_S3                         00000196-R   987    (11)
T37_S3_X                       0000031F-R   1054   (11)
T37_S4                         0000032A-R   1055   (11)
T37_S4_X                       00000455-R   1099   (11)
T37_S5                         00000460-R   1100   (11)
T37_S5_X                       0000054B-R   1136   (11)
T44_S1                         000000BA-R   2247   (24)
T44_S1_X                       00000138-R   2267   (24)
T44_S2                         00000143-R   2268   (24)
T44_S2_X                       000001EE-R   2298   (24)
T44_S3                         000001F9-R   2299   (24)
T44_S3_X                       000004E0-R   2407   (24)
TEST_032                       0000004C-R   127    (4)     127     (4)
TEST_032_X                     0000032B-R   253    (4)    #-130    (4)     156     (4)     221     (4)
TEST_033                       0000003B-R   286    (6)     286     (6)
TEST_033_X                     000001AF-R   354    (6)    #-289    (6)    #-298    (6)     318     (6)
TEST_034                       0000003B-R   400    (8)     400     (8)
TEST_034_X                     00000279-R   508    (8)    #-403    (8)     432     (8)
TEST_035                       00000048-R   548    (9)     548     (9)
TEST_035_X                     0000020E-R   635    (9)    #-551    (9)    #-552    (9)    #-556    (9)     582     (9)
TEST_036                       00000044-R   688    (10)    688     (10)
TEST_036_X                     000004F9-R   884    (10)   #-691    (10)   #-692    (10)   #-696    (10)    722     (10)
TEST_037                       00000048-R   925    (11)    925     (11)
TEST_037_X                     00000559-R   1137   (11)   #-928    (11)   #-929    (11)   #-933    (11)    959     (11)
TEST_038                       00000044-R   1180   (12)    1180    (12)
TEST_038_X                     00000198-R   1232   (12)   #-1183   (12)   #-1187   (12)    1217    (12)
TEST_039                       00000040-R   1267   (14)    1267    (14)
TEST_039_X                     000002B3-R   1371   (14)   #-1275   (14)
TEST_040                       0000003C-R   1415   (16)    1415    (16)
TEST_040_X                     00000255-R   1484   (16)   #-1418   (16)   #-1420   (16)
TEST_041                       00000040-R   1536   (18)    1536    (18)
TEST_041_X                     0000083E-R   1883   (18)   #-1539   (18)   #-1543   (18)   #-1545   (18)   #-1579   (18)
                                                          #-1590   (18)   #-1603   (18)   #-1626   (18)   #-1636   (18)
                                                          #-1649   (18)   #-1670   (18)   #-1678   (18)   #-1698   (18)
```

```
                                                  #-1706   (18)   #-1732   (18)   #-1749   (18)   #-1771   (18)
                                                  #-1779   (18)   #-1794   (18)   #-1799   (18)   #-1821   (18)
                                                  #-1834   (18)   #-1844   (18)
TEST_042                 00000044-R   1922   (20)    1922   (20)
TEST_042_X               00000391-R   2066   (20) #-1925   (20)   #-1929   (20)   #-1931   (20)   #-1963   (20)
                                                  #-1972   (20)   #-1988   (20)   #-2009   (20)   #-2018   (20)
                                                  #-2033   (20)   #-2055   (20)   #-2064   (20)
TEST_043                 00000038-R   2104   (22)    2104   (22)
TEST_043_X               0000024B-R   2173   (22) #-2107   (22)   #-2109   (22)
TEST_044                 0000003C-R   2229   (24)    2229   (24)
TEST_044_X               00000520-R   2411   (24) #-2232   (24)   #-2234   (24)     2265   (24)
TIME_OUT                 00000000-XR                 1584   (18)     1585   (18)     1630   (18)     1631   (18)
                                                     1672   (18)     1673   (18)     1700   (18)     1701   (18)
                                                     1736   (18)     1737   (18)     1967   (20)     1968   (20)
                                                     2059   (20)     2060   (20)
WORD_COUNT               00000000-XR              #-1091   (11)   #-2401   (24)   #-246    (4)    #-493    (8)
                                                  #-496    (8)    #-499    (8)    #-827    (10)
WRITE                   =0000000A                 #-1013   (11)   #-1021   (11)   #-1031   (11)   #-1330   (14)
                                                  #-181    (4)    #-211    (4)    #-2291   (24)   #-2327   (24)
                                                  #-456    (8)    #-626    (9)    #-775    (10)
WRITE_BUFFER             00000000-XR              #-1000   (11)   #-1003   (11)     1007   (11)   #-1270   (14)
                                                     1323   (14)     1354   (14)   #-170    (4)      174    (4)
                                                  #-2276   (24)     2285   (24)   #-2313   (24)     2321   (24)
                                                  #-2378   (24)   #-2399   (24)   #-244    (4)    #-444    (8)
                                                      448   (8)    #-489    (8)       491   (8)    #-497    (8)
                                                      596   (9)       620   (9)       736   (10)   #-762    (10)
                                                  #-765    (10)      769   (10)      973   (11)
WRTCHK                  =00000002                 #-1107   (11)   #-1344   (14)   #-842    (10)   #-851    (10)
                                                  #-865    (10)   #-868    (10)
```

```
                                     +--------------------------+
                                     ! Macros Cross Reference !
                                     +--------------------------+

MACRO            SIZE          DEFINITION         REFERENCES...
-----            ----          ----------         -------------
$ASNPUSH         1             1557    (18)       1557  (18)      1714  (18)      1949  (20)      2041  (20)
$CANTIM_S        1             1588    (18)       1588  (18)      1601  (18)      1624  (18)      1634  (18)      1647  (18)
                                                  1668  (18)      1676  (18)      1696  (18)      1704  (18)      1712  (18)
                                                  1730  (18)      1740  (18)      1973  (20)      2065  (20)
$D1_ABTEST       1             345     (6)        345   (6)
$D1_BDATA        1             120     (4)        1173  (12)      120   (4)       1260  (14)      1408  (16)      1529  (18)
                                                  1915  (20)      2097  (22)      2222  (24)      279   (6)       393   (8)
                                                  541   (9)       681   (10)      918   (11)
$D1_BSUBT        1             138     (4)        1055  (11)      1100  (11)      138   (4)       159   (4)       189   (4)
                                                  2247  (24)      2268  (24)      2299  (24)      414   (8)       435   (8)
                                                  564   (9)       585   (9)       704   (10)      725   (10)      750   (10)
                                                  834   (10)      941   (11)      962   (11)      987   (11)
$D1_BTEST        1             127     (4)        1180  (12)      1267  (14)      127   (4)       1415  (16)      1536  (18)
                                                  1922  (20)      2104  (22)      2229  (24)      286   (6)       400   (8)
                                                  548   (9)       688   (10)      925   (11)
$D1_ERR          1             147     (4)        1018  (11)      1037  (11)      1043  (11)      1052  (11)      1077  (11)
                                                  1083  (11)      1089  (11)      1097  (11)      1122  (11)      1128  (11)
                                                  1134  (11)      1196  (12)      1215  (12)      1224  (12)      1230  (12)
                                                  1287  (14)      1320  (14)      1336  (14)      1350  (14)      1366  (14)
                                                  1431  (16)      1447  (16)      1454  (16)      1458  (16)      147   (4)
                                                  1474  (16)      1482  (16)      154   (4)       1553  (18)      1577  (18)
                                                  1587  (18)      1600  (18)      1623  (18)      1633  (18)      1646  (18)
                                                  1667  (18)      1675  (18)      1695  (18)      1703  (18)      1729  (18)
                                                  1739  (18)      1747  (18)      1769  (18)      1777  (18)      1792  (18)
                                                  1797  (18)      1819  (18)      1832  (18)      1842  (18)      1855  (18)
                                                  186   (4)       1876  (18)      1881  (18)      1939  (20)      1961  (20)
                                                  1970  (20)      1986  (20)      2007  (20)      2016  (20)      2031  (20)
                                                  2053  (20)      2062  (20)      2118  (22)      2136  (22)      2155  (22)
                                                  2165  (22)      219   (4)       2256  (24)      2263  (24)      2282  (24)
                                                  2296  (24)      2318  (24)      232   (4)       2333  (24)      2341  (24)
                                                  2352  (24)      2368  (24)      2383  (24)      2397  (24)      2405  (24)
                                                  241   (4)       250   (4)       309   (6)       316   (6)       344   (6)
                                                  351   (6)       423   (8)       430   (8)       461   (8)       480   (8)
                                                  503   (8)       573   (9)       580   (9)       607   (9)       631   (9)
                                                  713   (10)      720   (10)      747   (10)      780   (10)      799   (10)
                                                  805   (10)      817   (10)      823   (10)      831   (10)      857   (10)
                                                  863   (10)      875   (10)      881   (10)      950   (11)      957   (11)
                                                  984   (11)
$D1_ERR_S        1             147     (4)        1018  (11)      1037  (11)      1043  (11)      1052  (11)      1077  (11)
                                                  1083  (11)      1089  (11)      1097  (11)      1122  (11)      1128  (11)
                                                  1134  (11)      1196  (12)      1215  (12)      1224  (12)      1230  (12)
                                                  1287  (14)      1320  (14)      1336  (14)      1350  (14)      1366  (14)
                                                  1431  (16)      1447  (16)      1454  (16)      1458  (16)      147   (4)
                                                  1474  (16)      1482  (16)      154   (4)       1553  (18)      1577  (18)
                                                  1587  (18)      1600  (18)      1623  (18)      1633  (18)      1646  (18)
                                                  1667  (18)      1675  (18)      1695  (18)      1703  (18)      1729  (18)
                                                  1739  (18)      1747  (18)      1769  (18)      1777  (18)      1792  (18)
                                                  1797  (18)      1819  (18)      1832  (18)      1842  (18)      1855  (18)
                                                  186   (4)       1876  (18)      1881  (18)      1939  (20)      1961  (20)
```

```
                                                 1970   (20)       1986   (20)       2007   (20)       2016   (20)       2031   (20)
                                                 2053   (20)       2062   (20)       2118   (22)       2136   (22)       2155   (22)
                                                 2165   (22)       219    (4)        2256   (24)       2263   (24)       2282   (24)
                                                 2296   (24)       2318   (24)       232    (4)        2333   (24)       2341   (24)
                                                 2352   (24)       2368   (24)       2383   (24)       2397   (24)       2405   (24)
                                                 241    (4)        250    (4)        309    (6)        316    (6)        344    (6)
                                                 351    (6)        423    (8)        430    (8)        461    (8)        480    (8)
                                                 503    (8)        573    (9)        580    (9)        607    (9)        631    (9)
                                                 713    (10)       720    (10)       747    (10)       780    (10)       799    (10)
                                                 805    (10)       817    (10)       823    (10)       831    (10)       857    (10)
                                                 863    (10)       875    (10)       881    (10)       950    (11)       957    (11)
                                                 984    (11)
$D1_ESCTEST          1             156    (4)    1217   (12)       156    (4)        221    (4)        2265   (24)       318    (6)
                                                 432    (8)        582    (9)        722    (10)       959    (11)
$D1_ESUBT            1             158    (4)    1054   (11)       1099   (11)       1136   (11)       158    (4)        188    (4)
                                                 2267   (24)       2298   (24)       2407   (24)       252    (4)        434    (8)
                                                 507    (8)        584    (9)        634    (9)        724    (10)       749    (10)
                                                 833    (10)       883    (10)       961    (11)       986    (11)
$D1_ETEST            1             253    (4)    1137   (11)       1232   (12)       1371   (14)       1484   (16)       1883   (18)
                                                 2066   (20)       2173   (22)       2411   (24)       253    (4)        354    (6)
                                                 508    (8)        635    (9)        884    (10)
$D1_EXTEST           1             130    (4)    1183   (12)       1187   (12)       1275   (14)       130    (4)        1418   (16)
                                                 1420   (16)       1539   (18)       1543   (18)       1545   (18)       1579   (18)
                                                 1590   (18)       1603   (18)       1626   (18)       1636   (18)       1649   (18)
                                                 1670   (18)       1678   (18)       1698   (18)       1706   (18)       1732   (18)
                                                 1749   (18)       1771   (18)       1779   (18)       1794   (18)       1799   (18)
                                                 1821   (18)       1834   (18)       1844   (18)       1925   (20)       1929   (20)
                                                 1931   (20)       1963   (20)       1972   (20)       1988   (20)       2009   (20)
                                                 2018   (20)       2033   (20)       2055   (20)       2064   (20)       2107   (22)
                                                 2109   (22)       2232   (24)       2234   (24)       289    (6)        298    (6)
                                                 403    (8)        551    (9)        552    (9)        556    (9)        691    (10)
                                                 692    (10)       696    (10)       928    (11)       929    (11)       933    (11)
$D1_PRINT_S          1             1279   (14)   1279   (14)       1433   (16)       1460   (16)       1555   (18)       1941   (20)
                                                 2120   (22)       2123   (22)       2170   (22)       2244   (24)       2308   (24)
                                                 2356   (24)       2408   (24)
$D1_SBTTL            2             73     (2)    1138   (11)       1233   (12)       1372   (14)       1485   (16)       1884   (18)
                                                 2067   (20)       2174   (22)       254    (4)        355    (6)        509    (8)
                                                 636    (9)        73     (2)        885    (10)
$D2_BDATA            1             120    (4)    1173   (12)       120    (4)        1260   (14)       1408   (16)       1529   (18)
                                                 1915   (20)       2097   (22)       2222   (24)       279    (6)        393    (8)
                                                 541    (9)        681    (10)       918    (11)
$D2_BTEST            1             127    (4)    1180   (12)       1267   (14)       127    (4)        1415   (16)       1536   (18)
                                                 1922   (20)       2104   (22)       2229   (24)       286    (6)        400    (8)
                                                 548    (9)        688    (10)       925    (11)
$D2_SBTTL            1             73     (2)    1138   (11)       1233   (12)       1372   (14)       1485   (16)       1884   (18)
                                                 2067   (20)       2174   (22)       254    (4)        355    (6)        509    (8)
                                                 636    (9)        73     (2)        885    (10)
$DEF                 1             629    (9)
$DEFINI              1             63     (2)    63     (2)        68     (2)        69     (2)        70     (2)
$DS_ABORT            1             345    (6)    345    (6)
$DS_ASKSTR_S         1             1434   (16)   1434   (16)       1461   (16)       2121   (22)       2124   (22)       2171   (22)
                                                 2245   (24)       2309   (24)       2357   (24)       2409   (24)
$DS_BGNDATA          1             120    (4)    1173   (12)       120    (4)        1260   (14)       1408   (16)       1529   (18)
                                                 1915   (20)       2097   (22)       2222   (24)       279    (6)        393    (8)
                                                 541    (9)        681    (10)       918    (11)
$DS_BGNMOD           1             61     (2)    61     (2)
```

```
$DS_BGNSUB        1          138    (4)     1055   (11)     1100   (11)     138    (4)      159    (4)      189    (4)
                                            2247   (24)     2268   (24)     2299   (24)     414    (8)      435    (8)
                                            564    (9)      585    (9)      704    (10)     725    (10)     750    (10)
                                            834    (10)     941    (11)     962    (11)     987    (11)
$DS_BGNTEST       1          127    (4)     1180   (12)     1267   (14)     127    (4)      1415   (16)     1536   (18)
                                            1922   (20)     2104   (22)     2229   (24)     286    (6)      400    (8)
                                            548    (9)      688    (10)     925    (11)
$DS_BITDEF        2          64     (2)     64     (2)
$DS_BNERROR       1          629    (9)     629    (9)
$DS_BOPER         1          1419   (16)    1419   (16)     1544   (18)     1930   (20)     2108   (22)     2233   (24)
$DS_BREAK         1          253    (4)     1137   (11)     1232   (12)     1371   (14)     1484   (16)     1883   (18)
                                            2066   (20)     2173   (22)     2411   (24)     253    (4)      354    (6)
                                            508    (8)      635    (9)      884    (10)
$DS_CHCDEF        1          65     (2)     65     (2)
$DS_CHDEF         1          65     (2)     65     (2)
$DS_CHIDEF        1          65     (2)     65     (2)
$DS_CHMDEF        2          65     (2)     65     (2)
$DS_CHSDEF        3          65     (2)     65     (2)
$DS_CKLOOP        1          148    (4)     1019   (11)     1038   (11)     1044   (11)     1053   (11)     1078   (11)
                                            1084   (11)     1090   (11)     1098   (11)     1123   (11)     1129   (11)
                                            1135   (11)     1197   (12)     1216   (12)     1225   (12)     1231   (12)
                                            1288   (14)     1321   (14)     1337   (14)     1351   (14)     1369   (14)
                                            1432   (16)     1448   (16)     1455   (16)     1459   (16)     1475   (16)
                                            148    (4)      1483   (16)     155    (4)      1554   (18)     1578   (18)
                                            1589   (18)     1602   (18)     1625   (18)     1635   (18)     1648   (18)
                                            1669   (18)     1677   (18)     1697   (18)     1705   (18)     1731   (18)
                                            1741   (18)     1748   (18)     1770   (18)     1778   (18)     1793   (18)
                                            1798   (18)     1820   (18)     1833   (18)     1843   (18)     1856   (18)
                                            187    (4)      1877   (18)     1882   (18)     1940   (20)     1962   (20)
                                            1971   (20)     1987   (20)     2008   (20)     2017   (20)     2032   (20)
                                            2054   (20)     2063   (20)     2119   (22)     2137   (22)     2156   (22)
                                            2166   (22)     220    (4)      2257   (24)     2264   (24)     2283   (24)
                                            2297   (24)     2319   (24)     233    (4)      2334   (24)     2342   (24)
                                            2353   (24)     2369   (24)     2384   (24)     2398   (24)     2406   (24)
                                            242    (4)      251    (4)      310    (6)      317    (6)      352    (6)
                                            424    (8)      431    (8)      462    (8)      481    (8)      504    (8)
                                            574    (9)      581    (9)      608    (9)      632    (9)      714    (10)
                                            721    (10)     748    (10)     781    (10)     800    (10)     806    (10)
                                            818    (10)     824    (10)     832    (10)     858    (10)     864    (10)
                                            876    (10)     882    (10)     951    (11)     958    (11)     985    (11)
$DS_DSADEF        5          63     (2)     63     (2)
$DS_ENDDATA       1          125    (4)     1178   (12)     125    (4)      1265   (14)     1413   (16)     1534   (18)
                                            1920   (20)     2102   (22)     2227   (24)     284    (6)      398    (8)
                                            546    (9)      686    (10)     923    (11)
$DS_ENDMOD        1          2412   (24)    2412   (24)
$DS_ENDSUB        1          158    (4)     1054   (11)     1099   (11)     1136   (11)     158    (4)      188    (4)
                                            2267   (24)     2298   (24)     2407   (24)     252    (4)      434    (8)
                                            507    (8)      584    (9)      634    (9)      724    (10)     749    (10)
                                            833    (10)     883    (10)     961    (11)     986    (11)
$DS_ENDTEST       1          253    (4)     1137   (11)     1232   (12)     1371   (14)     1484   (16)     1883   (18)
                                            2066   (20)     2173   (22)     2411   (24)     253    (4)      354    (6)
                                            508    (8)      635    (9)      884    (10)
$DS_ENVDEF        2          61     (2)     61     (2)
$DS_ERRHARD_S     1          146    (4)     1017   (11)     1036   (11)     1041   (11)     1050   (11)     1076   (11)
                                            1081   (11)     1087   (11)     1096   (11)     1121   (11)     1126   (11)
                                            1132   (11)     1195   (12)     1214   (12)     1223   (12)     1228   (12)
```

```
                                           1286  (14)    1319  (14)    1335  (14)    1349  (14)    1365  (14)
                                           1430  (16)    1446  (16)    1452  (16)    1457  (16)    146   (4)
                                           1473  (16)    1480  (16)    153   (4)     1552  (18)    1576  (18)
                                           1586  (18)    1598  (18)    1622  (18)    1632  (18)    1644  (18)
                                           1666  (18)    1674  (18)    1694  (18)    1702  (18)    1728  (18)
                                           1738  (18)    1745  (18)    1768  (18)    1776  (18)    1790  (18)
                                           1796  (18)    1818  (18)    1831  (18)    1840  (18)    185   (4)
                                           1854  (18)    1875  (18)    1879  (18)    1938  (20)    1960  (20)
                                           1969  (20)    1984  (20)    2006  (20)    2015  (20)    2029  (20)
                                           2052  (20)    2061  (20)    2117  (22)    2135  (22)    2154  (22)
                                           2163  (22)    218   (4)     2255  (24)    2262  (24)    2281  (24)
                                           2295  (24)    231   (4)     2317  (24)    2332  (24)    2339  (24)
                                           2350  (24)    2367  (24)    2382  (24)    239   (4)     2396  (24)
                                           2404  (24)    249   (4)     308   (6)     315   (6)     343   (6)
                                           350   (6)     422   (8)     429   (8)     460   (8)     479   (8)
                                           502   (8)     572   (9)     579   (9)     606   (9)     630   (9)
                                           712   (10)    719   (10)    746   (10)    779   (10)    798   (10)
                                           803   (10)    816   (10)    821   (10)    830   (10)    856   (10)
                                           861   (10)    874   (10)    879   (10)    949   (11)    956   (11)
                                           983   (11)
$DS_ESCAPE          1       156    (4)     1217  (12)    156   (4)     221   (4)     2265  (24)    318   (6)
                                           432   (8)     582   (9)     722   (10)    959   (11)
$DS_EXIT            1       130    (4)     1183  (12)    1187  (12)    1275  (14)    130   (4)     1418  (16)
                                           1420  (16)    1539  (18)    1543  (18)    1545  (18)    1579  (18)
                                           1590  (18)    1603  (18)    1626  (18)    1636  (18)    1649  (18)
                                           1670  (18)    1678  (18)    1698  (18)    1706  (18)    1732  (18)
                                           1749  (18)    1771  (18)    1779  (18)    1794  (18)    1799  (18)
                                           1821  (18)    1834  (18)    1844  (18)    1925  (20)    1929  (20)
                                           1931  (20)    1963  (20)    1972  (20)    1988  (20)    2009  (20)
                                           2018  (20)    2033  (20)    2055  (20)    2064  (20)    2107  (22)
                                           2109  (22)    2232  (24)    2234  (24)    289   (6)     298   (6)
                                           403   (8)     551   (9)     552   (9)     556   (9)     691   (10)
                                           692   (10)    696   (10)    928   (11)    929   (11)    933   (11)
$DS_PARDEF          1       66     (2)     66    (2)
$DS_PRINTF_S        1       1279   (14)    1279  (14)    1433  (16)    1460  (16)    1555  (18)    1941  (20)
                                           2120  (22)    2123  (22)    2170  (22)    2244  (24)    2308  (24)
                                           2356  (24)    2408  (24)
$DS_QADEF           2       629    (9)     629   (9)
$DS_SBTTL           1       73     (2)     1138  (11)    1233  (12)    1372  (14)    1485  (16)    1884  (18)
                                           2067  (20)    2174  (22)    254   (4)     355   (6)     509   (8)
                                           636   (9)     73    (2)     885   (10)
$DS_SECDEF          1       67     (2)     67    (2)
$DS_WAITMS_S        1       324    (6)     1851  (18)    1995  (20)    324   (6)     353   (6)
$DS_WAITUS_S        1       1757   (18)    1757  (18)    1807  (18)
$ENRKA_SYMDEF       7       70     (2)     70    (2)
$EQU                1       629    (9)     61    (2)     629   (9)     64    (2)     65    (2)     66    (2)
                                           71    (2)
$EQULS1             1       629    (9)     61    (2)     629   (9)     65    (2)     71    (2)
$EQULST             1       61     (2)     61    (2)     629   (9)     65    (2)     71    (2)
$GBLINI             2       61     (2)     61    (2)     629   (9)     64    (2)     65    (2)     66    (2)
                                           71    (2)
$HDR_LAYOUTDEF      2       69     (2)     69    (2)
$IDC_REGDEF         10      68     (2)     68    (2)
$PUSHADR            1       147    (4)     1018  (11)    1037  (11)    1043  (11)    1052  (11)    1077  (11)
                                           1083  (11)    1089  (11)    1097  (11)    1122  (11)    1128  (11)
                                           1134  (11)    1196  (12)    1215  (12)    1224  (12)    1230  (12)
```

```
                                           1287  (14)   1293  (14)   1320  (14)   1336  (14)   1350  (14)
                                           1366  (14)   1431  (16)   1435  (16)   1447  (16)   1454  (16)
                                           1458  (16)   1462  (16)   147   (4)    1474  (16)   1482  (16)
                                           154   (4)    1553  (18)   1557  (18)   1577  (18)   1584  (18)
                                           1587  (18)   1600  (18)   1623  (18)   1630  (18)   1633  (18)
                                           1646  (18)   1667  (18)   1672  (18)   1675  (18)   1695  (18)
                                           1700  (18)   1703  (18)   1714  (18)   1729  (18)   1736  (18)
                                           1739  (18)   1747  (18)   1757  (18)   1769  (18)   1777  (18)
                                           1792  (18)   1797  (18)   1807  (18)   1819  (18)   1832  (18)
                                           1842  (18)   1851  (18)   1855  (18)   186   (4)    1876  (18)
                                           1881  (18)   1939  (20)   1949  (20)   1961  (20)   1967  (20)
                                           1970  (20)   1986  (20)   1995  (20)   2007  (20)   2016  (20)
                                           2031  (20)   2041  (20)   2053  (20)   2059  (20)   2062  (20)
                                           2118  (22)   2122  (22)   2125  (22)   2136  (22)   2155  (22)
                                           2165  (22)   2172  (22)   219   (4)    2246  (24)   2256  (24)
                                           2263  (24)   2282  (24)   2296  (24)   2310  (24)   2318  (24)
                                           232   (4)    2333  (24)   2341  (24)   2352  (24)   2358  (24)
                                           2368  (24)   2383  (24)   2397  (24)   2405  (24)   241   (4)
                                           2410  (24)   250   (4)    309   (6)    316   (6)    324   (6)
                                           344   (6)    351   (6)    353   (6)    423   (8)    430   (8)
                                           461   (8)    480   (8)    503   (8)    573   (9)    580   (9)
                                           607   (9)    631   (9)    713   (10)   720   (10)   747   (10)
                                           780   (10)   799   (10)   805   (10)   817   (10)   823   (10)
                                           831   (10)   857   (10)   863   (10)   875   (10)   881   (10)
                                           950   (11)   957   (11)   984   (11)
$PUSHTWO        1          1588  (18)      1588  (18)   1601  (18)   1624  (18)   1634  (18)   1647  (18)
                                           1668  (18)   1676  (18)   1696  (18)   1704  (18)   1712  (18)
                                           1730  (18)   1740  (18)   1973  (20)   2065  (20)
$QIOPUSH        1          1557  (18)      1557  (18)   1714  (18)   1949  (20)   2041  (20)
$READEF_S       1          1292  (14)      1292  (14)   1584  (18)   1630  (18)   1672  (18)   1700  (18)
                                           1736  (18)   1967  (20)   2059  (20)
$SETIMR_S       1          1557  (18)      1557  (18)   1714  (18)   1949  (20)   2041  (20)
$VIELD          1          61    (2)       61    (2)    65    (2)    66    (2)
$VIELD1         1          629   (9)       61    (2)    65    (2)    66    (2)
BDEVFOUND       1          130   (4)       1183  (12)   130   (4)    1418  (16)   1539  (18)   1925  (20)
                                           2107  (22)   2232  (24)   289   (6)    403   (8)    551   (9)
                                           691   (10)   928   (11)
DRV_TBL         3          71    (2)       71    (2)
ENRKADEF        1          62    (2)       62    (2)
GET_LUN         1          129   (4)       1182  (12)   1273  (14)   129   (4)    1417  (16)   1538  (18)
                                           1924  (20)   2106  (22)   2231  (24)   288   (6)    402   (8)
                                           550   (9)    690   (10)   927   (11)
GET_SECLUN      1          295   (6)       295   (6)
STARS           1          139   (4)       1056  (11)   1101  (11)   139   (4)    1558  (18)   1591  (18)
                                           160   (4)    1604  (18)   1637  (18)   1650  (18)   1679  (18)
                                           1707  (18)   1750  (18)   1780  (18)   1800  (18)   1822  (18)
                                           1845  (18)   1857  (18)   190   (4)    1942  (20)   1974  (20)
                                           1989  (20)   2019  (20)   2034  (20)   2248  (24)   2269  (24)
                                           2300  (24)   2370  (24)   301   (6)    325   (6)    415   (8)
                                           436   (8)    463   (8)    565   (9)    586   (9)    609   (9)
                                           705   (10)   726   (10)   751   (10)   835   (10)   942   (11)
                                           963   (11)   988   (11)
```

```
                         +------------------------+
                         ! Performance indicators !
                         +------------------------+

Phase                      Page faults    CPU Time       Elapsed Time
-----                      -----------    --------       ------------
Initialization                      25    00:00:00.15    00:00:00.68
Command processing                 877    00:00:00.91    00:00:01.63
Pass 1                            3350    00:01:02.18    00:01:33.69
Symbol table sort                    6    00:00:01.45    00:00:01.79
Pass 2                             906    00:00:14.49    00:00:34.28
Symbol table output                 67    00:00:00.46    00:00:00.71
Psect synopsis output                7    00:00:00.08    00:00:00.20
Cross-reference output             324    00:00:08.83    00:00:17.90
Assembler run totals              5565    00:01:28.55    00:02:30.93

The working set limit was 1024 pages.
563482 bytes (1101 pages) of virtual memory were used to buffer the intermediate code.
There were 50 pages of symbol table space allocated to hold 795 non-local and 236 local symbols.
2413 source lines were read in Pass 1, producing 98 object records in Pass 2.
134 pages of virtual memory were used to define 62 macros.

                        +--------------------------+
                        ! Macro library statistics !
                        +--------------------------+

Macro library name                          Macros defined
------------------                          --------------
$1$DUA9:[BOWSER.TAPE]ENRKA.MLB;16                    9
SYS$COMMON:[SYSLIB]LIB.MLB;3                         0
$1$DUA9:[DIAGLIB.LIBRARY.BLISS]DIAG.MLB;1           47
SYS$COMMON:[SYSLIB]STARLET.MLB;3                    12
TOTALS (all libraries)                              68

1162 GETS were required to define 68 macros.

There were no errors, warnings or information messages.

MACRO/LIST/CROSS ENRKA3
```