RK611 UNIBUS RK6 DR PRT2 AH-9122F-MC APR 1982
RK06,RK07 CZR6IF0 FICHE 1 OF 2 COPYRIGHT@ 76-82
MADE IN USA

RK611 UNIBUS RK6 DR PRT2 AH-9122F-MC
RK06,RK07 CZR6IF0 FICHE 2 OF 2

.REM %

IDENTIFICATION

PRODUCT CODE:	AC-9120F-MC
PRODUCT NAME:	CZR6IF0 UNIBUS RK6 DR PRT2
DATE:	JANUARY 1982
MAINTAINER:	STORAGE SYSTEMS SOFTWARE TEST APPLICATIONS
AUTHOR:	B. T. LEBLANC

```
     44
     45
     46                                        REVISION HISTORY
     47
     48                                  REVISION        FIXES                                   DATE
     49
     50                                  CZR6IF          IMPLEMENTED XXDP LOAD MEDIA OPTION      JAN 82
     51
```

TABLE OF CONTENTS

```
                                1.0     ABSTRACT

                                        THIS PROGRAM PERFORMS PART 2 OF THE DRIVE DIAGNOSTICS TO
                                        INSURE THAT THE DISK IS CAPABLE OF PERFORMING READ AND WRITE
                                        DATA OPERATIONS IN BOTH 20 AND 22 SECTOR FORMATS.
                                        WORST CASE PATTERNS, SPIRAL WRITING AND READING, AND ALL
                                        OFFSET OPERATIONS ARE PERFORMED.
                                        ERROR DETECTION LOGIC IS CHECKED BY SOFTWARE ERROR FORCING.

                                        AFTER A SUCCESSFUL RUN  (WITH NO ERRORS) OF PART 2, THE DRIVE
                                        IS READY FOR PART 3 OF THE DRIVE DIAGNOSTICS.

                                        TESTING IS BASED ON A HIERARCHY APPROACH STARTING WITH
                                        BASIC LOGIC TESTS AND PROCEEDING THRU DYNAMIC TESTING.  THE
                                        TESTS WILL BE KEPT SMALL TO FACILITATE SCOPING LOOPS.

                                        THIS PROGRAM WILL TEST RK06 & RK07 WITHOUT NEED OF
                                        OPERATOR INTERVENTION.

                                                *****CAUTION*****

                                        HALTING THIS PROGRAM ANYWHERE BUT AT THE END OF A PASS,
                                        MAY LEAVE THE HEADERS IN THE DISK CARTRIDGE IN AN
                                        UNDETERMINED STATE.


                                2.0     REQUIREMENTS

                                2.1     HARDWARE

                                        THE FOLLOWING HARDWARE IS REQUIRED TO RUN THE DISK DIAGNOSTIC:

                                                PDP-11
                                                CONSOLE TELETYPE
                                                16K MEMORY
                                                KW11-L OR KW11-P CLOCK
                                                RK06 UNIBUS CONTROLLER (RK611)
                                                1 TO 8 RK06/RK07 DRIVES

                                        NOTES:  1.  IF NEITHER KW11-L OR P CLOCK IS USED, ALL TIMING
                                                    TESTS WILL BE BYPASSED.  A MESSAGE AT THE
                                                    BEGINNING OF THE TESTS WILL CONFIRM THIS.

                                                2.  A 22 SECTOR FORMATTED PACK IS REQ'D, BUT WILL
                                                    BE A RESULT OF RUNNING DRIVE DIAGNOSTOC PART 1
                                                    (SEE BELOW).


                                2.2     PRELIMINARY TESTING & PROGRAMS

                                        THE RK611 DISKLESS CONTROLLER DIAGNOSTICS (ALL PARTS) SHOULD
                                        FIRST RUN SUCCESSFULLY FOLLOWED BY THE RK06 DRIVE DIAGNOSTIC- PART 1.


                                3.0     PROGRAM CONSIDERATIONS
```

3.1 PDP-11 FAMILY COMPATIBILITY

THIS PROGRAM CAN BE USED BY THE PDP-11/04,05,10,20,
34,35,40,45,50, & 70.

IT IS COMPATABLE WITH THE LSI-11 INSTRUCTION SET AND CAN TEST
THE RK06 ONLY IF THE DRIVE CONTROLLER FOR THE LSI-11 IS
DESIGNED TO BE DIAGNOSTICALLY COMPATABLE WITH THE RK611.

3.2 XXDP

THIS PROGRAM CAN BC CHAINED BY XXDP & WILL NOT OVERLAY THE
LOADER.

CHAIN MODE OPERATION (MONITOR)

1. THE INPUT DIALOGUE IS BYPASSED.
2. THE BUSS ADDRESS & CONTROLLER INTERRUPT VECTOR IS
   DEFAULTED.
3. DRIVE 0 WILL NOT BE TESTED.
4. ALL OTHER DRIVES IN THE 'DRIVE PRESENT' CONDITION WILL
   BE TESTED.

NOTE: THE DRIVE PRESENT CONDITION IS:

   A. HEADS MANUALLY LOADED
   B. CORRECT PORT SELECTED
   C. WRITE LOCK DISABLED
   D. DRIVE READY INDICATOR ON

DUMP MODE OPERATION (MANUAL)

1. INPUT DIALOGUE IF STARTED FROM 220.
2. DRIVE 0 CAN BE TESTED, BUT THE OPERATOR IS FIRST GIVEN
   A MESSAGE TO REPLACE THE PACK IN DR0 WITH A SCRATCH
   PACK & TYPE <CR> WHEN DONE.

3.3 ACT/APT

THIS PROGRAM IS ACT COMPATIBLE. IT IS APT
COMPATABLE TO THE EXTENT THAT APT HOOKS WILL BE IN THE
PROGRAM & WILL WORK THRU THE 'UPTON INTERFACE'.

FOR OTHER INTERFACES, APT MAY ONLY LOAD & START THE PROGRAM.
I.E. LOAD & DUMP MODE.

AUTOMATIC MODE (MONITOR)

1. THE INPUT DIALOGUE IS BYPASSED.
2. THE BUSS ADDRESS & CONTROLLER INTERRUPT VECTOR IS

DEFAULTED.
3. ALL DRIVES IN THE 'DRIVE PRESENT' CONDITION WILL BE TESTED.

NOTE: THE DRIVE PRESENT CONDITION IS:

A. HEADS MANUALLY LOADED
B. CORRECT PORT SELECTED
C. WRITE LOCK DISABLED
D. DRIVE READY INDICATOR ON

DUMP MODE (MANUAL): INPUT DIALOGUE IF STARTED FROM 220.

3.3.1 APT ETABLE DEFINITIONS

THE FOLLOWING DEFINITIONS ARE VALID FOR SPECIFYING APT ENVIRONMENTAL TABLE (ETABLE) ENTRIES. VIA RUNNING THE APT UTILITY PROGRAM 'TSP':

1. SOFTWARE ENVIRONMENT:
   =1 IF APT SCRIPT MODE
   =0 IF STANDALONE MODE

2. ENVIRONMENT MODE:BYTE
   BIT 7 = 1 ETABLE DOES SIZING
   = 0 PROGRAM DOES SIZING
   BIT 6 = 1 SPOOL MESSAGES TO APT IF SCRIPT MODE
   = 0 DON'T SPOOL TO APT
   BIT 5 = 1 SUPPRESS CONSOLE OUTPUT
   = 0 ALLOW CONSOLE OUTPUT
   BITS 4-0 NOT USED

3. SWITCH 1 (SOFTWARE SWITCH REGISTER)
   IF ENVIRONMENT MODE BIT 7 (SIZING BIT) IS SET TO 1, THE SOFTWARE SWITCH REGISTER WILL BE USED, INSTEAD OF THE HARDWARE CONSOLE SWITCH REGISTER. REGARDLESS OF WHICH ONE IS USED, ALL BITS DEFINED IN SECTIONS 4.3 & 4.4 (SWITCH REGISTER OPTIONS) MAY USED WHEN RUNNING IN STANDALONE MODE.
   IN APT SCRIPT MODE, HOWEVER, BIT 14 (LOOP ON TEST) MUST ALWAYS BE SET TO 0.

4. SWITCH 2 (USER SWITCH REGISTER)
   NOT USED

5. CPU OPTIONS:
   NOT USED

6. MEMORY TYPES 1-4 AND MAX MEMORY ADDRESSES
   NOT USED

7. INTERRUPT VECTOR 1:
   USED WHEN ENVIRONMENT MODE BIT 7=1. DEFAULT = 210

8. BUS PRIORITY 1:
   USED WHEN ENVIRONMENT MODE BIT 7=1. DEFAULT = 5

9. INTERRUPT VECTOR 2:
NOT USED

10. BUS PRIORITY 2:
NOT USED

11. BASE ADDRESS:
USED WHEN ENVIRONMENT MODE BIT 7 = 1. DEFAULT = 177440

12. DEVICE MAP:
USED WHEN ENVIRONMENT MODE BIT 7 = 1. EACH BIT SET TO 1 IN BITS 0-7 WILL SELECT THE CORRESPONDING DRIVE TO BE TESTED. BITS 8-15 ARE NOT USED.

13. CONTROLLER DESCRIPTOR WORDS:
NOT USED

14. DEVICE DESCRIPTOR CODES (IN WORDS):
NOT USED

3.4 DUAL ACCESS

THIS PROGRAM WILL NOT TEST OR SUPPORT DUAL-ACCESS. A DRIVE EQUIPED WITH DUAL ACCESS MUST BE SWITCHED TO THE PORT UNDER TEST TO PREVENT CONTENTION WITH THE OTHER PORT.

DUAL ACCESS TESTS WILL BE INCORPORATED IN A SEPARATE PROGRAM AT A LATER DATE.

3.5 MEMORY MANAGEMENT

MEMORY MANAGEMENT NOT USED

3.6 PARITY CHECK ENABLED

IF THE MEMORY PARITY CHECK OPTION IS AVAILABLE ON THE SYSTEM, THE PROGRAM WILL RUN WITH MEMORY CHECK ENABLED.

3.7 BAD SECTOR

THE PROGRAM WILL COMPARE DATA ERRORS WITH THE BAD SECTOR INFORMATION CONTAINED ON CYLINDER 410, HEAD 2. PRINTOUTS OF DATA ERRORS DUE TO BAD SECTORS/TRACKS WILL BE MASKED OUT.

3.8 EXECUTION TIME

THE EXECUTION TIMES SHOWN BELOW ARE BASED ON THE PDP 11/50.

TOTAL TIME: 1 MIN, 30 SEC

3.9 FAULT ISOLATION

TO BE DETERMINED.

3.10 ERROR CORRECTION AND FAILURE RATE ANALYSIS

THIS PROGRAM WILL NOT DO ERROR CORRECTION OR FAILURE RATE ANALYSIS.

3.11 DEFAULT UNIBUS ADDRESSES & VECTORS

THE FOLLOWING IS A LIST OF ALL DEFAULT ADDRESSES & VECTORS OF ALL HARDWARE TO BE USED & THEIR MEMORY ADDRESSES WHERE THEY CAN BE CHANGED.

	LOCATION	DEFAULT CONTENTS
RK06 BUSS ADDRESS	1264	177440
CONTROLLER INTERRUPT VECTOR	1314	210
CONTROLLER PRIORITY	1316	240
P-CLOCK STATUS REG	1320	172540
P-CLOCK SET BUFFER	1322	172542
P-CLOCK READ BUFFER	1324	172544
L-CLOCK STATUS REG	1326	177546
L-CLOCK INTERRUPT VECTOR	1330	100
P-CLOCK INTERRUPT VECTOR	1332	104
TTY KB STATUS REG	1144	177560
TTY KB BUFFER	1146	177562
TTY PRINTER STATUS REG	1150	177564
TTY PRINTER BUFFER	1152	177566

4.0 OPERATING PROCEDURE & CONTROL FUNCTIONS

4.1 PROGRAM LOADING

THE PROGRAM CAN BE LOADED FROM PAPER TAPE USING STANDARD PROCEDURE FOR ABSOLUTE LOADER TAPES; OR FROM ANY MEDIA SUPPORTED BY XXDP.

4.1.1 LOAD THE STARTING ADDRESS (SEE SEC 4.2).

4.1.2 SET SWITCH REGISTERS AS DESIRED (SEE SEC 4.3).

4.1.3 SET DRIVES TO BE TESTED IN THE 'LOAD' CONDITION & WITH THE APPROPRIATE PORT SELECTED & WRITE LOCK DISABLED. DRIVES NOT TO BE TESTED MUST HAVE BOTH PORTS DESELECTED.

NOTE: THE DRIVE WILL NOT RESPOND TO THE 'START SPINDLE' COMMAND IF THE RUN/STOP SWITCH IS IN THE 'STOP' POSITION.

4.1.4 PRESS 'START'

THE PROGRAM WILL IDENTIFY ITSELF AND WILL BEGIN A DIALOGUE WITH THE OPERATOR TO DETERMINE DRIVES TO BE TESTED (SEE SEC 4.5).

THE PROGRAM BEGINS TESTING ONLY THOSE DRIVES SPECIFIED BY THE INPUT DIALOGUE. IF A SPECIFIED DRIVE CANNOT BE FOUND BY THE PROGRAM IT WILL BE FLAGGED AS AN ERROR THAT THE DRIVE WAS NOT AVAILABLE. THEN BEGINNING WITH THE LOWEST NUMERICAL DRIVE AND PROCEEDING IN SEQUENTIAL ORDER, ALL VALID DRIVES WILL BE TESTED. ONE PASS THROUGH THE TEST SEQUENCE WILL BE PERFORMED ON EACH DRIVE BEFORE MOVING TO THE NEXT DRIVE IN SEQUENCE. THE DRIVE TO BE TESTED WILL BE TYPED AT THE BEGINNING OF EACH PASS. "END OF PASS" WILL BE TYPED AFTER TESTING ALL DRIVES.

4.2 STARTING LOCATIONS

LOCATION 200 - STARTING ADDRESS TO DEFAULT THE BUSS ADDRESS & THE CONTROLLER INTERRUPT VECTOR & TEST ALL DRIVES IN THE 'DRIVE PRESENT' CONDITION.

NOTE: THE DRIVE PRESENT CONDITION IS:

A. HEADS MANUALLY LOADED
B. CORRECT PORT SELECTED
C. WRITE LOCK DISABLED
D. DRIVE READY INDICATOR ON

LOCATION 220 - STARTING ADDRESS TO INPUT TESTING PARAMETERS VIA THE INPUT DIALOGUE. BUSS ADDRESS & CONT. INTERRUPT VECTOR INPUTTED ONLY ON 1ST PASS.

IMPORTANT: FOR VARIATIONS OF THE ABOVE, SEE XXDP, ACT/APT CONSIDERATIONS IN SECTIONS 3.2 & 3.3.

ALSO, THE PROGRAM WILL AUTOMATICALLY DETERMINE WHETHER THE DRIVE IS AN RK06 OR RK07.

4.3 SWITCH REGISTER

THE SWITCHES ARE USED TO PROVIDE CONTROL FUNCTIONS.

SWITCH	FUNCTION
15	HALT ON ERROR
14	LOOP ON TEST
13	INHIBIT ERROR TYPEOUT
12	BYPASS DRIVE AFTER 20 ERRORS
11	INHIBIT ITERATION
10	BELL ON ERROR
9	LOOP ON ERROR
8	LOOP ON TEST IN SW<07:00>

4.3.1 SW<15>

THE PROGRAM HALTS ON ENCOUNTERING AN ERROR, AFTER TYPING OUT
THE ERROR MESSAGE AND PERTINENT INFORMATION.
PRESSING "CONTINUE" CONTINUES NORMAL OPERATION OF THE PROGRAM.

4.3.2 SW<14>

THE PROGRAM LOOPS ON THE TEST THAT IS BEING EXECUTED WHEN
THE SWITCH IS PUT ON. THIS SWITCH IS NORMALLY USED ALONG
WITH SW15.

4.3.3 SW<13>

THIS SWITCH INHIBITS ALL ERROR MESSAGES. NORMALLY USED WHEN
LOOPING ON TEST (SW14) OR LOOPING ON ERROR (SW9).
WITH SWITCH <13> SET, SWITCH <15> SHOULD NOT BE SET.

4.3.4 SW<12>

THIS SWITCH BYPASSES A GIVEN DRIVE AFTER 20 ERRORS HAVE
BEEN DETECTED.

4.3.5 SW<11>

EACH TEST WILL BE EXECUTED ONLY ONCE. NORMALLY AFTER THE
FIRST PASS, EACH SUBTEST IS ITERATED A NUMBER OF TIMES
(USUALLY 50, 5 IN SOME CASES). SETTING THIS SWITCH INHIBITS
ITERATIONS, SO THAT QUICK PASSES CAN BE MADE.

4.3.6 SW<10>

RINGS A BELL ON ERROR. USEFUL WHEN ERROR TYPEOUT IS INHIBITED.

4.3.7 SW<09>

THIS SWITCH PROVIDES THE TIGHTEST POSSIBLE SCOPE LOOP FOR ERRORS. IF THE PROGRAM DETECTS AN ERROR, IT WILL LOOP BACK TO THE BEGINNING OF TEST.

4.3.8 SW<08>

THIS SWITCH IS USED TO SELECT A PARTICULAR TEST (AS PER SW<00-7>) FOR EXECUTION AND SUBSEQUENT LOOPING. THUS IF TEST 15 IS TO BE SELECTED THE SWITCH SETTING WOULD BE 000415. IT SHOULD BE NOTED THAT BEFORE SELECTING & LOOPING TEST 15, ALL THE PREVIOUS TESTS (1-14) WILL BE EXECUTED.

4.4 'SOFTWARE' SWITCH REGISTER

IF THE PROGRAM IS BEING RUN ON A SWITCHLESS PROCESSOR (I.E. AN 11/04 OR 11/34) THE PROGRAM WILL DETERMINE THAT THE HARDWARE SWITCH REGISTER IS NOT PRESENT AND WILL USE A 'SOFTWARE' SWITCH REGISTER. THE 'SOFTWARE' SWITCH REGISTER IS LOCATED AT LOCATION 176 (8). THE SETTINGS OF THE 'SOFTWARE' SWITCHES ARE CONTROLLED THROUGH A KEYBOARD ROUTINE WHICH IS CALLED BY TYPING A 'CONTROL G'. THE PROGRAM WILL RECOGNIZE THE 'CONTROL G' AT ANY TIME EXCEPT WHEN THE PROGRAM IS AT A HIGHER PRIORITY PROCESSING AN RK06 INTERRUPT. THE 'SOFTWARE' SWITCH VALUES ARE ENTERED AS AN OCTAL NUMBER IN RESPONSE TO THE PROMPT FROM THE SWITCH ENTRY ROUTINE:

SWR = NNNNNN NEW =

EACH TIME SWITCH SETTING ARE ENTERED, THE ENTIRE SWITCH REGISTER IMAGE MUST BE ENTERED. LEADING ZEROS ARE NOT REQUIRED., 'RUBOUT' AND 'CONTROL U' FUNCTIONS MAY BE USED TO CORRECT TYPING ERRORS DURING SWITCH ENTRY.

ON PROCESSORS WITH HARDWARE SWITCH REGISTERS, THE 'SOFTWARE' SWITCH REGISTER MAY BE USED. IF THE PROGRAM FINDS ALL 16 SWITCHES IN THE 'UP' POSITION, ALL SWITCH REGISTER REFERENCES WILL BE TO THE 'SOFTWARE' REGISTER AND THE PROCEDURES DESCRIBED ABOVE MUST BE FOLLOWED.

4.5 INPUT DIALOGUE

THE DIALOGUE WILL BE DONE INTERACTIVELY. THE PROGRAM WILL REQUEST A PARAMETER BY CONSOLE TYPEOUT. THE PARAMETER MAY THEN BE ENTERED AS SPECIFIED BELOW OR ALLOWED TO DEFAULT BY A CARRIAGE RETURN. UNRECOGNIZED OR ILLEGAL RESPONSES WILL BE ECHOED BACK FOLLOWED BY "?". THE PROPER RESPONSE MAY THEN BE ENTERED.

IMPORTANT: FOR VARIATIONS OF THE ABOVE, SEE XXDP, ACT/APT CONSIDERATIONS IN SECTIONS 3.2 & 3.4.

548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603

4.5.1 DRIVE SELECTION

THE REQUEST WILL BE:

DRIVES TO BE TESTED:

THE DEFAULT RESPONSE IS CARRIAGE RETURN TO TEST ALL DRIVES IN THE 'DRIVE PRESENT' CONDITION.

THE OPERATOR CAN ALSO TYPE IN THE SPECIFIC DRIVE NUMBERS TO BE TESTED, SEPARATED BY COMMAS & TERMINATED BY A CARRIAGE RETURN.

E.G. DRIVES TO BE TESTED: 1,2,4,6

IMPORTANT: FOR VARIATIONS OF THE ABOVE, SEE XXDP, ACT/APT CONSIDERATIONS IN SECTIONS 3.2 & 3.3.

4.5.2 BUS ADDRESS

THE REQUEST WILL BE:

TYPE IN BUSS ADDRESS IF NOT 177440

THE DEFAULT IS A CARRIAGE RETURN

4.5.3 CONTROLLER INTERRUPT VECTOR

THE REQUEST WILL BE:

TYPE IN CONTROLLER INTERRUPT VECTOR IF NOT 210

THE DEFAULT IS A CARRIAGE RETURN.

4.5.4 EXAMPLE OF PROGRAM DIALOGUE

THE EXAMPLE SHOWN IS FOR A PROGRAM STARTED AT ADDRESS 220. ALL OPERATOR RESPONSES ARE UNDERLINED.

```
CZR6IF0 UNIBUS RK6 DR PRT2

DRIVES TO BE TESTED: 1,3<CR>
                     -------

TYPE IN BUSS ADDRESS IF NOT 177440 <CR>
                                   ----

TYPE IN CONTROLLER INTERRUPT VECTOR IF NOT 210 <CR>
                                               ----

WILL TEST DRIVES:
1
```

```
                                        3

                                        DRIVE 1

                                        (THE REST IS IDENTICAL TO THE EXAMPLE SHOWN IN 4.6 BELOW)


                              4.6       PROGRAM EXAMPLE

                                        THE FOLLOWING IS AN EXAMPLE OF A PROGRAM STARTED AT THE
                                        DEFAULT ADDRESS (200) & WITH 2 DRIVES ON THE LINE.

                                        CZR6IF0 UNIBUS RK6 DR PRT2

                                        WILL TEST DRIVES:
                                        0
                                        1

                                        DRIVE 0

                                        DRIVE SERIAL NO. AAA
                                        CARTRIDGE SERIAL NO. BBB

                                        DRIVE 1

                                        DRIVE SERIAL NO. CCC
                                        CARTRIDGE SERIAL NO. DDD

                                        END PASS #1


                                        WILL TEST DRIVES:
                                        0
                                        1

                                        DRIVE 0

                                        DRIVE 1

                                        END PASS # 2

                                           (ETC)

                                    THE ABOVE ASSUMES NO ERRORS DETECTED.
                                    THE NUMBER OF PASSES IS DETERMINED BY ACT/APT/XXDP

                                    IMPORTANT:  FOR VARIATIONS OF THE ABOVE, SEE XXDP, ACT/APT
                                                CONSIDERATIONS IN SECTIONS 3.2 & 3.3.

                              4.7       HALTING THE PROGRAM

                                        THE PROGRAM PROVIDES A METHOD OF HALTING ITSELF SUCH THAT
                                        THE CARTRIDGE AND/OR DRIVE IS NOT LEFT IN ON UNDETERMINED
                                        STATE; IE: HEADS UNLOADED OR INVALID FORMAT.

                                        TO PROPERLY HALT, TYPE CONTROL-C (^C) ON THE CONSOLE.
```

```
660
661                                     IF HEADS ARE LOADED & FORMATTING IS VALID,
662                                     THE PROGRAM WILL:
663
664                                     1.  ECHO ^C
665                                     2.  TYPE ''CPU HALTED''
666                                     3.  HALT THE PROGRAM
667
668                                     IF HEADS ARE NOT LOADED AND/OR FORMATTING IS INVALID,
669                                     THE PROGRAM WILL:
670
671                                     1.  ECHO ^C
672                                     2.  TYPE 'HALT PENDING, PLEASE WAIT'
673                                     3.  DO THE TEST(S) THAT LOADS HEADS AND/OR FORMATS
674                                         THE INVALID CYLINDERS
675                                     4.  TYPE 'CPU HALTED'
676                                     5.  HALT THE PROGRAM
677
678
679                                     NOTES:
680
681                                     1.  THE ABOVE EXAMPLE IS FOR THE PROGRAM RUNNING IN DUMP
682                                         MODE (MANUAL).  IF THE PROGRAM IS RUNNING IN CHAIN/AUTO
683                                         MODE VIA XXDP,ACT,APT;  IT WILL FIRST LOAD HEADS
684                                         AND/OR FORMAT CORRECTLY,  IF REQ'D, THEN IT WILL
685                                         JUMP ON TO THE MONITOR WHERE THE NEXT PROGRAM CAN BE
686                                         CALLED IN.
687
688                                         THE TYPEOUTS WILL BE ''ABORT PENDING - PLEASE WAIT''
689                                                          &  'PROGRAM ABORTING''
690
691                                     2.  OPERATING THE 'CONTINUE' SWITCH ON THE CPU CONSOLE WILL RETURN THE
692                                         PROGRAM TO TEST 1 WHERE TESTING WILL BEGIN WITH THE 1'ST DRIVE AGAIN.
693
694
695
696
697                         5.0     DRIVE DIAGNOSTIC FUNCTIONAL DESCRIPTION
698
699                         5.1     GENERAL
700
701                                 A.  WRITE TESTS
702
703                                     THESE TESTS CHECK THE ABILITY OF THE DRIVE TO WRITE & READ
704                                     WORSE CASE PATTERNS;  PERFORM ALL OFFSETS & PERFORM ALL
705                                     SPIRAL WRITING.
706
707                                 B.  SERVO & SPINDLE TIMING TESTS
708
709                                     THESE TESTS CHECK & TYPE HEAD LOAD, UNLOAD & INDEX TIMING,
710                                     ALSO MIN, MAX, AND AVERAGE SEEK TIMES, AND MAX VELOCITY
711                                     OF THE HEADS ARE MEASURED & TYPED.
712
713                         5.2     TEST DESCRIPTIONS
714
715                                 ********************************************************************************
```

```
                        BASIC CONTROLLER TESTS, SIZING & SETUP
                        ;*****************************************************************

                        TEST 1  REFERENCE ALL CONTROLLER REGISTERS

                                THIS TEST VERIFIES THAT ALL THE CONTROLLER REGISTERS
                                CAN BE ACCESSED.  THE INABILITY  TO BE ACCESSED WILL
                                RESULT IN A TIMEOUT TRAP WITH  AN ERROR MESSAGE.  ANY
                                ERROR IN THIS TEST WILL RESULT IN ABORTING ALL OTHER
                                TESTS AND JUMPING TO  'END OF PASS'


                        TEST 2  SIZE THE BUSS

                                THIS TEST IS ENTERED ONLY IF 'DRIVE SELECTION' IS DEFAULTED
                                EITHER BY RUNNING IN THE AUTO MODE OR A 200 START IN THE
                                MANUAL MODE.
                                EVERY DRIVE FROM 0 THRU 7 IS ADDRESSED.
                                CONTROLLER ERROR (CERR) IS EXAMINED AND IF NOT SET, THE
                                DRIVE WILL BE TESTED AS AN RK06.  IF SET, THE PROGRAM WILL BYPASS
                                TESTING THAT DRIVE ONLY IF THE ERROR WAS A RESULT OF
                                MDS, UFE OR NED BEING SET; OR BOTH NED & DRA RESET IN-
                                DICATING THE OTHER PORT IS ACCESSED.
                                IF CERR IS DUE TO DTYE, THE DRIVE WILL BE TESTED AS AN RK07


                        TEST 3  VERIFY OPERATOR DRIVE SELECTIONS

                                THIS TEST IS ENTERED ONLY IF DRIVE SELECTION IS NOT
                                DEFAULTED.  EVERY DRIVE FROM 0 TO 7 IS ADDRESSED &
                                CONTROLLER ERROR (CERR) IS EXAMINED. IF NOT SET, THE
                                PROGRAM WILL ASSUME THE DRIVE IS PRESENT AS AN RK06.
                                IF CERR WAS SET, THAT DRIVE WILL BE BYPASSED
                                ONLY IF THE ERROR WAS A RESULT OF MDS OR UFE SET OR BOTH
                                NED & DRA RESET (WRONG PORT).  IF CERR IS  A RESULT OF
                                NED ONLY, IT IS CHECKED AGAINT THE INPUTTED INFOR TO
                                VERIFY IT WAS NOT SPECIFIED.
                                IF CERR IS DUE TO DTYE, THE DRIVE WILL BE TESTED AS
                                AN RK07.


                        TEST 4  FIND NEXT DRIVE TO BE TESTED

                                THIS TEST FINDS THE NEXT DRIVE PRESENT & PUTS THAT
                                ADDRESS IN 'DRVAD' & $TMP4 IS SET TO CDT IF THE DRIVE
                                DRIVE IS AN RK07.
                                THROUGHOUT THE FOLLOWING TESTS, THE DRIVE TESTED IS
                                THE DRIVE WHOSE ADDRESS IS IN 'DRVAD'.


                        TEST 5  PRINT DRIVE SERIAL NUMBER

                                THIS TEST READS & PRINTS THE DRIVE SERIAL # FROM MSG A, WORD 11
                                IN DECIMAL & IS PERFORMED ON THE 1ST PASS ONLY
```

```
                                        TEST 6  SET VV WITH PACK COMMAND

                                                IF VV IS RESET, THE PACK COMMAND IS USED TO SET IT.


                                        TEST 7  READ & SAVE BAD SECTOR INFO & TYPE PACK SERIAL #

                                                THIS TEST VERIFIES THAT CYL 632 (1456 FOR RK07), TRACK 2 CAN BE READ.
                                                THIS AREA CONTAINS BAD SECTOR INFO WHICH IS WRITTEN BY THE
                                                FACTORY DURING MANF.  ALL BAD SECTOR INFO (BSE) WILL BE STORED
                                                AT THIS TIME TO MASK FUTURE READ HEADER OR DATA ERROR PRINTOUTS.
                                                IF BSE INFO CANNOT BE READ, OR IF AFTER READING THE BSE INFO
                                                IT IS DETERMINED THAT AN ALIGNMENT CARTRIDGE IS USED,
                                                A MESSAGE WILL BE TYPED INDICATING THAT ALL
                                                FUTURE FORMAT AND READ-WRITE TESTS WILL BE BYPASSED.
                                                THIS IS DONE SO AS NOT TO DESTROY BSE INFO OR AN ALIGNMENT PACK BY WRITI

                                                THE PACK SERIAL # IS TYPED IN OCTAL & FOR THE FIRST PASS ONLY.


                                        ********************************************************************************
                                        WRITE TESTS
                                        ********************************************************************************


                                        TEST 10 BASIC WRITE DATA TEST; 1 WORD

                                                THIS TEST VERIFIES THE ABILITY OF THE DRIVE TO WRITE JUST ONE WORD,
                                                ALL SECTORS ON CYL 0 ARE GIVEN IDENTICAL HEADERS &
                                                A WRITE COMMAND IS ISSUED. READ & WRITE CHECK COMMANDS ARE NOT
                                                PERFORMED. THIS TEST PROVIDES THE TIGHTEST POSSIBLE SCOPE LOOP
                                                FOR A WRITE ERROR.


                                        TEST 11 BASIC WRITE DATA TEST; FULL SECTOR

                                                THIS TEST VERIFIES THE ABILITY OF THE DRIVE TO WRITE
                                                A FULL SECTOR.  ALL ZEROS ARE WRITTEN BY THE WRITE DATA COMMAND
                                                & CHECKED BY A RD DATA COMMAND.  A FURTHER CHECK IS PERFORMED
                                                BY THE WRT CHK COMMAND.
                                                THE ABOVE IS REPEATED FOR AN ALL ONES PATTERN.


                                        TEST 12 20 SECTOR FORMAT TEST

                                                DATA IS WRITTEN ON A FULL TRACK IN 20 SECTOR FORMAT.
                                                MSG B0,B1 ARE CHECKED FOR ANY ERROR CONDITION.
                                                CYLINDER,TRACK, SECTOR 0 IS USED.


                                        TEST 13 TEST OFFSET & RTC LOGIC

                                                THE HEADS ARE FIRST OFFSET BY OFFSET COMMANDS.
                                                THIS TEST CHECKS THE RTC LOGIC BY VERIFYING THAT THE
                                                'OFFSET ON' BIT (MSG A,00) RESETS AND THE OFFSET REG
```

```
                                        BECOMES THE CYL DIFF INFO WHEN A SEEK CMD TO A
                                        DIFFERENT CYLINDER IS ISSUED
                                        IT ALSO TESTS THAT DRIVE CLEAR & SEEK TO SELF WILL NOT
                                        CLEAR THE 'OFFSET ON' BIT OR THE OFFSET REG.
                                        ALL OFFSET POSITIONS IN BOTH DIRECTIONS ARE CHECKED


                                TEST 14 TEST READ DATA AT ALL HEAD OFFSET POSITIONS

                                        THIS TEST VERIFIES THAT THE HEAD OFFSET LOGIC IS OPERATIONAL BY
                                        WRITING ALL 1'S PATTERNS ON CYLINDER 0, HEAD 0. THEN
                                        PERFORMING READ DATA FROM CENTERLINE AND MOVING OUT + AND - OFFSET
                                        POSITIONS UNTIL A FAILURE OCCURES. THE OFFSET POSITIONS
                                        ARE TYPED OUT.
                                        OFFSET CODES ARE ALSO VERIFIED BY READING MSG A, STATUS 00 & 10.
                                        ALL HEADS ARE TESTED AT CYL 0.

                                        IF THERE ARE NO FAILURES AT ALL, THIS INDICATES THAT

                                                A. HEADS DID NOT MOVE AT ALL
                                        OR      B. THE COMBINATION OF DISC SURFACE, HEADS, R/W AMP
                                                   ARE EXCEPTIONALLY GOOD.

                                        NOTE THAT THE OFFSET FAILURE IS NOT AN ERROR,
                                        BUT AN INDICATION OF SURFACE, HEAD, & R/W ELECTRONICS QUALITY ONLY


                                TEST 15 WRITE WITH HEADS OFFSET

                                        THIS TEST VERIFIES THAT WHEN ATTEMPTING TO
                                        WRITE WITH HEADS OFFSET THAT THE OFFSET WILL CLEAR
                                        & THE DRIVE WILL WRITE
                                        SINCE THE WRITE COMMAND HAS AN IMPLIED RTC.
                                        THIS TEST IS PERFORMED FOR MAX POS & NEG OFFSETS ONLY


                                TEST 16 TEST CURRENT CROSS-OVER CYLINDERS

                                        THIS TEST VERIFIES THAT THE DRIVE CAN WRITE & READ OFF
                                        CURRENT CHANGE CYLINDERS X & Y IN THE FOLLOWING WAY:

                                        SPIRAL WRITING IS PERFORMED FROM CYLINDER X TO CYLINDER Y
                                        WITH A DATA PATTERN FILLING THE ENTIRE 2 CYLINDERS.

                                        A WRITE CHECK IS THEN PREFORMED TO VERIFY DATA WAS PROPERLY WRITTEN.
                                        THIS TEST IS PERFORMED FOR ALL 3 HEADS.

                                        CYLINDER X:  63  127  191  255  319  383  RK06
                                        CYLINDER Y:  64  128  192  256  320  384  RK06

                                        CYLINDER X:  127 255  383  511  639  767  RK07
                                        CYLINDER Y:  128 256  384  512  640  768  RK07

                                        THE ABOVE CYL NUMBERS ARE IN DECIMAL.
```

```
884
885                                     TEST 17 TEST HEAD SWITCHING TIME
886
887                                             TESTS THE ABILITY TO SWITCH HEADS IN LESS THEN 10MS WHEN HEADS SPIRAL.
888
889                                             1. SECTOR 23(8) IS FIRST LOCATED AND A WRITE DATA COMMAND OF 512 WORDS
890                                                TO SECTOR 25(8) IS ISSUED.
891                                             2. THE PROGRAM NOW KNOWS THAT THE DRIVE WILL NOT HAVE TO TRAVEL
892                                                A FULL REVOLUTION BEFORE FINDING SECTOR 25(8).
893                                             3  SINCE EACH SECTOR TAKES APPROX. 1.2MS, THE TIME BETWEEN
894                                                THE START OF THE WRITE COMMAND (FROM SECTOR 25(8), HEAD 0; TO
895                                                SECTOR 0, HEAD 1) AND CONTROLLER READY SHOULD BE APPROX 6MS
896
897                                             THE ABOVE IS REPEATED FOR HEAD SWITCHING BETWEEN 1 TO 2
898
899                                             THIS TEST IS BYPASSED IF NEITHER L OR P CLOCK IS PRESENT
900
901     .
902                                     TEST 20 DRIVE OFF TRACK TEST
903
904                                             THIS TEST CHECKS FOR SERVO OSCILLATIONS DURING SETTLING TIME BEYOND
905                                             THE ALLOTED 3MS.
906
907                                             1.  INITIALLY, EVERY CYLINDER IS FORMATTED WITH IDENTICAL HEADERS
908                                                 (UNIQUE TO EACH CYLINDER)
909                                             2.  A FULL SECTOR WRITE COMMAND IS ISSUED BY A SINGLE CYL SEEK FROM 0 TO
910                                                 AS HEADERS ARE IDENTICAL, THE NEXT SECTOR TO COME UNDER THE
911                                                 HEADS WILL IMMEDIATELY BE WRITTEN.
912                                             3.  IF THERE IS OSCILLATION SENSED BY READING THE TRIBITS,
913                                                 DRIVE OFF TRACK ERROR WILL SET.
914
915                                             IN THIS MANNER OSCILLATING SEEKS ARE PERFORMED BETWEEN ALL MAJOR CYLINDE
916                                             100 OSCILLATIONS ARE PERFORMED AT EACH MAJOR CYLINDER
917                                             BEFORE DOING THE NEXT CYLINDER
918
919
920                                     ANY TEST THAT MODIFIES STANDARD FORMATTING IS FOLLOWED BY A
921                                     'CLEAN UP' TEST TO PUT THOSE CYLINDERS BACK TO STANDARD
922                                     FORMAT.
923
924             6.0     ERROR REPORTING
925
926
927             6.1     ERROR INTERPRETATION
928
929                                     WHENEVER AN ERROR MESSAGE IS PRINTED OUT, ALL REGISTERS
930                                     AND OTHER DATA PERTAINING TO THE ERROR ARE ALSO GIVEN.
931                                     MSG A(00), MSG B(01), RKER, RKBA...ETC, INDICATE THE
932                                     CONTENTS OF THE CORRESPONDING REGISTERS AT THE TIME OF
933                                     ERROR.
934
935                                     EVERY ERROR MESSAGE CONTAINS A PC.  THIS PC INDICATES THE
936                                     POSITION IN PROGRAM WHERE THE ERROR CALL IS LOCATED.  THE
937                                     ERROR MESSAGE, BECAUSE OF PRACTICAL CONSIDERATIONS IS MADE
938                                     SHORT AND MEANINGFUL.  THE USER IS ADVISED TO LOOK UP THE
939                                     PC IN THE PROGRAM LISTING, WHERE HE WILL FIND MORE INFORMATION
```

ABOUT THE ERROR. IN MANY INSTANCES, A SINGLE FAULT WILL
GIVE RISE TO MORE THAN ONE ERROR REPORT. A LITTLE DELIBERATION
AND CAREFUL EXAMINATION OF THE DATA GIVEN WILL BE CERTAINLY
VERY HELPFUL IN PINPOINTING THE FAULT. A BRIEF EXPLANATION
OF WHAT IS BEING CHECKED IN THE TEST IS GIVEN AT THE
BEGINNING OF EVERY TEST. ALL THE NUMBERS GIVEN WITH ERROR
MESSAGES ARE IN OCTAL.

NOTE

NO ERROR LOGGING OR OPERATION HISTORY IS PROVIDED.

6.2 ERROR PRINTOUT EXAMPLES:

EXAMPLE #1

MESSAGE A0 ERROR
AFTER START SPINDLE CMD & FWD SET

```
  TEST NO.  PC
  000014  016530
        EXPECT
A0      B0      A1      B1      A2      B2      B3
030144  100000  013704  000001
        ACTUAL
  140144  100000  101744  000001
RKCS1   RKCS2   RKASOF  RKER    RKDS    RKDC
040200  000100  010000  000000  000000  000000
```

THE ABOVE EXAMPLE SHOWS EXPECTED & ACTUAL DATA FOR
MESSAGE REGISTERS A0, B0, A1 & B1.

MESSAGES A2, B2 & B3 WILL BE TYPED OUT ONLY AS REQUIRED IF
THE CYLINDER DIFFERENCE/OFFSET, CYLINDER ADDRESS &
HEAD & SECTOR INFORMATION IS A VARIABLE PARAMETER OF THE
TEST.

EXAMPLE #2:

NO ATTN IN RKASOF
AFTER UNLOAD COMMAND

```
  TEST NO.  PC
  000003  014330
RKMR2   RKMR3   RKER    RKDS    RKCS1   RKCS2   RKASOF
000144  100000  000000  100101  000206  000104  000000
```

[END OF DOCUMENT]

```
996                           %
```

```
                                                    I  2
 997
 998
 999
1000                                ;*** PGM REV 032 ***
1001                                .NLIST  CND,MC,MD
1002                                .LIST   ME
1003                                .ENABL  ABS,AMA
1004
1005                                ;DEFINE SYSMAC MACROS
1006
1007            167400              $SWR=   167400              ;DEFINE SWITCHES 15,14,13,11,10,9,8
1008            000001              $TN=    1                   ;SET FIRST TEST NO. TO 1
1009
1010
1011
1012                                .TITLE  CZR6IF0 UNIBUSS RK6 DR PRT2
1013                                ;*COPYRIGHT (C) 1976,1982
1014                                ;*DIGITAL EQUIPMENT CORP.
1015                                ;*MAYNARD, MASS. 01754
1016                                ;*
1017                                ;*PROGRAM BY GARY PAPAZIAN
1018                                ;*
1019                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
1020                                ;*PACKAGE (MAINDEC-11-DZQAC-C5), JAN, 1981.
1021                                ;*
1022
1023                                .SBTTL  OPERATIONAL SWITCH SETTINGS
1024                                ;*
1025                                ;*      SWITCH                  USE
1026                                ;*      ------          --------------------
1027                                ;*        15            HALT ON ERROR
1028                                ;*        14            LOOP ON TEST
1029                                ;*        13            INHIBIT ERROR TYPEOUTS
1030                                ;*        12            ABORT DRIVE AFTER 20 ERRORS
1031                                ;*        11            INHIBIT ITERATIONS
1032                                ;*        10            BELL ON ERROR
1033                                ;*         9            LOOP ON ERROR
1034                                ;*         8            LOOP ON TEST IN SWR<7:0>
1035
1036
1037                                .SBTTL  SUMMARY OF STARTING LOCATIONS
1038                                ;*
1039                                ;*
1040                                ;*              200     DEFAULT PARAMETERS
1041                                ;*              204     DEFAULT PARAMETERS & BYPASS WRITE TESTS
1042                                ;*              214     DEFAULT PARAMETERS & BYPASS TIMING TESTS
1043                                ;*              220     INPUT PARAMETERS
1044                                ;*              224     INPUT PARAMETERS & BYPASS WRITE TESTS
1045                                ;*              230     INPUT PARAMETERS & BYPASS TIMING TESTS
1046                                ;*              240     ODT11
1047                                ;*
```

```
                        .SBTTL  BASIC DEFINITIONS

                        ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100                  STACK=  1100
                        .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                        .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL

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

                        ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                  R0=     %0              ;;GENERAL REGISTER
000001                  R1=     %1              ;;GENERAL REGISTER
000002                  R2=     %2              ;;GENERAL REGISTER
000003                  R3=     %3              ;;GENERAL REGISTER
000004                  R4=     %4              ;;GENERAL REGISTER
000005                  R5=     %5              ;;GENERAL REGISTER
000006                  R6=     %6              ;;GENERAL REGISTER
000007                  R7=     %7              ;;GENERAL REGISTER
000006                  SP=     %6              ;;STACK POINTER
000007                  PC=     %7              ;;PROGRAM COUNTER

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

                        ;*''SWITCH REGISTER'' SWITCH DEFINITIONS
100000                  SW15=   100000
040000                  SW14=   40000
020000                  SW13=   20000
010000                  SW12=   10000
004000                  SW11=   4000
002000                  SW10=   2000
001000                  SW09=   1000
000400                  SW08=   400
000200                  SW07=   200
000100                  SW06=   100
000040                  SW05=   40
000020                  SW04=   20
000010                  SW03=   10
000004                  SW02=   4
```

```
                    000002          SW01=   2
                    000001          SW00=   1
                                    .EQUIV  SWC9,SW9
                                    .EQUIV  SW08,SW8
                                    .EQUIV  SW07,SW7
                                    .EQUIV  SW06,SW6
                                    .EQUIV  SW05,SW5
                                    .EQUIV  SW04,SW4
                                    .EQUIV  SW03,SW3
                                    .EQUIV  SW02,SW2
                                    .EQUIV  SW01,SW1
                                    .EQUIV  SW00,SW0

                                    ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
                    100000          BIT15=  100000
                    040000          BIT14=  40000
                    020000          BIT13=  20000
                    010000          BIT12=  10000
                    004000          BIT11=  4000
                    002000          BIT10=  2000
                    001000          BIT09=  1000
                    000400          BIT08=  400
                    000200          BIT07=  200
                    000100          BIT06=  100
                    000040          BIT05=  40
                    000020          BIT04=  20
                    000010          BIT03=  10
                    000004          BIT02=  4
                    000002          BIT01=  2
                    000001          BIT00=  1
                                    .EQUIV  BIT09,BIT9
                                    .EQUIV  BIT08,BIT8
                                    .EQUIV  BIT07,BIT7
                                    .EQUIV  BIT06,BIT6
                                    .EQUIV  BIT05,BIT5
                                    .EQUIV  BIT04,BIT4
                                    .EQUIV  BIT03,BIT3
                                    .EQUIV  BIT02,BIT2
                                    .EQUIV  BIT01,BIT1
                                    .EQUIV  BIT00,BIT0

                                    ;*BASIC ''CPU'' TRAP VECTOR ADDRESSES
                    000004          ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
                    000010          RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
                    000014          TBITVEC=14              ;;''T'' BIT
                    000014          TRTVEC= 14              ;;TRACE TRAP
                    000014          BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
                    000020          IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
                    000024          PWRVEC= 24              ;;POWER FAIL
                    000030          EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
                    000034          TRAPVEC=34              ;;''TRAP'' TRAP
                    000060          TKVEC=  60              ;;TTY KEYBOARD VECTOR
                    000064          TPVEC=  64              ;;TTY PRINTER VECTOR
                    000240          PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR

                                    .SBTTL  RK06 CONTROLLER REGISTER DEFINITION
```

```
1160
1161                            ;       $BASE=177440
1162
1163            000000          RKCS1=  0               ;CONTROL AND STATUS REGISTER 1
1164            000002          RKWC=   2               ;WORD COUNT REGISTER
1165            000004          RKBA=   4               ;BUS ADDRESS REGISTER
1166            000006          RKDA=   6               ;DESIRED TRACK SECTOR REGISTER
1167            000010          RKCS2=  10              ;CONTROL AND STATUS REGISTER 2
1168            000012          RKDS=   12              ;DRIVE STATUS REGISTER
1169            000014          RKER=   14              ;ERROR REGISTER
1170            000016          RKASOF= 16              ;ATTENTION SUMMARY AND OFFSET REGISTER
1171            000020          RKDC=   20              ;DESIRED CYLINDER REGISTER
1172            000024          RKDB=   24              ;DATA BUFFER
1173            000026          RKMR1=  26              ;MAINTENANCE REGISTER 1
1174            000034          RKMR2=  34              ;MAINTENANCE REGISTER 2 (MESSAGE LINE A)
1175            000036          RKMR3=  36              ;MAINTENANCE REGISTER 3 (MESSAGE LINE B)
1176            000030          RKECPS= 30              ;ECC POSITION INFORMATION
1177            000032          RKECPT= 32              ;ECC PATTERN INFORMATION
1178
1179                            .SBTTL  CONTROL AND STATUS REGISTER 1 BITS (RKCS1:0)
1180
1181                            ;       DRIVE COMMANDS
1182
1183            000001          SELDRV= 1               ;SELECT DRIVE (GET STATUS)
1184            000003          PACK=   3               ;PACK ACKNOWLEDGE
1185            000005          CLEAR=  5               ;DRIVE CLEAR
1186            000007          UNLOAD= 7               ;UNLOAD
1187            000011          SRTSPL= 11              ;START SPINDLE
1188            000013          RECAL=  13              ;RECALIBRATE
1189            000015          OFFSET= 15              ;OFFSET
1190            000017          SEEK=   17              ;SEEK
1191            000021          RDDATA= 21              ;READ DATA
1192            000023          WRDATA= 23              ;WRITE DATA
1193            000025          RDHEAD= 25              ;READ HEADER
1194            000027          WRHEAD= 27              ;WRITE HEADER AND DATA
1195            000031          WRTCHK= 31              ;WRITE CHECK
1196
1197            000001          GO=     BIT0            ;GO BIT
1198            000100          IE=     BIT6            ;INTERRUPT ENABLE
1199            000200          RDY=    BIT7            ;CONTROLLER READY
1200            000400          BA16=   BIT8            ;BUS ADDRESS BIT 16
1201            001000          BA17=   BIT9            ;BUS ADDRESS BIT 17
1202            002000          CDT=    BIT10           ;CONTROLLER DRIVE TYPE (0=RK06,1=RK07)
1203            004000          CTO=    BIT11           ;CONTROLLER TIMEOUT
1204            010000          CFMT=   BIT12           ;CONTROLLER DRIVE FORMAT (0=22 SECTOR, 1=20 SECTOR)
1205            020000          DCPAR=  BIT13           ;SERCON PARITY ERROR DETECTED BY CONTROLLER
1206            040000          DI=     BIT14           ;DRIVE INTERRUPT
1207            100000          CERR=   BIT15           ;CONTROLLER ERROR
```

```
                100000          CCLR=   BIT15                   ;CONTROLLER CLEAR
                
                                .SBTTL  CONTROL AND STATUS REGISTER 2 BITS (RKCS2:10)
                
                000007          DRVMSK= 7                       ;MASK FOR DRIVE SELECTION CODE
                000010          RLS=    BIT3                    ;DESELECT OR RELEASE DRIVE IN BITS 0-2
                000020          BAI=    BIT4                    ;BUS ADDRESS INCREMENT INHIBIT
                000040          SCLR=   BIT5                    ;SUBSYSTEM CLEAR CONTROLLER AND ALL DRIVES
                000100          IR=     BIT6                    ;INPUT READY
                000200          OR=     BIT7                    ;OUTPUT READY
                000400          UFE=    BIT8                    ;UNIT FIELD ERROR
                001000          MDS=    BIT9                    ;MULTIPLE DRIVE SELECT
                002000          PGE=    BIT10                   ;PROGRAMMING ERROR
                004000          NEM=    BIT11                   ;NON-EXISTENT MEMORY
                010000          NED=    BIT12                   ;NON-EXISTENT DRIVE
                020000          UPE=    BIT13                   ;UNIBUS PARITY ERROR
                040000          WCE=    BIT14                   ;WRITE CHECK ERROR
                100000          DLT=    BIT15                   ;DATA LATE ERROR
                
                                .SBTTL  ERROR REGISTER BIT DEFINITION (RKER:14)
                
                000001          ILF=    BIT0                    ;ILLEGAL FUNCTION CODE
                000002          SKI=    BIT1                    ;SEEK INCOMPLETE
                000004          NXF=    BIT2                    ;NON-EXECUTABLE FUNCTION
                000010          DRPAR=  BIT3                    ;DRIVE DETECTED SERCON PARITY ERROR
                000020          FMTE=   BIT4                    ;FORMAT ERROR
                000040          DTYE=   BIT5                    ;DRIVE TYPE ERROR
                000100          ECH=    BIT6                    ;ECC HARD
                000200          BSE=    BIT7                    ;BAD SECTOR ERROR
                000400          HVRC=   BIT8                    ;HEADER VRC ERROR
                001000          COE=    BIT9                    ;CYLINDER ADDRESS OVERFLOW ERROR
                002000          IDAE=   BIT10                   ;INVALID DISK ADDRESS ERROR: HEAD/CYL
                004000          WLE=    BIT11                   ;WRITE LOCK ERROR
                010000          DTE=    BIT12                   ;DRIVE TIMING ERROR
                020000          OPI=    BIT13                   ;OPERATION (SEARCH) INCOMPLETE
                040000          UNS=    BIT14                   ;DRIVE UNSAFE
                100000          DCK=    BIT15                   ;DATA CHECK
                
                                .SBTTL  STATUS REGISTER BIT DEFINITION (RKDS:12)
                
                000001          DRA=    BIT0                    ;DRIVE AVAILABLE (CONTROLLER IS SET IF
                                                                ;   THIS BIT IS RESET)
                000004          OFST=   BIT2                    ;DRIVE OFFSET
                000010          ACLO=   BIT3                    ;AC LOW
                000020          DCLO=   BIT4                    ;DC LOW
                000040          DROT=   BIT5                    ;DRIVE OFF TRACK
                000100          VV=     BIT6                    ;VOLUME VALID
                000200          DRDY=   BIT7                    ;DRIVE READY
                000400          DDT=    BIT8                    ;DRIVE TYPE (0=RK06,1=RK07)
                004000          WRL=    BIT11                   ;WRITE LOCK
                020000          PIP=    BIT13                   ;POSITIONING IN PROGRESS
                040000          DSC=    BIT14                   ;DRIVE STATUS CHANGE
                100000          SVAL=   BIT15                   ;STATUS VALID
                
                                .SBTTL  MAINTENANCE REGISTER 1 BIT DEFINITION (RKMR1:22)
```

```
1264            000017                  MESMSK= 17                      ;MESSAGE MASK
1265            000020                  PAT=    BIT4                    ;FORCE EVEN PARITY ON SERCON MESSAGE LINES
1266            000040                  DMD=    BIT5                    ;DIAGNOSTIC MODE
1267            000100                  MSP=    BIT6                    ;MAINTENANCE SECTOR PULSE
1268            000200                  MIND=   BIT7                    ;MAINTENANCE INDEX
1269            000400                  MCLK=   BIT8                    ;MAINTENANCE CLOCK
1270            001000                  MERD=   BIT9                    ;MAINTENANCE ENCODED READ DATA
1271            002000                  MEWD=   BIT10                   ;MAINTENANCE ENCODED WRITE DATA
1272            004000                  PCA=    BIT11                   ;PRECOMPENSATION ADVANCE
1273            010000                  PCD=    BIT12                   ;PRECOMPENSATION DELAY
1274            020000                  ECCW=   BIT13                   ;ECC WORD IS BEING READ OR WRITTEN
1275            040000                  WRTGAT= BIT14                   ;WRITE GATE
1276            100000                  RDGATE= BIT15                   ;READ GATE
1277
1278                                    .SBTTL  DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE A (RKMR2:34)
1279
1280            000040                  D.DRA=  BIT5                    ;DRIVE AVAILABLE
1281            000100                  D.VV=   BIT6                    ;VOLUME VALID
1282            000200                  D.DRDY= BIT7                    ;DRIVE READY
1283            000400                  D.DDT=  BIT8                    ;DRIVE TYPE (0=RK06,1=RK07)
1284            001000                  D.FORM= BIT9                    ;DRIVE FORMAT
1285            002000                  D.OFF=  BIT10                   ;OFFSET ON
1286            004000                  D.WRL=  BIT11                   ;WRITE LOCK
1287            010000                  D.SPIN= BIT12                   ;SPINDLE ON
1288            020000                  D.PIP=  BIT13                   ;POSITIONING IN PROGRESS
1289            040000                  D.DSC=  BIT14                   ;DRIVE STATUS CHANGE
1290
1291                                    .SBTTL  DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE A (RKMR2:34)
1292
1293            000020                  D.SSP=  BIT4                    ;SERVO SIG PRES
1294            000040                  D.HDHM= BIT5                    ;HEADS HOME
1295            000100                  D.BRHM= BIT6                    ;BRUSHES HOME
1296            000200                  D.DOOR= BIT7                    ;DOOR INTERLOCKED
1297            000400                  D.CART= BIT8                    ;CARTRAGE INTERLOCK
1298            001000                  D.SPOK= BIT9                    ;SPEED OK
1299            002000                  D.FWD=  BIT10                   ;FORWARD
1300            004000                  D.REV=  BIT11                   ;REVERSE
1301            010000                  D.LOAD= BIT12                   ;HEADS LOADING
1302            020000                  D.RTZ=  BIT13                   ;RETURN TO ZERO
1303            040000                  D.UNLD= BIT14                   ;HEADS UNLOADING
1304
1305                                    .SBTTL  DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE B (RKMR3:36)
1306
1307            000040                  D.IDAE= BIT5                    ;INVALID DISK ADDRESS ERROR:HEAD/CYL
1308            000100                  D.ACLO= BIT6                    ;AC LOW
1309            000200                  D.FLT=  BIT7                    ;DRIVE FAULT
1310            000400                  D.ILF=  BIT8                    ;ILLEGAL FUNCTION CODE
1311            001000                  D.PAR=  BIT9                    ;DRIVE DETECTED SERCON PARITY ERROR
1312            002000                  D.SKI=  BIT10                   ;SEEK INCOMPLETE
1313            004000                  D.WLE=  BIT11                   ;WRITE LOCK ERROR
1314            010000                  D.SPLS= BIT12                   ;SPEED LOSS
1315            020000                  D.DROT= BIT13                   ;DRIVE OFF TRACK
1316            040000                  D.UNS=  BIT14                   ;R/W UNSAFE
1317
1318                                    .SBTTL  DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE B (RKMR3:36)
1319
```

```
1320    000020          D.SECT= BIT4                    ;SECTOR ERROR
1321    000040          D.WCUR= BIT5                    ;WRITE CURRENT AND NO WRITE GATE
1322    000100          D.WGAT= BIT6                    ;WRITE GATE AND NO TRANSISTIONS
1323    000200          D.HDFL= BIT7                    ;HEAD FAULT
1324    000400          D.MHD=  BIT8                    ;MULTIPLE HEAD SELECT
1325    001000          D.XERR= BIT9                    ;INDEX ERROR
1326    002000          D.TIB=  BIT10                   ;TRIBIT ERROR
1327    004000          D.PLO=  BIT11                   ;PLO ERROR
1328    010000          D.NMOV= BIT12                   ;SEEK AND NO MOTION
1329    020000          D.LIMD= BIT13                   ;LIMIT DETECT ON SEEK
1330    040000          D.SUNS= BIT14                   ;SERVO UNSAFE
1331
1332                    .SBTTL  COMMON MASKS AND OTHER BITS: MESSAGE A (RKMR2:34)
1333
1334    000007          M.DRV=  7                       ;DRIVE CODE, ALL BYTES
1335    077770          M.SER=  77770                   ;DRIVE SERIAL #, BYTE 11
1336
1337                    .SBTTL  COMMON MASKS AND OTHER BITS: MESSAGE B (RKMR3:36)
1338
1339    000003          M.ID=   3                       ;BYTE ID, ALL BYTES
1340    040000          M.ALGN= BIT14                   ;ALIGN SIGN, BYTE 10
1341    000760          M.SECT= 760                     ;SECTOR COUNT, BYTE 11
1342    007000          M.HEAD= 7000                    ;HEAD DECODE, BYTE 11
1343    100000          M.PAR=  BIT15                   ;PARITY, MESS A/B, ALL BYTES
```

```
                                .SBTTL  TRAP CATCHER

        000000                          .=0
                                ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
                                ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
                                ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
        000174                          .=174
000174  000000                 DISPREG: .WORD  0              ;;SOFTWARE DISPLAY REGISTER
000176  000000                 SWREG:   .WORD  0              ;;SOFTWARE SWITCH REGISTER
                                .SBTTL  STARTING ADDRESS(ES)
000200  000137  012556                  JMP    @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM
        000220                          .=220
000220  000137  012546                  JMP    PARSRT          ;INPUT ALL PARAMETERS & START TESTING

        000240                          .=240
000240  000137  055672                  JMP    0.ODT           ;ENTER ODT11

                                .SBTTL  ACT11 HOOKS

                                ;;**************************************************************
                                ;HOOKS REQUIRED BY ACT11
        000244                          $SVPC=.                ;SAVE PC
        000046                          .=46
000046  031636                  $ENDAD                         ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
        000052                          .=52
000052  100000                  .WORD   100000                 ;;2)SET LOC.52 TO 100000
        000244                  .=$SVPC                        ;; RESTORE PC
        001000                  .=1000
                                .SBTTL  APT PARAMETER BLOCK

                                ;;**************************************************************
                                ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                                ;;**************************************************************
        001000                          .$X=.   ;;SAVE CURRENT LOCATION
        000024                          .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
000024  000200                  200     ;;FOR APT START UP
        000044                  .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
000044  001000                  $APTHDR ;;POINT TO APT HEADER BLOCK
        001000                  .=.$X   ;;RESET LOCATION COUNTER
                                ;;**************************************************************
                                ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
                                ;INTERFACE SPEC.

001000                          $APTHD:
001000  000000                  $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
001002  001210                  $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
001004  000430                  $TSTM:  .WORD   280.    ;;RUN TIM OF LONGEST TEST
001006  001130                  $PASTM: .WORD   600.    ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
001010  001130                  $UNITM: .WORD   600.    ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
001012  000042                          .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)


                                .LIST   MD

                                ;
```

```
1400                            ;USE LOOP X TO OMIT SUBCLR
1401                            ;
1402
1403                            .MACRO  LOOP    A
1404                                    SCOP1
1405                                    MOV     #STACK,SP       ;RESTORE STK PTR
1406
1407                            .IF     B       A
1408                                    JSR     PC,SUBCLR
1409                                    ERROR   24              ;CERR AFTER SCLR
1410
1411                            .ENDC
1412                            .ENDM   LOOP
1413
1414                            ;
1415                            ;THIS MACRO FILLS EXPECTED MSG A0, B0, A1, B1, A2, B2 & B3 WITH STANDARD BITS
1416                            ;A=D.DSC AFTER ATTN OR 0 AFTER DRIVE CLEAR OR ANY IMPLIED SEEKS
1417                            ;NOTE: A CAN BE ANY BIT COMBINATION DESIRED.
1418                            ;
1419                            .MACRO  F.EAB   A
1420
1421                                    MOV     #<A!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
1422                                    CLR     E.B0            ;EXPECTED MSG B0
1423                                    MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
1424                                    MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
1425                                    CLR     E.A2            ;EXPECTED MSG A2
1426                                    MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
1427                                    MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
1428                            .ENDM   F.EAB
1429
1430                            ;
1431                            ;THIS MACRO ASSUMES DRIVE MSG A0, B0, A1, B1 WILL ALWAYS BE TESTED
1432                            ;USE    A,C,D,E FOR MSG A0, B0, A1, B1 ERROR NUMBERS RESP.
1433                            ;USE    G=T.A2  TO READ MSG A2 & PUT INFO INTO 'CYLDIF'
1434                            ;       H=T.B2  TO READ MSG B2 & PUT INFO INTO 'CYLADD'
1435                            ;       I=T.B3  TO READ MSG B3 & PUT INFO INTO 'SECTOR' & 'HEADA'
1436                            ;
1437                            ;USE    F=<ERROR DESCRIPTION>
1438                            ;
1439                            .MACRO  CHECK   A,C,D,E,F,G,H,I
1440
1441                                    JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
1442                                    .WORD   G!H!I           ;& MSGS SPECIFIED HERE
1443                                    ERROR   A               ;MSG A0 ERROR F
1444                                    ERROR   C               ;MSH B0 ERROR
1445                                    ERROR   D               ;MSG A1 ERROR
1446                                    ERROR   E               ;MSG B1 ERROR
1447                            .ENDM   CHECK
1448
1449                            ;
1450                            ;A=CYL DIFF/OFFSET ERROR #
1451                            ;B=CYL ADDR ERROR #
1452                            ;C=<ERROR DESCRIPTION>
1453                            ;
1454                            .MACRO  CWD2    A,B,C,?D,?E
1455
```

```
                        MOV     #2,RKMR1(R5)    ;SELECT WORD 2
                        JSR     PC,GSTAT
                        TST     CYLDIF          ;SEE IF MSG A2=0
                        BEQ     D               ;BR IF YES
                        ERROR   A               ;MSG A2 NOT CLEARED C
                D:      TST     CYLADD          ;SEE IF MSG B2=0
                        BEQ     E               ;BR IF YES
                        ERROR   B               ;MSG B2 NOT CLEARED C
                E:
                .ENDM   CWD2

                .MACRO  DRCLR   ?A

                        MOV     #CCLR,RKCS1(R5)
                        MOV     $UNIT,RKCS2(R5) ;DRIVE#
                        MOV     #CLEAR,HCS1
                        JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
                        ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
                        JSR     PC,TSTATN       ;TEST FOR ATTN
                        BR      A
                        ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
                A:

                .ENDM   DRCLR


                ;
                ;USE CALIB      X   TO OMIT CHECKING MSGS A0, B0, A1, B1, A2 & B2
                ;
                .MACRO  CALIB   A,?C

                        MOV     #CCLR,RKCS1(R5)
                        MOV     $UNIT,RKCS2(R5)
                        MOV     #RECAL,HCS1
                        JSR     PC,DOCMD        ;DO RECAL CMD & GET CONTR RDY
                        ERROR   124             ;RDY NOT SET AFTER RECAL CMD

                        MOV     #1,RKMR1(R5)    ;SELECT WORD 1
                        JSR     PC,GSTAT
                        BIT     #D.RTZ,HMR2
                        BNE     C
                        ERROR   244             ;RTZ NOT SET DURING RECAL CMD
                C:      MOV     T10,TEMP2       ;SETUP TIMEOUT
                        JSR     PC,FATT1        ;FIND ATTN
                        ERROR   55              ;NO ATTN AFTER RECAL CMD
                .IF B   A
                        F.EAB   DSC
                        CHECK   221,275,222,276,<AFTER RECAL CMD>,T.A2,T.B2,T.B3
                        CWD2    47,50,<AFTER RECAL CMD>
                .ENDC
                        DRCLR

                .ENDM   CALIB

                ;
                ;QUICK START SPINDLE
```

```
1512                                    ;
1513                                    .MACRO  QKSRT
1514
1515                                            JSR     PC,SUBCLR
1516                                            ERROR   24              ;CERR AFTER SCLR
1517
1518                                            MOV     #SRTSPL,HCS1
1519                                            JSR     PC,DOCMD        ;DO START SPINDLE CMD & GET CONTR RDY
1520                                            ERROR   121             ;RDY NOT SET AFTER ST SPIN CMD.
1521
1522                                            MOV     T100,TEMP2      ;SETUP TIMEOUT
1523                                            JSR     PC,FATT1        ;FIND ATTN
1524                                            ERROR   74              ;NO ATTN AFTER ST SPIN CMD.
1525
1526                                            CLR     UNLD
1527
1528                                    .ENDM   QKSRT
1529
1530                                    ;
1531                                    ; A=WRHEAD/<CFMT!WRHEAD>
1532                                    ; USE WRHDR     <A>,X   TO OMIT CHECKING A0, B0, A1 & B1
1533                                    ;
1534                                    .MACRO  WRHDR   A,C,K,?D
1535
1536                                            MOV     #<A>,HCS1
1537                                            JSR     PC,DATCMD       ;DO DATA X FOR CMD & GET CONTR RDY
1538                                            ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
1539                                    .IF     B       K
1540                                            JSR     PC,GSTAT        ;GET FRESH STATUS
1541                                    .ENDC
1542                                    .IF     NB      K
1543                                            MOV     #<CFMT!SELDRV>,HCS1
1544                                            JSR     PC,DOCMD
1545                                            ERROR   117             ;NO RDY AFTER SELDRV CMD
1546                                    .ENDC
1547                                            BIT     #CERR,HCS1
1548                                            BEQ     D
1549                                            ERROR   201             ;CERR AFTER WRITE HEADER CMD
1550                                            TYPE    ,MSG26          ;ABORTING BAL OF TESTS
1551                                            JMP     $EOP
1552                                    D:
1553                                    .IF B   C
1554                                            F.EAB   0
1555                                            CHECK   277,267,300,270,<AFTER WRITE HEADER CMD>,0,0,0
1556                                    .ENDC
1557
1558                                    .ENDM   WRHDR
1559
1560                                    ;
1561                                    ; A=RDHEAD/<CFMT!RDHEAD>
1562                                    ;USE RDHDR <A>,X TO OMIT CHECKING A0, B0, A1, B1
1563                                    ;
1564                                    .MACRO  RDHDR   A,C,?D,?E
1565
1566                                            MOV     #RHTAB,R0
1567                                            MOV     #<A>,HCS1
```

```
1568                                    JSR     PC,DATCMD       ;DO DATA X FOR CMD & GET CONTR RDY
1569                                    ERROR   171             ;NO RDY AFTER READ HEADER CMD
1570                                    BIT     #CERR,HCS1
1571                                    BEQ     D
1572                                    ERROR   174             ;CERR AFTER READ HEADER CMD
1573                                    TYPE    ,MSG26          ;ABORTING BAL OF TESTS
1574                                    JMP     $EOP
1575
1576                            D:      MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
1577                                    MOV     RKDB(R5),(R0)+  ;2'ND WORD
1578                                    MOV     RKDB(R5),(R0)+  ;3'RD WORD
1579
1580
1581                                    BIT     #DLT,RKCS2(R5)
1582                                    BEQ     E
1583                                    JSR     PC,GSTAT
1584                                    ERROR   173             ;DLT AFTER READ HEADER CMD
1585                                    TYPE    ,MSG26          ;ABORTING BAL OF TESTS
1586                                    JMP     $EOP
1587                            E:
1588                            .IF     B       C
1589                                    F.EAB   0
1590                                    CHECK 301,271,302,272,<AFTER READ HEADER CMD>,T.A2,T.B2,0
1591                            .ENDC
1592
1593                            .ENDM   RDHDR
1594
1595
1596                            .MACRO  HDCHK3  ?A
1597
1598                                    RDHDR   RDHEAD
1599                                    CMP     RHTAB,TOCYL     ;CHECK WORD 0 ONLY, CYL#
1600                                    BEQ     A               ;BR IF SAME
1601                                    ERROR   51              ;WRONG CYL# ON HEADER
1602                            A:
1603
1604                            .ENDM   HDCHK3
1605
1606                            ;
1607                            ; A=TOCYL/FRCYL , B=HEAD#, C = 0 FOR 22 SECTOR, 1 FOR 20 SECTOR
1608                            ;
1609                            .MACRO  HDTBL   A,B,C
1610
1611                                    MOV     A,CALADD        ;SETUP
1612                                    MOV     #B,HEAD         ;TO FILL
1613                                    MOV     #C,FORMAT       ;HEADER
1614                                    JSR     PC,FHDTAB       ;TABLE
1615
1616                            .ENDM   HDTBL
1617
1618                            ;
1619                            ;QUICK SEEK.    ENTER WITH CYL# IN RKDC
1620                            ;
1621                            .MACRO  QKSEEK  ?A
1622
1623                                    MOV     #SEEK,HCS1
```

```
1624                                    JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
1625                                    ERROR   131             ;NO RDY AFTER SEEK CMD
1626
1627                                    MOV     T50000,TEMP1    ;SETUP TIMEOUT
1628                                    JSR     PC,FATT2        ;FIND ATTN
1629                                    ERROR   132             ;NO ATTN AFTER SEEK CMD
1630
1631                                    BIT     #CERR,HCS1
1632                                    BEQ     A
1633                                    ERROR   210             ;CERR AFTER SEEK CMD
1634
1635                                A:
1636
1637                                .ENDM   QKSEEK
1638
1639
1640                                ;
1641                                ;A=WRDATA/<CFMT!WRDATA>
1642                                ;C=ADDR TO JMP TO ATTEMPT TO WRITE ON ANOTHER SECTOR
1643                                ;D=ADDR TO JMP TO BYPASS TEST
1644                                ;E: IF BLANK WILL CHECK A0, B0, A1 & B1 AT THE END OF WRITING
1645                                ;E: IF NON BLANK WILL OMIT CHECKING A0 THRU B1
1646                                ;
1647
1648                                .MACRO  WDATA   A,C,D,E,J,K,?F,?G,?H,?I
1649
1650                                        MOV     #<A>,HCS1
1651                                        JSR     PC,DATCMD       ;DO DATA X FOR CMD & GET CONTR RDY
1652                                        ERROR   11              ;NO RDY AFTER WRITE DATA CMD
1653                                .IF     B       K
1654                                        JSR     PC,GSTAT        ;GET FRESH STATUS
1655                                .ENDC
1656                                .IF     NB      K
1657                                        MOV     #<CFMT!SELDRV>,HCS1
1658                                        JSR     PC,DOCMD
1659                                        ERROR   117             ;NO RDY AFTER SELDRV CMD
1660                                .ENDC
1661                                        BIT     #CERR,HCS1
1662                                        BEQ     I               ;BR IF NO ERRORS
1663
1664                                        BIT     #BSE,HER        ;SEE IF BAD SECTOR FLAG
1665                                        BEQ     G               ;BR IF NO
1666                                        JSR     PC,TRUERR       ;ELSE SEE IF SECTOR LISTED IN BSE TABLE
1667                                        BR      H               ;RETURN HERE IF NO
1668                                .IF     B       J
1669
1670                                        INC     SECTOR          ;RETURN HERE IF YES
1671                                        CMP     SECTOR,#10.     ;ARE 10 CONSEC. SECTORS BAD
1672                                        BNE     F               ;BR IF NO
1673                                        ERROR   46              ;ABORTING TEST DETECTED 10 BAD SECTORS
1674                                        JMP     D               ;BYPASS TEST
1675                                F:      MOV     #CCLR,RKCS1(R5) ;TRY ANOTHER SECTOR
1676                                        JMP     C
1677                                .ENDC
1678                                .IF     NB      J
1679                                        JMP     3$              ;RET HERE IF YES
```

```
1680                                    .ENDC
1681                                    G:      ERROR   12              ;CERR WITH WRITE DATA CMD
1682                                            F.EAB   0
1683                                            CHECK   52,23,53,25,<AFTER WRITE DATA CMD>,T.A2,T.B2,0
1684                                            TYPE    ,MSG26          ;ABORTING BAL OF TESTS
1685                                            JMP     $EOP
1686                                    H:      ERROR   63              ;BAD SECTOR NOT LISTED IN TABLE
1687                                    I:
1688                                    .IF     B       E
1689                                            F.EAB   0
1690                                            CHECK   52,23,53,25,<AFTER WRITE DATA CMD>,T.A2,T.B2,0
1691                                    .ENDC
1692                                    .ENDM   WDATA
1693
1694
1695                                    ;
1696                                    ;A=RDDATA/<CFMT!RDDATA>
1697                                    ;USE RDATA      <A>,X TO OMIT CHECKING A0, B0, A1 & B1
1698                                    ;
1699
1700                                    .MACRO  RDATA   A,C,K,?D,?E,?F,?G,?H
1701
1702                                            MOV     #<A>,HCS1
1703                                            JSR     PC,DATCMD       ;DO DATA X FOR CMD & GET CONTR RDY
1704                                            ERROR   13              ;NO RDY AFTER READ DATA CMD
1705                                    .IF     B       K
1706                                            JSR     PC,GSTAT        ;GET FRESH STATUS
1707                                    .ENDC
1708                                    .IF     NB      K
1709                                            MOV     #<CFMT!SELDRV>,HCS1
1710                                            JSR     PC,DOCMD
1711                                            ERROR   117             ;NO RDY AFTER SELDRV CMD
1712                                    .ENDC
1713                                            BIT     #CERR,HCS1
1714                                            BEQ     G
1715                                            BIT     #BSE,HER        ;SEE IF BAD SECTOR
1716                                            BEQ     E
1717                                            ERROR   65              ;DETECTED BSE IN READ BUT NOT IN WRITE CMD.
1718                                            BR      H
1719                                    D:      TYPE    ,MSG26          ;ABORTING BAL OF TESTS
1720                                            JMP     $EOP
1721
1722                                    E:      BIT     #DCK,HER        ;SEE IF DATA CHECK ERROR
1723                                            BEQ     F
1724                                            ERROR   21              ;DATA CHECK ERROR AFTER READ CMD (ECC)
1725                                            BR      H
1726
1727                                    F:      ERROR   14              ;CERR AFTER READ DATA CMD.
1728
1729                                    H:      F.EAB   0
1730                                            CHECK   54,26,56,30,<AFTER READ DATA CMD>,T.A2,T.B2,0
1731                                            BR      D
1732                                    G:
1733                                    .IF     B       C
1734                                            F.EAB   0
1735                                            CHECK   54,26,56,30,<AFTER READ DATA CMD>,T.A2,T.B2,0
```

```
1736                                    .ENDC
1737                                    .ENDM   RDATA
1738
1739
1740                                    ;
1741                                    ;A=WRTCHK/<CFMT!WRTCHK>
1742                                    ;C=EXPECTED DATA FOR TYPEOUT
1743                                    ;USE WRCHK      <A>,DATA0,X TO OMIT CHECKING A0, B0, A1 & B1
1744                                    ;
1745
1746                                    .MACRO  WRCHK   A,C,D,K,?E,?F
1747
1748                                            MOV     #<A>,HCS1
1749                                            JSR     PC,DATCMD       ;DO DATA X FOR CMD & GET CONTR RDY
1750                                            ERROR   15              ;NO RDY AFTER WRITE CHECK CMD
1751                                    .IF     B       K
1752                                            JSR     PC,GSTAT        ;GET FRESH STATUS
1753                                    .ENDC
1754                                    .IF     NB      K
1755                                            MOV     #<CFMT!SELDRV>,HCS1
1756                                            JSR     PC,DOCMD
1757                                            ERROR   117             ;NO RDY AFTER SELDRV CMD
1758                                    .ENDC
1759                                            BIT     #CERR,HCS1
1760                                            BEQ     F
1761                                            BIT     #WCE,HCS2       ;SEE IF WRITE CHECK ERROR
1762                                            BEQ     E
1763                                            MOV     RKDB(R5),WD1    ;ACTUAL WORD FOR PRINTOUT
1764                                            MOV     C,WD2           ;EXPECTED WORD FOR TYPEOUT
1765                                            ERROR   16              ;WCE AFTER WRITE CMD
1766                                            BR      F
1767
1768                                    E:      ERROR   22              ;CERR AFTER WRITE CHECK CMD
1769                                            F.EAB   0
1770                                            CHECK   57,31,60,32,<AFTER WRITE CHECK CMD>,T.A2,T.B2,0
1771                                            TYPE    ,MSG26          ;ABORTING BAL OF TESTS
1772                                            JMP     $EOP
1773
1774                                    F:
1775                                    .IF     B       D
1776                                            F.EAB   0
1777                                            CHECK   57,31,60,32,<AFTER WRITE CHECK CMD>,T.A2,T.B2,0
1778                                    .ENDC
1779                                    .ENDM   WRCHK
1780
1781
1782                                    .MACRO  OFSET ?A
1783
1784                                            MOV     #A,$ESCAPE
1785                                            MOV     #OFFSET,HCS1
1786                                            JSR     PC,DOCMD        ;DO RECAL CMD & GET CONTR RDY
1787                                            ERROR   33              ;NO RDY AFTER OFFSET CMD
1788
1789                                            MOV     #<D.PIP!D.SPIN!D.OFF!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
1790                                            CLR     E.B0
1791                                            MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
```

```
                        MOV     #1,E.B1
                        CHECK   35,61,36,62,<DURING OFFSET CMD>,0,0,0

                A:      CLR     $ESCAPE
                        MOV     T5000,TEMP1     ;SETUP TIMEOUT
                        JSR     PC,FATT2        ;FIND ATTN
                        ERROR   34              ;NO ATTN AFTER OFFSET CMD

                        F.EAB   <D.DSC!D.OFF>
                        CHECK   260,261,37,40,<AFTER OFFSET CMD>,T.A2,T.B2,0

                .ENDM   OFSET


                .MACRO  EOPGM

                        SCOPE
                        CLR     $ESCAPE
                        MOV     #1,$TIMES
                        MOV     #STACK,SP
                        INC     $DEVCT          ;INCR COUNT FOR # DRIVES CHECKED
                        CMP     DRIVS,$DEVCT    ;ARE ALL DRIVES PRESENT TESTED?
                        BEQ     1$              ;BR IF YES
                        CLR     BSERR           ;CLEAR BAD SECTOR ERROR FLAG
                        JMP     NUDRV           ;ELSE TEST NEXT DRIVE PRESENT
                1$:     CLR     BSERR           ;CLEAR BAD SECTOR ERROR FLAG
                        BR      $EOP1+2
                $EOP1:  SCOPE
                .ENDM   EOPGM
                ;
                ;A= ERROR #
                ;B = ERROR CONDITION
                ;
                .MACRO  OFFDIR  A,B,?C,?D
                        MOV     R2,RKASOF(R5)   ;REFRESH RKASOF

                        BIT     #BIT7,R2
                        BNE     C               ;BR IF NEG OFFSET

                        CMP     R2,CYLDIF       ;CHECK POS OFFSET
                        BEQ     D
                        ERROR   A               ;OFFSET IN A2 NOT = RKASOF
                        BR      D               ;B

                C:      CMP     R1,CYLDIF       ;CHECK NEG OFFSET
                        BEQ     D
                        ERROR   A               ;OFFSET IN A2 NOT = RKASOF
                                                ;B
                D:
                .ENDM   OFFDIR

                        .NLIST  MD
```

```
                                .SBTTL  COMMON TAGS

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

                        001100          .=1100
001100                  $CMTAG:                         ;;START OF COMMON TAGS
001100  000000                  .WORD   0
001102     000          $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001103     000          $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001104  000000          $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001106  000000          $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001110  000000          $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001112  000000          $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001114     000          $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001115     001          $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001116  000000          $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001120  000000          $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001122  000000          $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001124  000000          $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001126  000000          $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001130  000000                  .WORD   0               ;;RESERVED--NOT TO BE USED
001132  000000                  .WORD   0
001134     000          $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001135     000          $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001136  000000                  .WORD   0
001140  177570          SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001142  177570          DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
001144  177560          $TKS:   177560                  ;;TTY KBD STATUS
001146  177562          $TKB:   177562                  ;;TTY KBD BUFFER
001150  177564          $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
001152  177566          $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001154     000          $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001155     002          $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001156     012          $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A 'LINE FEED"
001157     000          $TPFLG: .BYTE   0               ;;'TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001160  000000          $TMP0:  .WORD   0               ;;USER DEFINED
001162  000000          $TMP1:  .WORD   0               ;;USER DEFINED
001164  000000          $TMP2:  .WORD   0               ;;USER DEFINED
001166  000000          $TMP3:  .WORD   0               ;;USER DEFINED
001170  000000          $TMP4:  .WORD   0               ;;USER DEFINED
001172  000000          $TMP5:  .WORD   0               ;;USER DEFINED
001174  000000          $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
001176  000000          $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
001200  177607  000377  $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
001204     077          $QUES:  .ASCII  /?/             ;;QUESTION MARK
001205     015          $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
001206  000012          $LF:    .ASCIZ  <12>            ;;LINE FEED
                                ;;*****************************************************************
                                .SBTTL  APT MAILBOX-ETABLE

                                ;;*****************************************************************
                                .EVEN
001210                  $MAIL:                          ;;APT MAILBOX
001210  000000          $MSGTY: .WORD   AMSGTY          ;;MESSAGE TYPE CODE
```

```
1901  001212  000000                $FATAL: .WORD   AFATAL  ;;FATAL ERROR NUMBER
1902  001214  000000                $TESTN: .WORD   ATESTN  ;;TEST NUMBER
1903  001216  000000                $PASS:  .WORD   APASS   ;;PASS COUNT
1904  001220  000000                $DEVCT: .WORD   ADEVCT  ;;DEVICE COUNT
1905  001222  000000                $UNIT:  .WORD   AUNIT   ;;I/O UNIT NUMBER
1906  001224  000000                $MSGAD: .WORD   AMSGAD  ;;MESSAGE ADDRESS
1907  001226  000000                $MSGLG: .WORD   AMSGLG  ;;MESSAGE LENGTH
1908  001230                        $ETABLE:                ;;APT ENVIRONMENT TABLE
1909  001230     000                $ENV:   .BYTE   AENV    ;;ENVIRONMENT BYTE
1910  001231     000                $ENVM:  .BYTE   AENVM   ;;ENVIRONMENT MODE BITS
1911  001232  000000                $SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
1912  001234  000000                $USWR:  .WORD   AUSWR   ;;USER SWITCHES
1913  001236  000000                $CPUOP: .WORD   ACPUOP  ;;CPU TYPE,OPTIONS
1914                                :*                      BITS 15-11=CPU TYPE
1915                                :*                              11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
1916                                :*                              11/70=06,PDQ=07,Q=10
1917                                :*                      BIT 10=REAL TIME CLOCK
1918                                :*                      BIT  9=FLOATING POINT PROCESSOR
1919                                :*                      BIT  8=MEMORY MANAGEMENT
1920  001240     000                $AMS1:  .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
1921  001241     000                $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
1922                                :*                      MEM.TYPE BYTE   --  (HIGH BYTE)
1923                                :*                              900 NSEC CORE=001
1924                                :*                              300 NSEC BIPOLAR=002
1925                                :*                              500 NSEC MOS=003
1926  001242  000000                $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
1927                                :*                        MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
1928  001244     000                $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
1929  001245     000                $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
1930  001246  000000                $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
1931  001250     000                $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
1932  001251     000                $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
1933  001252  000000                $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
1934  001254     000                $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
1935  001255     000                $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
1936  001256  000000                $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
1937  001260  000000                $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
1938  001262  000000                $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
1939  001264  177440                $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
1940  001266  000000                $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
1941  001270  000000                $CDW1:  .WORD   ACDW1   ;;CONTROLLER DESCRIPTION WORD#1
1942  001272  000000                $CDW2:  .WORD   ACDW2   ;;CONTROLLER DESCRIPTION WORD#2
1943  001274  000000                $DDW0:  .WORD   ADDW0   ;;DEVICE DESCRIPTOR WORD#0
1944  001276  000000                $DDW1:  .WORD   ADDW1   ;;DEVICE DESCRIPTOR WORD#1
1945  001300  000000                $DDW2:  .WORD   ADDW2   ;;DEVICE DESCRIPTOR WORD#2
1946  001302  000000                $DDW3:  .WORD   ADDW3   ;;DEVICE DESCRIPTOR WORD#3
1947  001304  000000                $DDW4:  .WORD   ADDW4   ;;DEVICE DESCRIPTOR WORD#4
1948  001306  000000                $DDW5:  .WORD   ADDW5   ;;DEVICE DESCRIPTOR WORD#5
1949  001310  000000                $DDW6:  .WORD   ADDW6   ;;DEVICE DESCRIPTOR WORD#6
1950  001312  000000                $DDW7:  .WORD   ADDW7   ;;DEVICE DESCRIPTOR WORD#7
1951  001314                        $ETEND:
1952                                .MEXIT
1953          177440                ABASE=  177440                  ;DEFAULT BUSS ADDRESS
1954  001314  000210                RKVEC:  210                     ;DEFAULT CONTROLLER INTERRUPT VECTOR
1955  001316  000240                RKPRI:  PR5                     ;PRIORITY
1956  001320  172540                PKS:    172540                  ;P-CLOCK STATUS REG
```

```
1957  001322  172542          PKSB:   172542          ;P-CLOCK SET BUFFER
1958  001324  172544          PKRB:   172544          ;P-CLOCK READ BUFFER
1959  001326  177546          LKS:    177546          ;L-CLOCK STATUS REG.
1960
1961  001330  000100          LCVEC:  100             ;L-CLOCK INTERRUPT VECTOR
1962  001332  000104          PCVEC:  104             ;P-CLOCK INTERRUPT VECTOR.
1963
1964          000114          MEMVEC= 114             ;MEMORY PARITY VECTOR
1965          172100          MEMBAS= 172100          ;MEMORY PARITY OPTION CSR START ADDR
1966  001334  000000          TRAPPC: 0               ;PC FOR MEM CHECK ENABLE TRAP
1967
1968  001336  000000          PARAM:  0               ;1 FOR 220 START, NO DEFAULT
1969  001340  000000          FTITLE: 0               ;FLAG FOR PRINTING OUT 1ST PROGRAM TITLE
1970
1971  001342  000000          DRVPTR: 0               ;CONTAINS THE POINTER TO THE DRIVE FLAG
1972                                                  ;(DRIV0-DRIV7) OF THE DRIVE TO BE CHECKED NEXT.
1973  001344  000000          FRCYL:  0               ;FROM CYLINDER
1974  001346  000000          TOCYL:  0               ;TO CYLINDER
1975  001350  000000          CCYL:   0               ;CURRENT CYL, USED IN N SQUARE TEST
1976  001352  000000          PCYL:   0               ;PREV CYL., USED IN N SQUARE TEST
1977  001354  000000          CALDIF: 0               ;CALC CYL DIFF USED IN N SQUARE TEST
1978  001356  000000          CYLDIF: 0               ;CYL DIFF, RIGHT JUSTIFIED FROM RKMR3
1979  001360  000000          CYLADD: 0               ;CYL ADDR, RIGHT JUSTIFIED FROM RKMR3
1980  001362  000000          CALADD: 0               ;CYL ADDR USED IN FHDTAB ROUTINE
1981
1982  001364  000074          HZ:     60.             ;60 FOR 60 CPS
1983                                                  ;50 FOR 50 CPS
1984  001366  000000          COUNT:  0               ;LOADED TO 50 OR 60 TO COUNT TO 1 SEC
1985                                                  ;OR ANY OTHER NUMBER TO COUNT OFF FRACTIONAL SECOND
1986  001370  000000          SEC:    0               ;SECOND COUNTER
1987  001372  000000          TIMUP:  0               ;FLAG TO INDICATE TIME IS UP
1988  001374  000000          SECNT:  0               ;SECTOR COUNT
1989  001376  000000          PSEC:   0               ;PREVIOUS SECTOR
1990  001400  000000          ESEC:   0               ;EXPECTED SECTOR
1991  001402  0C0000          SECTOR: 0               ;SECTOR COUNT, RIGHT JUSTIFIED FROM RKMR3
1992
1993  001404  000001          T1:     1               ;TIMEOUT CONSTANTS
1994  001406  000012          T10:    10.
1995  001410  000062          T50:    50.
1996  001412  000764          T500:   500.
1997  001414  000144          T100:   100.
1998  001416  011610          T5000:  5000.
1999  001420  141520          T50000: 50000.
2000
2001  001422  000077          CYL:    63.             ;CYLINDER NUMBERS USED IN
2002  001424  000177                  127.            ;CURRENT CROSSOVER TEST FOR RK06
2003  001426  000277                  191.
2004  001430  000377                  255.
2005  001432  000477                  319.
2006  001434  000577                  383.
2007
2008  001436  000177          CYL7:   127.            ;FOR RK07
2009  001440  000377                  255.
2010  001442  000577                  383.
2011  001444  000777                  511.
2012  001446  001177                  639.
```

```
 2013  001450  001377                    767.
 2014
 2015  001452  000000            WD1:    0                ;ACTUAL HEADER/DATA WORD
 2016  001454  000000            WD2:    0                ;EXPECTED DATA WORD
 2017
 2018  001456  000000            OFFERR: 0                ;SET WHEN WRITE CHECK ERROR ON OFFSET
 2019
 2020
 2021  001460  000000            HEAD:   0                ;HEAD NUMBER
 2022  001462  000000            HEADA:  0                ;HEAD # FROM H.B3, RT. JUSTIFIED
 2023  001464  000000            HD1:    0                ;SHIFTED HEAD# FOR FORMATTER ROUTINE
 2024  001466  000000            FORMAT: 0                ;FORMAT TYPE
 2025  001470  000000            FMT1:   0                ;SHIFTED FORMAT FOR FORMATTER ROUTINE
 2026  001472  000000            WDCNT:  0                ;WORD COUNT
 2027
 2028  001474  000000            DATA0:  0                ;ALL 0'S
 2029  001476  052525            DATA01: 52525            ;0101 PATT
 2030  001500  177777            DATA1:  177777           ;ALL 1'S
 2031  001502  133467            DPAT1:  133467
 2032  001504  070627            DPAT2:  70627
 2033
 2034  001506  000000            WORD:   0                ;HEADER/DATA WORD
 2035  001510  000000            HDWD:   0                ;HEADER WORD FROM RKDB
 2036                                                      ,
 2037  001512  000000            BSERR:  0                ;CANNOT READ BSE INFO WHEN SET
 2038  001514  000000            LIMERR: 0                ;LIMIT DETECT ERROR FLAG
 2039  001516  000000            BYPCERR:0                ;SET TO 1 TO BYPASS CKCERR IN 'GSTAT1'
 2040  001520  000000            CHKFLG: 0                ;WORDS TO BE TESTED
 2041
 2042  001522  000102            HDTAB:  .BLKW 66.        ;CALCULATED HEADER WORD TABLE
 2043  001726  000102            RHTAB:  .BLKW 66.        ;FILLED AFTER READ HEADER CMD
 2044  002132  000102            SRTTAB: .BLKW 66.        ;ABOVE RHTAB SORTED STARTING FORM
 2045                                                      ;SECTOR 0 BY SORT ROUTINE
 2046  002336  000400            BSE20H: .BLKW 256.       ;20 SECTOR HARDWARE BSE INFO
 2047  003336  000400            BSE22H: .BLKW 256.       ;22 SECTOR HARDWARE BSE INFO.
 2048  004336  000400            BSE20S: .BLKW   256.     ;20 SECTOR SOFTWARE BSE INFO.
 2049  005336  000400            BSE22S: .BLKW   256.     ;22 SECTOR SOFTWARE BSE INFO.
 2050  006336  000400            RDTAB:  .BLKW 256.       ;FILLED AFTER READ DATA CMD
 2051
 2052  007336  000000            UNLD:   0                ;SET TO 0 IF HEADS ARE LOADED
 2053                                                      ;SET TO 1 IF HEADS UNLOADED
 2054  007340  000000            BADHDR: 0                ;SET TO 0 IF FORMATTING OK
 2055                                                      ;SET TO 1 IF FORMATTING ALTERED
 2056  007342  000000            HPEND:  0                ;SET TO 0 IF HALT NOT PENDING
 2057                                                      ;SET TO 1 IF HALT PENDING
 2058
 2059                                                      ;THE ABOVE 3 FLAGS ARE USED
 2060                                                      ;BY 'STOP' ROUTINE TO BRING
 2061                                                      ;THE CPU TO A VALID HALT.
 2062
 2063
 2064  007344    001   002   004  ATTN:   .BYTE 1,2,4,10,20,40,100,200    ;ATN 0-7 RESP.
 2065  007347    010   020   040
 2066  007352    100   200
 2067                                    .EVEN
 2068
```

```
                                ;
                                ;THE FOLLOWING ARE HOLDING REGISTERS FOR THE RK611 REGISTERS
                                ;THEY ARE LOADED AFTER RDY IS REC'D FROM WRDY ROUTINE.
                                ;
007354  000000                  HCS1:   0                       ;HOLD RKCS1
007356  000000                  HCS2:   0                       ;HOLD RKCS2
007360  000000                  HWC:    0                       ;HOLD RKWC
007362  000000                  HBA:    0                       ;ETC.
007364  000000                  HDA:    0
007366  000000                  HDS:    0
007370  000000                  HER:    0
007372  000000                  HASOF:  0
007374  000000                  HDC:    0
007376  000000                  HDB:    0
007400  000000                  HMR1:   0
007402  000000                  HMR2:   0
007404  000000                  HMR3:   0
007406  000000                  HPOS:   0
007410  000000                  HPAT:   0

007412  000000                  TEMP1:  0                       ;TEMPORARY STORAGE.
007414  000000                  TEMP2:  0
007416  000000                  TEMP3:  0
007420  000000                  TEMP4:  0
007422  000000                  TEMP5:  0
                                ;
                                ;THE FOLLOWING ARE HOLDING REGISTERS FOR MSGA(0-3) & MSGB(0-3).
                                ;
007424  000000                  H.A0:   0
007426  000000                  H.B0:   0
007430  000000                  H.A1:   0
007432  000000                  H.B1:   0
007434  000000                  H.A2:   0
007436  000000                  H.B2:   0
007440  000000                  H.A3:   0
007442  000000                  H.B3:   0
                                ;
                                ;THE FOLLOWING ARE 'EXPECTED' REGISTER FOR THE ABOVE.
                                ;
007444  000000                  E.A0:   0
007446  000000                  E.B0:   0
007450  000000                  E.A1:   0
007452  000000                  E.B1:   0
007454  000000                  E.A2:   0
007456  000000                  E.B2:   0
007460  000000                  E.A3:   0
007462  000000                  E.B3:   0
                                ;
                                ;THE FOLLOWING ARE IDENTIFIERS FOR DRIVE MSG WORDS TO BE TESTED.
                                ;
        000001                  T.A2=BIT0                       ;TEST MSG A2 IF SET
        000002                  T.B2=BIT1
        000004                  T.B3=BIT2
```

```
                                ;
                                ;ALL THE FLAGS BELOW ARE CLEARED INITIALLY BY THE CLRFLG ROUTINE.
                                ;
007464  000000                  DDUMP:  0                       ;FLAG - SET WHEN IN DDP DUMP MODE
007466  000000                  DDPCH:  0                       ;FLAG - SET WHEN IN DDP CHAIN MODE
007470  000000                  ACT11:  0                       ;FLAG - SET WHEN IN ACT11 MODE OF OPERATION
007472  000000                  PPTP:   0                       ;FLAG - SET WHEN PROGRAM LOADED BY PAPER TAPE
007474  000000                  DRIVS:  0                       ;CONTAINS THE NUMBER OF DRIVES PRESENT

                                ;THE FLAGS BELOW ARE SET TO 1 TO INDICATE THAT A PARTICULAR DRIVE
                                ;IS PRESENT AND IS TO BE TESTED.

007476  000000                  DRIV0:  0                       ;FLAG SET TO 1 WHEN DRIVE 0 PRESENT
007500  000000                  DRIV1:  0                       ;FOR DRIVE 1
007502  000000                  DRIV2:  0                       ;FOR DRIVE 2
007504  000000                  DRIV3:  0                       ;FOR DRIVE 3
007506  000000                  DRIV4:  0                       ;FOR DRIVE 4
007510  000000                  DRIV5:  0                       ;FOR DRIVE 5
007512  000000                  DRIV6:  0                       ;FOR DRIVE 6
007514  000000                  DRIV7:  0                       ;FOR DRIVE 7

007516  000000                  LCLKF:  0                       ;L-CLOCK FLAG PRESENT FLAG
007520  000000                  PCLKF:  0                       ;P-CLOCK FLAG PRESENT FLAG
007522  000000                  DOTIM:  0                       ;SET IF EITHER CLOCK PRESENT FOR TIMING TESTS.
007524  000000                  SIZFLG: 0                       ;SET IF DEFAULT DO SIZING IN TEST 1
```

```
                                .SBTTL  ERROR POINTER TABLE

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

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


007526                          $ERRTB:

                                ;ERROR 1
007526  046405                          EM2                     ;DR # IN RKCS2 CANNOT BE READ BACK CORRECTLY IN RKMR2
007530  052041                          DH1
007532  054124                          DT1
007534  054726                          DF1

                                ;ERROR 2
007536  046624                          EM5                     ;DETECTED MDS
007540  052041                          DH1
007542  054124                          DT1
007544  054726                          DF1

                                ;ERROR 3
007546  046645                          EM6                     ;DETECTED UFE
007550  052041                          DH1
007552  054124                          DT1
007554  054726                          DF1

                                ;ERROR 4
007556  046666                          EM7                     ;DETECTED DRA & NED RESET (WRONG PORT SELECTED?)
007560  052041                          DH1
007562  054124                          DT1
007564  054726                          DF1
                                ;ERROR 5
007566  000000                          0
007570  000000                          0
007572  000000                          0
007574  000000                          0

                                ;ERROR 6
007576  047031                          EM9                     ;DR NOT PRESENT BUT SPECIFIED BY OPERATOR
007600  052041                          DH1
007602  054124                          DT1
007604  054726                          DF1

                                ;ERROR 7
007606  047105                          EM10                    ;ABORT TEST, COULD NOT REFERENCE CONTROLLER REGISTER
007610  052041                          DH1
007612  054124                          DT1
007614  054726                          DF1
```

```
                                ;ERROR 10
007616  047170                          EM11            ;DRA & NED BOTH SET
007620  052041                          DH1
007622  054124                          DT1
007624  054726                          DF1
                                ;ERR 11
007626  047234                          EM12            ;NO RDY
007630  053024                          DH27            ;AFTER WRITE DATA CMD
007632  054124                          DT1
007634  055056                          DF10
                                ;ERR 12
007636  047630                          EM21            ;CERR SET
007640  053024                          DH27
007642  054124                          DT1
007644  055056                          DF10
                                ;ERR 13
007646  047234                          EM12            ;NO RDY
007650  052774                          DH26            ;AFTER READ DATA CMD
007652  054124                          DT1
007654  055056                          DF10
                                ;ERR 14
007656  047630                          EM21            ;CERR SET
007660  052774                          DH26
007662  054124                          DT1
007664  055056                          DF10
                                ;ERR 15
007666  047234                          EM12            ;NO RDY
007670  053154                          DH32            ;AFTER WRITE CHECK CMD
007672  054124                          DT1
007674  055056                          DF10
                                ;ERR 16
007676  051016                          EM80            ;WRITE CHECK ERROR SET
007700  053154                          DH32            ;AFTER WRITE CHECK CMD
007702  054236                          DT6
007704  054746                          DF3
                                ;ERR 17
007706  051055                          EM81            ;WRITE CHECK CMD NOT FUNCTIONING
007710  053703                          DH52            ;WITH INTENTIONAL MISCOMPARE
007712  054124                          DT1
007714  055056                          DF10
                                ;ERR 20
007716  051121                          EM82            ;READ DATA NOT COMPARE WITH WRITE DATA
007720  052774                          DH26            ;AFTER READ DATA CMD
007722  054236                          DT6
007724  054746                          DF3
                                ;ERR 21
007726  051173                          EM83            ;DATA CHECK ERROR
007730  052774                          DH26
007732  054124                          DT1
007734  055056                          DF10
                                ;ERR 22
007736  047630                          EM21            ;CERR SET
007740  053154                          DH32            ;AFTER WRITE CHECK CMD
007742  054124                          DT1
007744  055056                          DF10
```

```
2263                                    ;ERR 23
2264  007746  047545                          EM18                    ;MSG B0 ERROR
2265  007750  053024                          DH27                    ;AFTER WRITE DATA CMD
2266  007752  054462                          DT13
2267  007754  055206                          DF21
2268                                    ;ERROR 24
2269  007756  047630                          EM21                    ;CERR SET
2270  007760  052644                          DH21                    ;AFTER SCLR
2271  007762  054124                          DT1
2272  007764  055056                          DF10
2273                                    ;ERR 25
2274  007766  047607                          EM20                    ;MSG B1 ERROR
2275  007770  053024                          DH27
2276  007772  054462                          DT13
2277  007774  055206                          DF21
2278                                    ;ERR 26
2279  007776  047545                          EM18
2280  010000  052774                          DH26                    ;AFTER READ DATA CMD
2281  010002  054462                          DT13
2282  010004  055206                          DF21
2283
2284                                    ;ERROR 27
2285  010006  050057                          EM24                    :VOL VALID NOT SET
2286  010010  052566                          DH19                    ;AFTER PACK CMD
2287  010012  054124                          DT1
2288  010014  055056                          DF10
2289                                    ;ERR 30
2290  010016  047607                          EM20                    ;MSG B1 ERROR
2291  010020  052774                          DH26                    ;AFTER READ DATA CMD.
2292  010022  054462                          DT13
2293  010024  055206                          DF21
2294                                    ;ERR 31
2295  010026  047545                          EM18                    ;MSG B0 ERROR
2296  010030  053154                          DH32                    ;AFTER WRITE CHECK CMD
2297  010032  054462                          DT13
2298  010034  055206                          DF21
2299                                    ;ERR 32
2300  010036  047607                          EM20                    ;MSG B1 ERROR
2301  010040  053154                          DH32
2302  010042  054462                          DT13
2303  010044  055206                          DF21
2304                                    ;ERR 33
2305  010046  047234                          EM12                    ;CONTR NOT READY
2306  010050  052724                          DH24                    ;AFTER OFFSET CMD
2307  010052  054124                          DT1
2308  010054  055056                          DF10
2309                                    ;ERR 34
2310  010056  047272                          EM13                    ;NO ATTN
2311  010060  052724                          DH24
2312  010062  054124                          DT1
2313  010064  055056                          DF10
2314                                    ;ERR 35
2315  010066  047524                          EM17                    ;MSG A0 ERROR
2316  010070  053737                          DH53                    ;DURING OFFSET COMMAND
2317  010072  054462                          DT13
2318  010074  055206                          DF21
```

```
 2319                                   ;ERR 36
 2320  010076  047566                          EM19                    ;MSG A1 ERROR
 2321  010100  053737                          DH53
 2322  010102  054462                          DT13
 2323  010104  055206                          DF21
 2324                                   ;ERR 37
 2325  010106  047566                          EM19                    ;MSG A1 ERROR
 2326  010110  052724                          DH24                    ;AFTER OFFSET CMD
 2327  010112  054462                          DT13
 2328  010114  055206                          DF21
 2329                                   ;ERR 40
```

```
2330  010116  047607                    EM20                    ;MSG B1 ERROR
2331  010120  052724                    DH24
2332  010122  054462                    DT13
2333  010124  055206                    DF21
2334                          ;ERR 41
2335  010126  047314                    EM14                    ;UNEXP MEM PCRITY TRAP
2336  010130  052232                    DH8                     ;TEST #, TRAP PC
2337  010132  054164                    DT3
2338  010134  054742                    DF2
2339                          ;ERR 42
2340  010136  050532                    EM41                    ;CYL ADDR IN B2 DID NOT REMAIN CLEARED
2341  010140  052724                    DH24
2342  010142  054542                    DT14
2343  010144  055242                    DF22
2344                          ;ERR 43
2345  010146  051321                    EM85
2346  010150  052672                    DH22
2347  010152  054124                    DT1
2348  010154  055056                    DF10
2349                          ;ERR 44
2350  010156  047352                    EM15                    ;WCE AT CYL 411,TRK 2, SEC 21
2351  010160  052041                    DH1
2352  010162  054124                    DT1
2353  010164  054772                    DF4
2354                          ;ERR 45
2355  010166  051321                    EM85                    ;OFFSET BIT IN RKMR2 CLEARED
2356  010170  053650                    DH51                    ;AFTER SEEK TO SELF
2357  010172  054124                    DT1
2358  010174  055056                    DF10
2359                          ;ERR 46
2360  010176  050112                    EM25                    ;DETECTED 10 BAD SECTORS
2361  010200  053024                    DH27                    ;AFTER WRITE DATA CMD.
2362  010202  054124                    DT1
2363  010204  055056                    DF10
2364                          ;ERROR 47
2365  010206  050425                    EM39                    ;CYL DIFF/OFFSET IN RKMR2 NOT CLEARED
2366  010210  052542                    DH17                    ;AFTER RECAL CMD
2367  010212  054542                    DT14
2368  010214  055242                    DF22
2369                          ;ERROR 50
2370  010216  050474                    EM40                    ;CYL ADDR IN RKMR3 NOT CLEARED
2371  010220  052542                    DH17                    ;AFTER RECAL COMD
2372  010222  054542                    DT14
2373  010224  055242                    DF22
2374                          ;ERR 51
2375  010226  051463                    EM93                    ;WRONG CYL# IN HEADER WORD (MISPOSITION)
2376  010230  052751                    DH25                    ;AFTER SEEK CMD
2377  010232  054416                    DT9
2378  010234  055162                    DF20
2379                          ;ERR 52
2380  010236  047524                    EM17                    ;MSG A0 ERROR
2381  010240  053024                    DH27                    ;AFTER WRITE DATA CMD
2382  010242  054462                    DT13
2383  010244  055206                    DF21
2384                          ;ERR 53
2385  010246  047566                    EM19                    ;MSG A1 ERROR
```

```
 2386  010250  053024                    DH27
 2387  010252  054462                    DT13
 2388  010254  055206                    DF21
 2389                          ;ERR 54
 2390  010256  047524                    EM17          ;MSG A0 ERROR
 2391  010260  052774                    DH26          ;AFTER READ DATA CMD
 2392  010262  054462                    DT13
 2393  010264  055206                    DF21
 2394                          ;ERROR 55
 2395  010266  047272                    EM13          ;NO ATTN
 2396  010270  052542                    DH17          ;AFTER RECAL CMD
 2397  010272  054124                    DT1
 2398  010274  055056                    DF10
 2399                          ;ERR 56
 2400  010276  047566                    EM19          ;MSG A1 ERROR
 2401  010300  052774                    DH26
 2402  010302  054462                    DT13
 2403  010304  055206                    DF21
 2404                          ;ERR 57
 2405  010306  047524                    EM17          ;MSG A0 ERROR
 2406  010310  053154                    DH32          ;AFTER WRITE CHECK CMD
 2407  010312  054462                    DT13
 2408  010314  055206                    DF21
 2409                          ;ERR 60
 2410  010316  047566                    EM19          ;MSG A1 ERROR
 2411  010320  053154                    DH32
 2412  010322  054462                    DT13
 2413  010324  055206                    DF21
 2414                          ;ERR 61
 2415  010326  047545                    EM18          ;MSG B0 ERROR
 2416  010330  053737                    DH53          ;DURING OFFSET CMD
 2417  010332  054462                    DT13
 2418  010334  055206                    DF21
 2419                          ;ERR 62
 2420  010336  047607                    EM20          ;MSG B1 ERROR
 2421  010340  053737                    DH53
 2422  010342  054462                    DT13
 2423  010344  055206                    DF21
 2424                          ;ERR 63
 2425  010346  050164                    EM26          ;BSE ERROR IN WRITE CMD NOT ON BSE TABLE
 2426  010350  053024                    DH27          ;AFTER WRITE DATA CMD
 2427  010352  054124                    DT1
 2428  010354  055056                    DF10
 2429                          ;ERR 64
 2430  010356  051422                    EM88          ;DID NOT FIND SECTOR 0 FROM INDEX
 2431  010360  053765                    DH54          ;AFTER FORMAT CHANGE AND READY REC'D
 2432  010362  054124                    DT1
 2433  010364  055056                    DF10
 2434                          ;ERR 65
 2435  010366  050243                    EM27          ;DETECTED BSE IN READ BUT NOT IN WRITE CMD.
 2436  010370  052041                    DH1
 2437  010372  054124                    DT1
 2438  010374  054726                    DF1
 2439                          ;ERR 66
 2440  010376  000000                    0
 2441  010400  000000                    0
```

```
 2442  010402  000000                    0
 2443  010404  000000                    0
 2444                          ;ERR 67
 2445  010406  000000                    0
 2446  010410  000000                    0
 2447  010412  000000                    0
 2448  010414  000000                    0
 2449                          ;ERROR 70
 2450  010416  000000                    0
 2451  010420  000000                    0
 2452  010422  000000                    0
 2453  010424  000000                    0
 2454                          ;ERR 71
 2455  010426  000000                    0
 2456  010430  000000                    0
 2457  010432  000000                    0
 2458  010434  000000                    0
 2459                          ;ERROR 72
 2460  010436  000000                    0
 2461  010440  000000                    0
 2462  010442  000000                    0
 2463  010444  000000                    0
 2464                          ;ERR 73
 2465  010446  000000                    0
 2466  010450  000000                    0
 2467  010452  000000                    0
 2468  010454  000000                    0
 2469                          ;ERR 74
 2470  010456  047272                    EM13        ;NO ATTN
 2471  010460  052326                    DH10        ;AT END OF HEAD LOADING
 2472  010462  054124                    DT1
 2473  010464  055056                    DF10
 2474                          ;ERR 75
 2475  010466  047652                    EM22        ;NO DRIVS IN $DEVM
 2476  010470  052041                    DH1
 2477  010472  054124                    DT1
 2478  010474  054726                    DF1
 2479                          ;ERR 76
 2480  010476  047757                    EM23        ;NO DRIVS ON BUSS
 2481  010500  052041                    DH1
 2482  010502  054124                    DT1
 2483  010504  054726                    DF1
 2484                          ;ERR 77
 2485  010506  000000                    0
 2486  010510  000000                    0
 2487  010512  000000                    0
 2488  010514  000000                    0
 2489                          ;ERR 100
 2490  010516  000000                    0
 2491  010520  000000                    0
 2492  010522  000000                    0
 2493  010524  000000                    0
 2494                          ;ERROR 101
 2495  010526  051544                    EM94        ;OFFSET NOT CLEARED
 2496  010530  053523                    DH47        ;AFTER READ HEADER WITH MOVEMENT
 2497  010532  054542                    DT14
```

```
2498  010534  055242                    DF22
2499                              ;ERROR 102
2500  010536  051600                    EM95          ;FORMAT NOT SET
2501  010540  053024                    DH27          ;AFTER WRITE DATA CMD
2502  010542  054124                    DT1
2503  010544  055056                    DF10
2504                              ;ERR 103
2505  010546  051600                    EM95
2506  010550  053154                    DH32          ;AFTER WRITE CHECK CMD
2507  010552  054124                    DT1
2508  010554  055056                    DF10
2509                              ;ERR 104
2510  010556  050425                    EM39          ;OFFSET NOT RESET
2511  010560  054063                    DH57          ;AFTER WRITE CMD WITH OFFSET
2512  010562  054542                    DT14
2513  010564  055242                    DF22
2514                              ;ERR 105
2515  010566  050474                    EM40          ;CYL ADDR NOT 0
2516  010570  054063                    DH57
2517  010572  054542                    DT14
2518  010574  055242                    DF22
2519                              ;ERR 106
2520  010576  051634                    EM96          ;CANNOT FIND SECTOR 23(8)
2521  010600  052041                    DH1
2522  010602  054124                    DT1
2523  010604  054726                    DF1
2524                              ;ERR 107
2525  010606  051665                    EM97          ;HEAD SWITCHING TOO LONG
2526  010610  053024                    DH27          ;AFTER WRITE DTA CMD
2527  010612  054124                    DT1
2528  010614  055116                    DF15
2529                              ;ERR 110
2530  010616  000000                    0
2531  010620  000000                    0
2532  010622  000000                    0
2533  010624  000000                    0
2534                              ;ERR 111
2535  010626  000000                    0
2536  010630  000000                    0
2537  010632  000000                    0
2538  010634  000000                    0
2539                              ;ERR 112
2540  010636  051752                    EM100         ;DRIVE OFF TRACK SET
2541  010640  053024                    DH27          ;AFTER WRITE DATA CMD
2542  010642  054170                    DT4
2543  010644  055032                    DF6
2544                              ;ERR 113
2545  010646  050362                    EM36          ;CYL ADDR IN RKMR3 INCORRECT
2546  010650  053024                    DH27
2547  010652  054170                    DT4
2548  010654  055032                    DF6
2549                              ;ERROR 114
2550  010656  051364                    EM86          ;OFFSET IN A2 NOT = RKASOF
2551  010660  052724                    DH24          ;AFTER OFFSET CMD
2552  010662  054542                    DT14
2553  010664  055242                    DF22
```

```
2554
2555  010666  051364                ;ERR 115
                                          EM86
2556  010670  052672                         DH22                ;AFTER DRIVE CLEAR CMD
2557  010672  054542                         DT14
2558  010674  055242                         DF22
2559                              ;ERROR 116
2560  010676  047234                         EM12                ;CONT NOT RDY
2561  010700  052566                         DH19                ;AFTER PACK CMD
2562  010702  054124                         DT1
2563  010704  055056                         DF10
2564                              ;ERROR 117
2565  010706  047234                         EM12                ;CONT NOT RDY
2566  010710  052611                         DH20                ;AFTER SEL DR CMD
2567  010712  054124                         DT1
2568  010714  055056                         DF10
2569                              ;ERROR 120
2570  010716  047234                         EM12
2571  010720  052644                         DH21                ;AFTER SUBSYS CLEAR
2572  010722  054124                         DT1
2573  010724  055056                         DF10
2574                              ;ERROR 121
2575  010726  047234                         EM12
2576  010730  052253                         DH9                 ;AFTER START SPINDLE CMD
2577  010732  054124                         DT1
2578  010734  055056                         DF10
2579                              ;ERROR 122
2580  010736  052007                         EM101               ;DID NOT GO TO CYL 10
2581  010740  053075                         DH30                ;AFTER READ HEADER CMD
2582  010742  054542                         DT14
2583  010744  055242                         DF22
2584                              ;ERROR 123
2585  010746  051364                         EM86                ;A2 OFFSET NOT = RKASOF
2586  010750  053650                         DH51                ;AFTER SEEK TO SELF
2587  010752  054542                         DT14
2588  010754  055242                         DF22
2589                              ;ERROR 124
2590  010756  047234                         EM12
2591  010760  052542                         DH17                ;AFTER RECAL CMD
2592  010762  054124                         DT1
2593  010764  055056                         DF10
2594                              ;ERR 125
2595  010766  050727                         EM73                ;CTO SET
2596  010770  051230                         EM84                ;WHILE WAITING FOR OR REC'D CONTR RDY. MSG A&B BAD
2597  010772  054124                         DT1
2598  010774  055006                         DF5
2599                              ;ERR 126
2600  010776  050775                         EM79                ;NED SET
2601  011000  051230                         EM84
2602  011002  054124                         DT1
2603  011004  055006                         DF5
2604                              ;ERR 127
2605  011006  046624                         EM5                 ;MDS SET
2606  011010  051230                         EM84
2607  011012  054124                         DT1
2608  011014  055006                         DF5
2609                              ;ERROR 130
```

```
 2610  011016  000000                    0
 2611  011020  000000                    0
 2612  011022  000000                    0
 2613  011024  000000                    0
 2614                          ;ERROR 131
 2615  011026  047234                    EM12                ;NO RDY
 2616  011030  052751                    DH25                ;AFTER SEEK CMD
 2617  011032  054124                    DT1
 2618  011034  055056                    DF10
 2619                          ;ERROR 132
 2620  011036  047272                    EM13                ;NO ATTN
 2621  011040  052751                    DH25
 2622  011042  054124                    DT1
 2623  011044  055056                    DF10
 2624                          ;ERROR 133
 2625  011046  000000                    0
 2626  011050  000000                    0
 2627  011052  000000                    0
 2628  011054  000000                    0
 2629                          ;ERROR 134
 2630  011056  000000                    0
 2631  011060  000000                    0
 2632  011062  000000                    0
 2633  011064  000000                    0
 2634                          ;ERROR 135
 2635  011066  000000                    0
 2636  011070  000000                    0
 2637  011072  000000                    0
 2638  011074  000000                    0
 2639                          ;ERROR 136
 2640  011076  000000                    0
 2641  011100  000000                    0
 2642  011102  000000                    0
 2643  011104  000000                    0
 2644                          ;ERROR 137
 2645  011106  050425                    EM39                ;CYL DIFF/OFFSET IN RKMR2 NOT CLEARED
 2646  011110  052751                    DH25
 2647  011112  054124                    DT1
 2648  011114  055056                    DF10
 2649                          ;ERR 140
 2650  011116  047524                    EM17                ;MSG A0 ERROR
 2651  011120  053650                    DH51                ;AFTER SEEK TO SELF
 2652  011122  054462                    DT13
 2653  011124  055206                    DF21
 2654                          ;ERR 141
 2655  011126  047545                    EM18
 2656  011130  053650                    DH51
 2657  011132  054462                    DT13
 2658  011134  055206                    DF21
 2659                          ;ERR 142
 2660  011136  047566                    EM19
 2661  011140  053650                    DH51
 2662  011142  054462                    DT13
 2663  011144  055206                    DF21
 2664                          ;ERR 143
 2665  011146  047607                    EM20
```

```
 2666  011150  053650                    DH51
 2667  011152  054462                    DT13
 2668  011154  055206                    DF21
 2669                          ;ERROR 144
 2670  011156  000000                    0
 2671  011160  000000                    0
 2672  011162  000000                    0
 2673  011164  000000                    0
 2674                          ;ERROR 145
 2675  011166  000000                    0
 2676  011170  000000                    0
 2677  011172  000000                    0
 2678  011174  000000                    0
 2679                          ;ERROR 146
 2680  011176  000000                    0
 2681  011200  000000                    0
 2682  011202  000000                    0
 2683  011204  000000                    0
 2684                          ;ERROR 147
 2685  011206  000000                    0
 2686  011210  000000                    0
 2687  011212  000000                    0
 2688  011214  000000                    0
 2689                          ;ERROR 150
 2690  011216  000000                    0
 2691  011220  000000                    0
 2692  011222  000000                    0
 2693  011224  000000                    0
 2694                          ;ERROR 151
 2695  011226  047234                    EM12        ;NO RDY
 2696  011230  052672                    DH22        ;AFTER CLEAR CMD
 2697  011232  054124                    DT1
 2698  011234  055056                    DF10
 2699                          ;ERROR 152
 2700  011236  000000                    0
 2701  011240  000000                    0
 2702  011242  000000                    0
 2703  011244  000000                    0
 2704                          ;ERROR 153
 2705  011246  000000                    0
 2706  011250  000000                    0
 2707  011252  000000                    0
 2708  011254  000000                    0
 2709                          ;ERROR 154
 2710  011256  050600                    EM55        ;ATTN NOT CLEARED
 2711  011260  052672                    DH22
 2712  011262  054124                    DT1
 2713  011264  055056                    DF10
 2714                          ;ERROR 155
 2715  011266  000000                    0
 2716  011270  000000                    0
 2717  011272  000000                    0
 2718  011274  000000                    0
 2719                          ;ERROR 156
 2720  011276  000000                    0
 2721  011300  000000                    0
```

```
 2722  011302  000000                        0
 2723  011304  000000                        0
 2724                              ;ERROR 157
 2725  011306  000000                        0
 2726  011310  000000                        0
 2727  011312  000000                        0
 2728  011314  000000                        0
 2729                              ;ERROR 160
 2730  011316  000000                        0
 2731  011320  000000                        0
 2732  011322  000000                        0
 2733  011324  000000                        0
 2734                              ;ERROR 161
 2735  011326  000000                        0
 2736  011330  000000                        0
 2737  011332  000000                        0
 2738  011334  000000                        0
 2739                              ;ERROR 162
 2740  011336  000000                        0
 2741  011340  000000                        0
 2742  011342  000000                        0
 2743  011344  000000                        0
 2744                              ;ERROR 163
 2745  011346  000000                        0
 2746  011350  000000                        0
 2747  011352  000000                        0
 2748
 2749
 2750
 2751
 2752
 2753
 2754
 2755
 2756
 2757
 2758
 2759
 2760
 2761
 2762
 2763
 2764
 2765
 2766
 2767
 2768
 2769  011354  000000                        0
 2770                              ;ERROR 164
 2771  011356  000000                        0
 2772  011360  000000                        0
 2773  011362  000000                        0
 2774  011364  000000                        0
 2775                              ;ERROR 165
 2776  011366  000000                        0
 2777  011370  000000                        0
```

```
 2778  011372  000000                    0
 2779  011374  000000                    0
 2780                            ;ERROR 166
 2781  011376  000000                    0
 2782  011400  000000                    0
 2783  011402  000000                    0
 2784  011404  000000                    0
 2785                            ;ERROR 167
 2786  011406  000000                    0
 2787  011410  000000                    0
 2788  011412  000000                    0
 2789  011414  000000                    0
 2790                            ;ERROR 170
 2791  011416  000000                    0
 2792  011420  000000                    0
 2793  011422  000000                    0
 2794  011424  000000                    0
 2795                            ;ERROR 171
 2796  011426  047234                    EM12            ;NO RDY
 2797  011430  053075                    DH30            ;AFTER READ HEADER CMD
 2798  011432  054124                    DT1
 2799  011434  055056                    DF10
 2800                            ;ERROR 172
 2801  011436  000000                    0
 2802  011440  000000                    0
 2803  011442  000000                    0
 2804  011444  000000                    0
 2805                            ;ERROR 173
 2806  011446  050633                    EM63            ;DLT SET
 2807  011450  053075                    DH30
 2808  011452  054124                    DT1
 2809  011454  055116                    DF15
 2810                            ;ERROR 174
 2811  011456  047630                    EM21            ;CERR SET
 2812  011460  053075                    DH30
 2813  011462  054124                    DT1
 2814  011464  055116                    DF15
 2815                            ;ERROR 175
 2816  011466  050425                    EM39            ;CYL DIFF NOT CLEARED
 2817  011470  052326                    DH10            ;AT END OF HEAD LOADING
 2818  011472  054124                    DT1
 2819  011474  055056                    DF10
 2820                            ;ERROR 176
 2821  011476  050474                    EM40            ;CYL ADDR NOT CLEARED.
 2822  011500  052326                    DH10
 2823  011502  054124                    DT1
 2824  011504  055056                    DF10
 2825                            ;ERROR 177
 2826  011506  000000                    0
 2827  011510  000000                    0
 2828  011512  000000                    0
 2829  011514  000000                    0
 2830                            ;ERROR 200
 2831  011516  047234                    EM12            ;NO RDY
 2832  011520  053206                    DH39            ;AFTER WRITE HEADER CMD
 2833  011522  054124                    DT1
```

```
 2834  011524  055116                    DF15
 2835                              ;ERROR 201
 2836  011526  047630                    EM21                ;CERR SET
 2837  011530  053206                    DH39
 2838  011532  054124                    DT1
 2839  011534  055116                    DF15
 2840                              ;ERROR 202
 2841  011536  050654                    EM65                ;READ HEADER ERROR
 2842  011540  052041                    DH1
 2843  011542  054302                    DT7
 2844  011544  055076                    DF14
 2845                              ;ERROR 203
 2846  011546  000000                    0
 2847  011550  000000                    0
 2848  011552  000000                    0
 2849  011554  000000                    0
 2850                              ;ERROR 204
 2851  011556  000000                    0
 2852  011560  000000                    0
 2853  011562  000000                    0
 2854  011564  000000                    0
 2855                              ;ERROR 205
 2856  011566  000000                    0
 2857  011570  000000                    0
 2858  011572  000000                    0
 2859  011574  000000                    0
 2860                              ;ERROR 206
 2861  011576  000000                    0
 2862  011600  000000                    0
 2863  011602  000000                    0
 2864  011604  000000                    0
 2865                              ;ERROR 207
 2866  011606  050362                    EM36                ;CYL ADDR IN RKMR3 INCORRECT
 2867  011610  052751                    DH25                ;AFTER SEEK CMD
 2868  011612  054170                    DT4
 2869  011614  055032                    DF6
 2870                              ;ERROR 210
 2871  011616  047630                    EM21                ;CERR SET
 2872  011620  052751                    DH25
 2873  011622  054124                    DT1
 2874  011624  055056                    DF10
 2875                              ;ERROR 211
 2876  011626  000000                    0
 2877  011630  000000                    0
 2878  011632  000000                    0
 2879  011634  000000                    0
 2880                              ;ERROR 212
 2881  011636  000000                    0
 2882  011640  000000                    0
 2883  011642  000000                    0
 2884  011644  000000                    0
 2885                              ;ERROR 213
 2886  011646  000000                    0
 2887  011650  000000                    0
 2888  011652  000000                    0
 2889  011654  000000                    0
```

```
2890                                    ;ERROR 214
2891  011656  000000                            0
2892  011660  000000                            0
2893  011662  000000                            0
2894  011664  000000                            0
2895                                    ;ERROR 215
2896  011666  000000                            0
2897  011670  000000                            0
2898  011672  000000                            0
2899  011674  000000                            0
2900                                    ;ERROR 216
2901  011676  000000                            0
2902  011700  000000                            0
2903  011702  000000                            0
2904  011704  000000                            0
2905                                    ;ERROR 217
2906  011706  000000                            0
2907  011710  000000                            0
2908  011712  000000                            0
2909  011714  000000                            0
2910                                    ;ERROR 220
2911  011716  000000                            0
2912  011720  000000                            0
2913  011722  000000                            0
2914  011724  000000                            0
2915                                    ;ERROR 221
2916  011726  047524                            EM17                    ;MSG A0 ERROR
2917  011730  052542                            DH17
2918  011732  054462                            DT13
2919  011734  055206                            DF21
2920                                    ;ERROR 222
2921  011736  047566                            EM19                    ;MSG A1 ERROR
2922  011740  052542                            DH17
2923  011742  054462                            DT13
2924  011744  055206                            DF21
2925                                    ;ERROR 223
2926  011746  000000                            0
2927  011750  000000                            0
2928  011752  000000                            0
2929  011754  000000                            0
2930                                    ;ERROR 224
2931  011756  000000                            0
2932  011760  000000                            0
2933  011762  000000                            0
2934  011764  000000                            0
2935                                    ;ERROR 225
2936  011766  000000                            0
2937  011770  000000                            0
2938  011772  000000                            0
2939  011774  000000                            0
2940                                    ;ERROR 226
2941  011776  047234                            EM12                    ;NO RDY
2942  012000  052774                            DH26                    ;AFTER READ DATA CMD
2943  012002  054124                            DT1
2944  012004  055056                            DF10
2945                                    ;ERROR 227
```

```
2946  012006  047630                    EM21                    ;CERR SET
2947  012010  052774                    DH26
2948  012012  054124                    DT1
2949  012014  055116                    DF15
2950                              ;ERROR 230
2951  012016  047461                    EM16                    ;CANNOT READ BSE INFO
2952  012020  052372                    DH13                    ;ON SEC 10, 12, 14, 16, 18, 20
2953  012022  054124                    DT1
2954  012024  055136                    DF17
2955                              ;ERROR 231
2956  012026  047461                    EM16
2957  012030  052456                    DH14                    ;ON SEC 11, 13, 15, 17, 19, 21
2958  012032  054124                    DT1
2959  012034  055136                    DF17
2960                              ;ERROR 232
2961  012036  000000                    0
2962  012040  000000                    0
2963  012042  000000                    0
2964  012044  000000                    0
2965                              ;ERROR 233
2966  012046  047461                    EM16                    ;CANNOT READ BSE INFO
2967  012050  053322                    DH42                    ;ON SECT 0,2,4,6,8
2968  012052  054124                    DT1
2969  012054  055136                    DF17
2970                              ;ERROR 234
2971  012056  047461                    EM16
2972  012060  053373                    DH43                    ;ON SECT 1,3,5,7,9
2973  012062  054124                    DT1
2974  012064  055136                    DF17
2975                              ;ERROR 235
2976  012066  050676                    EM69                    ;ALIGN CARTRIDGE USED
2977  012070  053444                    DH44                    ;WILL BYPASS FORMAT & ALL R/W TESTS
2978  012072  054124                    DT1
2979  012074  055056                    DF10
2980                              ;ERROR 236
2981  012076  000000                    0
2982  012100  000000                    0
2983  012102  000000                    0
2984  012104  000000                    0
2985                              ;ERROR 237
2986  012106  000000                    0
2987  012110  000000                    0
2988  012112  000000                    0
2989  012114  000000                    0
2990                              ;ERROR 240
2991  012116  000000                    0
2992  012120  000000                    0
2993  012122  000000                    0
2994  012124  000000                    0
2995                              ;ERROR 241
2996  012126  000000                    0
2997  012130  000000                    0
2998  012132  000000                    0
2999  012134  000000                    0
3000                              ;ERROR 242
3001  012136  000000                    0
```

```
3002  012140  000000                    0
3003  012142  000000                    0
3004  012144  000000                    0
3005
3006                              ;ERROR 243
3007  012146  050362                    EM36                ;CYL ADDR IN RKMR3 INCORRECT
3008  012150  052751                    DH25                ;AFTER SEEK CMD
3009  012152  054350                    DT8
3010  012154  055032                    DF6
3011                              ;ERR 244
3012  012156  050750                    EM74                ;RTZ NOT SET
3013  012160  053275                    DH41                ;DURING RECAL CMD
3014  012162  054124                    DT1
3015  012164  055056                    DF10
3016                              ;ERR 245
3017  012166  000000                    0
3018  012170  000000                    0
3019  012172  000000                    0
3020  012174  000000                    0
3021                              ;ERR 246
3022  012176  000000                    0
3023  012200  000000                    0
3024  012202  000000                    0
3025  012204  000000                    0
3026                              ;ERR 247
3027  012206  000000                    0
3028  012210  000000                    0
3029  012212  000000                    0
3030  012214  000000                    0
3031                              ;ERR 250
3032  012216  000000                    0
3033  012220  000000                    0
3034  012222  000000                    0
3035  012224  000000                    0
3036                              ;ERR 251
3037  012226  000000                    0
3038  012230  000000                    0
3039  012232  000000                    0
3040  012234  000000                    0
3041                              ;ERR 252
3042  012236  000000                    0
3043  012240  000000                    0
3044  012242  000000                    0
3045  012244  000000                    0
3046                              ;ERR 253
3047  012246  000000                    0
3048  012250  000000                    0
3049  012252  000000                    0
3050  012254  000000                    0
3051                              ;ERR 254
3052  012256  000000                    0
3053  012260  000000                    0
3054  012262  000000                    0
3055  012264  000000                    0
3056                              ;ERR 255
3057  012266  000000                    0
```

```
3058  012270  000000                    0
3059  012272  000000                    0
3060  012274  000000                    0
3061                            ;ERR 256
3062  012276  000000                    0
3063  012300  000000                    0
3064  012302  000000                    0
3065  012304  000000                    0
3066                            ;ERR 257
3067  012306  000000                    0
3068  012310  000000                    0
3069  012312  000000                    0
3070  012314  000000                    0
3071                            ;ERR 260
3072  012316  047524                    EM17                ;MSG A0 ERROR
3073  012320  052724                    DH24                ;AFTER OFFSET CMD
3074  012322  054462                    DT13
3075  012324  055206                    DF21
3076                            ;ERR 261
3077  012326  047545                    EM18                ;MSG B0 ERROR
3078  012330  052724                    DH24
3079  012332  054462                    DT13
3080  012334  055206                    DF21
3081                            ;FRR 262
3082  012336  000000                    0
3083  012340  000000                    0
3084  012342  000000                    0
3085  012344  000000                    0
3086                            ;ERR 263
3087  012346  000000                    0
3088  012350  000000                    0
3089  012352  000000                    0
3090  012354  000000                    0
3091                            ;ERR 264
3092  012356  000000                    0
3093  012360  000000                    0
3094  012362  000000                    0
3095  012364  000000                    0
3096                            ;ERR 265
3097  012366  047545                    EM18                ;MSG B0 ERROR
3098  012370  052672                    DH22                ;AFTER DRIVE CLEAR CMD
3099  012372  054462                    DT13
3100  012374  055206                    DF21
3101                            ;ERR 266
3102  012376  047607                    EM20                ;MSG B1 ERROR
3103  012400  052672                    DH22
3104  012402  054462                    DT13
3105  012404  055206                    DF21
3106                            ;ERR 267
3107  012406  047545                    EM18                ;MSG B0 ERROR
3108  012410  053206                    DH39                ;AFTER WRITE HEADER CMD
3109  012412  054462                    DT13
3110  012414  055206                    DF21
3111                            ;ERR 270
3112  012416  047607                    EM20                ;MSG B1 ERROR
3113  012420  053206                    DH39
```

```
3114  012422  054462                    DT13
3115  012424  055206                    DF21
3116                            ;ERR 271
3117  012426  047545                    EM18
3118  012430  053075                    DH30              ;AFTER RD. HDR. CMD.
3119  012432  054462                    DT13
3120  012434  055206                    DF21
3121                            ;ERR 272
3122  012436  047607                    EM20
3123  012440  053075                    DH30
3124  012442  054462                    DT13
3125  012444  055206                    DF21
3126                            ;ERR 273
3127  012446  047524                    EM17              ;MSG A0 ERROR
3128  012450  052672                    DH22              ;AFTER DRV CLR CMD
3129  012452  054462                    DT13
3130  012454  055206                    DF21
3131                            ;ERR 274
3132  012456  047566                    EM19              ;MSG A1 ERROR
3133  012460  052672                    DH22
3134  012462  054462                    DT13
3135  012464  055206                    DF21
3136                            ;ERR 275
3137  012466  047545                    EM18              ;MSG B0 ERROR
3138  012470  052542                    DH17              ;AFTER RECAL CMD
3139  012472  054462                    DT13
3140  012474  055206                    DF21
3141                            ;ERR 276
3142  012476  047607                    EM20              ;MSG B1 ERROR
3143  012500  052542                    DH17
3144  012502  054462                    DT13
3145  012504  055206                    DF21
3146                            ;ERR 277
3147  012506  047524                    EM17              ;MSG A0 ERROR
3148  012510  053206                    DH39              ;AFTER WRITE HEADER CMD
3149  012512  054462                    DT13
3150  012514  055206                    DF21
3151                            ;ERR 300
3152  012516  047566                    EM19              ;MSG A1 ERROR
3153  012520  053206                    DH39
3154  012522  054462                    DT13
3155  012524  055206                    DF21
3156                            ;ERR 301
3157  012526  047524                    EM17
3158  012530  053075                    DH30              ;AFT RD HDR. CMD
3159  012532  054462                    DT13
3160  012534  055206                    DF21
3161                            ;ERR 302
3162  012536  047566                    EM19
3163  012540  053075                    DH30
3164  012542  054462                    DT13
3165  012544  055206                    DF21
3166
```

```
3167
3168                                    .SBTTL  PROGRAM SETUP
3169
3170  012546  012737  000001  001336  PARSRT: MOV     #1,PARAM        ;SET FLAG FOR 22U START
3171  012554  000402                          BR      PRGSRT          ;START PROGRAM
3172
3173  012556  005037  001336          START:  CLR     PARAM           ;CLEAR FOR 200 START
3174  012562  000005                  PRGSRT: RESET                   ;CLEAR ALL INT ENABLE & INIT
3175  012564  012706  001100                  MOV     #STACK,SP       ;SETUP STACK POINTER
3176  012570  012746  000000                  MOV     #PR0,-(SP)      ;PSW LOADED TO BE
3177  012574  012746  012602                  MOV     #1$,-(SP)       ;LSI-11 COMPATABLE
3178  012600  000002                          RTI                     ;ENABLE ALL INTERRUPTS
3179
3180  012602  004737  041222          1$:     JSR     PC,$TKINT       ;SETUP KB VECTOR ADDR, PRIORITY 4
3181                                                                   ;& TURN ON KB INTERRUPT
3182
3183
3184                                          ;*** CPU PRIORITY LEVEL NOW AT 0 ***
3185                                          ;*** ANY DEVICE WHICH SETS ITS   ***
3186                                          ;*** INTERRUPT ENABLE BIT WILL   ***
3187                                          ;*** SERVICED.                   ***
3188
3189                                          ;CLOCK INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 6 (IN 'ST5')
3190                                          ;RK06 CONTROLLER INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 5 IN 'SETINT')
3191                                          ;KEYBOARD INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 4 (SEE ABOVE)
3192
3193                                          ;ALL 'SYSMAC' TRAPS WILL CHANGE CPU PRIORITY TO LEVEL 7 (SEE BELOW)
3194
3195                                                                  ;SYSMAC 'SETUP'
3196                                    .SBTTL  INITIALIZE THE COMMON TAGS
3197                                    ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
3198  012606  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
3199  012612  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
3200  012614  022706  001140                  CMP     #SWR,R6 ;;DONE?
3201  012620  001374                          BNE     .-6             ;;LOOP BACK IF NO
3202  012622  012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
3203                                    ;;INITIALIZE A FEW VECTORS
3204  012626  012737  037256  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
3205  012634  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
3206  012642  012737  037536  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
3207  012650  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
3208  012656  012737  043452  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
3209  012664  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
3210  012672  012737  037012  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
3211  012700  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
3212  012706  013737  031604  031576          MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
3213  012714  005037  001174                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
3214  012720  005037  001176                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
3215  012724  112737  000001  001115          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
3216  012732  012737  012732  001106          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
3217  012740  012737  012740  001110          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
3218                                    ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
3219                                    ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
3220  012746  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
3221  012752  012737  013006  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
3222  012760  012737  177570  001140          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
```

```
3223  012766  012737  177570  001142          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
3224  012774  022777  177777  166136          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
3225  013002  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
3226                                                                  ;;AND  THE HARDWARE SWR IS NOT = -1
3227  013004  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
3228  013006  012716  013014          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
3229  013012  000002                          RTI
3230  013014  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
3231  013022  012737  000174  001142          MOV     #DISPREG,DISPLAY
3232  013030  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
3233
3234  013034  005037  001216                  CLR     $PASS           ;;CLEAR PASS COUNT
3235  013040  132737  000200  001231          BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
3236  013046  001403                          BEQ     67$             ;;YES,USE NON-APT SWITCH
3237  013050  012737  001232  001140          MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
3238  013056                          67$:
3239  013056  012737  000000  000032          MOV     #PR0,EMTVEC+2   ;EMT VECTOR TO PRIORITY 0
3240  013064  012737  000000  000036          MOV     #PR0,TRAPVEC+2  ;TRAP VECTOR TO PRIORITY 0
3241  013072  012737  013136  000004  MEMPAR: MOV     #1$,ERRVEC      ;SETUP TIMEOUT VECTOR
3242  013100  012737  000340  000006          MOV     #PR7,ERRVEC+2
3243
3244  013106  012701  172100                  MOV     #MEMBAS,R1      ;ADDR OF MEM CSR
3245  013112  005011                  3$:     CLR     (R1)            ;SEE IF CAN REFERENCE
3246  013114  012711  000001                  MOV     #1,(R1)         ;SET ENABLE BIT IF YES
3247  013120  012737  036714  000114          MOV     #MEMERR,MEMVEC  ;LOAD VECTOR IF NO TIMEOUT
3248  013126  012737  000340  000116          MOV     #PR7,MEMVEC+2
3249  013134  000401                          BR      2$
3250
3251  013136  022626                  1$:     CMP     (SP)+,(SP)+     ;ADJ STACK
3252  013140  062701  000002          2$:     ADD     #2,R1           ;TRY NEXT CSR
3253  013144  020127  172140                  CMP     R1,#MEMBAS+40   ;SEE IF TRIED ALL
3254  013150  001360                          BNE     3$              ;BR IF NO
3255  013152  012737  000006  000004          MOV     #ERRVEC+2,ERRVEC        ;RESTORE TRAP CATCHER
3256  013160  005037  000006                  CLR     ERRVEC+2
3257
3258  013164  004737  031724                  JSR     PC,CLRFLG       ;CLEAR DDUMP THRU SIZFLG
3259  013170  005037  001220                  CLR     $DEVCT
3260  013174  005037  001222                  CLR     $UNIT
3261
3262                                  ;
3263                                  ;FIND OUT IF XXDP, ACT, APT; CHAIN OR DUMP MODE
3264                                  ;
3265
3266  013200  005737  000042          START1: TST     42
3267  013204  001015                          BNE     1$              ;BR IF AUTO
3268  013206  004737  031744                  JSR     PC,TITLE        ;MANUAL, TYPE PROG ID
3269  013212  123727  000041  000013          CMPB    41,#13          ;13=LOADED BY XXDP
3270  013220  001011                          BNE     2$
3271  013222  005237  007464                  INC     DDUMP           ;SET RK06 DUMP MODE FLAG
3272  013226  104401  044403                  TYPE    ,MSG2           ;REPLACE DR0 PACK W/SCRATCH & DO<CR>
3273  013232  000000                          HALT                    ;HALT
3274  013234  000137  013250                  JMP     ST2
3275  013240  000137  013314          1$:     JMP     ST3
3276  013244  005237  007472          2$:     INC     PPTP            ;SET ACT/APT/PTP DUMP MODE FLAG
3277
3278                                  ;
```

```
                                        ;CHECK IF ALL PARAMETERS DEFAULTED.  IF NOT, BEGIN INPUT DIALOGUE
                                        ;WITH OPERATOR.  THE REPLY TO 'DRIVES TO BE TESTED' SHOULD BE
                                        ;DRIVE NOS.  SEPERATED BY COMMAS & TERMINATED BY <CR>
                                        ;       EX:     DRIVES TO BE TESTED:  1,2,4<CR>
                                        ;
013250  005737  001336          ST2:    TST     PARAM
013254  001002                          BNE     1$              ;BR IF 220 START
013256  000137  013346                  JMP     ST4             ;200 START, DEFAULT & SIZE THE BUSS
013262  104401  044562          1$:     TYPE    ,MSG3           ;DRIVES TO BE TESTED
013266  004737  032024                  JSR     PC,GDRVS        ;GET DR NOS.
013272  104401  044614                  TYPE    ,MSG4           ;BUSS ADDR
013276  004737  032164                  JSR     PC,GBA          ;GET BA
013302  104401  044661                  TYPE    ,MSG5           ;CONT INT VECTOR
013306  004737  032212                  JSR     PC,GINT         ;GET INT VECTOR
013312  000427                          BR      ST5

                                        ;
                                        ;AUTO MODE
                                        ;CHECK IF LOADED BY XXDP OR OTHER.  SET FLAGS & NO INPUT DIALOGUE.
                                        ;DEFAULT ALL PARAMETERS.  TEST ONLY THOSE DRIVES THAT ARE READY
                                        ;ON THE BUSS
                                        ;
013314  123727  000041  000013  ST3:    CMPB    41,#13          ;13=LOADED BY XXDP
013322  001007                          BNE     1$
013324  005237  007466                  INC     DDPCH           ;SET RK06 CHAIN MODE FLAG
013330  004737  031744                  JSR     PC,TITLE
013334  104401  044776                  TYPE    ,MSG7           ;DR0 NOT TSTD
013340  000402                          BR      ST4
013342  005237  007470          1$:     INC     ACT11           ;SET ACT AUTO FLAG.

013346  012737  177440  001264  ST4:    MOV     #177440,$BASE   ;DEFAULT VALUE
013354  012737  000210  001314          MOV     #210,RKVEC      ;DEFAULT VALUE
013362  004737  032244                  JSR     PC,SETINT
013366  005237  007524                  INC     SIZFLG          ;DO ''SIZE THE BUSS'' TEST

013372  005037  007336          ST5:    CLR     UNLD            ;INITIALIZE FLAGS
013376  005037  007340                  CLR     BADHDR          ;USED IN 'STOP' ROUTINE
013402  005037  007342                  CLR     HPEND           ;FOR VALID PROGRAM HALTS
013406  005037  001176                  CLR     $ESCAPE
013412  005037  001170                  CLR     $TMP4           ;CLEAR RK07 FLAG
013416  012737  007476  001342          MOV     #DRIV0,DRVPTR   ;SETUP
013424  005037  001220                  CLR     $DEVCT          ;NO. OF DRVS DONE
013430  005037  001222                  CLR     $UNIT           ;CURRENT DRV UNDER TEST
013434  012737  013502  000004          MOV     #1$,ERRVEC      ;SETUP TIMEOUT ERROR VECTOR
013442  005777  165660                  TST     @LKS            ;SEE IF L-CLOCK THERE
013446  005237  007516                  INC     LCLKF           ;PRESENT, SET FLAG.
013452  013700  001330                  MOV     LCVEC,R0        ;VECTOR ADDR
013456  012737  013544  000004          MOV     #2$,ERRVEC
013464  005777  165630                  TST     @PKS            ;SEE IF P-CLOCK THERE
013470  005237  007520                  INC     PCLKF           ;PRESENT, SET FLAG
013474  013700  001332                  MOV     PCVEC,R0        ;VECTOR ADDR
013500  000412                          BR      3$

013502  022626          1$:     CMP     (SP)+,(SP)+     ;L-CLOCK NOT THERE, CLEAR STACK
```

```
 3335  013504  012737  013550  000004          MOV     #4$,ERRVEC
 3336  013512  005777  165602                  TST     @PKS            ;SEE IF P-CLOCK THERE
 3337  013516  005237  007520                  INC     PCLKF           ;PRESENT, SET FLAG
 3338  013522  013700  001332                  MOV     PCVEC,R0        ;VECTOR ADDR
 3339  013526  005237  007522          3$:     INC     DOTIM           ;INDICATES TIMING TESTS CAN BE DONE
 3340  013532  012720  036136                  MOV     #CLOCK,(R0)+    ;SERVICE ROUTINE FOR CLOCKS
 3341  013536  012710  000300                  MOV     #PR6,(R0)
 3342  013542  000407                          BR      TST1            ;;GO TO NEXT TEST
 3343
 3344  013544  022626                  2$:     CMP     (SP)+,(SP)+     ;P-CLOCK NOT THERE, CLEAR STACK
 3345  013546  000767                          BR      3$
 3346
 3347  013550  022626                  4$:     CMP     (SP)+,(SP)+     ;NEITHER CLOCK THERE, CLEAR STACK
 3348  013552  005037  007522                  CLR     DOTIM           ;TIMING TESTS CANNOT BE DONE.
 3349  013556  104401  045337                  TYPE    ,MSG13          ;HEAD SW. TEST BYPASSED
 3350
 3351
```

```
                                        .SBTTL BASIC CONTROLLER TESTS, SIZING & SETUP

                                        ;;*****************************************************************
                                        ;*TEST 1        REFERENCE ALL CONTROLLER REGISTERS
                                        ;*
                                        ;*      THIS TEST VERIFIES THAT ALL THE CONTROLLER REGISTERS
                                        ;*      CAN BE ACCESSED.  THE INABILITY  TO BE ACCESSED WILL
                                        ;*      RESULT IN A TIMEOUT TRAP WITH  AN ERROR MESSAGE.  ANY
                                        ;*      ERROR IN THIS TEST WILL RESULT IN ABORTING ALL OTHER
                                        ;*      TESTS AND JUMPING TO  'END OF PASS'
                                        ;*
                                        ;;*****************************************************************
013562  000004                  TST1:   SCOPE
013564  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
013572  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
                                
013576  012746  000000                  MOV     #PR0,-(SP)      ;RESET PSW TO PRIORITY 0
013602  012746  013610                  MOV     #5$,-(SP)       ;& MAKE IT LSI COMPATABLE
013606  000002                          RTI
013610                          5$:
```

```
  3372
```

```
  3373  013610  012737  013734  000004          MOV     #1$,ERRVEC      ;SETUP TIMOUT ERROR VECTOR
```

```
3374  013616  013705  001264                  MOV     $BASE,R5        ;SETUP INDEX REG.
3375  013622  005765  000000                  TST     RKCS1(R5)       ;REFERENCE ALL THE
3376  013626  005765  000010                  TST     RKCS2(R5)       ;CONTROLLER REGISTERS
3377  013632  005765  000002                  TST     RKWC(R5)
3378  013636  005765  000004                  TST     RKBA(R5)
3379  013642  005765  000006                  TST     RKDA(R5)
3380  013646  005765  000012                  TST     RKDS(R5)        ;TIMEOUTS IN THIS SECTION
3381  013652  005765  000014                  TST     RKER(R5)        ;INDICATE THAT THE CONTROLLER
3382  013656  005765  000016                  TST     RKASOF(R5)      ;REGISTERS CANNOT BE READ.
3383  013662  005765  000020                  TST     RKDC(R5)        ;TESTING SHOULD NOT PROCEED
3384  013666  005765  000024                  TST     RKDB(R5)        ;UNTIL THIS IS REMEDIED.
3385  013672  005765  000026                  TST     RKMR1(R5)
3386  013676  005765  000034                  TST     RKMR2(R5)
3387  013702  005765  000036                  TST     RKMR3(R5)
3388  013706  005765  000030                  TST     RKECPS(R5)
3389  013712  005765  000032                  TST     RKECPT(R5)
3390
3391  013716  012737  036626  000004          MOV     #BADTMO,ERRVEC  ;SETUP TIMEOUT HANDLER
3392  013724  012737  000340  000006          MOV     #PR7,ERRVEC+2
3393  013732  000404                          BR      TST2            ;;GO TO NEXT TEST
3394
3395  013734  022626                  1$:     CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
3396  013736  104007                          ERROR   7               ;ABORT-COULD NOT REFERENCE CONTROLLER REGISTER
3397  013740  000137  031550                  JMP     $EOP1
3398
3399
3400                                  ;;**********************************************************************
3401                                  ;*TEST 2         SIZE THE BUSS
3402                                  ;*
3403                                  ;*      THIS TEST IS ENTERED ONLY IF 'DRIVE SELECTION' IS DEFAULTED
3404                                  ;*      EITHER BY RUNNING IN THE AUTO MODE OR A 200 START IN THE
3405                                  ;*      MANUAL MODE.
3406                                  ;*      EVERY DRIVE FROM 0 THRU 7 IS ADDRESSED.
3407                                  ;*      CONTROLLER ERROR (CERR) IS EXAMINED AND IF NOT SET, THE
3408                                  ;*      DRIVE WILL BE TESTED AS AN RK06.  IF SET, THE PROGRAM WILL BYPASS
3409                                  ;*      TESTING THAT DRIVE ONLY IF THE ERROR WAS A RESULT OF
3410                                  ;*      MDS, UFE OR NED BEING SET; OR BOTH NED & DRA RESET IN-
3411                                  ;*      DICATING THE OTHER PORT IS ACCESSED.
3412                                  ;*      IF CERR DUE TO DTYE, DRIVE WILL BE TESTED AS RK07
3413                                  ;*
3414                                  ;;**********************************************************************
3415  013744  000004                  TST2:   SCOPE
3416  013746  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
3417  013754  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
3418
3419  013760  005237  001516                  INC     BYPCERR         ;DO NOT TEST CERR IN 'FRDY'
3420
3421
3422  013764  132737  000200  001231          BITB    #BIT7,$ENVM     ;SEE IF USE APT SELECTED DRIVES
3423  013772  001002                          BNE     14$             ;BR IF YES
3424  013774  000137  014114                  JMP     12$             ;ELSE DO NORM SIZING OR VERIFY
3425
3426  014000  104401  045105          14$:    TYPE    ,MSG10          ;WILL TEST DRIVES
3427  014004  005037  007474                  CLR     DRIVS           ;# OF DRIVES PRESENT
3428  014010  005000                          CLR     R0              ;DRV ADDR
3429  014012  012701  007476                  MOV     #DRIV0,R1       ;DRV FLAG
```

```
3430  014016  013702  001266                  MOV     $DEVM,R2        ;APT DEVICE MAP
3431
3432  014022  032702  000001          15$:    BIT     #BIT0,R2        ;SEE IF DRV IN DEVICE MAP
3433  014026  001410                          BEQ     16$             ;BR IF NO
3434  014030  005237  007474                  INC     DRIVS           ;ELSE INCR DRIVE COUNT
3435  014034  005211                          INC     (R1)            ;& SET DRIVE PRESENT FLAG
3436  014036  104401  001205                  TYPE    ,$CRLF
3437  014042  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
3438                                                                  ;;TYPE DRIVE #
3439  014044  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
3440  014046     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
3441  014047     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
3442
3443  014050  005721          16$:    TST     (R1)+           ;ADV POINTER TO NEXT FLAG
3444  014052  005200                  INC     R0              ;INC DRIVE #
3445  014054  022700  000010          CMP     #8.,R0          ;ALL 8 TESTED?
3446  014060  001402                  BEQ     17$             ;BR IF YES
3447
3448  014062  006002                  ROR     R2              ;ELSE GET NEXT BIT OFF DEVICE MAP
3449  014064  000756                  BR      15$             ;& TRY AGAIN
3450
3451  014066  005737  007474          17$:    TST     DRIVS           ;SEE IF MORE DRIVES PRESENT
3452  014072  001402                  BEQ     18$             ;BR IF NO
3453  014074  000137  014600          JMP     VERIFY          ;ELSE EXIT TEST & SETUP FOR RK07'S
3454
3455  014100  104075          18$:    ERROR   75              ;NO DRIVES FOUND IN $DEVM
3456  014102  000000                  HALT                    ;SETUP CORRECTLY & PRESS 'CONTINUE'
3457  014104  000137  013372          JMP     ST5             ;TO TRY AGAIN
3458  014110  000137  014600          20$:    JMP     VERIFY          ;DO NOT SIZE, GO TO NEXT TEST
3459
3460  014114  012765  000040  000010  12$:    MOV     #SCLR,RKCS2(R5) ;SUBSYSTEM CLEAR
3461  014122  013737  001406  007412          MOV     T10,TEMP1       ;SET TIMEOUT
3462  014130  004737  032356                  JSR     PC,FRDY         ;FIND RDY
3463  014134  104120                          ERROR   120             ;RDY NOT SET BY END OF SCLR
3464  014136  005737  007524                  TST     SIZFLG          ;SIZE BUS?
3465  014142  001762                          BEQ     20$             ;BR IF NO
3466  014144  104401  045105                  TYPE    ,MSG10          ;WILL TEST DRIVES
3467  014150  005037  007474                  CLR     DRIVS           ;# OF DRIVES PRESENT
3468  014154  005000                          CLR     R0              ;DRV ADDR
3469  014156  012701  007476                  MOV     #DRIV0,R1       ;DRV FLAG
3470  014162                          1$:
3471  014162  104415                          SCOP1
3472  014164  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
3473
3474  014170  012765  000040  000010          MOV     #SCLR,RKCS2(R5) ;SUBSYSTEM CLEAR
3475  014176  013737  001406  007412          MOV     T10,TEMP1       ;SET TIMEOUT
3476  014204  004737  032356                  JSR     PC,FRDY         ;FIND RDY
3477  014210  104120                          ERROR   120             ;RDY NOT SET BY END OF SCLR
3478  014212  010065  000010                  MOV     R0,RKCS2(R5)    ;SELECT THE DRIVE ADDR
3479  014216  012737  000001  007354          MOV     #SELDRV,HCS1
3480  014224  053737  001170  007354          BIS     $TMP4,HCS1      ;ADD CDT IF RK07
3481  014232  013765  007354  000000          MOV     HCS1,RKCS1(R5)  ;GET STATUS
3482  014240  013737  001420  007412          MOV     T50000,TEMP1
3483  014246  004737  033046                  JSR     PC,DLY          ;DO DELAY TO CATCH MDS
3484  014252  013737  001406  007412          MOV     T10,TEMP1
3485  014260  004737  032356                  JSR     PC,FRDY         ;FIND RDY
```

```
3486  014264  104117                        ERROR   117             ;NO RDY AFTER SELECT DRIVE CMD.
3487  014266  032737  100000  007354        BIT     #CERR,HCS1
3488  014274  001056                        BNE     2$
3489  014276  013737  007402  007412        MOV     HMR2,TEMP1
3490  014304  042737  177770  007412        BIC     #^C<DRVMSK>,TEMP1
3491  014312  020037  007412                CMP     R0,TEMP1        ;S/B SAME
3492  014316  001024                        BNE     3$
3493  014320  005700                        TST     R0
3494  014322  001007                        BNE     4$
3495  014324  005737  007466                TST     DDPCH           ;SEE IF XXDP CHAIN MODE
3496  014330  001022                        BNE     5$
3497  014332  123727  000041  000013        CMPB    41,#13          ;IS DRIVE 0 TO BE TESTED
3498  014340  001416                        BEQ     5$              ;BRANCH IF NOT
3499  014342  005237  007474        4$:     INC     DRIVS           ;INC DRIVE COUNT.
3500  014346  005211                        INC     (R1)            ;SET DRIVE PRESENT FLAG
3501  014350  053711  001170                BIS     $TMP4,(R1)      ;ADD CDT IF RK07
3502  014354  104401  001205                TYPE    ,$CRLF
3503  014360  010046                        MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
3504                                                                ;;TYPE DR #
3505  014362  104403                        TYPOS                   ;;GO TYPE--OCTAL ASCII
3506  014364     001                        .BYTE   1               ;;TYPE 1 DIGIT(S)
3507  014365     000                        .BYTE   0               ;;SUPPRESS LEADING ZEROS
3508  014366  000403                        BR      5$
3509
3510  014370  004737  033064        3$:     JSR     PC,BYP          ;TYPE BYPASS DR #
3511  014374  104001                        ERROR   1               ;WRITTEN DR # DOES NOT MATCH RKMR2 DR #
3512
3513  014376  005721                5$:     TST     (R1)+           ;SHIFT PTR TO NEXT DR. FLAG
3514  014400  005200                        INC     R0              ;INC DR #
3515  014402  005037  001170                CLR     $TMP4           ;CLEAR RK07 FLAG
3516  014406  022700  000010                CMP     #8.,R0
3517  014412  001263                        BNE     1$              ;MORE LEFT.
3518  014414  005737  007474                TST     DRIVS
3519  014420  001065                        BNE     10$
3520  014422  104076                        ERROR   76              ;NO DRIVES FOUND ON BUSS
3521  014424  000000                        HALT                    ;SETUP CORRECTLY
3522  014426  000137  013372                JMP     ST5             ;AND PRESS 'CONTINUE'
3523  014432  032737  000040  007370  2$:   BIT     #DTYE,HER
3524  014440  001405                        BEQ     13$
3525  014442  012737  002000  001170        MOV     #CDT,$TMP4      ;ADD CDT
3526  014450  000137  014162                JMP     1$              ;TRY AGAIN
3527
3528  014454  032737  001000  007356  13$:   BIT     #MDS,HCS2
3529  014462  001015                        BNE     6$
3530  014464  032737  000400  007356        BIT     #UFE,HCS2
3531  014472  001015                        BNE     7$
3532  014474  032737  000001  007366        BIT     #DRA,HDS
3533  014502  001015                        BNE     8$
3534  014504  032737  010000  007356        BIT     #NED,HCS2
3535  014512  001424                        BEQ     9$
3536  014514  000730                        BR      5$
3537
3538  014516  004737  033064        6$:     JSR     PC,BYP          ;TYPE BYP DR #
3539  014522  104002                        ERROR   2               ;MDS DETECTED
3540  014524  000724                        BR      5$
3541
```

```
3542  014526  004737  033064          7$:     JSR     PC,BYP
3543  014532  104003                          ERROR   3               ;UFE DETECTED
3544  014534  000720                          BR      5$
3545
3546  014536  032737  010000  007356  8$:     BIT     #NED,HCS2
3547  014544  001676                          BEQ     4$
3548  014546  104401  045460                  TYPE    ,MSG15          ;DRV#
3549  014552  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
3550                                                                  ;;TYPE DR#
3551  014554  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
3552  014556     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
3553  014557     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
3554  014560  104010                          ERROR   10              ;DRA & NED BOTH SET
3555  014562  000705                          BR      5$
3556
3557  014564  004737  033064          9$:     JSR     PC,BYP
3558  014570  104004                          ERROR   4               ;NO DRA & NO NED = OTHER PORT SELECTED
3559  014572  000701                          BR      5$
3560  014574  000137  015172          10$:    JMP     NUDRV
3561
3562  014600                          VERIFY:
3563                                  ;;*********************************************************************
3564                                  ;*TEST 3        VERIFY OPERATOR DRIVE SELECTIONS
3565                                  ;*
3566                                  ;*      THIS TEST IS ENTERED ONLY IF DRIVE SELECTION IS NOT
3567                                  ;*      DEFAULTED.  EVERY DRIVE FROM 0 TO 7 IS ADDRESSED &
3568                                  ;*      CONTROLLER ERROR (CERR) IS EXAMINED. IF NOT SET, THE
3569                                  ;*      PROGRAM WILL ASSUME THE DRIVE IS PRESENT AS AN RK06.
3570                                  ;*      IF CERR WAS SET, THAT DRIVE WILL BE BYPASSED
3571                                  ;*      ONLY IF THE ERROR WAS A RESULT OF MDS OR UFE SET OR BOTH
3572                                  ;*      NED & DRA RESET (WRONG PORT).  IF CERR IS  A RESULT OF
3573                                  ;*      NED ONLY, IT IS CHECKED AGAINT THE INPUTTED INFOR TO
3574                                  ;*      VERIFY IT WAS NOT SPECIFIED.
3575                                  ;*      IF CERR DUE TO DTYE, THE DRIVE WILL BE TESTED AS AN RK07.
3576                                  ;*
3577                                  ;;*********************************************************************
3578  014600  000004                  TST3:   SCOPE
3579  014602  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
3580  014610  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
3581  014614  005000                          CLR     R0              ;DRIVE ADDR
3582  014616  012701  007476                  MOV     #DRIV0,R1       ;DRIVE FLAG
3583  014622                          1$:
3584  014622  104415                          SCOP1
3585  014624  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
3586
3587  014630  012765  000040  000010          MOV     #SCLR,RKCS2(R5) ;SUBSYSTEM CLEAR
3588  014636  013737  001406  007412          MOV     T10,TEMP1       ;SET TIME OUT
3589  014644  004737  032356                  JSR     PC,FRDY         ;FIND RDY
3590  014650  104120                          ERROR   120             ;NO RDY AFTER SCLR
3591  014652  010065  000010                  MOV     R0,RKCS2(R5)    ;DRV ADDR
3592  014656  012737  000001  007354          MOV     #SELDRV,HCS1
3593  014664  053737  001170  007354          BIS     $TMP4,HCS1      ;ADD CDT IF RK07
3594  014672  013765  007354  000000          MOV     HCS1,RKCS1(R5)  ;GET STATUS
3595  014700  013737  001420  007412          MOV     T50000,TEMP1
3596  014706  004737  033046                  JSR     PC,DLY          ;DO DELAY TO CATCH MDS
3597  014712  013737  001406  007412          MOV     T10,TEMP1
```

```
3598  014720  004737  032356                    JSR     PC,FRDY         ;FIND RDY
3599  014724  104117                            ERROR   117             ;NO RDY AFTER SELECT DRIVE CMD.
3600  014726  032737  100000  007354            BIT     #CERR,HCS1
3601  014734  001036                            BNE     2$
3602  014736  013737  007402  007412            MOV     HMR2,TEMP1
3603  014744  042737  177770  007412            BIC     #^C<DRVMSK>,TEMP1
3604  014752  020037  007412                    CMP     R0,TEMP1        ;S/B SAME
3605  014756  001014                            BNE     3$
3606  014760  005711                    11$:    TST     (R1)
3607  014762  001402                            BEQ     4$
3608  014764  053711  001170                    BIS     $TMP4,(R1)      ;SET CDT IF RK07.
3609  014770  005721                    4$:     TST     (R1)+           ;SHIFT PTR TO NEXT DR FLAG
3610  014772  005200                            INC     R0              ;INC DR#
3611  014774  005037  001170                    CLR     $TMP4           ;CLEAR CDT FOR NEXT DRIVE
3612  015000  022700  000010                    CMP     #8.,R0
3613  015004  001306                            BNE     1$              ;MORE LEFT
3614  015006  000475                            BR      TST4            ;;GO TO NEXT TEST
3615
3616  015010  004737  033064            3$:     JSR     PC,BYP          ;TRY BYPASS DRIVE#
3617  015014  104001                            ERROR   1               ;WRITTEN DR# DOES NOT MATCH RKMR2 DR#
3618  015016  005711                            TST     (R1)
3619  015020  001763                            BEQ     4$              ;BRANCH IF NOT SPEC BY INPUT
3620  015022  005337  007474            12$:    DEC     DRIVS           ;DECREMENT TOTAL DRIVS
3621  015026  005011                            CLR     (R1)            ;CLEAR DRIVE FLAG
3622  015030  000757                            BR      4$
3623
3624  015032  032737  000040  007370    2$:     BIT     #DTYE,HER
3625  015040  001405                            BEQ     13$
3626  015042  012737  002000  001170            MOV     #CDT,$TMP4      ;ADD CDT
3627  015050  000137  014622                    JMP     1$              ;TRY AGAIN
3628
3629  015054  032737  001000  007356    13$:    BIT     #MDS,HCS2
3630  015062  001027                            BNE     6$
3631  015064  032737  000400  007356            BIT     #UFE,HCS2
3632  015072  001027                            BNE     7$
3633  015074  032737  000001  007366            BIT     #DRA,HDS
3634  015102  001005                            BNE     8$
3635  015104  032737  010000  007356            BIT     #NED,HCS2
3636  015112  001423                            BEQ     9$
3637  015114  000404                            BR      10$
3638  015116  032737  010000  007356    8$:     BIT     #NED,HCS2
3639  015124  001715                            BEQ     11$
3640  015126  005711                    10$:    TST     (R1)
3641  015130  001717                            BEQ     4$
3642
3643  015132  004737  033064                    JSR     PC,BYP          ;TYPE BYPASS DRIVE#
3644  015136  104006                            ERROR   6
3645  015140  000730                            BR      12$
3646
3647  015142  004737  033064            6$:     JSR     PC,BYP          ;TYPE BYPASS DRIVE#
3648  015146  104002                            ERROR   2               ;MDS DETECTED
3649  015150  000724                            BR      12$
3650
3651  015152  004737  033064            7$:     JSR     PC,BYP
3652  015156  104003                            ERROR   3               ;UFE DETECTED
3653  015160  000720                            BR      12$
```

```
3654
3655  015162  004737  033064          9$:     JSR     PC,BYP
3656  015166  104004                          ERROR   4               ;DRA & NED RESET - OTHER PORT SELECTED
3657  015170  000714                          BR      12$
3658
3659
3660
3661                                          ;
3662                                          ;THIS PART OF THE PROGRAM WILL BE REPEATED FOR EACH
3663                                          ;DRIVE PRESENT
3664                                          ;
3665                                          ;'$UNIT' CONTAINS THE ADDRESS OF THE DRIVE CURRENTLY
3666                                          ;UNDER TEST
3667
3668  015172  005037  001516          NUDRV:  CLR     BYPCERR         ;ENTER HERE FROM LAST TEST
3669                                                                  ;ALLOW CHECKING CERR IN 'FRDY'
3670  015176  005037  001170                  CLR     $TMP4           ;CLEAR RK07 FLAG
3671
3672                                          ;;**************************************************************
3673                                          ;*TEST 4        FIND NEXT DRIVE TO BE TESTED
3674                                          ;*
3675                                          ;*      THIS TEST FINDS THE NEXT DRIVE PRESENT & PUTS THAT
3676                                          ;*      ADDRESS IN '$UNIT' & $TMP4 IS SET TO CDT IF DRIVE IS RK07.
3677                                          ;*      THROUGHOUT THE FOLLOWING TESTS, THE DRIVE TESTED IS
3678                                          ;*      THE DRIVE WHOSE ADDRESS IS IN '$UNIT'.
3679                                          ;*
3680                                          ;;**************************************************************
3681  015202  000004                  TST4:   SCOPE
3682  015204  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
3683  015212  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
3684  015216  012737  000004  001214          MOV     #$TN-1,$TESTN
3685  015224  012737  000004  001102          MOV     #$TN-1,$TSTNM
3686
3687  015232  005737  007474                  TST     DRIVS           ;ANY DRIVES PRESENT?
3688  015236  001004                          BNE     4$              ;YES BRANCH
3689  015240  104401  045624                  TYPE    ,MSG27          ;ALL DRIVES TESTED
3690  015244  000137  031550                  JMP     $EOP1           ;NO, GO TO END
3691
3692  015250  013701  001342          4$:     MOV     DRVPTR,R1       ;ADDR OF NEXT DRIVE FLAG
3693  015254  005737  001220                  TST     $DEVCT          ;IS FIRST DRIVE BEING CHECKED
3694  015260  001402                          BEQ     2$              ;YES, BRANCH
3695  015262  005237  001222          1$:     INC     $UNIT           ;INCR DRIVE ADDR TO NEXT DRIVE
3696  015266  005711                  2$:     TST     (R1)            ;IS DRIVE PRESENT?
3697  015270  001002                          BNE     5$              ;BR IF YES
3698  015272  005721                          TST     (R1)+           ;ELSE FIND NEXT DRIVE
3699  015274  000772                          BR      1$
3700  015276  005737  007466          5$:     TST     DDPCH           ;DDP CHAIN MODE?
3701  015302  001405                          BEQ     3$              ;BR IF NO
3702  015304  005737  001222                  TST     $UNIT           ;ELSE SEE IF DRV 0
3703  015310  001002                          BNE     3$              ;BR IF NO
3704  015312  005721                          TST(R1)+                ;ELSE FIND NEXT DRIVE PRESENT
3705  015314  000762                          BR      1$
3706
3707  015316  032721  002000          3$:     BIT     #CDT,(R1)+      ;SEE IF DRIVE UNDER TEST IS RK07
3708  015322  001403                          BEQ     6$              ;BR IF NO
3709  015324  012737  002000  001170          MOV     #CDT,$TMP4      ;ELSE SET RK07 FLAG
```

```
3710  015332  010137  001342          6$:     MOV     R1,DRVPTR       ;STORE POINTER TO NEXT DR FLAG
3711  015336  104401  045460                  TYPE    ,MSG15          ;'DRIVE'
3712  015342  013700  001222                  MOV     $UNIT,R0
3713  015346  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
3714                                                                  ;;DRIVE #
3715  015350  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
3716  015352     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
3717  015353     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
3718
3719                                  ;       TYPE    ,$CRLF
3720
3721  015354  005737  001170                  TST     $TMP4           ;SEE IF RK07 UNDER TEST
3722  015360  001017                          BNE     7$              ;BR IF YES
3723  015362  012737  000632  015460          MOV     #632,LC         ;ELSE LOAD RK06 PARAMETERS
3724  015370  005037  015470                  CLR     E.DDT
3725  015374  012737  001000  015462          MOV     #1000,MC1
3726  015402  012737  000777  015464          MOV     #777,MASK
3727  015410  012737  160017  015466          MOV     #160017,MASK1
3728  015416  000425                          BR      TST5            ;;GOTO NEXT TEST
3729
3730  015420  012737  001456  015460  7$:     MOV     #1456,LC        ;LOAD RK07 PARAMETERS
3731  015426  012737  000400  015470          MOV     #D.DDT,E.DDT
3732  015434  012737  002000  015462          MOV     #2000,MC1
3733  015442  012737  001777  015464          MOV     #1777,MASK
3734  015450  012737  140017  015466          MOV     #140017,MASK1
3735  015456  000405                          BR      TST5            ;;GOTO NEXT TEST
3736
3737  015460  000000                  LC:     0                       ;LAST CYL
3738  015462  000000                  MC1:    0                       ;MAJ CYL + 1 SHIFT
3739  015464  000000                  MASK:   0
3740  015466  000000                  MASK1:  0
3741  015470  000000                  E.DDT:  0                       ;EXPECTED DRIVE TYPE TO E.A0
3742
3743  015472                          PFSRT:                          ;ENTER HERE FOR POWER FAIL RESTART
3744                                  ;;**************************************************************
3745                                  ;*TEST 5        PRINT DRIVE SERIAL NUMBER
3746                                  ;*
3747                                  ;*      THIS TEST READS & PRINTS THE DRIVE SERIAL # FROM MSG A, WORD 11
3748                                  ;*      IN BCD & IS PERFORMED ON THE 1ST PASS ONLY
3749                                  ;*
3750                                  ;;**************************************************************
3751  015472  000004                  TST5:   SCOPE
3752  015474  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
3753  015502  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
3754
3755  015506  005737  001216                  TST     $PASS
3756  015512  001042                          BNE     TST6            ;;GO TO NEXT IF NOT FIRST PASS
3757  015514  004737  034252                  JSR     PC,SUBCLR       ;DO SUBSYS CLEAR
3758  015520  104024                          ERROR   24              ;CERR AFTER SCLR
3759
3760  015522  104401  045472                  TYPE    ,MSG16          ;DRIVE SERIAL NO.
3761  015526  012765  000003  000026          MOV     #3,RKMR1(R5)    ;SELECT BYTE 3
3762  015534  004737  033722                  JSR     PC,GSTAT        ;GET STATUS
3763  015540  013701  007402                  MOV     HMR2,R1         ;GET SERIAL #
3764  015544  012704  042736                  MOV     #$OCTVL,R4      ;GET ADDR CHAR BUFF
3765  015550  010446                          MOV     R4,-(SP)        ;STORE ON STACK FOR $SUPRS
```

```
3766  015552  012703  000003            MOV     #3,R3           ;SETUP CHAR COOUNT
3767  015556  006101                    ROL     R1              ;INITIALIZE BIT POSITIONS
3768  015560  006101                    ROL     R1
3769  015562  006101            1$:     ROL     R1              ;GET NEXT 4 BITS
3770  015564  006101                    ROL     R1
3771  015566  006101                    ROL     R1
3772  015570  006101                    ROL     R1
3773  015572  010100                    MOV     R1,R0           ;GET WORKING COPY
3774  015574  042700  177760            BIC     #177760,R0      ;CLEAR ALL BUT LOW 4 BITS
3775  015600  052700  000060            BIS     #60,R0          ;CONVERT TO ASCII DIGIT
3776  015604  110024                    MOVB    R0,(R4)+        ;PUT ASCII DIGIT INTO CHAR BUFF
3777  015606  005303                    DEC     R3
3778  015610  001364                    BNE     1$              ;BR IF ALL 3 CHARS NOT DONE
3779  015612  105014                    CLRB    (R4)            ;ELSE INSERT NULL TERMINATOR
3780
3781  015614  004737  043204            JSR     PC,$SUPRS       ;TYPE
3782                            ;       TYPE    ,$CRLF
3783                            ;       TYPE    ,$CRLF
3784
3785                            ;;**************************************************************
3786                            ;*TEST 6        SET VV WITH PACK COMMAND
3787                            ;*
3788                            ;*      IF VV IS RESET, THE PACK COMMAND IS USED TO SET IT.
3789                            ;*
3790                            ;;**************************************************************
3791  015620  000004            TST6:   SCOPE
3792  015622  012737  000001  001174    MOV     #1,$TIMES       ;;DO 1 ITERATION
3793  015630  012706  001100            MOV     #STACK,SP       ;RESTORE STK PTR
3794
3795  015634  004737  034252            JSR     PC,SUBCLR
3796  015640  104024                    ERROR   24              ;CERR AFTER SCLR
3797
3798  015642  032737  000100  007402    BIT     #D.VV,HMR2
3799  015650  001021                    BNE     TST7            ;;GO TO NEXT TEST IF VV SET
3800
3801  015652  104415                    SCOP1
3802  015654  012706  001100            MOV     #STACK,SP       ;RESTORE STK PTR
3803
3804  015660  004737  034252            JSR     PC,SUBCLR
3805  015664  104024                    ERROR   24              ;CERR AFTER SCLR
3806
3807  015666  012737  000003  007354    MOV     #PACK,HCS1
3808  015674  004737  032262            JSR     PC,DOCMD        ;DO PACK CMD & GET CONTR RDY
3809  015700  104116                    ERROR   116             ;RDY NOT SET AFTER PACK CMD
3810
3811  015702  032737  000100  007402    BIT     #D.VV,HMR2
3812  015710  001001                    BNE     TST7            ;;GO TO NEXT TEST IF VV NOW SET
3813  015712  104027                    ERROR   27              ;PACK DID NOT SET V.V.
3814
3815
3816
3817                            ;;**************************************************************
3818                            ;*TEST 7        READ & SAVE BAD SECTOR INFO & TYPE PACK SERIAL #
3819                            ;*
3820                            ;*      THIS TEST VERIFIES THAT CYL 632 (1456 FOR RK07), TRACK 2 CAN BE READ.
3821                            ;*      THIS AREA CONTAINS BAD SECTOR INFO WHICH IS WRITTEN BY THE
```

```
                                        ;*      FACTORY DURING MANF.  ALL BAD SECTOR INFO (BSE) WILL BE STORED
                                        ;*      AT THIS TIME TO MASK FUTURE READ HEADER OR DATA ERROR PRINTOUTS.
                                        ;*
                                        ;*      SECTORS 0,2,4,6,8 CONTAIN IDENTICAL INFO FOR 22 SECTOR HARDWARE DETECTED BAD SEC
                                        ;*      SECTORS 10,12,14,16,18,20 CONTAIN IDENTICAL INFO FOR 22 SECTOR SOFTWARE DETECTED
                                        ;*
                                        ;*      SECTORS 1,3,5,7,9 CONTAIN IDENTICAL INFO FOR 20 SECTOR HARDWARE DETECTED BAD SEC
                                        ;*      SECTORS 11,13,15,17,19,21 CONTAIN IDENTICAL INFO FOR 20 SECTOR SOFTWARE DETECTED
                                        ;*
                                        ;*      IF BSE INFO CANNOT BE READ, OR IF AFTER READING THE BSE INFO
                                        ;*      IT IS DETERMINED THAT AN ALIGNMENT CARTRIDGE IS USED,
                                        ;*      A MESSAGE WILL BE TYPED INDICATING THAT ALL
                                        ;*      FUTURE FORMAT AND READ-WRITE TESTS WILL BE BYPASSED.
                                        ;*      THIS IS DONE SO AS NOT TO DESTROY BSE INFO OR AN ALIGNMENT PACK BY WRITING
                                        ;*
                                        ;*      THE PACK SERIAL # IS TYPED IN OCTAL & FOR THE FIRST PASS ONLY.
                                        ;*
                                        ;;*****************************************************************
015714  000004                TST7:   SCOPE
015716  012737  000001  001174        MOV     #1,$TIMES       ;;DO 1 ITERATION
015724  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR

015730  004737  034252                JSR     PC,SUBCLR
015734  104024                        ERROR   24              ;CERR AFTER SCLR


015736  012765  100000  000000        MOV     #CCLR,RKCS1(R5)
015744  013765  001222  000010        MOV     $UNIT,RKCS2(R5)
015752  012737  000013  007354        MOV     #RECAL,HCS1
015760  004737  032262                JSR     PC,DOCMD        ;DO RECAL CMD & GET CONTR RDY
015764  104124                        ERROR   124             ;RDY NOT SET AFTER RECAL CMD

015766  012765  000001  000026        MOV     #1,RKMR1(R5)    ;SELECT WORD 1
015774  004737  033722                JSR     PC,GSTAT
016000  032737  020000  007402        BIT     #D.RTZ,HMR2
016006  001001                        BNE     64$
016010  104244                        ERROR   244             ;RTZ NOT SET DURING RECAL CMD
016012  013737  001406  007414  64$:  MOV     T10,TEMP2       ;SETUP TIMEOUT
016020  004737  032672                JSR     PC,FATT1        ;FIND ATTN
016024  104055                        ERROR   55              ;NO ATTN AFTER RECAL CMD

016026  012765  100000  000000        MOV     #CCLR,RKCS1(R5)
016034  013765  001222  000010        MOV     $UNIT,RKCS2(R5) ;DRIVE#
016042  012737  000005  007354        MOV     #CLEAR,HCS1
016050  004737  032262                JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
016054  104151                        ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
016056  004737  032640                JSR     PC,TSTATN       ;TEST FOR ATTN
016062  000401                        BR      65$
016064  104154                        ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
016066                          65$:


016066  004737  034252                JSR     PC,SUBCLR
016072  104024                        ERROR   24              ;CERR AFTER SCLR
```

```
3878  016074  005037  007414                  CLR     TEMP2           ;SECTOR CTR
3879  016100  005037  007416                  CLR     TEMP3           ;0=22 SECTOR HARDWARE DETECTED TABLE
3880                                                                  ;1=22 SECTOR SOFTWARE DETECTED TABLE
3881                                                                  ;2=20 SECTOR HARDWARE DETECTED TABLE
3882                                                                  ;3=20 SECTOR SOFTWARE DETECTED TABLE
3883  016104  012737  003336  007420            MOV     #BSE22H,TEMP4   ;STORE 22 SECTOR HARDWARE BSE ADDR.
3884  016112  013765  007420  000004            MOV     TEMP4,RKBA(R5)
3885  016120  012737  001000  007422            MOV     #1000,TEMP5     ;TRACK 2, SECTOR 0
3886  016126  013765  007422  000006            MOV     TEMP5,RKDA(R5)
3887
3888  016134  013765  015460  000020  1$:       MOV     LC,RKDC(R5)     ;LAST CYL
3889  016142  012765  177400  000002            MOV     #-256.,RKWC(R5) ;LOAD WORD CT
3890  016150  012737  000021  007354            MOV     #RDDATA,HCS1
3891  016156  004737  032320                    JSR     PC,DATCMD       ;DO READ DATA CMD & GET CONTR RDY
3892  016162  104226                            ERROR   226             ;NO RDY AFTER READ DATA CMD
3893  016164  004737  033722                    JSR     PC,GSTAT        ;GET FRESH DATA
3894  016170  032737  100000  007354            BIT     #CERR,HCS1
3895  016176  001504                            BEQ     8$
3896  016200  104227                            ERROR   227             ;CERR AFTER READ DATA CMD
3897
3898  016202  012737  010340  007444            MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
3899  016210  005037  007446                    CLR     E.B0            ;EXPECTED MSG B0
3900  016214  012737  001720  007450            MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
3901  016222  012737  000001  007452            MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
3902  016230  005037  007454                    CLR     E.A2            ;EXPECTED MSG A2
3903  016234  012737  000002  007456            MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
3904  016242  012737  000003  007462            MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
3905
3906  016250  004737  033100                    JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
3907  016254  000000                            .WORD   0!0!0           ;& MSGS SPECIFIED HERE
3908  016256  104054                            ERROR   54              ;MSG A0 ERROR AFTER READ DATA CMD
3909  016260  104026                            ERROR   26              ;MSH B0 ERROR
3910  016262  104056                            ERROR   56              ;MSG A1 ERROR
3911  016264  104030                            ERROR   30              ;MSG B1 ERROR
3912
3913  016266  004737  034252                    JSR     PC,SUBCLR
3914  016272  104024                            ERROR   24              ;CERR AFTER SUBCLR
3915
3916  016274  005237  007414                    INC     TEMP2
3917  016300  023727  007414  000005            CMP     TEMP2,#5        ;READ ALL 5 SECTORS?
3918  016306  001023                            BNE     5$
3919  016310  005737  007416                    TST     TEMP3
3920  016314  001002                            BNE     2$
3921  016316  104233                            ERROR   233             ;CANT READ SECTORS 0,2,4,6,8
3922  016320  000430                            BR      3$
3923
3924  016322  023727  007416  000001  2$:       CMP     TEMP3,#1
3925  016330  001002                            BNE     4$
3926  016332  104230                            ERROR   230             ;CANT READ SECTORS 10,12...
3927  016334  000422                            BR      3$
3928
3929  016336  023727  007416  000002  4$:       CMP     TEMP3,#2
3930  016344  001002                            BNE     6$
3931  016346  104234                            ERROR   234             ;CANT READ SECTORS 1,3,5 ...
3932  016350  000414                            BR      3$
3933
```

```
3934  016352  104231                    6$:   ERROR   231             ;CANT READ SECTORS 11,13,15 ...
3935  016354  000412                          BR      3$
3936
3937  016356  013765  007420  000004    5$:   MOV     TEMP4,RKBA(R5)  ;RESTORE TABLE ADDR
3938  016364  062737  000002  007422          ADD     #2,TEMP5        ;READ 2 SECTORS FROM LAST
3939  016372  013765  007422  000006          MOV     TEMP5,RKDA(R5)
3940  016400  000655                          BR      1$
3941
3942  016402  005237  001512            3$:   INC     BSERR           ;SET BSE FLAG
3943  016406  000550                          BR      TST10           ;;GO TO NEXT TEST
3944  016410  005737  003344            8$:   TST     BSE22H+6        ;TEST CARTRIDGE TYPE
3945  016414  001404                          BEQ     9$              ;BRANCH IF DATA CARTRIDGE
3946  016416  104235                          ERROR   235             ;ALIGNMENT CARTRIDGE USED
3947  016420  005237  001512                  INC     BSERR           ;SET BSE ERROR FLAG
3948  016424  000476                          BR      10$
3949
3950  016426  005237  007416            9$:   INC     TEMP3
3951  016432  023727  007416  000001        CMP     TEMP3,#1
3952  016440  001020                        BNE     11$
3953  016442  005037  007414                CLR     TEMP2           ;SECTOR CTR
3954  016446  012737  005336  007420        MOV     #BSE22S,TEMP4   ;STORE 22 SECTOR SOFTWARE BSE ADDR
3955  016454  013765  007420  000004        MOV     TEMP4,RKBA(R5)
3956  016462  012737  001012  007422        MOV     #1012,TEMP5     ;TRACK 2, SECTOR 12(8)
3957  016470  013765  007422  000006        MOV     TEMP5,RKDA(R5)
3958  016476  000137  016134                JMP     1$              ;REPEAT
3959
3960  016502  023727  007416  000002  11$:   CMP     TEMP3,#2
3961  016510  001020                        BNE     12$
3962  016512  005037  007414                CLR     TEMP2           ;SECTOR CTR
3963  016516  012737  002336  007420        MOV     #BSE20H,TEMP4   ;STORE 20 SECTOR HARDWARE BSE ADDR.
3964  016524  013765  007420  000004        MOV     TEMP4,RKBA(R5)
3965  016532  012737  001001  007422        MOV     #1001,TEMP5     ;TRACK 2, SECTOR 1
3966  016540  013765  007422  000006        MOV     TEMP5,RKDA(R5)
3967  016546  000137  016134                JMP     1$              ;REPEAT
3968
3969  016552  023727  007416  000003  12$:   CMP     TEMP3,#3
3970  016560  001020                        BNE     10$
3971  016562  005037  007414                CLR     TEMP2           ;SECTOR CTR
3972  016566  012737  004336  007420        MOV     #BSE20S,TEMP4   ;STORE 20 SECTOR SOFTWARE BSE ADDR
3973  016574  013765  007420  000004        MOV     TEMP4,RKBA(R5)
3974  016602  012737  001013  007422        MOV     #1013,TEMP5     ;TRACK 2, SECTOR 13(8)
3975  016610  013765  007422  000006        MOV     TEMP5,RKDA(R5)
3976  016616  000137  016134                JMP     1$              ;REPEAT
3977
3978  016622  005737  001216  10$:   TST     $PASS
3979  016626  001040                        BNE     TST10           ;;GO TO NEXT TST IF NOT 1'ST PASS
3980  016630  104401  045516                TYPE    ,MSG17          ;CART SERIAL #
3981  016634  012746  003336                MOV     #BSE22H,-(SP)
3982  016640  004737  042634                JSR     PC,$DB2O        ;CONVERT DBL BINARY WORD TO OCTAL
3983  016644  004737  043204                JSR     PC,$SUPRS       ;TYPE SERIAL #
3984  016650  104401  001205                TYPE    ,$CRLF
3985  016654  104401  001205                TYPE    ,$CRLF
3986
3987  016660  004737  034252                JSR     PC,SUBCLR
3988  016664  104024                        ERROR   24              ;CERR AFTER SCLR
3989                                                                ;GO BACK TO CYL 0
```

```
        016666  012737  000017  007354              MOV     #SEEK,HCS1
        016674  004737  032262                      JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
        016700  104131                              ERROR   131             ;NO RDY AFTER SEEK CMD

        016702  013737  001420  007412              MOV     T50000,TEMP1    ;SETUP TIMEOUT
        016710  004737  032766                      JSR     PC,FATT2        ;FIND ATTN
        016714  104132                              ERROR   132             ;NO ATTN AFTER SEEK CMD

        016716  032737  100000  007354              BIT     #CERR,HCS1
        016724  001401                              BEQ     66$
        016726  104210                              ERROR   210             ;CERR AFTER SEEK CMD

        016730                              66$:




                                            .SBTTL WRITE TESTS


                                            ;;*****************************************************************
                                            ;*TEST 10        BASIC WRITE DATA TEST; 1 WORD
                                            ;*
                                            ;*      THIS TEST VERIFIES THE ABILITY OF THE DRIVE TO WRITE JUST ONE WORD,
                                            ;*      ALL SECTORS ON CYL 0 ARE GIVEN IDENTICAL HEADERS &
                                            ;*      A WRITE COMMAND IS ISSUED. READ & WRITE CHECK COMMANDS ARE NOT
                                            ;*      PERFORMED. THIS TEST PROVIDES THE TIGHTEST POSSIBLE SCOPE LOOP
                                            ;*      FOR A WRITE ERROR.
                                            ;*
                                            ;;*****************************************************************
        016730  000004                      TST10:  SCOPE
        016732  012737  000001  001174              MOV     #1,$TIMES       ;;DO 1 ITERATION
        016740  012706  001100                      MOV     #STACK,SP       ;RESTORE STK PTR

        016744  005737  001512                      TST     BSERR           ;SEE IF ALIGN CART
        016750  001406                              BEQ     2$              ;BR IF NO
        016752  104401  046054                      TYPE    ,MSG40          ;BSE OR ALIGN CART USED
        016756  104401  C.5546                      TYPE    ,MSG26          ;ABORTING BAL OF TESTS
        016762  000137  031476                      JMP     $EOP

        016766  004737  034252              2$:     JSR     PC,SUBCLR
        016772  104024                              ERROR   24              ;CERR AFTER SCLR

        016774  005237  007340                      INC     BADHDR          ;USED FOR VALID HALT

        017000  012700  001522                      MOV     #HDTAB,R0       ;MAKE ALL CYL 0 HEADERS IDENTICAL

        017004  005020                      1$:     CLR     (R0)+           ;HEADER WORD 0: CYL 0
        017006  C12720  140000                      MOV     #140000,(R0)+   ;HEADER WORD 1: SECTOR 0
        017012  012720  140000                      MOV     #140000,(R0)+   ;HEADER WORD 2: XOR OF 1 & 2

        017016  020027  001726                      CMP     R0,#HDTAB+132.  ;ALL HEADERS DONE? (22X6=132)
        017022  001370                              BNE     1$              ;BR IF NO

        017024  012765  001522  000004              MOV     #HDTAB,RKBA(R5) ;HEADER TABLE
        017032  012765  177676  000002              MOV     #-66.,RKWC(R5)  ;WORD COUNT
```

```
4046
4047  017040  012737  000027  007354          MOV     #<WRHEAD>,HCS1
4048  017046  004737  032320                  JSR     PC,DATCMD       ;DO DATA X FOR CMD & GET CONTR RDY
4049  017052  104200                          ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
4050  017054  004737  033722                  JSR     PC,GSTAT        ;GET FRESH STATUS
4051  017060  032737  100000  007354          BIT     #CERR,HCS1
4052  017066  001405                          BEQ     64$
4053  017070  104201                          ERROR   201             ;CERR AFTER WRITE HEADER CMD
4054  017072  104401  045546                  TYPE    ,MSG26          ;ABORTING BAL OF TESTS
4055  017076  000137  031476                  JMP     $EOP
4056  017102                          64$:
4057
4058
4059  017102  104415                          SCOP1
4060  017104  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
4061
4062  017110  004737  034252                  JSR     PC,SUBCLR
4063  017114  104024                          ERROR   24              ;CERR AFTER SCLR
4064
4065  017116  005037  001402                  CLR     SECTOR
4066  017122  013765  001402  000006  3$:     MOV     SECTOR,RKDA(R5) ;TRACK/SECTOR #
4067  017130  012765  001500  000004          MOV     #DATA1,RKBA(R5) ;DATA TO BE ALL 1'S
4068  017136  012765  177777  000002          MOV     #-1,RKWC(R5)    ;WORD COUNT=1
4069
4070
4071  017144  012737  000023  007354          MOV     #<WRDATA>,HCS1
4072  017152  004737  032320                  JSR     PC,DATCMD       ;DO DATA X FOR CMD & GET CONTR RDY
4073  017156  104011                          ERROR   11              ;NO RDY AFTER WRITE DATA CMD
4074  017160  004737  033722                  JSR     PC,GSTAT        ;GET FRESH STATUS
4075  017164  032737  100000  007354          BIT     #CERR,HCS1
4076  017172  001465                          BEQ     68$             ;BR IF NO ERRORS
4077
4078  017174  032737  000200  007370          BIT     #BSE,HER        ;SEE IF BAD SECTOR FLAG
4079  017202  001421                          BEQ     66$             ;BR IF NO
4080  017204  004737  035736                  JSR     PC,TRUERR       ;ELSE SEE IF SECTOR LISTED IN BSE TABLE
4081  017210  000455                          BR      67$             ;RETURN HERE IF NO
4082
4083  017212  005237  001402                  INC     SECTOR          ;RETURN HERE IF YES
4084  017216  023727  001402  000012          CMP     SECTOR,#10.     ;ARE 10 CONSEC. SECTORS BAD
4085  017224  001003                          BNE     65$             ;BR IF NO
4086  017226  104046                          ERROR   46              ;ABORTING TEST DETECTED 10 BAD SECTORS
4087  017230  000137  017432                  JMP     5$              ;BYPASS TEST
4088  017234  012765  100000  000000  65$:    MOV     #CCLR,RKCS1(R5) ;TRY ANOTHER SECTOR
4089  017242  000137  017122                  JMP     3$
4090  017246  104012                  66$:    ERROR   12              ;CERR WITH WRITE DATA CMD
4091
4092  017250  012737  010340  007444          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
4093  017256  005037  007446                  CLR     E.B0            ;EXPECTED MSG B0
4094  017262  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
4095  017270  012737  000001  007452          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
4096  017276  005037  007454                  CLR     E.A2            ;EXPECTED MSG A2
4097  017302  012737  000002  007456          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
4098  017310  012737  000003  007462          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
4099
4100  017316  004737  033100                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
4101  017322  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
```

```
                                                ERROR   52              ;MSG A0 ERROR AFTER WRITE DATA CMD
                                                ERROR   23              ;MSH B0 ERROR
                                                ERROR   53              ;MSG A1 ERROR
                                                ERROR   25              ;MSG B1 ERROR
                                                TYPE    ,MSG26          ;ABORTING BAL OF TESTS
                                                JMP     $EOP
                                        67$:    ERROR   63              ;BAD SECTOR NOT LISTED IN TABLE
                                        68$:

                                                MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
                                                CLR     E.B0            ;EXPECTED MSG B0
                                                MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
                                                MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
                                                CLR     E.A2            ;EXPECTED MSG A2
                                                MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
                                                MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3

                                                JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
                                                .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
                                                ERROR   52              ;MSG A0 ERROR AFTER WRITE DATA CMD
                                                ERROR   23              ;MSH B0 ERROR
                                                ERROR   53              ;MSG A1 ERROR
                                                ERROR   25              ;MSG B1 ERROR
                                        5$:
                                        ;;*****************************************************************
                                        ;*TEST 11       BASIC WRITE DATA TEST; FULL SECTOR
                                        ;*
                                        ;*      THIS TEST VERIFIES THE ABILITY OF THE DRIVE TO WRITE
                                        ;*      A FULL SECTOR.  ALL ZEROS ARE WRITTEN BY THE WRITE DATA COMMAND
                                        ;*      & CHECKED BY A RD DATA COMMAND.  A FURTHER CHECK IS PERFORMED
                                        ;*      BY THE WRT CHK COMMAND.
                                        ;*      THE ABOVE IS REPEATED FOR AN ALL ONES PATTERN.
                                        ;*
                                        ;;*****************************************************************
                                        TST11:  SCOPE
                                                MOV     #1,$TIMES       ;;DO 1 ITERATION
                                                MOV     #STACK,SP       ;RESTORE STK PTR

                                                JSR     PC,SUBCLR
                                                ERROR   24              ;CERR AFTER SCLR

                                                MOV     #HDTAB,RKBA(R5) ;RESTORE TO 22 SECTOR
                                                MOV     #-66.,RKWC(R5)  ;STANDARD FORMAT
                                                CLR     TOCYL

                                                MOV     TOCYL,CALADD    ;SETUP
                                                MOV     #0,HEAD         ;TO FILL
                                                MOV     #0,FORMAT       ;HEADER
                                                JSR     PC,FHDTAB       ;TABLE


                                                MOV     #<WRHEAD>,HCS1
                                                JSR     PC,DATCMD       ;DO DATA X FOR CMD & GET CONTR RDY
                                                ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
                                                JSR     PC,GSTAT        ;GET FRESH STATUS
                                                BIT     #CERR,HCS1
```

```
                                        BEQ     64$
                                        ERROR   201             ;CERR AFTER WRITE HEADER CMD
                                        TYPE    ,MSG26          ;ABORTING BAL OF TESTS
                                        JMP     $EOP
                                64$:
                                        CLR     BADHDR          ;USED FOR VALID HALT
                                        SCOP1
                                        MOV     #STACK,SP       ;RESTORE STK PTR

                                        JSR     PC,SUBCLR
                                        ERROR   24              ;CERR AFTER SCLR

                                        CLR     SECTOR
                                8$:     MOV     SECTOR,RKDA(R5) ;SETUP SECTOR
                                        MOV     #DATA0,RKBA(R5) ;WRITE ALL 0'S
                                        MOV     DATA0,R0
                                1$:     BIS     #BAI,RKCS2(R5)
                                        MOV     #-256.,RKWC(R5)

                                        MOV     #<WRDATA>,HCS1
                                        JSR     PC,DATCMD       ;DO DATA X FOR CMD & GET CONTR RDY
                                        ERROR   11              ;NO RDY AFTER WRITE DATA CMD
                                        JSR     PC,GSTAT        ;GET FRESH STATUS
                                        BIT     #CERR,HCS1
                                        BEQ     68$             ;BR IF NO ERRORS

                                        BIT     #BSE,HER        ;SEE IF BAD SECTOR FLAG
                                        BEQ     66$             ;BR IF NO
                                        JSR     PC,TRUERR       ;ELSE SEE IF SECTOR LISTED IN BSE TABLE
                                        BR      67$             ;RETURN HERE IF NO

                                        INC     SECTOR          ;RETURN HERE IF YES
                                        CMP     SECTOR,#10.     ;ARE 10 CONSEC. SECTORS BAD
                                        BNE     65$             ;BR IF NO
                                        ERROR   46              ;ABORTING TEST DETECTED 10 BAD SECTORS
                                        JMP     7$              ;BYPASS TEST
                                65$:    MOV     #CCLR,RKCS1(R5) ;TRY ANOTHER SECTOR
                                        JMP     8$
                                66$:    ERROR   12              ;CERR WITH WRITE DATA CMD

                                        MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
                                        CLR     E.B0            ;EXPECTED MSG B0
                                        MOV     #<D.SPOK!D.CART!D.DOUR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
                                        MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
                                        CLR     E.A2            ;EXPECTED MSG A2
                                        MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
                                        MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3

                                        JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
                                        .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
                                        ERROR   52              ;MSG A0 ERROR AFTER WRITE DATA CMD
                                        ERROR   23              ;MSH B0 ERROR
                                        ERROR   53              ;MSG A1 ERROR
                                        ERROR   25              ;MSG B1 ERROR
                                        TYPE    ,MSG26          ;ABORTING BAL OF TESTS
```

```
 4214  020040  000137  031476                JMP     $EOP
 4215  020044  104063                67$:    ERROR   63              ;BAD SECTOR NOT LISTED IN TABLE
 4216  020046                        68$:
 4217
 4218  020046  012737  010340  007444        MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
 4219  020054  005037  007446                CLR     E.B0            ;EXPECTED MSG B0
 4220  020060  012737  001720  007450        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
 4221  020066  012737  000001  007452        MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 4222  020074  005037  007454                CLR     E.A2            ;EXPECTED MSG A2
 4223  020100  012737  000002  007456        MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 4224  020106  012737  000003  007462        MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 4225
 4226  020114  004737  033100                JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
 4227  020120  000003                        .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 4228  020122  104052                        ERROR   52              ;MSG A0 ERROR AFTER WRITE DATA CMD
 4229  020124  104023                        ERROR   23              ;MSH B0 ERROR
 4230  020126  104053                        ERROR   53              ;MSG A1 ERROR
 4231  020130  104025                        ERROR   25              ;MSG B1 ERROR
 4232  020132  104415                        SCOP1
 4233  020134  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR
 4234
 4235  020140  004737  034252                JSR     PC,SUBCLR
 4236  020144  104024                        ERROR   24              ;CERR AFTER SCLR
 4237
 4238  020146  013765  001402  000006        MOV     SECTOR,RKDA(R5) ;SETUP SECTOR
 4239  020154  012765  006336  000004        MOV     #RDTAB,RKBA(R5)
 4240  020162  012765  177400  000002        MOV     #-256.,RKWC(R5)
 4241
 4242
 4243  020170  012737  000021  007354        MOV     #<RDDATA>,HCS1
 4244  020176  004737  032320                JSR     PC,DATCMD       ;DO DATA X FOR CMD & GET CONTR RDY
 4245  020202  104013                        ERROR   13              ;NO RDY AFTER READ DATA CMD
 4246  020204  004737  033722                JSR     PC,GSTAT        ;GET FRESH STATUS
 4247  020210  032737  100000  007354        BIT     #CERR,HCS1
 4248  020216  001454                        BEQ     72$
 4249  020220  032737  000200  007370        BIT     #BSE,HER        ;SEE IF BAD SECTOR
 4250  020226  001406                        BEQ     70$
 4251  020230  104065                        ERROR   65              ;DETECTED BSE IN READ BUT NOT IN WRITE CMD.
 4252  020232  000413                        BR      73$
 4253  020234  104401  045546        69$:    TYPE    ,MSG26          ;ABORTING BAL OF TESTS
 4254  020240  000137  031476                JMP     $EOP
 4255
 4256  020244  032737  100000  007370  70$:  BIT     #DCK,HER        ;SEE IF DATA CHECK ERROR
 4257  020252  001402                        BEQ     71$
 4258  020254  104021                        ERROR   21              ;DATA CHECK ERROR AFTER READ CMD (ECC)
 4259  020256  000401                        BR      73$
 4260
 4261  020260  104014                71$:    ERROR   14              ;CERR AFTER READ DATA CMD.
 4262
 4263  020262                        73$:
 4264
 4265  020262  012737  010340  007444        MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
 4266  020270  005037  007446                CLR     E.B0            ;EXPECTED MSG B0
 4267  020274  012737  001720  007450        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
 4268  020302  012737  000001  007452        MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 4269  020310  005037  007454                CLR     E.A2            ;EXPECTED MSG A2
```

```
4270  020314  012737  000002  007456          MOV     #2,E.B2          ;MSG ID FOR EXPECTED MSG B2
4271  020322  012737  000003  007462          MOV     #3,E.B3          ;MSG ID FOR EXPECTED MSG B3
4272
4273  020330  004737  033100                  JSR     PC,CHKMSG        ;CHECK MSGS A0, B0, A1, B1
4274  020334  000003                          .WORD   T.A2!T.B2!0              ;& MSGS SPECIFIED HERE
4275  020336  104054                          ERROR   54               ;MSG A0 ERROR AFTER READ DATA CMD
4276  020340  104026                          ERROR   26               ;MSH B0 ERROR
4277  020342  104056                          ERROR   56               ;MSG A1 ERROR
4278  020344  104030                          ERROR   30               ;MSG B1 ERROR
4279  020346  000732                          BR      69$
4280  020350                          72$:
4281
4282  020350  012737  010340  007444          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
4283  020356  005037  007446                  CLR     E.B0             ;EXPECTED MSG B0
4284  020362  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
4285  020370  012737  000001  007452          MOV     #1,E.B1          ;MSG ID FOR EXPECTED MSG B1
4286  020376  005037  007454                  CLR     E.A2             ;EXPECTED MSG A2
4287  020402  012737  000002  007456          MOV     #2,E.B2          ;MSG ID FOR EXPECTED MSG B2
4288  020410  012737  000003  007462          MOV     #3,E.B3          ;MSG ID FOR EXPECTED MSG B3
4289
4290  020416  004737  033100                  JSR     PC,CHKMSG        ;CHECK MSGS A0, B0, A1, B1
4291  020422  000003                          .WORD   T.A2!T.B2!0              ;& MSGS SPECIFIED HERE
4292  020424  104054                          ERROR   54               ;MSG A0 ERROR AFTER READ DATA CMD
4293  020426  104026                          ERROR   26               ;MSH B0 ERROR
4294  020430  104056                          ERROR   56               ;MSG A1 ERROR
4295  020432  104030                          ERROR   30               ;MSG B1 ERROR
4296
4297  020434  012701  006336                  MOV     #RDTAB,R1
4298  020440  011137  001452          2$:     MOV     (R1),WD1         ;ACTUAL WORD FOR TYPEOUT
4299  020444  010037  001454                  MOV     R0,WD2           ;EXPECTED DATA FOR TYPEOUT
4300
4301  020450  020021                          CMP     R0,(R1)+         ;COMPARE READ DATA TABLE AGAINST
4302  020452  001401                          BEQ     3$               ;THE 'SHOULD BE' VALUE
4303  020454  104020                          ERROR   20               ;READ DATA DID NOT COMPARE WITH WRITE DATA
4304
4305  020456  020127  007336          3$:     CMP     R1,#RDTAB+512.   ;SEE IF REACHED END OF TABLE
4306  020462  001366                          BNE     2$               ;BR IF NO & DO NEXT WORD
4307
4308  020464  020037  001474                  CMP     R0,DATA0         ;SEE IF DID ALL 0'S
4309  020470  001401                          BEQ     4$               ;BR IF YES
4310  020472  000412                          BR      5$
4311
4312  020474  012765  001500  000004  4$:     MOV     #DATA1,RKBA(R5) ;WRITE ALL 1'S
4313  020502  013700  001500                  MOV     DATA1,R0
4314  020506  013765  001402  000006          MOV     SECTOR,RKDA(R5)
4315  020514  000137  017630                  JMP     1$
4316
4317  020520                          5$:
4318  020520  104415                          SCOP1
4319  020522  012706  001100                  MOV     #STACK,SP        ;RESTORE STK PTR
4320
4321  020526  004737  034252                  JSR     PC,SUBCLR
4322  020532  104024                          ERROR   24               ;CERR AFTER SCLR
4323
4324  020534  052765  000020  000010          BIS     #BAI,RKCS2(R5)  ;THIS PORTION OF THE TEST CHECKS
4325  020542  012765  001500  000004          MOV     #DATA1,RKBA(R5) ;OUT THE WRITE CHECK CMD
```

```
 4326  020550  012765  177400  000002          MOV     #-256.,RKWC(R5) ;ALL 1'S WERE PREVIOUSLY WRITTEN
 4327  020556  013765  001402  000006          MOV     SECTOR,RKDA(R5)
 4328
 4329  020564  012737  000031  007354          MOV     #<WRTCHK>,HCS1
 4330  020572  004737  032320                  JSR     PC,DATCMD       ;DO DATA X FOR CMD & GET CONTR RDY
 4331  020576  104015                          ERROR   15              ;NO RDY AFTER WRITE CHECK CMD
 4332  020600  004737  033722                  JSR     PC,GSTAT        ;GET FRESH STATUS
 4333  020604  032737  100000  007354          BIT     #CERR,HCS1
 4334  020612  001453                          BEQ     75$
 4335  020614  032737  040000  007356          BIT     #WCE,HCS2       ;SEE IF WRITE CHECK ERROR
 4336  020622  001410                          BEQ     74$
 4337  020624  016537  000024  001452          MOV     RKDB(R5),WD1    ;ACTUAL WORD FOR PRINTOUT
 4338  020632  013737  001500  001454          MOV     DATA1,WD2               ;EXPECTED WORD FOR TYPEOUT
 4339  020640  104016                          ERROR   16              ;WCE AFTER WRITE CMD
 4340  020642  000437                          BR      75$
 4341
 4342  020644  104022                  74$:    ERROR   22              ;CERR AFTER WRITE CHECK CMD
 4343
 4344  020646  012737  010340  007444          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
 4345  020654  005037  007446                  CLR     E.B0            ;EXPECTED MSG B0
 4346  020660  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
 4347  020666  012737  000001  007452          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 4348  020674  005037  007454                  CLR     E.A2            ;EXPECTED MSG A2
 4349  020700  012737  000002  007456          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 4350  020706  012737  000003  007462          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 4351
 4352  020714  004737  033100                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
 4353  020720  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 4354  020722  104057                          ERROR   57              ;MSG A0 ERROR AFTER WRITE CHECK CMD
 4355  020724  104031                          ERROR   31              ;MSH B0 ERROR
 4356  020726  104060                          ERROR   60              ;MSG A1 ERROR
 4357  020730  104032                          ERROR   32              ;MSG B1 ERROR
 4358  020732  104401  045546                  TYPE    ,MSG26          ;ABORTING BAL OF TESTS
 4359  020736  000137  031476                  JMP     $EOP
 4360
 4361  020742                          75$:
 4362
 4363  020742  012737  010340  007444          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
 4364  020750  005037  007446                  CLR     E.B0            ;EXPECTED MSG B0
 4365  020754  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
 4366  020762  012737  000001  007452          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 4367  020770  005037  007454                  CLR     E.A2            ;EXPECTED MSG A2
 4368  020774  012737  000002  007456          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 4369  021002  012737  000003  007462          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 4370
 4371  021010  004737  033100                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
 4372  021014  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 4373  021016  104057                          ERROR   57              ;MSG A0 ERROR AFTER WRITE CHECK CMD
 4374  021020  104031                          ERROR   31              ;MSH B0 ERROR
 4375  021022  104060                          ERROR   60              ;MSG A1 ERROR
 4376  021024  104032                          ERROR   32              ;MSG B1 ERROR
 4377
 4378  021026  104415                          SCOP1
 4379  021030  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
 4380
 4381  021034  004737  034252                  JSR     PC,SUBCLR
```

```
 4382  021040  104024                          ERROR   24              ;CERR AFTER SCLR
 4383
 4384  021042  012765  001474  000004          MOV     #DATA0,RKBA(R5) ;SETUP TO CHECK AGAINST WRONG DATA
 4385  021050  012765  177400  000002          MOV     #-256.,RKWC(R5)
 4386  021056  013765  001402  000006          MOV     SECTOR,RKDA(R5)
 4387  021064  012737  000031  007354          MOV     #WRTCHK,HCS1
 4388  021072  004737  032320                  JSR     PC,DATCMD       ;DO WRITE CHECK CMD. & GET CONTR RDY.
 4389  021076  104015                          ERROR   15              ;NO RDY AFTER WRITE CHECK CMD
 4390  021100  004737  033722                  JSR     PC,GSTAT        ;GET FRESH STATUS
 4391  021104  032737  040000  007356          BIT     #WCE,HCS2       ;EXPECT MISCOMPARE
 4392  021112  001001                          BNE     6$
 4393  021114  104017                          ERROR   17              ;WRITE CHECK CMD NOT FUNCTIONING
 4394                                                                  ;WITH INTENTIONAL MISCOMPARE
 4395  021116                          6$:
 4396
 4397  021116  012737  010340  007444          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.AC  ;EXPECTED MSG A0
 4398  021124  005037  007446                  CLR     E.B0            ;EXPECTED MSG B0
 4399  021130  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
 4400  021136  012737  000001  007452          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 4401  021144  005037  007454                  CLR     E.A2            ;EXPECTED MSG A2
 4402  021150  012737  000002  007456          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 4403  021156  012737  000003  007462          MOV     #3,E.B3         ,MSG ID FOR EXPECTED MSG B3
 4404
 4405  021164  004737  033100                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
 4406  021170  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
 4407  021172  104057                          ERROR   57              ;MSG A0 ERROR AFT WRT CHK CMD
 4408  021174  104031                          ERROR   31              ;MSH B0 ERROR
 4409  021176  104060                          ERROR   60              ;MSG A1 ERROR
 4410  021200  104032                          ERROR   32              ;MSG B1 ERROR
 4411  021202                          7$:
 4412
 4413                                  ;;******************************************************************
 4414                                  ;*TEST 12       20 SECTOR FORMAT TEST
 4415                                  ;*
 4416                                  ;*      ALL 1'S ARE WRITTEN ON A FULL SECTOR IN 20 SECTOR FORMAT.
 4417                                  ;*      MSG B0,B1 ARE CHECKED FOR ANY ERROR CONDITION.
 4418                                  ;*      CYL 0, TRACK 0, & SECTOR 0 IS USED.
 4419                                  ;*
 4420                                  ;******************************************************************
 4421  021202  000004                  TST12:  SCOPE
 4422  021204  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
 4423  021212  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
 4424
 4425  021216  004737  034252                  JSR     PC,SUBCLR
 4426  021222  104024                          ERROR   24              ;CERR AFTER SCLR
 4427
 4428  021224  012765  001522  000004          MOV     #HDTAB,RKBA(R5) ;HEADER WORD TABLE
 4429  021232  012765  177704  000002          MOV     #-60.,RKWC(R5)  ;WORD COUNT FOR 20 SECTOR FMT
 4430  021240  005037  001346                  CLR     TOCYL
 4431  021244  005237  007340                  INC     BADHDR          ;USED FOR VALID HALT
 4432
 4433
 4434  021250  013737  001346  001362          MOV     TOCYL,CALADD    ;SETUP
 4435  021256  012737  000000  001460          MOV     #0,HEAD         ;TO FILL
 4436  021264  012737  000001  001466          MOV     #1,FORMAT       ;HEADER
 4437  021272  004737  035250                  JSR     PC,FHDTAB       ;TABLE
```

```
                                                MOV     #<CFMT!WRHEAD>,HCS1
021304  004737  032320                          JSR     PC,DATCMD       ;DO DATA X FOR CMD & GET CONTR RDY
021310  104200                                  ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
021312  012737  010001  007354                  MOV     #<CFMT!SELDRV>,HCS1
021320  004737  032262                          JSR     PC,DOCMD
021324  104117                                  ERROR   117             ;NO RDY AFTER SELDRV CMD
021326  032737  100000  007354                  BIT     #CERR,HCS1
021334  001405                                  BEQ     64$
021336  104201                                  ERROR   201             ;CERR AFTER WRITE HEADER CMD
021340  104401  045546                          TYPE    ,MSG26          ;ABORTING BAL OF TESTS
021344  000137  031476                          JMP     $EOP
021350                                  64$:
021350  104415                                  SCOP1
021352  012706  001100                          MOV     #STACK,SP       ;RESTORE STK PTR
021356  004737  034252                          JSR     PC,SUBCLR
021362  104024                                  ERROR   24              ;CERR AFTER SCLR
021364  005037  001402                          CLR     SECTOR
021370  013765  001402  000006          4$:     MOV     SECTOR,RKDA(R5)
021376  012765  001500  000004                  MOV     #DATA1,RKBA(R5) ;WRITE ALL 1'S
021404  052765  000020  000010                  BIS     #BAI,RKCS2(R5)  ;BUSS ADDR INCR INHIBIT
021412  012765  177400  000002                  MOV     #-256.,RKWC(R5)  ;DO FULL SECTOR
021420  012737  010023  007354                  MOV     #<CFMT!WRDATA>,HCS1
021426  004737  032320                          JSR     PC,DATCMD       ;DO DATA X FOR CMD & GET CONTR RDY
021432  104011                                  ERROR   11              ;NO RDY AFTER WRITE DATA CMD
021434  012737  010001  007354                  MOV     #<CFMT!SELDRV>,HCS1
021442  004737  032262                          JSR     PC,DOCMD
021446  104117                                  ERROR   117             ;NO RDY AFTER SELDRV CMD
021450  032737  100000  007354                  BIT     #CERR,HCS1
021456  001465                                  BEQ     68$             ;BR IF NO ERRORS
021460  032737  000200  007370                  BIT     #BSE,HER        ;SEE IF BAD SECTOR FLAG
021466  001421                                  BEQ     66$             ;BR IF NO
021470  004737  035736                          JSR     PC,TRUERR       ;ELSE SEE IF SECTOR LISTED IN BSE TABLE
021474  000455                                  BR      67$             ;RETURN HERE IF NO
021476  005237  001402                          INC     SECTOR          ;RETURN HERE IF YES
021502  023727  001402  000012                  CMP     SECTOR,#10.     ;ARE 10 CONSEC. SECTORS BAD
021510  001003                                  BNE     65$             ;BR IF NO
021512  104046                                  ERROR   46              ;ABORTING TEST DETECTED 10 BAD SECTORS
021514  000137  022420                          JMP     3$              ;BYPASS TEST
021520  012765  100000  000000  65$:    MOV     #CCLR,RKCS1(R5) ;TRY ANOTHER SECTOR
021526  000137  021370                  JMP     4$
021532  104012                  66$:    ERROR   12              ;CERR WITH WRITE DATA CMD
021534  012737  010340  007444                  MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
021542  005037  007446                          CLR     E.B0            ;EXPECTED MSG B0
021546  012737  001720  007450                  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
021554  012737  000001  007452                  MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
021562  005037  007454                          CLR     E.A2            ;EXPECTED MSG A2
```

```
 4494  021566  012737  000002  007456          MOV     #2,E.B2          ;MSG ID FOR EXPECTED MSG B2
 4495  021574  012737  000003  007462          MOV     #3,E.B3          ;MSG ID FOR EXPECTED MSG B3
 4496
 4497  021602  004737  033100                  JSR     PC,CHKMSG        ;CHECK MSGS A0, B0, A1, B1
 4498  021606  000003                          .WORD   T.A2!T.B2!0              ;& MSGS SPECIFIED HERE
 4499  021610  104052                          ERROR   52               ;MSG A0 ERROR AFTER WRITE DATA CMD
 4500  021612  104023                          ERROR   23               ;MSH B0 ERROR
 4501  021614  104053                          ERROR   53               ;MSG A1 ERROR
 4502  021616  104025                          ERROR   25               ;MSG B1 ERROR
 4503  021620  104401  045546                  TYPE    ,MSG26           ;^PORTING BAL OF TESTS
 4504  021624  000137  031476                  JMP     $EOP
 4505  021630  104063                  67$:    ERROR   63               ;BAD SECTOR NOT LISTED IN TABLE
 4506  021632                          68$:
 4507  021632  012737  010001  007354          MOV     #<CFMT!SELDRV>,HCS1
 4508  021640  004737  032262                  JSR     PC,DOCMD
 4509  021644  104117                          ERROR   117              ;RDY NOT FOUND AFTER SELDRV CMD
 4510  021646  032737  001000  007402          BIT     #D.FORM,HMR2
 4511  021654  001001                          BNE     1$
 4512  021656  104102                          ERROR   102              ;FORMAT NOT SET
 4513
 4514  021660                          1$:
 4515
 4516  021660  012737  010340  007444          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
 4517  021666  005037  007446                  CLR     E.B0             ;EXPECTED MSG B0
 4518  021672  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
 4519  021700  012737  000001  007452          MOV     #1,E.B1          ;MSG ID FOR EXPECTED MSG B1
 4520  021706  005037  007454                  CLR     E.A2             ;EXPECTED MSG A2
 4521  021712  012737  000002  007456          MOV     #2,E.B2          ;MSG ID FOR EXPECTED MSG B2
 4522  021720  012737  000003  007462          MOV     #3,E.B3          ;MSG ID FOR EXPECTED MSG B3
 4523
 4524  021726  004737  033100                  JSR     PC,CHKMSG        ;CHECK MSGS A0, B0, A1, B1
 4525  021732  000000                          .WORD   0!0!0            ;& MSGS SPECIFIED HERE
 4526  021734  104052                          ERROR   52               ;MSG A0 ERROR AFTER WRITE DATA CMD
 4527  021736  104023                          ERROR   23               ;MSH B0 ERROR
 4528  021740  104053                          ERROR   53               ;MSG A1 ERROR
 4529  021742  104025                          ERROR   25               ;MSG B1 ERROR
 4530  021744  012765  177400  000002          MOV     #-256.,RKWC(R5)
 4531  021752  012765  001500  000004          MOV     #DATA1,RKBA(R5)
 4532  021760  052765  000020  000010          BIS     #BAI,RKCS2(R5)
 4533  021766  013765  001402  000006          MOV     SECTOR,RKDA(R5)
 4534
 4535  021774  012737  010031  007354          MOV     #<CFMT!WRTCHK>,HCS1
 4536  022002  004737  032320                  JSR     PC,DATCMD        ;DO DATA X FOR CMD & GET CONTR RDY
 4537  022006  104015                          ERROR   15               ;NO RDY AFTER WRITE CHECK CMD
 4538  022010  012737  010001  007354          MOV     #<CFMT!SELDRV>,HCS1
 4539  022016  004737  032262                  JSR     PC,DOCMD
 4540  022022  104117                          ERROR   117              ;NO RDY AFTER SELDRV CMD
 4541  022024  032737  100000  007354          BIT     #CERR,HCS1
 4542  022032  001453                          BEQ     70$
 4543  022034  032737  040000  007356          BIT     #WCE,HCS2        ;SEE IF WRITE CHECK ERROR
 4544  022042  001410                          BEQ     69$
 4545  022044  016537  000024  001452          MOV     RKDB(R5),WD1     ;ACTUAL WORD FOR PRINTOUT
 4546  022052  013737  001500  001454          MOV     DATA1,WD2                ;EXPECTED WORD FOR TYPEOUT
 4547  022060  104016                          ERROR   16               ;WCE AFTER WRITE CMD
 4548  022062  000437                          BR      70$
 4549
```

```
 4550  022064  104022                  69$:    ERROR   22              ;CERR AFTER WRITE CHECK CMD
 4551
 4552  022066  012737  010340  007444          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
 4553  022074  005037  007446                  CLR     E.B0            ;EXPECTED MSG B0
 4554  022100  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
 4555  022106  012737  000001  007452          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 4556  022114  005037  007454                  CLR     E.A2            ;EXPECTED MSG A2
 4557  022120  012737  000002  007456          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 4558  022126  012737  000003  007462          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 4559
 4560  022134  004737  033100                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
 4561  022140  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 4562  022142  104057                          ERROR   57              ;MSG A0 ERROR AFTER WRITE CHECK CMD
 4563  022144  104031                          ERROR   31              ;MSH B0 ERROR
 4564  022146  104060                          ERROR   60              ;MSG A1 ERROR
 4565  022150  104032                          ERROR   32              ;MSG B1 ERROR
 4566  022152  104401  045546                  TYPE    ,MSG26          ;ABORTING BAL OF TESTS
 4567  022156  000137  031476                  JMP     $EOP
 4568
 4569  022162                          70$:
 4570  022162  012737  010001  007354          MOV     #<CFMT!SELDRV>,HCS1
 4571  022170  004737  032262                  JSR     PC,DOCMD
 4572  022174  104117                          ERROR   117             ;NO RDY AFTER SELDRV CMD
 4573  022176  032737  001000  007402          BIT     #D.FORM,HMR2
 4574  022204  001001                          BNE     2$
 4575  022206  104103                          ERROR   103             ;FORMAT NOT SET
 4576
 4577  022210                          2$:
 4578
 4579  022210  012737  010340  007444          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
 4580  022216  005037  007446                  CLR     E.B0            ;EXPECTED MSG B0
 4581  022222  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
 4582  022230  012737  000001  007452          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 4583  022236  005037  007454                  CLR     E.A2            ;EXPECTED MSG A2
 4584  022242  012737  000002  007456          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 4585  022250  012737  000003  007462          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 4586
 4587  022256  004737  033100                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
 4588  022262  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
 4589  022264  104057                          ERROR   57              ;MSG A0 ERROR AFTER WRITE CHECK CMD
 4590  022266  104031                          ERROR   31              ;MSH B0 ERROR
 4591  022270  104060                          ERROR   60              ;MSG A1 ERROR
 4592  022272  104032                          ERROR   32              ;MSG B1 ERROR
 4593  022274  104415                          SCOP1
 4594  022276  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
 4595
 4596  022302  004737  034252                  JSR     PC,SUBCLR
 4597  022306  104024                          ERROR   24              ;CERR AFTER SCLR
 4598
 4599  022310  012765  001522  000004          MOV     #HDTAB,RKBA(R5) ;RESTORE CYL 0 TO 22 SECTOR FMT
 4600  022316  012765  177676  000002          MOV     #-66., RKWC(R5)
 4601  022324  005037  001346                  CLR     TOCYL
 4602
 4603
 4604  022330  013737  001346  001362          MOV     TOCYL,CALADD    ;SETUP
 4605  022336  012737  000000  001460          MOV     #0,HEAD         ;TO FILL
```

```
 4606  022344  012737  000000  001466          MOV     #0,FORMAT       ;HEADER
 4607  022352  004737  035250                  JSR     PC,FHDTAB       ;TABLE
 4608
 4609
 4610  022356  012737  000027  007354          MOV     #<WRHEAD>,HCS1
 4611  022364  004737  032320                  JSR     PC,DATCMD       ;DO DATA X FOR CMD & GET CONTR RDY
 4612  022370  104200                          ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
 4613  022372  004737  033722                  JSR     PC,GSTAT        ;GET FRESH STATUS
 4614  022376  032737  100000  007354          BIT     #CERR,HCS1
 4615  022404  001405                          BEQ     71$
 4616  022406  104201                          ERROR   201             ;CERR AFTER WRITE HEADER CMD
 4617  022410  104401  045546                  TYPE    ,MSG26          ;ABORTING BAL OF TESTS
 4618  022414  000137  031476                  JMP     $EOP
 4619  022420                          71$:
 4620
 4621  022420  005037  007340          3$:     CLR     BADHDR          ;USED FOR VALID HALT
 4622
 4623                                  ;;*****************************************************************
 4624                                  ;*TEST 13       TEST OFFSET & RTC LOGIC
 4625                                  ;*
 4626                                  ;*      THE HEADS ARE FIRST OFFSET BY OFFSET COMMANDS.
 4627                                  ;*      THIS TEST CHECKS THE RTC LOGIC BY VERIFYING THAT THE
 4628                                  ;*      'OFFSET ON' BIT (MSG A,00) RESETS AND THE OFFSET REG
 4629                                  ;*      BECOMES THE CYL DIFF INFO WHEN A SEEK CMD TO A
 4630                                  ;*      DIFFERENT CYLINDER IS ISSUED
 4631                                  ;*      IT ALSO TESTS THAT DRIVE CLEAR & SEEK TO SELF WILL NOT
 4632                                  ;*      CLEAR THE 'OFFSET ON' BIT OR THE OFFSET REG.
 4633                                  ;*      ALL OFFSET POSITIONS IN BOTH DIRECTIONS ARE CHECKED
 4634                                  ;*
 4635                                  ;;*****************************************************************
 4636  022424  000004                  TST13:  SCOPE
 4637  022426  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
 4638  022434  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
 4639
 4640  022440  012702  000001                  MOV     #1,R2           ;MIN POS OFFSET
 4641
 4642  022444  004737  034252          1$:     JSR     PC,SUBCLR
 4643  022450  104024                          ERROR   24              ;CERR AFTER SCLR
 4644
 4645  022452  010265  000016                  MOV     R2,RKASOF(R5)   ;SET OFFSET
 4646
 4647  022456  012737  022544  001176          MOV     #64$,$ESCAPE
 4648  022464  012737  000015  007354          MOV     #OFFSET,HCS1
 4649  022472  004737  032262                  JSR     PC,DOCMD        ;DO RECAL CMD & GET CONTR RDY
 4650  022476  104033                          ERROR   33              ;NO RDY AFTER OFFSET CMD
 4651
 4652  022500  012737  032140  007444          MOV     #<D.PIP!D.SPIN!D.OFF!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
 4653  022506  005037  007446                  CLR     E.B0
 4654  022512  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
 4655  022520  012737  000001  007452          MOV     #1,E.B1
 4656
 4657  022526  004737  033100                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
 4658  022532  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
 4659  022534  104035                          ERROR   35              ;MSG A0 ERROR DURING OFFSET CMD
 4660  022536  104061                          ERROR   61              ;MSH B0 ERROR
 4661  022540  104036                          ERROR   36              ;MSG A1 ERROR
```

```
4662  022542  104062                          ERROR   62              ;MSG B1 ERROR
4663
4664  022544  005037  001176          64$:    CLR     $ESCAPE
4665  022550  013737  001416  007412          MOV     T5000,TEMP1     ;SETUP TIMEOUT
4666  022556  004737  032766                  JSR     PC,FATT2        ;FIND ATTN
4667  022562  104034                          ERROR   34              ;NO ATTN AFTER OFFSET CMD
4668
4669
4670  022564  012737  052340  007444          MOV     #<D.DSC!D.OFF!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
4671  022572  005037  007446                  CLR     E.B0            ;EXPECTED MSG B0
4672  022576  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
4673  022604  012737  000001  007452          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
4674  022612  005037  007454                  CLR     E.A2            ;EXPECTED MSG A2
4675  022616  012737  000002  007456          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
4676  022624  012737  000003  007462          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
4677
4678  022632  004737  033100                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
4679  022636  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
4680  022640  104260                          ERROR   260             ;MSG A0 ERROR AFTER OFFSET CMD
4681  022642  104261                          ERROR   261             ;MSH B0 ERROR
4682  022644  104037                          ERROR   37              ;MSG A1 ERROR
4683  022646  104040                          ERROR   40              ;MSG B1 ERROR
4684
4685  022650  005737  001360                  TST     CYLADD
4686  022654  001401                          BEQ     17$
4687  022656  104042                          ERROR   42              ;CYL ADDR IN B2 WAS NOT 0
4688                                                                  ;AFTER OFFSET CMD FROM CYL 0
4689  022660  042737  001000  001356  17$:    BIC     #1000,CYLDIF    ;GET RID OF HIGH BIT
4690  022666  010265  000016                  MOV     R2,RKASOF(R5)   ;REFRESH RKASOF
4691
4692  022672  032702  000200                  BIT     #BIT7,R2
4693  022676  001005                          BNE     65$             ;BR IF NEG OFFSET
4694
4695  022700  020237  001356                  CMP     R2,CYLDIF       ;CHECK POS OFFSET
4696  022704  001406                          BEQ     66$
4697  022706  104114                          ERROR   114             ;OFFSET IN A2 NOT = RKASOF
4698  022710  000404                          BR      66$             ;AFTER OFFSET CMD
4699
4700  022712  020137  001356          65$:    CMP     R1,CYLDIF       ;CHECK NEG OFFSET
4701  022716  001401                          BEQ     66$
4702  022720  104114                          ERROR   114             ;OFFSET IN A2 NOT = RKASOF
4703                                                                  ;AFTER OFFSET CMD
4704  022722                          66$:
4705
4706  022722  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
4707  022730  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
4708  022736  012737  000005  007354          MOV     #CLEAR,HCS1
4709  022744  004737  032262                  JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
4710  022750  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
4711  022752  004737  032640                  JSR     PC,TSTATN       ;TEST FOR ATTN
4712  022756  000401                          BR      67$
4713  022760  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
4714  022762                          67$:
4715
4716  022762  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
4717  022770  004737  033722                  JSR     PC,GSTAT
```

```
 4718  022774  032737  002000  007402          BIT     #D.OFF,HMR2
 4719  023002  001001                          BNE     4$
 4720  023004  104043                          ERROR   43              ;OFFSET BIT IN RKMR2 CLEARED
 4721                                                                  ;AFTER DRIVE CLEAR CMD & SELECT DRV CMD
 4722  023006  012737  012340  007444  4$:     MOV     #<D.OFF!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED A0
 4723  023014  005037  007446                  CLR     E.B0
 4724  023020  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
 4725  023026  012737  000001  007452          MOV     #1,E.B1
 4726
 4727  023034  004737  033100                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
 4728  023040  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 4729  023042  104273                          ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
 4730  023044  104265                          ERROR   265             ;MSH B0 ERROR
 4731  023046  104274                          ERROR   274             ;MSG A1 ERROR
 4732  023050  104266                          ERROR   266             ;MSG B1 ERROR
 4733  023052  042737  001000  001356          BIC     #1000,CYLDIF    ;GET RID OF HIGH BIT
 4734  023060  010265  000016                  MOV     R2,RKASOF(R5)   ;REFRESH RKASOF
 4735
 4736  023064  032702  000200                  BIT     #BIT7,R2
 4737  023070  001005                          BNE     68$             ;BR IF NEG OFFSET
 4738
 4739  023072  020237  001356                  CMP     R2,CYLDIF       ;CHECK POS OFFSET
 4740  023076  001406                          BEQ     69$
 4741  023100  104115                          ERROR   115             ;OFFSET IN A2 NOT = RKASOF
 4742  023102  000404                          BR      69$             ;AFTER DRIVE CLEAR CMD
 4743
 4744  023104  020137  001356          68$:    CMP     R1,CYLDIF       ;CHECK NEG OFFSET
 4745  023110  001401                          BEQ     69$
 4746  023112  104115                          ERROR   115             ;OFFSET IN A2 NOT = RKASOF
 4747                                                                  ;AFTER DRIVE CLEAR CMD
 4748  023114                          69$:
 4749
 4750  023114  012737  000017  007354          MOV     #SEEK,HCS1
 4751  023122  004737  032262                  JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
 4752  023126  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD
 4753
 4754  023130  013737  001420  007412          MOV     T50000,TEMP1    ;SETUP TIMEOUT
 4755  023136  004737  032766                  JSR     PC,FAT?2        ;FIND ATTN
 4756  023142  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
 4757
 4758  023144  032737  100000  007354          BIT     #CERR,HCS1
 4759  023152  001401                          BEQ     70$
 4760  023154  104210                          ERROR   210             ;CERR AFTER SEEK CMD
 4761
 4762  023156                          70$:
 4763
 4764
 4765  023156  032737  002000  007402          BIT     #D.OFF,HMR2
 4766  023164  001001                          BNE     7$
 4767  023166  104045                          ERROR   45              ;OFFSET BIT CLEARED IN RKMR2 AFTER SEEK TO SELF.
 4768
 4769  023170                          7$:
 4770
 4771  023170  012737  052340  007444          MOV     #<D.DSC!D.OFF!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
 4772  023176  005037  007446                  CLR     E.B0            ;EXPECTED MSG B0
 4773  023202  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
```

```
4774  023210  012737  000001  007452        MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
4775  023216  005037  007454                CLR     E.A2            ;EXPECTED MSG A2
4776  023222  012737  000002  007456        MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
4777  023230  012737  000003  007462        MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
4778
4779  023236  004737  033100                JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
4780  023242  000003                        .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
4781  023244  104140                        ERROR   140             ;MSG A0 ERROR AFTER SEEK TO SELF
4782  023246  104141                        ERROR   141             ;MSH B0 ERROR
4783  023250  104142                        ERROR   142             ;MSG A1 ERROR
4784  023252  104143                        ERROR   143             ;MSG B1 ERROR
4785  023254  042737  001000  001356        BIC     #1000,CYLDIF    ;GET RID OF HIGH BIT
4786  023262  010265  000016                MOV     R2,RKASOF(R5)   ;REFRESH RKASOF
4787
4788  023266  032702  000200                BIT     #BIT7,R2
4789  023272  001005                        BNE     71$             ;BR IF NEG OFFSET
4790
4791  023274  020237  001356                CMP     R2,CYLDIF       ;CHECK POS OFFSET
4792  023300  001406                        BEQ     72$
4793  023302  104123                        ERROR   123             ;OFFSET IN A2 NOT = RKASOF
4794  023304  000404                        BR      72$             ;AFTER SEEK TO SELF
4795
4796  023306  020137  001356        71$:    CMP     R1,CYLDIF       ;CHECK NEG OFFSET
4797  023312  001401                        BEQ     72$
4798  023314  104123                        ERROR   123             ;OFFSET IN A2 NOT = RKASOF
4799                                                                ;AFTER SEEK TO SELF
4800  023316                        72$:
4801
4802  023316  004737  034252                JSR     PC,SUBCLR
4803  023322  104024                        ERROR   24              ;CERR AFTER SCLR
4804
4805  023324  012737  000012  001346        MOV     #10.,TOCYL
4806  023332  012765  000012  000020        MOV     #10.,RKDC(R5)   ;SETUP CYL 10
4807                                                                ;DO ACTUAL IMPLIED SEEK TO CYL 10 TO VERIFY
4808                                                                ;OFFSET BIT IN RKMR2 CLEARED
4809
4810
4811  023340  012700  001726                MOV     #RHTAB,R0
4812  023344  012737  000025  007354        MOV     #<RDHEAD>,HCS1
4813  023352  004737  032320                JSR     PC,DATCMD       ;DO DATA X FOR CMD & GET CONTR RDY
4814  023356  104171                        ERROR   171             ;NO RDY AFTER READ HEADER CMD
4815  023360  032737  100000  007354        BIT     #CERR,HCS1
4816  023366  001405                        BEQ     74$
4817  023370  104174                        ERROR   174             ;CERR AFTER READ HEADER CMD
4818  023372  104401  045546                TYPE    ,MSG26          ;ABORTING BAL OF TESTS
4819  023376  000137  031476                JMP     $EOP
4820
4821  023402  016520  000024        74$:    MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
4822  023406  016520  000024                MOV     RKDB(R5),(R0)+  ;2'ND WORD
4823  023412  016520  000024                MOV     RKDB(R5),(R0)+  ;3'RD WORD
4824
4825
4826  023416  032765  100000  000010        BIT     #DLT,RKCS2(R5)
4827  023424  001407                        BEQ     75$
4828  023426  004737  033722                JSR     PC,GSTAT
4829  023432  104173                        ERROR   173             ;DLT AFTER READ HEADER CMD
```

```
 4830  023434  104401  045546                TYPE    ,MSG26          ;ABORTING BAL OF TESTS
 4831  023440  000137  031476                JMP     $EOP
 4832  023444                        75$:
 4833
 4834  023444  012737  010340  007444        MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
 4835  023452  005037  007446                CLR     E.B0            ;EXPECTED MSG B0
 4836  023456  012737  001720  007450        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
 4837  023464  012737  000001  007452        MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 4838  023472  005037  007454                CLR     E.A2            ;EXPECTED MSG A2
 4839  023476  012737  000002  007456        MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 4840  023504  012737  000003  007462        MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 4841
 4842  023512  004737  033100                JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
 4843  023516  000003                        .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 4844  023520  104301                        ERROR   301             ;MSG A0 ERROR AFTER READ HEADER CMD
 4845  023522  104271                        ERROR   271             ;MSH B0 ERROR
 4846  023524  104302                        ERROR   302             ;MSG A1 ERROR
 4847  023526  104272                        ERROR   272             ;MSG B1 ERROR
 4848
 4849  023530  023737  001726  001346        CMP     RHTAB,TOCYL     ;CHECK WORD 0 ONLY, CYL#
 4850  023536  001401                        BEQ     73$             ;BR IF SAME
 4851  023540  104051                        ERROR   51              ;WRONG CYL# ON HEADER
 4852  023542                        73$:
 4853
 4854
 4855  023542  032737  002000  007402        BIT     #D.OFF,HMR2
 4856  023550  001401                        BEQ     9$
 4857  023552  104101                        ERROR   101             ;OFFSET NOT CLEARED AFTER READ HEADER WITH MOVEMENT
 4858
 4859  023554  023727  001360  000012  9$:   CMP     CYLADD,#10.
 4860  023562  001401                        BEQ     10$
 4861  023564  104122                        ERROR   122             ;DID NOT GO TO CYL 10
 4862
 4863  023566  005737  001356        10$:    TST     CYLDIF
 4864  023572  001401                        BEQ     16$
 4865  023574  104101                        ERROR   101             ;OFFSET NOT CLEARED IN RKMR2
 4866
 4867  023576  004737  034252        16$:    JSR     PC,SUBCLR
 4868  023602  104024                        ERROR   24              ;CERR AFTER SCLR
 4869
 4870  023604  012737  000017  007354        MOV     #SEEK,HCS1
 4871  023612  004737  032262                JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
 4872  023616  104131                        ERROR   131             ;NO RDY AFTER SEEK CMD
 4873
 4874  023620  013737  001420  007412        MOV     T50000,TEMP1    ;SETUP TIMEOUT
 4875  023626  004737  032766                JSR     PC,FATT2        ;FIND ATTN
 4876  023632  104132                        ERROR   132             ;NO ATTN AFTER SEEK CMD
 4877
 4878  023634  032737  100000  007354        BIT     #CERR,HCS1
 4879  023642  001401                        BEQ     76$
 4880  023644  104210                        ERROR   210             ;CERR AFTER SEEK CMD
 4881
 4882  023646                        76$:
 4883
 4884
 4885  023646  032702  000200                BIT     #BIT7,R2        ;SEE IF DOING NEG OFFSETS
```

```
                                BNE     18$             ;BR IF YES

                                INC     R2
                                CMP     R2,#61          ;SEE IF JUST DID MAX POS OFFSET
                                BEQ     20$             ;BR IF YES
                                JMP     1$              ;ELSE DO NEXT POS OFFSET

                        20$:    MOV     #201,R2         ;SETUP NEG OFFSET FOR RKASOF
                                MOV     #101,R1         ;SETUP NEG OFFSET OFOR MSG A
                                JMP     1$              ;DO NEG OFFSET

                        18$:    INC     R1
                                INC     R2
                                CMP     R2,#261         ;SEE IF ALL NEG OFFSETS DONE
                                BEQ     TST14           ;;GO TO NEXT TST
                                JMP     1$              ;DO ANOTHER

;;*****************************************************************
;*TEST 14        TEST READ DATA AT ALL HEAD OFFSET POSITIONS
;*
;*      THIS TEST VERIFIES THAT THE HEAD OFFSET LOGIC IS OPERATIONAL BY
;*      WRITING ALL 1'S PATTERNS ON CYLINDER 0, HEAD 0. THEN
;*      PERFORMING READ DATA FROM CENTERLINE AND MOVING OUT + AND - OFFSET
;*      POSITIONS UNTIL A FAILURE OCCURES.  THE FAILING OFFSET POSITIONS
;*      ARE TYPED OUT.
;*      OFFSET CODES ARE ALSO VERIFIED BY READING MSG A, STAIUS 00 & 10.
;*
;*      ALL HEADS ARE TESTED AT CYLINDER 0
;*      IF THERE ARE NO FAILURES AT ALL, THIS INDICATES THAT
;*
;*              A. HEADS DID NOT MOVE AT ALL
;*      OR      B. THE COMBINATION OF DISC SURFACE, HEADS, R/W AMP
;*                 ARE EXCEPTIONALLY GOOD.
;*
;*      NOTE THAT THE OFFSET FAILURE IS NOT AN ERROR,
;*      BUT AN INDICATION OF SURFACE, HEAD & R/W ELECTRONICS QUALITY ONLY.
;*
;;*****************************************************************
TST14:  SCOPE
                                MOV     #1,$TIMES       ;;DO 1 ITERATION
                                MOV     #STACK,SP       ;RESTORE STK PTR

                                JSR     PC,SUBCLR
                                ERROR   24              ;CERR AFTER SCLR

                                CLR     SECTOR
                        11$:    MOV     SECTOR,RKDA(R5)
                                MOV     #DATA1,RKBA(R5) ;WRITE ALL 1'S
                                BIS     #BAI,RKCS2(R5)  ;BUS ADDR INCR INHIB
                                MOV     #-256.,RKWC(R5) ;SECTOR 0 ONLY
                                                        ;WILL DO IMPLIED SEEK TO CYL 0
                                                        ;WAS ON CYL 1 FROM LAST TEST

                                MOV     #<WRDATA>,HCS1
                                JSR     PC,DATCMD       ;DO DATA X FOR CMD & GET CONTR RDY
                                ERROR   11              ;NO RDY AFTER WRITE DATA CMD
```

```
4942  024014  004737  033722                    JSR     PC,GSTAT        ;GET FRESH STATUS
4943  024020  032737  100000  007354            BIT     #CERR,HCS1
4944  024026  001465                            BEQ     67$             ;BR IF NO ERRORS
4945
4946  024030  032737  000200  007370            BIT     #BSE,HER        ;SEE IF BAD SECTOR FLAG
4947  024036  001421                            BEQ     65$             ;BR IF NO
4948  024040  004737  035736                    JSR     PC,TRUERR       ;ELSE SEE IF SECTOR LISTED IN BSE TABLE
4949  024044  000455                            BR      66$             ;RETURN HERE IF NO
4950
4951  024046  005237  001402                    INC     SECTOR          ;RETURN HERE IF YES
4952  024052  023727  001402  000012            CMP     SECTOR,#10.     ;ARE 10 CONSEC. SECTORS BAD
4953  024060  001003                            BNE     64$             ;BR IF NO
4954  024062  104046                            ERROR   46              ;ABORTING TEST DETECTED 10 BAD SECTORS
4955  024064  000137  025462                    JMP     10$             ;BYPASS TEST
4956  024070  012765  100000  000000  64$:      MOV     #CCLR,RKCS1(R5) ;TRY ANOTHER SECTOR
4957  024076  000137  023750                    JMP     11$
4958  024102  104012                    65$:    ERROR   12              ;CERR WITH WRITE DATA CMD
4959
4960  024104  012737  010340  007444            MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
4961  024112  005037  007446                    CLR     E.B0            ;EXPECTED MSG B0
4962  024116  012737  001720  007450            MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
4963  024124  012737  000001  007452            MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
4964  024132  005037  007454                    CLR     E.A2            ;EXPECTED MSG A2
4965  024136  012737  000002  007456            MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
4966  024144  012737  000003  007462            MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
4967
4968  024152  004737  033100                    JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
4969  024156  000003                            .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
4970  024160  104052                            ERROR   52              ;MSG A0 ERROR AFTER WRITE DATA CMD
4971  024162  104023                            ERROR   23              ;MSH B0 ERROR
4972  024164  104053                            ERROR   53              ;MSG A1 ERROR
4973  024166  104025                            ERROR   25              ;MSG B1 ERROR
4974  024170  104401  045546                    TYPE    ,MSG26          ;ABORTING BAL OF TESTS
4975  024174  000137  031476                    JMP     $EOP
4976  024200  104063                    66$:    ERROR   63              ;BAD SECTOR NOT LISTED IN TABLE
4977  024202                            67$:
4978
4979  024202  012737  010340  007444            MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
4980  024210  005037  007446                    CLR     E.B0            ;EXPECTED MSG B0
4981  024214  012737  001720  007450            MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
4982  024222  012737  000001  007452            MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
4983  024230  005037  007454                    CLR     E.A2            ;EXPECTED MSG A2
4984  024234  012737  000002  007456            MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
4985  024242  012737  000003  007462            MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
4986
4987  024250  004737  033100                    JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
4988  024254  000003                            .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
4989  024256  104052                            ERROR   52              ;MSG A0 ERROR AFTER WRITE DATA CMD
4990  024260  104023                            ERROR   23              ;MSH B0 ERROR
4991  024262  104053                            ERROR   53              ;MSG A1 ERROR
4992  024264  104025                            ERROR   25              ;MSG B1 ERROR
4993  024266  012765  001500  000004            MOV     #DATA1,RKBA(R5)
4994  024274  052765  000020  000010            BIS     #BAI,RKCS2(R5)
4995  024302  012765  177400  000002            MOV     #-256.,RKWC(R5)
4996  024310  013765  001402  000006            MOV     SECTOR,RKDA(R5)
4997
```

```
 4998  024316  012737  000031  007354          MOV     #<WRTCHK>,HCS1
 4999  024324  004737  032320                  JSR     PC,DATCMD       ;DO DATA X FOR CMD & GET CONTR RDY
 5000  024330  104015                          ERROR   15              ;NO RDY AFTER WRITE CHECK CMD
 5001  024332  004737  033722                  JSR     PC,GSTAT        ;GET FRESH STATUS
 5002  024336  032737  100000  007354          BIT     #CERR,HCS1
 5003  024344  001453                          BEQ     69$
 5004  024346  032737  040000  007356          BIT     #WCE,HCS2       ;SEE IF WRITE CHECK ERROR
 5005  024354  001410                          BEQ     68$
 5006  024356  016537  000024  001452          MOV     RKDB(R5),WD1    ;ACTUAL WORD FOR PRINTOUT
 5007  024364  013737  001500  001454          MOV     DATA1,WD2               ;EXPECTED WORD FOR TYPEOUT
 5008  024372  104016                          ERROR   16              ;WCE AFTER WRITE CMD
 5009  024374  000437                          BR      69$
 5010
 5011  024376  104022                  68$:    ERROR   22              ;CERR AFTER WRITE CHECK CMD
 5012
 5013  024400  012737  010340  007444          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.AC  ;EXPECTED MSG A0
 5014  024406  005037  007446                  CLR     E.B0            ;EXPECTED MSG B0
 5015  024412  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
 5016  024420  012737  000001  007452          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 5017  024426  005037  007454                  CLR     E.A2            ;EXPECTED MSG A2
 5018  024432  012737  000002  007456          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 5019  024440  012737  000003  007462          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 5020
 5021  024446  004737  033100                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
 5022  024452  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 5023  024454  104057                          ERROR   57              ;MSG A0 ERROR AFTER WRITE CHECK CMD
 5024  024456  104031                          ERROR   31              ;MSH B0 ERROR
 5025  024460  104060                          ERROR   60              ;MSG A1 ERROR
 5026  024462  104032                          ERROR   32              ;MSG B1 ERROR
 5027  024464  104401  045546                  TYPE    ,MSG26          ;ABORTING BAL OF TESTS
 5028  024470  000137  031476                  JMP     $EOP
 5029
 5030  024474                          69$:
 5031
 5032  024474  012737  010340  007444          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
 5033  024502  005037  007446                  CLR     E.B0            ;EXPECTED MSG B0
 5034  024506  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
 5035  024514  012737  000001  007452          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 5036  024522  005037  007454                  CLR     E.A2            ;EXPECTED MSG A2
 5037  024526  012737  000002  007456          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 5038  024534  012737  000003  007462          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 5039
 5040  024542  004737  033100                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
 5041  024546  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 5042  024550  104057                          ERROR   57              ;MSG A0 ERROR AFTER WRITE CHECK CMD
 5043  024552  104031                          ERROR   31              ;MSH B0 ERROR
 5044  024554  104060                          ERROR   60              ;MSG A1 ERROR
 5045  024556  104032                          ERROR   32              ;MSG B1 ERROR
 5046
 5047  024560  104401  045033                  TYPE    ,MSG8           ;READ WITH OFFSET TEST
 5048  024564  005001                          CLR     R1              ;HEAD #
 5049
 5050  024566  012700  000001          9$:     MOV     #1,R0           ;INIT OFFSET COUNTER
 5051  024572  104401  045072                  TYPE    ,MSG9           ;HEAD #
 5052  024576  010146                          MOV     R1,-(SP)        ;;SAVE R1 FOR TYPEOUT
 5053                                                                  ;;TYPE HEAD #
```

```
5054  024600  104403                    TYPOS                   ;;GO TYPE--OCTAL ASCII
5055  024602     00'                    .BYTE   1               ;;TYPE 1 DIGIT(S)
5056  024603     000                    .BYTE   0               ;;SUPPRESS LEADING ZEROS
5057  024604  104401  00 205            TYPE    ,$CRLF
5058
5059  024610  005037  001456      1$:   CLR     OFFERR          ;WRITE CHECK ERROR FLAG
5060
5061  024614  004737  034252            JSR     PC,SUBCLR
5062  024620  104024                    ERROR   24              ;CERR AFTER SCLR
5063
5064  024622  010065  000016            MOV     R0,RKASOF(R5)   ;OFFSET VALUE
5065  024626  000301                    SWAB    R1
5066  024630  010165  000006            MOV     R1,RKDA(R5)     ;HEAD NO.
5067  024634  000301                    SWAB    R1
5068
5069  024636  012737  024724  001176    MOV     #70$,$ESCAPE
5070  024644  012737  000015  007354    MOV     #OFFSET,HCS1
5071  024652  004737  032262            JSR     PC,DOCMD        ;DO RECAL CMD & GET CONTR RDY
5072  024656  104033                    ERROR   33              ;NO RDY AFTER OFFSET CMD
5073
5074  024660  012737  032140  007444    MOV     #<D.PIP!D.SPIN!D.OFF!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
5075  024666  005037  007446            CLR     E.B0
5076  024672  012737  001720  007450    MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
5077  024700  012737  000001  007452    MOV     #1,E.B1
5078
5079  024706  004737  033100            JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
5080  024712  000000                    .WORD   0!0!0           ;& MSGS SPECIFIED HERE
5081  024714  104035                    ERROR   35              ;MSG A0 ERROR DURING OFFSET CMD
5082  024716  104061                    ERROR   61              ;MSH B0 ERROR
5083  024720  104036                    ERROR   36              ;MSG A1 ERROR
5084  024722  104062                    ERROR   62              ;MSG B1 ERROR
5085
5086  024724  005037  001176      70$:  CLR     $ESCAPE
5087  024730  013737  001416  007412    MOV     T5000,TEMP1     ;SETUP TIMEOUT
5088  024736  004737  032766            JSR     PC,FATT2        ;FIND ATTN
5089  024742  104034                    ERROR   34              ;NO ATTN AFTER OFFSET CMD
5090
5091
5092  024744  012737  052340  007444    MOV     #<D.DSC!D.OFF!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
5093  024752  005037  007446            CLR     E.B0            ;EXPECTED MSG B0
5094  024756  012737  001720  007450    MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5095  024764  012737  000001  007452    MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5096  024772  005037  007454            CLR     E.A2            ;EXPECTED MSG A2
5097  024776  012737  000002  007456    MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5098  025004  012737  000003  007462    MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5099
5100  025012  004737  033100            JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
5101  025016  000003                    .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5102  025020  104260                    ERROR   260             ;MSG A0 ERROR AFTER OFFSET CMD
5103  025022  104261                    ERROR   261             ;MSH B0 ERROR
5104  025024  104037                    ERROR   37              ;MSG A1 ERROR
5105  025026  104040                    ERROR   40              ;MSG B1 ERROR
5106
5107
5108  025030  012765  100000  000000    MOV     #CCLR,RKCS1(R5)
5109  025036  013765  001222  000010    MOV     $UNIT,RKCS2(R5) ;DRIVE#
```

```
5110  025044  012737  000005  007354          MOV     #CLEAR,HCS1
5111  025052  004737  032262                  JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
5112  025056  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
5113  025060  004737  032640                  JSR     PC,TSTATN       ;TEST FOR ATTN
5114  025064  000401                          BR      71$
5115  025066  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5116  025070                          71$:
5117
5118  025070  012765  001500  000004          MOV     #DATA1,RKBA(R5)
5119  025076  052765  000020  000010          BIS     #BAI,RKCS2(R5)
5120  025104  012765  177400  000002          MOV     #-256.,RKWC(R5)
5121  025112  013765  001402  000006          MOV     SECTOR,RKDA(R5)
5122  025120  012737  000031  007354          MOV     #WRTCHK,HCS1
5123  025126  004737  032320                  JSR     PC,DATCMD       ;DO WRITE CHECK CMD. & GET CONTR RDY.
5124  025132  104015                          ERROR   15              ;NO RDY AFTER WRITE CHECK CMD
5125  025134  004737  033722                  JSR     PC,GSTAT        ;GET FRESH STATUS
5126  025140  032737  040000  007356          BIT     #WCE,HCS2
5127  025146  001421                          BEQ     2$
5128
5129  025150  016537  000024  001452          MOV     RKDB(R5),WD1    ;GET MISCOMPARED WORD
5130  025156  005237  001456                  INC     OFFERR          ;BAD WRITE CHK ERROR=SET ERR FLG.
5131
5132  025162  005737  001456                  TST     OFFERR
5133  025166  001411                          BEQ     2$
5134  025170  104401  046012                  TYPE    ,MSG39          ;WRITE CHECK FAILURE AT OFFSET
5135  025174  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
5136                                                                  ;;TYPE OFFSET VALUE
5137  025176  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
5138  025200     006                          .BYTE   6               ;;TYPE 6 DIGITS
5139  025201     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
5140  025202  104401  001205                  TYPE    ,$CRLF
5141  025206  104401  001205                  TYPE    ,$CRLF
5142
5143  025212  032700  000200          2$:     BIT     #BIT7,R0        ;SEE IF OFFSET IS + OR -
5144  025216  001023                          BNE     5$              ;BR IF - OFFSET
5145
5146  025220  020027  000060                  CMP     R0,#60
5147  025224  001412                          BEQ     4$
5148  025226  005737  001456                  TST     OFFERR
5149  025232  001404                          BEQ     3$
5150  025234  012700  000200          8$:     MOV     #200,R0         ;SETUP FOR NEG OFFSET
5151  025240  000137  024610                  JMP     1$
5152
5153  025244  005200                  3$:     INC     R0
5154  025246  000137  024610                  JMP     1$
5155
5156  025252  005737  001456          4$:     TST     OFFERR
5157  025256  001366                          BNE     8$              ;DO NEG OFFSETS
5158  025260  104401  045654                  TYPE    ,MSG37          ;NO WRITE CHECK ERROR AT MAX POS OFFSET
5159                                                                  ;NOTE! EITHER HEADS DID NOT MOVE
5160                                                                  ;OR READ/WRITE AMP IS EXCEPTIONALLY GOOD.
5161  025264  000763                          BR      8$              ;DO NEG OFFSETS
5162
5163  025266  020027  000260          5$:     CMP     R0,#260
5164  025272  001404                          BEQ     6$
5165  025274  005737  001456                  TST     OFFERR
```

```
5166  025300  001072                        BNE     TST15           ;;GO TO NEXT TST
5167  025302  000760                        BR      3$
5168
5169  025304  005737  001456          6$:   TST     OFFERR
5170  025310  001002                        BNE     7$
5171  025312  104401  045732                TYPE    ,MSG38          ;NO WRITE CHECK ERROR AT MAX NEG OFFSET
5172                                                                  ;NOTE! EITHER HEADS DID NOT MOVE
5173                                                                  ;OR READ/WRITE AMP IS EXECEPTIONALLY GOOD.
5174  025316                          7$:
5175
5176  025316  012765  100000  000000        MOV     #CCLR,RKCS1(R5)
5177  025324  013765  001222  000010        MOV     $UNIT,RKCS2(R5)
5178  025332  012737  000013  007354        MOV     #RECAL,HCS1
5179  025340  004737  032262                JSR     PC,DOCMD        ;DO RECAL CMD & GET CONTR RDY
5180  025344  104124                        ERROR   124             ;RDY NOT SET AFTER RECAL CMD
5181
5182  025346  012765  000001  000026        MOV     #1,RKMR1(R5)    ;SELECT WORD 1
5183  025354  004737  033722                JSR     PC,GSTAT
5184  025360  032737  020000  007402        BIT     #D.RTZ,HMR2
5185  025366  001001                        BNE     72$
5186  025370  104244                        ERROR   244             ;RTZ NOT SET DURING RECAL CMD
5187  025372  013737  001406  007414  72$:  MOV     T10,TEMP2       ;SETUP TIMEOUT
5188  025400  004737  032672                JSR     PC,FATT1        ;FIND ATTN
5189  025404  104055                        ERROR   55              ;NO ATTN AFTER RECAL CMD
5190
5191  025406  012765  100000  000000        MOV     #CCLR,RKCS1(R5)
5192  025414  013765  001222  000010        MOV     $UNIT,RKCS2(R5) ;DRIVE#
5193  025422  012737  000005  007354        MOV     #CLEAR,HCS1
5194  025430  004737  032262                JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
5195  025434  104151                        ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
5196  025436  004737  032640                JSR     PC,TSTATN       ;TEST FOR ATTN
5197  025442  000401                        BR      73$
5198  025444  104154                        ERRCR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5199  025446                          73$:
5200
5201
5202  025446  005201                        INC     R1              ;HEAD CTR
5203  025450  020127  000003                CMP     R1,#3           ;SEE IF ALL HEADS DONE
5204  025454  001402                        BEQ     10$             ;BR IF YES
5205  025456  000137  024566                JMP     9$              ;ELSE REPEAT ALL FOR NEXT HEAD
5206  025462  104401  045172          10$:  TYPE    ,MSG12          ;OFFSET FAILURES ARE NOT ERRORS
5207                                        ;;*****************************************************************
5208                                        ;*TEST 15       WRITE WITH HEADS OFFSET
5209                                        ;*
5210                                        ;*      THIS TEST VERIFIES THAT WHEN ATTEMPTING TO
5211                                        ;*      WRITE WITH HEADS OFFSET THAT THE OFFSET WILL CLEAR
5212                                        ;*      & THE DRIVE WILL WRITE
5213                                        ;*      SINCE THE WRITE COMMAND HAS AN IMPLIED RTC.
5214                                        ;*      THIS TEST IS PERFORMED FOR MAX POS & NEG OFFSETS ONLY
5215                                        ;*
5216                                        ;;*****************************************************************
5217  025466  000004                  TST15:  SCOPE
5218  025470  012737  000001  001174        MOV     #1,$TIMES       ;;DO 1 ITERATION
5219  025476  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR
5220
5221  025502  012700  000260                MOV     #260,R0         ;MAX NEG OFFSET
```

```
5222
5223  025506  004737  034252          1$:     JSR     PC,SUBCLR
5224  025512  104024                          ERROR   24              ;CERR AFTER SCLR
5225
5226  025514  010065  000016                  MOV     R0,RKASOF(R5)   ;SET OFFSET
5227
5228  025520  012737  025606  001176          MOV     #64$,$ESCAPE
5229  025526  012737  000015  007354          MOV     #OFFSET,HCS1
5230  025534  004737  032262                  JSR     PC,DOCMD        ;DO RECAL CMD & GET CONTR RDY
5231  025540  104033                          ERROR   33              ;NO RDY AFTER OFFSET CMD
5232
5233  025542  012737  032140  007444          MOV     #<D.PIP!D.SPIN!D.OFF!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
5234  025550  005037  007446                  CLR     E.B0
5235  025554  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
5236  025562  012737  000001  007452          MOV     #1,E.B1
5237
5238  025570  004737  033100                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
5239  025574  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
5240  025576  104035                          ERROR   35              ;MSG A0 ERROR DURING OFFSET CMD
5241  025600  104061                          ERROR   61              ;MSH B0 ERROR
5242  025602  104036                          ERROR   36              ;MSG A1 ERROR
5243  025604  104062                          ERROR   62              ;MSG B1 ERROR
5244
5245  025606  005037  001176          64$:    CLR     $ESCAPE
5246  025612  013737  001416  007412          MOV     T5000,TEMP1     ;SETUP TIMEOUT
5247  025620  004737  032766                  JSR     PC,FATT2        ;FIND ATTN
5248  025624  104034                          ERROR   34              ;NO ATTN AFTER OFFSET CMD
5249
5250
5251  025626  012737  052340  007444          MOV     #<D.DSC!D.OFF!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
5252  025634  005037  007446                  CLR     E.B0            ;EXPECTED MSG B0
5253  025640  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5254  025646  012737  000001  007452          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5255  025654  005037  007454                  CLR     E.A2            ;EXPECTED MSG A2
5256  025660  012737  000002  007456          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5257  025666  012737  000003  007462          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5258
5259  025674  004737  033100                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
5260  025700  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5261  025702  104260                          ERROR   260             ;MSG A0 ERROR AFTER OFFSET CMD
5262  025704  104261                          ERROR   261             ;MSH B0 ERROR
5263  025706  104037                          ERROR   37              ;MSG A1 ERROR
5264  025710  104040                          ERROR   40              ;MSG B1 ERROR
5265
5266
5267  025712  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
5268  025720  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
5269  025726  012737  000005  007354          MOV     #CLEAR,HCS1
5270  025734  004737  032262                  JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
5271  025740  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
5272  025742  004737  032640                  JSR     PC,TSTATN       ;TEST FOR ATTN
5273  025746  000401                          BR      65$
5274  025750  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5275  025752                          65$:
5276
5277
```

```
5278  025752  005037  001402                  CLR     SECTOR
5279  025756  013765  001402  000006  4$:     MOV     SECTOR,RKDA(R5)
5280  025764  012765  001474  000004          MOV     #DATA0,RKBA(R5) ;WRITE ALL 0'S
5281  025772  052765  000020  000010          BIS     #BAI,RKCS2(R5)  ;BUS ADDR INCR INH
5282  026000  012765  177400  000002          MOV     #-256.,RKWC(R5) ;FULL SECTOR
5283
5284  026006  012737  000023  007354          MOV     #<WRDATA>,HCS1
5285  026014  004737  032320                  JSR     PC,DATCMD       ;DO DATA X FOR CMD & GET CONTR RDY
5286  026020  104011                          ERROR   11              ;NO RDY AFTER WRITE DATA CMD
5287  026022  004737  033722                  JSR     PC,GSTAT        ;GET FRESH STATUS
5288  026026  032737  100000  007354          BIT     #CERR,HCS1
5289  026034  001465                          BEQ     69$             ;BR IF NO ERRORS
5290
5291  026036  032737  000200  007370          BIT     #BSE,HER        ;SEE IF BAD SECTOR FLAG
5292  026044  001421                          BEQ     67$             ;BR IF NO
5293  026046  004737  035736                  JSR     PC,TRUERR       ;ELSE SEE IF SECTOR LISTED IN BSE TABLE
5294  026052  000455                          BR      68$             ;RETURN HERE IF NO
5295
5296  026054  005237  001402                  INC     SECTOR          ;RETURN HERE IF YES
5297  026060  023727  001402  000012          CMP     SECTOR,#10.     ;ARE 10 CONSEC. SECTORS BAD
5298  026066  001003                          BNE     66$             ;BR IF NO
5299  026070  104046                          ERROR   46              ;ABORTING TEST DETECTED 10 BAD SECTORS
5300  026072  000137  027002                  JMP     3$              ;BYPASS TEST
5301  026076  012765  100000  000000  66$:     MOV     #CCLR,RKCS1(R5) ;TRY ANOTHER SECTOR
5302  026104  000137  025756                  JMP     4$
5303  026110  104012                  67$:    ERROR   12              ;CERR WITH WRITE DATA CMD
5304
5305  026112  012737  010340  007444          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
5306  026120  005037  007446                  CLR     E.B0            ;EXPECTED MSG B0
5307  026124  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5308  026132  012737  000001  007452          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5309  026140  005037  007454                  CLR     E.A2            ;EXPECTED MSG A2
5310  026144  012737  000002  007456          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5311  026152  012737  000003  007462          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5312
5313  026160  004737  033100                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
5314  026164  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5315  026166  104052                          ERROR   52              ;MSG A0 ERROR AFTER WRITE DATA CMD
5316  026170  104023                          ERROR   23              ;MSH B0 ERROR
5317  026172  104053                          ERROR   53              ;MSG A1 ERROR
5318  026174  104025                          ERROR   25              ;MSG B1 ERROR
5319  026176  104401  045546                  TYPE    ,MSG26          ;ABORTING BAL OF TESTS
5320  026202  000137  031476                  JMP     $EOP
5321  026206  104063                  68$:    ERROR   63              ;BAD SECTOR NOT LISTED IN TABLE
5322  026210                           69$:
5323
5324  026210  012737  010340  007444          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
5325  026216  005037  007446                  CLR     E.B0            ;EXPECTED MSG B0
5326  026222  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5327  026230  012737  000001  007452          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5328  026236  005037  007454                  CLR     E.A2            ;EXPECTED MSG A2
5329  026242  012737  000002  007456          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5330  026250  012737  000003  007462          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5331
5332  026256  004737  033100                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
5333  026262  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
```

```
5334  026264  104052                        ERROR   52              ;MSG A0 ERROR AFTER WRITE DATA CMD
5335  026266  104023                        ERROR   23              ;MSH B0 ERROR
5336  026270  104053                        ERROR   53              ;MSG A1 ERROR
5337  026272  104025                        ERROR   25              ;MSG B1 ERROR
5338
5339  026274  012765  000002  000026        MOV     #2,RKMR1(R5)    ;SELECT WORD 2
5340  026302  004737  033722                JSR     PC,GSTAT
5341  026306  005737  001356                TST     CYLDIF          ;SEE IF MSG A2=0
5342  026312  001401                        BEQ     70$             ;BR IF YES
5343  026314  104104                        ERROR   104             ;MSG A2 NOT CLEARED AFTER WRITE CMD WITH OFFSET
5344  026316  005737  001360        70$:    TST     CYLADD          ;SEE IF MSG B2=0
5345  026322  001401                        BEQ     71$             ;BR IF YES
5346  026324  104105                        ERROR   105             ;MSG B2 NOT CLEARED AFTER WRITE CMD WITH OFFSET
5347  026326                        71$:
5348
5349  026326  104415                        SCOP1
5350  026330  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR
5351
5352  026334  004737  034252                JSR     PC,SUBCLR
5353  026340  104024                        ERROR   24              ;CERR AFTER SCLR
5354
5355
5356  026342  012765  001474  000004        MOV     #DATA0,RKBA(R5)
5357  026350  052765  000020  000010        BIS     #BAI,RKCS2(R5)
5358  026356  012765  177400  000002        MOV     #-256.,RKWC(R5)
5359  026364  013765  001402  000006        MOV     SECTOR,RKDA(R5)
5360
5361  026372  012737  000031  007354        MOV     #<WRTCHK>,HCS1
5362  026400  004737  032320                JSR     PC,DATCMD       ;DO DATA X FOR CMD & GET CONTR RDY
5363  026404  104015                        ERROR   15              ;NO RDY AFTER WRITE CHECK CMD
5364  026406  004737  033722                JSR     PC,GSTAT        ;GET FRESH STATUS
5365  026412  032737  100000  007354        BIT     #CERR,HCS1
5366  026420  001453                        BEQ     73$
5367  026422  032737  040000  007356        BIT     #WCE,HCS2       ;SEE IF WRITE CHECK ERROR
5368  026430  001410                        BEQ     72$
5369  026432  016537  000024  001452        MOV     RKDB(R5),WD1    ;ACTUAL WORD FOR PRINTOUT
5370  026440  013737  001474  001454        MOV     DATA0,WD2               ;EXPECTED WORD FOR TYPEOUT
5371  026446  104016                        ERROR   16              ;WCE AFTER WRITE CMD
5372  026450  000437                        BR      73$
5373
5374  026452  104022                72$:    ERROR   22              ;CERR AFTER WRITE CHECK CMD
5375
5376  026454  012737  010340  007444        MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
5377  026462  005037  007446                CLR     E.B0            ;EXPECTED MSG B0
5378  026466  012737  001720  007450        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5379  026474  012737  000001  007452        MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5380  026502  005037  007454                CLR     E.A2            ;EXPECTED MSG A2
5381  026506  012737  000002  007456        MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5382  026514  012737  000003  007462        MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5383
5384  026522  004737  033100                JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
5385  026526  000003                        .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5386  026530  104057                        ERROR   57              ;MSG A0 ERROR AFTER WRITE CHECK CMD
5387  026532  104031                        ERROR   31              ;MSH B0 ERROR
5388  026534  104060                        ERROR   60              ;MSG A1 ERROR
5389  026536  104032                        ERROR   32              ;MSG B1 ERROR
```

```
 5390  026540  104401  045546                TYPE    ,MSG26          ;ABORTING BAL OF TESTS
 5391  026544  000137  031476                JMP     $EOP
 5392
 5393  026550                        73$:
 5394
 5395  026550  012737  010340  007444        MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
 5396  026556  005037  007446                CLR     E.B0            ;EXPECTED MSG B0
 5397  026562  012737  001720  007450        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
 5398  026570  012737  000001  007452        MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 5399  026576  005037  007454                CLR     E.A2            ;EXPECTED MSG A2
 5400  026602  012737  000002  007456        MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 5401  026610  012737  000003  007462        MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 5402
 5403  026616  004737  033100                JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
 5404  026622  000003                        .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 5405  026624  104057                        ERROR   57              ;MSG A0 ERROR AFTER WRITE CHECK CMD
 5406  026626  104031                        ERROR   31              ;MSH B0 ERROR
 5407  026630  104060                        ERROR   60              ;MSG A1 ERROR
 5408  026632  104032                        ERROR   32              ;MSG B1 ERROR
 5409
 5410  026634  020027  000260                CMP     R0,#260
 5411  026640  001004                        BNE     2$              ;BR IF JUST DID POS OFFSET
 5412  026642  012700  000060                MOV     #60,R0          ;ELSE SETUP FOR POS OFFSET
 5413  026646  000137  025506                JMP     1$
 5414
 5415  026652                        2$:
 5416
 5417  026652  012765  100000  000000        MOV     #CCLR,RKCS1(R5)
 5418  026660  013765  001222  000010        MOV     $UNIT,RKCS2(R5)
 5419  026666  012737  000013  007354        MOV     #RECAL,HCS1
 5420  026674  004737  032262                JSR     PC,DOCMD        ;DO RECAL CMD & GET CONTR RDY
 5421  026700  104124                        ERROR   124             ;RDY NOT SET AFTER RECAL CMD
 5422
 5423  026702  012765  000001  000026        MOV     #1,RKMR1(R5)    ;SELECT WORD 1
 5424  026710  004737  033722                JSR     PC,GSTAT
 5425  026714  032737  020000  007402        BIT     #D.RTZ,HMR2
 5426  026722  001001                        BNE     74$
 5427  026724  104244                        ERROR   244             ;RTZ NOT SET DURING RECAL CMD
 5428  026726  013737  001406  007414  74$:  MOV     T10,TEMP2       ;SETUP TIMEOUT
 5429  026734  004737  032672                JSR     PC,FATT1        ;FIND ATTN
 5430  026740  104055                        ERROR   55              ;NO ATTN AFTER RECAL CMD
 5431
 5432  026742  012765  100000  000000        MOV     #CCLR,RKCS1(R5)
 5433  026750  013765  001222  000010        MOV     $UNIT,RKCS2(R5) ;DRIVE#
 5434  026756  012737  000005  007354        MOV     #CLEAR,HCS1
 5435  026764  004737  032262                JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
 5436  026770  104151                        ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
 5437  026772  004737  032640                JSR     PC,TSTATN       ;TEST FOR ATTN
 5438  026776  000401                        BR      75$
 5439  027000  104154                        ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
 5440  027002                        75$:
 5441
 5442
 5443  027002                        3$:
 5444
```

```
                                ;;*****************************************************************
                                ;*TEST 16       TEST CURRENT CROSS-OVER CYLINDERS
                                ;*
                                ;*      THIS TEST VERIFIES THAT THE DRIVE CAN WRITE & READ OFF
                                ;*      CURRENT CHANGE CYLINDERS X & Y IN THE FOLLOWING WAY:
                                ;*
                                ;*      SPIRAL WRITING IS PERFORMED FROM CYLINDER X TO CYLINDER Y
                                ;*      WITH A DATA PATTERN FILLING THE ENTIRE 2 CYLINDERS.
                                ;*
                                ;*      A WRITE CHECK IS THEN PREFORMED TO VERIFY DATA WAS PROPERLY WRITTEN.
                                ;*      THIS TEST IS PERFORMED FOR ALL 3 HEADS.
                                ;*
                                ;*      CYLINDER X:  63  127  191  255  319  383  RK06
                                ;*      CYLINDER Y:  64  128  192  256  320  384  RK06
                                ;*
                                ;*      CYLINDER X:  127 255  383  511  639  767  RK07
                                ;*      CYLINDER Y:  128 256  384  512  640  768  RK07
                                ;*
                                ;*      ALL ABOVE CYLINDERS IN DECIMAL
                                ;*
                                ;*****************************************************************
027002  000004                  TST16:  SCOPE
027004  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
027012  012706  001100                  MOV     #STACK,SP

027016  005737  001170                  TST     $TMP4           ;SEE IF RK07
027022  001403                          BEQ     5$              ;BR IF NO
027024  012700  001436                  MOV     #CYL7,R0        ;ELSE LOAD UP RK07 TABLE
027030  000402                          BR      1$
027032  012700  001422          5$:     MOV     #CYL,R0         ;RK06 CYL ADDR TABLE

027036  004737  034252          1$:     JSR     PC,SUBCLR
027042  104024                          ERROR   24              ;CERR AFTER SCLR

027044  011065  000020                  MOV     (R0),RKDC(R5)   ;CYL #
027050  012765  001502  000004          MOV     #DPAT1,RKBA(R5) ;DATA PATTERN
027056  052765  000020  000010          BIS     #BAI,RKCS2(R5)  ;BUSS ADDR INCREMENT INHIBIT
027064  012765  076000  000002          MOV     #-6*22.*256.,RKWC(R5) ;WORD COUNT TO SPIRAL & FILL 2 CYLINDERS

027072  012737  000023  007354          MOV     #<WRDATA>,HCS1
027100  004737  032320                  JSR     PC,DATCMD       ;DO DATA X FOR CMD & GET CONTR RDY
027104  104011                          ERROR   11              ;NO RDY AFTER WRITE DATA CMD
027106  004737  033722                  JSR     PC,GSTAT        ;GET FRESH STATUS
027112  032737  100000  007354          BIT     #CERR,HCS1
027120  001451                          BEQ     67$             ;BR IF NO ERRORS

027122  032737  000200  007370          BIT     #BSE,HER        ;SEE IF BAD SECTOR FLAG
027130  001405                          BEQ     65$             ;BR IF NO
027132  004737  035736                  JSR     PC,TRUERR       ;ELSE SEE IF SECTOR LISTED IN BSE TABLE
027136  000441                          BR      66$             ;RETURN HERE IF NO
027140  000137  027620                  JMP     3$              ;RET HERE IF YES
027144  104012                  65$:    ERROR   12              ;CERR WITH WRITE DATA CMD

027146  012737  010340  007444          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
027154  005037  007446                  CLR     E.B0            ;EXPECTED MSG B0
```

```
5501  027160  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5502  027166  012737  000001  007452          MOV     #1,E.B1             ;MSG ID FOR EXPECTED MSG B1
5503  027174  005037  007454                  CLR     E.A2                ;EXPECTED MSG A2
5504  027200  012737  000002  007456          MOV     #2,E.B2             ;MSG ID FOR EXPECTED MSG B2
5505  027206  012737  000003  007462          MOV     #3,E.B3             ;MSG ID FOR EXPECTED MSG B3
5506
5507  027214  004737  033100                  JSR     PC,CHKMSG           ;CHECK MSGS A0, B0, A1, B1
5508  027220  000003                          .WORD   T.A2!T.B2!0                 ;& MSGS SPECIFIED HERE
5509  027222  104052                          ERROR   52                  ;MSG A0 ERROR AFTER WRITE DATA CMD
5510  027224  104023                          ERROR   23                  ;MSH B0 ERROR
5511  027226  104053                          ERROR   53                  ;MSG A1 ERROR
5512  027230  104025                          ERROR   25                  ;MSG B1 ERROR
5513  027232  104401  045546                  TYPE    ,MSG26              ;ABORTING BAL OF TESTS
5514  027236  000137  031476                  JMP     $EOP
5515  027242  104063                  66$:    ERROR   63                  ;BAD SECTOR NOT LISTED IN TABLE
5516  027244                           67$:
5517
5518  027244  012737  010340  007444          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
5519  027252  005037  007446                  CLR     E.B0                ;EXPECTED MSG B0
5520  027256  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5521  027264  012737  000001  007452          MOV     #1,E.B1             ;MSG ID FOR EXPECTED MSG B1
5522  027272  005037  007454                  CLR     E.A2                ;EXPECTED MSG A2
5523  027276  012737  000002  007456          MOV     #2,E.B2             ;MSG ID FOR EXPECTED MSG B2
5524  027304  012737  000003  007462          MOV     #3,E.B3             ;MSG ID FOR EXPECTED MSG B3
5525
5526  027312  004737  033100                  JSR     PC,CHKMSG           ;CHECK MSGS A0, B0, A1, B1
5527  027316  000003                          .WORD   T.A2!T.B2!0                 ;& MSGS SPECIFIED HERE
5528  027320  104052                          ERROR   52                  ;MSG A0 ERROR AFTER WRITE DATA CMD
5529  027322  104023                          ERROR   23                  ;MSH B0 ERROR
5530  027324  104053                          ERROR   53                  ;MSG A1 ERROR
5531  027326  104025                          ERROR   25                  ;MSG B1 ERROR
5532
5533  027330  011065  000020                  MOV     (R0),RKDC(R5)
5534  027334  012765  001502  000004          MOV     #DPAT1,RKBA(R5)
5535  027342  052765  000020  000010          BIS     #BAI,RKCS2(R5)
5536  027350  012765  076000  000002          MOV     #-6*22.*256.,RKWC(R5)
5537
5538  027356  012737  000031  007354          MOV     #<WRTCHK>,HCS1
5539  027364  004737  032320                  JSR     PC,DATCMD           ;DO DATA X FOR CMD & GET CONTR RDY
5540  027370  104015                          ERROR   15                  ;NO RDY AFTER WRITE CHECK CMD
5541  027372  004737  033722                  JSR     PC,GSTAT            ;GET FRESH STATUS
5542  027376  032737  100000  007354          BIT     #CERR,HCS1
5543  027404  001453                          BEQ     69$
5544  027406  032737  040000  007356          BIT     #WCE,HCS2           ;SEE IF WRITE CHECK ERROR
5545  027414  001410                          BEQ     68$
5546  027416  016537  000024  001452          MOV     RKDB(R5),WD1        ;ACTUAL WORD FOR PRINTOUT
5547  027424  013737  001502  001454          MOV     DPAT1,WD2                   ;EXPECTED WORD FOR TYPEOUT
5548  027432  104016                          ERROR   16                  ;WCE AFTER WRITE CMD
5549  027434  000437                          BR      69$
5550
5551  027436  104022                  68$:    ERROR   22                  ;CERR AFTER WRITE CHECK CMD
5552
5553  027440  012737  010340  007444          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
5554  027446  005037  007446                  CLR     E.B0                ;EXPECTED MSG B0
5555  027452  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5556  027460  012737  000001  007452          MOV     #1,E.B1             ;MSG ID FOR EXPECTED MSG B1
```

```
5557  027466  005037  007454                  CLR     E.A2            ;EXPECTED MSG A2
5558  027472  012737  000002  007456          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5559  027500  012737  000003  007462          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5560
5561  027506  004737  033100                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
5562  027512  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5563  027514  104057                          ERROR   57              ;MSG A0 ERROR AFTER WRITE CHECK CMD
5564  027516  104031                          ERROR   31              ;MSH B0 ERROR
5565  027520  104060                          ERROR   60              ;MSG A1 ERROR
5566  027522  104032                          ERROR   32              ;MSG B1 ERROR
5567  027524  104401  045546                  TYPE    ,MSG26          ;ABORTING BAL OF TESTS
5568  027530  000137  031476                  JMP     $EOP
5569
5570  027534                          69$:
5571
5572  027534  012737  010340  007444          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
5573  027542  005037  007446                  CLR     E.B0            ;EXPECTED MSG B0
5574  027546  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5575  027554  012737  000001  007452          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5576  027562  005037  007454                  CLR     E.A2            ;EXPECTED MSG A2
5577  027566  012737  000002  007456          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5578  027574  012737  000003  007462          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5579
5580  027602  004737  033100                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
5581  027606  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
5582  027610  104057                          ERROR   57              ;MSG A0 ERROR AFTER WRITE CHECK CMD
5583  027612  104031                          ERROR   31              ;MSH B0 ERROR
5584  027614  104060                          ERROR   60              ;MSG A1 ERROR
5585  027616  104032                          ERROR   32              ;MSG B1 ERROR
5586  027620  022027  000577          3$:     CMP     (R0)+,#383.     ;ALL CYLINDERS DONE?
5587  027624  001402                          BEQ     4$              ;BR IF YES
5588  027626  000137  027036                  JMP     1$              ;ELSE REPEAT
5589
5590  027632                          4$:
5591
5592  027632  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
5593  027640  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
5594  027646  012737  000013  007354          MOV     #RECAL,HCS1
5595  027654  004737  032262                  JSR     PC,DOCMD        ;DO RECAL CMD & GET CONTR RDY
5596  027660  104124                          ERROR   124             ;RDY NOT SET AFTER RECAL CMD
5597
5598  027662  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
5599  027670  004737  033722                  JSR     PC,GSTAT
5600  027674  032737  020000  007402          BIT     #D.RTZ,HMR2
5601  027702  001001                          BNE     70$
5602  027704  104244                          ERROR   244             ;RTZ NOT SET DURING RECAL CMD
5603  027706  013737  001406  007414  70$:    MOV     T10,TEMP2       ;SETUP TIMEOUT
5604  027714  004737  032672                  JSR     PC,FATT1        ;FIND ATTN
5605  027720  104055                          ERROR   55              ;NO ATTN AFTER RECAL CMD
5606
5607  027722  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
5608  027730  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
5609  027736  012737  000005  007354          MOV     #CLEAR,HCS1
5610  027744  004737  032262                  JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
5611  027750  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
5612  027752  004737  032640                  JSR     PC,TSTATN       ;TEST FOR ATTN
```

```
5613  027756  000401                          BR      71$
5614  027760  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5615  027762                          71$:
5616
5617
5618  027762                          2$:
5619
5620                                  ;;*****************************************************************
5621                                  ;*TEST 17       TEST HEAD SWITCHING TIME
5622                                  ;*
5623                                  ;*      TESTS THE ABILITY TO SWITCH HEADS IN LESS THEN 10MS WHEN HEADS SPIRAL.
5624                                  ;*
5625                                  ;*      1. SECTOR 23(8) IS FIRST LOCATED AND A WRITE DATA COMMAND OF 512 WORDS
5626                                  ;*         TO SECTOR 25(8) IS ISSUED.
5627                                  ;*      2. THE PROGRAM NOW KNOWS THAT THE DRIVE WILL NOT HAVE TO TRAVEL
5628                                  ;*         A FULL REVOLUTION BEFORE FINDING SECTOR 25(8).
5629                                  ;*      3. SINCE EACH SECTOR TAKES APPROX. 1.2MS, THE TIME BETWEEN
5630                                  ;*         THE START OF THE WRITE COMMAND (FROM SECTOR 25(8), HEAD 0; TO
5631                                  ;*         SECTOR 0, HEAD 1) AND CONTROLLER READY SHOULD BE APPROX 6MS
5632                                  ;*
5633                                  ;*      THE ABOVE IS REPEATED FOR HEAD SWITCHING BETWEEN 1 TO 2
5634                                  ;*
5635                                  ;*      THIS TEST IS BYPASSED IF NEITHER L OR P CLOCK IS PRESENT
5636                                  ;*
5637                                  ;;*****************************************************************
5638  027762  000004                  TST17:  SCOPE
5639  027764  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
5640  027772  012706  001100                  MOV     #STACK,SP
5641
5642  027776  005737  007522                  TST     DOTIM           ;BYPASS THIS TEST IF
```

```
  5643  030002  C01001                    BNE     1$              ;NEITHER L OR P CLOCK PRESENT
  5644  030004  000526                    BR      TST20           ;;GO TO NEXT TEST
```

```
5645
5646  030006  012737  000025  007422  1$:     MOV     #25,TEMP5       ;HEAD 0, SECTOR 21 TO BE PUT IN RKDA
5647  030014  004737  034252          2$:     JSR     PC,SUBCLR
5648  030020  104024                          ERROR   24              ;CERR AFTER SCLR
5649
5650  030022  004737  035036                  JSR     PC,FSEC23       ;FIND SECTOR 23
5651  030026  104106                          ERROR   106             ;CANNOT FIND SECTOR 23
5652  030030  000514                          BR      TST20           ;;GO TO NEXT TEST
5653
5654  030032  012765  001476  000004          MOV     #DATA01,RKBA(R5) ;DATA 0101
5655  030040  052765  000020  000010          BIS     #BAI,RKCS2(R5)  ;BUSS ADDR INCREMENT INHIBIT
5656  030046  012765  177000  000002          MOV     #-512.,RKWC(R5) ;WORD COUNT
5657  030054  013765  007422  000006          MOV     TEMP5,RKDA(R5)  ;HEAD & SECTOR
5658  030062  012737  000023  007354          MOV     #WRDATA,HCS1
5659  030070  053737  001170  007354          BIS     $TMP4,HCS1
5660  030076  013765  007354  000000          MOV     HCS1,RKCS1(R5)  ;DO WRITE DATA CMD
5661
5662  030104  012737  004000  007412          MOV     #4000,TEMP1     ;2000 IS IN VER 30,
5663                                                                  ;CHANGE TO 4000 7-DEC-77
5664
5665  030112  004737  033046                  JSR     PC,DLY          ;DO DELAY
5666
5667  030116  032765  000200  000000  7$:     BIT     #RDY,RKCS1(R5)  ;LOOK FOR CONTROLLER READY
5668  030124  001006                          BNE     8$
5669  030126  004737  032356                  JSR     PC,FRDY         ;FIND RDY AND GET FRESH STATUS
5670  030132  104011                          ERROR   11              ;NO RDY AFTER SEL DRV CMD
5671  030134  004737  033722                  JSR     PC,GSTAT
5672  030140  104107                          ERROR   107             ;HEAD SWITCHING LONGER THAN DELAY
5673
5674  030142  032737  100000  007354  8$:     BIT     #CERR,HCS1
5675  030150  001444                          BEQ     3$
5676  030152  104012                          ERROR   12              ;CERR AFTER WRITE DATA
5677
5678  030154  012737  010340  007444          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
5679  030162  005037  007446                  CLR     E.B0            ;EXPECTED MSG B0
5680  030166  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5681  030174  012737  000001  007452          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
5682  030202  005037  007454                  CLR     E.A2            ;EXPECTED MSG A2
5683  030206  012737  000002  007456          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
5684  030214  012737  000003  007462          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
5685
5686  030222  004737  033100                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
5687  030226  000000                          .WORD   0!0!0           ;& MSGS SPECIFIED HERE
5688  030230  104052                          ERROR   52              ;MSG A0 ERROR AFTER WRITE DATA CMD
5689  030232  104023                          ERROR   23              ;MSH B0 ERROR
5690  030234  104053                          ERROR   53              ;MSG A1 ERROR
5691  030236  104025                          ERROR   25              ;MSG B1 ERROR
5692
5693  030240  023727  007422  000425          CMP     TEMP5,#425      ;HEAD 1,SECTOR 21 DONE?
5694  030246  001405                          BEQ     TST20           ;;GO TO NEXT TEST
5695  030250  012737  000425  007422          MOV     #425,TEMP5
5696  030256  000137  030014                  JMP     2$              ;ELSE REPEAT FOR HEAD 1, SECTOR 21
5697  030262                          3$:
5698
5699                                  ;;*************************************************************************
5700                                  ;*TEST 20        DRIVE OFF TRACK TEST
```

```
                                        ;*
                                        ;*      THIS TEST CHECKS FOR SERVO OSCILLATIONS DURING SETTLING TIME BEYOND
                                        ;*      THE ALLOTED 3MS.
                                        ;*
                                        ;*      1.  INITIALLY, EVERY CYLINDER IS FORMATTED WITH IDENTICAL HEADERS
                                        ;*          (UNIQUE TO EACH CYLINDER)
                                        ;*      2.  A FULL SECTOR WRITE COMMAND IS ISSUED BY A SINGLE CYL SEEK FROM 0 TO 1.
                                        ;*          AS HEADERS ARE IDENTICAL, THE NEXT SECTOR TO COME UNDER THE
                                        ;*          HEADS WILL IMMEDIATELY BE WRITTEN.
                                        ;*      3.  IF THERE IS OSCILLATION SENSED BY READING THE TRIBITS,
                                        ;*          DRIVE OFF TRACK ERROR WILL SET.
                                        ;*
                                        ;*      IN THIS MANNER OSCILLATING SEEKS ARE PERFORMED BETWEEN ALL MAJOR CYLINDERS.
                                        ;*      100 OSCILLATIONS ARE PERFORMED AT EACH MAJOR CYLINDER
                                        ;*      BEFORE DOING THE NEXT CYLINDER
                                        ;*
                                        ;**************************************************************************
030262  000004                        TST20:  SCOPE
030264  012737  000001  001174                MOV     #1,$TIMES       ;;DO 1 ITERATION
030272  012706  001100                        MOV     #STACK,SP       ;RESTORE STK PTR
030276  005237  007340                        INC     BADHDR          ;USED FOR VALID HALT
030302  005037  001346                        CLR     TOCYL
030306  012737  100000  007422                MOV     #BIT15,TEMP5

030314  004737  034252                1$:     JSR     PC,SUBCLR
030320  104024                              ERROR   24              ;CERR AFTER SCLR

030322  012700  001522                        MOV     #HDTAB,R0       ;FORMAT HEADERS ON ALL MAJOR CYL.

030326  013720  001346                2$:     MOV     TOCYL,(R0)+     ;HEADER WORD 0: CYL #
030332  012720  140000                        MOV     #140000,(R0)+   ;HEADER WORD 1: ALL SECTOR 0
030336  012710  140000                        MOV     #140000,(R0)    ;HEADER WORD 2: XOR OF 0 & 1
030342  053720  001346                        BIS     TOCYL,(R0)+     ;ADD CYL # TO WORD 2

030346  020027  001726                        CMP     R0,#HDTAB+132.  ;ALL 22 SECTORS DONE? (22X6=132)
030352  001365                              BNE     2$              ;BR IF NO

030354  012765  001522  000004              MOV     #HDTAB,RKBA(R5)
030362  012765  177676  000002              MOV     #-66.,RKWC(R5)
030370  013765  001346  000020              MOV     TOCYL,RKDC(R5)

030376  012737  000027  007354              MOV     #<WRHEAD>,HCS1
030404  004737  032320                        JSR     PC,DATCMD       ;DO DATA X FOR CMD & GET CONTR RDY
030410  104200                              ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
030412  004737  033722                        JSR     PC,GSTAT        ;GET FRESH STATUS
030416  032737  100000  007354              BIT     #CERR,HCS1
030424  001405                              BEQ     64$
030426  104201                              ERROR   201             ;CERR AFTER WRITE HEADER CMD
030430  104401  045546                      TYPE    ,MSG26          ;ABORTING BAL OF TESTS
030434  000137  031476                      JMP     $EOP
030440                                  64$:

030440  006137  007422                        ROL     TEMP5           ;SET CARRY ONLY ONCE
030444  006137  001346                        ROL     TOCYL           ;SELECT NEXT MAJOR CYL
030450  023737  001346  015462              CMP     TOCYL,MC1       ;ALL MAJOR CYL FORMATTED?
```

```
5757  030456  001316                        BNE     1$                ;BR IF NO
5758  030460  005065  000020                CLR     RKDC(R5)          ;SETUP TO RETURN TO CYL 0
5759
5760  030464  012737  000017  007354        MOV     #SEEK,HCS1
5761  030472  004737  032262                JSR     PC,DOCMD          ;DO SEEK CMD & GET CONTR READY
5762  030476  104131                        ERROR   131               ;NO RDY AFTER SEEK CMD
5763
5764  030500  013737  001420  007412        MOV     T50000,TEMP1      ;SETUP TIMEOUT
5765  030506  004737  032766                JSR     PC,FATT2          ;FIND ATTN
5766  030512  104132                        ERROR   132               ;NO ATTN AFTER SEEK CMD
5767
5768  030514  032737  100000  007354        BIT     #CERR,HCS1
5769  030522  001401                        BEQ     65$
5770  030524  104210                        ERROR   210               ;CERR AFTER SEEK CMD
5771
5772  030526                          65$:
5773
5774  030526  005000                        CLR     R0                ;ITERATION COUNTER
5775  030530  012737  000001  001346        MOV     #1,TOCYL          ;SETUP TO CYL #
5776  030536  005037  001344                CLR     FRCYL
5777
5778  030542  104415                        SCOP1
5779  030544  012706  001100                MOV     #STACK,SP         ;RESTORE STK PTR
5780
5781  030550  013737  001346  001354        MOV     TOCYL,CALDIF      ;SETUP FOR ERROR PRINTOUT
5782
5783  030556  004737  034252          3$:   JSR     PC,SUBCLR
5784  030562  104024                        ERROR   24                ;CERR AFTER SCLR
5785
5786  030564  012737  031220  001176        MOV     #FORM,$ESCAPE
5787  030572  013765  001346  000020        MOV     TOCYL,RKDC(R5)    ;GO TO CYL #
5788  030600  012765  001500  000004        MOV     #DATA1,RKBA(R5)   ;ALL 1'S
5789  030606  052765  000020  000010        BIS     #BAI,RKCS2(R5)
5790  030614  012765  177400  000002        MOV     #-256.,RKWC(R5)   ; SECTOR TO BE ALL 1'S
5791
5792  030622  012737  000023  007354        MOV     #WRDATA,HCS1
5793  030630  004737  032320                JSR     PC,DATCMD         ;DO WRITE DATA CMD & GET CONTR RDY
5794  030634  104011                        ERROR   11                ;NO RDY AFTER WRITE DATA CMD.
5795
5796  030636  004737  033722                JSR     PC,GSTAT          ;GET FRESH STATUS
5797  030642  032737  020000  007404        BIT     #D.DROT,HMR3      ;SEE IF DRIVE OFF TRACK
5798  030650  001401                        BEQ     5$
5799  030652  104112                        ERROR   112               ;DRIVE OFF TRACK AFTER WRITE DATA CMD
5800
5801  030654  032737  100000  007354  5$:   BIT     #CERR,HCS1
5802  030662  001401                        BEQ     6$
5803  030664  104012                        ERROR   12                ;CERR SET AFTER WRITE DATA CMD
5804
5805  030666                          6$:
5806
5807  030666  012737  010340  007444        MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
5808  030674  005037  007446                CLR     E.B0              ;EXPECTED MSG B0
5809  030700  012737  001720  007450        MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
5810  030706  012737  000001  007452        MOV     #1,E.B1           ;MSG ID FOR EXPECTED MSG B1
5811  030714  005037  007454                CLR     E.A2              ;EXPECTED MSG A2
5812  030720  012737  000002  007456        MOV     #2,E.B2           ;MSG ID FOR EXPECTED MSG B2
```

```
 5813  030726  012737  000003  007462          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 5814
 5815  030734  004737  033100                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
 5816  030740  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 5817  030742  104052                          ERROR   52              ;MSG A0 ERROR AFTER WRITE DATA CMD
 5818  030744  104023                          ERROR   23              ;MSH B0 ERROR
 5819  030746  104053                          ERROR   53              ;MSG A1 ERROR
 5820  030750  104025                          ERROR   25              ;MSG B1 ERROR
 5821  030752  023737  001360  001346          CMP     CYLADD,TOCYL
 5822  030760  001401                          BEQ     7$
 5823  030762  104113                          ERROR   113             ;CYL ADDR IN RKMR3 NOT = RKDC
 5824
 5825  030764                          7$:
 5826  030764  104415                          SCOP1
 5827  030766  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
 5828
 5829  030772  004737  034252                  JSR     PC,SUBCLR
 5830  030776  104024                          ERROR   24              ;CERR AFTER SCLR
 5831
 5832                                          ;RETURN TO CYL 0
 5833  031000  012765  001500  000004          MOV     #DATA1,RKBA(R5)
 5834  031006  052765  000020  000010          BIS     #BAI,RKCS2(R5)
 5835  031014  012765  177400  000002          MOV     #-256.,RKWC(R5)
 5836
 5837  031022  012737  000023  007354          MOV     #WRDATA,HCS1
 5838  031030  004737  032320                  JSR     PC,DATCMD       ;DO WRITE DATA CMD & GET CONTR RDY
 5839  031034  104011                          ERROR   11              ;NO RDY AFTER WRITE DATA CMD
 5840
 5841  031036  004737  033722                  JSR     PC,GSTAT        ;GET FRESH STATUS
 5842  031042  032737  020000  007404          BIT     #D.DROT,HMR3
 5843  031050  001401                          BEQ     8$
 5844  031052  104112                          ERROR   112             ;DRIVE OFF TRACK AFTER WRITE DATA CMD
 5845
 5846  031054  032737  100000  007354  8$:     BIT     #CERR,HCS1
 5847  031062  001401                          BEQ     9$
 5848  031064  104012                          ERROR   12              ;CERR AFTER WRITE DATA CMD
 5849
 5850  031066                          9$:
 5851
 5852  031066  012737  010340  007444          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0  ;EXPECTED MSG A0
 5853  031074  005037  007446                  CLR     E.B0            ;EXPECTED MSG B0
 5854  031100  012737  001720  007450          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1  ;EXPECTED A1
 5855  031106  012737  000001  007452          MOV     #1,E.B1         ;MSG ID FOR EXPECTED MSG B1
 5856  031114  005037  007454                  CLR     E.A2            ;EXPECTED MSG A2
 5857  031120  012737  000002  007456          MOV     #2,E.B2         ;MSG ID FOR EXPECTED MSG B2
 5858  031126  012737  000003  007462          MOV     #3,E.B3         ;MSG ID FOR EXPECTED MSG B3
 5859
 5860  031134  004737  033100                  JSR     PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
 5861  031140  000003                          .WORD   T.A2!T.B2!0             ;& MSGS SPECIFIED HERE
 5862  031142  104052                          ERROR   52              ;MSG A0 ERROR AFTER WRITE DATA CMD
 5863  031144  104023                          ERROR   23              ;MSH B0 ERROR
 5864  031146  104053                          ERROR   53              ;MSG A1 ERROR
 5865  031150  104025                          ERROR   25              ;MSG B1 ERROR
 5866  031152  005737  001360                  TST     CYLADD
 5867  031156  001401                          BEQ     10$
 5868  031160  104042                          ERROR   42              ;NOT BACK TO CYL 0
```

```
5869
5870  031162  005200                    10$:    INC     R0
5871  031164  020027  000144                    CMP     R0,#100.        ;ALL ITERATIONS DONE?
5872  031170  001402                            BEQ     13$             ;BR IF YES
5873  031172  000137  030556                    JMP     3$              ;ELSE DO AGAIN
5874
5875  031176  005000                    13$:    CLR     R0              ;RESET ITERATION CTR
5876  031200  006337  001346                    ASL     TOCYL
5877  031204  023737  001346  015462            CMP     TOCYL,MC1       ;ALL MAJOR CYL DONE?
5878  031212  001402                            BEQ     FORM            ;BR IF YES
5879  031214  000137  030556                    JMP     3$              ;ELSE DO NEXT CYL
5880
5881  031220  005037  001176            FORM:   CLR     $ESCAPE
5882  031224  005037  001346                    CLR     TOCYL           ;RESTORE TO ORIG 22 SECTOR FORMAT.
5883  031230  012737  100000  007422            MOV     #BIT15,TEMP5
5884
5885  031236  004737  034252            12$:    JSR     PC,SUBCLR
5886  031242  104024                            ERROR   24              ;CERR AFTER SCLR
5887
5888  031244  012765  001522  000004            MOV     #HDTAB,RKBA(R5) ;REFORMAT ALL MAJOR CYLINDERS
5889  031252  012765  177676  000002            MOV     #-66.,RKWC(R5)
5890  031260  013765  001346  000020            MOV     TOCYL,RKDC(R5)
5891
5892  031266  013737  001346  001362            MOV     TOCYL,CALADD    ;SETUP
5893  031274  012737  000000  001460            MOV     #0,HEAD         ;TO FILL
5894  031302  012737  000000  001466            MOV     #0,FORMAT       ;HEADER
5895  031310  004737  035250                    JSR     PC,FHDTAB       ;TABLE
5896
5897
5898  031314  012737  000027  007354            MOV     #<WRHEAD>,HCS1
5899  031322  004737  032320                    JSR     PC,DATCMD       ;DO DATA X FOR CMD & GET CONTR RDY
5900  031326  104200                            ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
5901  031330  004737  033722                    JSR     PC,GSTAT        ;GET FRESH STATUS
5902  031334  032737  100000  007354            BIT     #CERR,HCS1
5903  031342  001405                            BEQ     64$
5904  031344  104201                            ERROR   201             ;CERR AFTER WRITE HEADER CMD
5905  031346  104401  045546                    TYPE    ,MSG26          ;ABORTING BAL OF TESTS
5906  031352  000137  031476                    JMP     $EOP
5907  031356                            64$:
5908
5909
5910  031356  006137  007422                    ROL     TEMP5
5911  031362  006137  001346                    ROL     TOCYL
5912  031366  023737  001346  015462            CMP     TOCYL,MC1       ;ALL MAJOR CYL REFORMATTED?
5913  031374  001320                            BNE     12$             ;BR IF NO
5914
5915  031376  005065  000020                    CLR     RKDC(R5)        ;SETUP TO RETURN TO CYL 0
5916  031402  005037  001176                    CLR     $ESCAPE
5917
5918  031406  012737  000017  007354            MOV     #SEEK,HCS1
5919  031414  004737  032262                    JSR     PC,DOCMD        ;DO SEEK CMD & GET CONTR READY
5920  031420  104131                            ERROR   131             ;NO RDY AFTER SEEK CMD
5921
5922  031422  013737  001420  007412            MOV     T50000,TEMP1    ;SETUP TIMEOUT
5923  031430  004737  032766                    JSR     PC,FATT2        ;FIND ATTN
5924  031434  104132                            ERROR   132             ;NO ATTN AFTER SEEK CMD
```

```
 5925
 5926  031436  032737  100000  007354          BIT     #CERR,HCS1
 5927  031444  001401                          BEQ     65$
 5928  031446  104210                          ERROR   210             ;CERR AFTER SEEK CMD
 5929
 5930  031450                          65$:
 5931
 5932
 5933  031450  005737  007342                  TST     HPEND           ;SEE IF HALT PENDING
 5934  031454  001406                          BEQ     4$              ;BR IF NO
 5935  031456  005037  007342                  CLR     HPEND           ;CLEAR FOR FUTURE FORMATTING
 5936  031462  005037  007340                  CLR     BADHDR          ;HEADERS NOW OK
 5937  031466  000137  036320                  JMP     STOP            ;GO BACK & HALT CPU
 5938
 5939  031472  005037  007340          4$:     CLR     BADHDR          ;HEADERS NOW OK
 5940
```

```
                        .SBTTL  END OF PASS ROUTINE

                        ;;**********************************************************************
                        ;*INCREMENT THE PASS NUMBER ($PASS)
                        ;*TYPE 'END PASS #XXXXX'' (WHERE XXXXX IS A DECIMAL NUMBER)
                        ;*IF THERES A MONITOR GO TO IT
                        ;*IF THERE ISN'T JUMP TO STAPT

031476                  $EOP:

031476  000004                    SCOPE
031500  005037  001176            CLR     $ESCAPE
031504  012737  000001  001174    MOV     #1,$TIMES
031512  012706  001100            MOV     #STACK,SP
031516  005237  001220            INC     $DEVCT          ;INCR COUNT FOR # DRIVES CHECKED
031522  023737  007474  001220    CMP     DRIVS,$DEVCT    ;ARE ALL DRIVES PRESENT TESTED?
031530  001404                    BEQ     1$              ;BR IF YES
031532  005037  001512            CLR     BSERR           ;CLEAR BAD SECTOR ERROR FLAG
031536  000137  015172            JMP     NUDRV           ;ELSE TEST NEXT DRIVE PRESENT
031542  005037  001512    1$:     CLR     BSERR           ;CLEAR BAD SECTOR ERROR FLAG
031546  000401                    BR      $EOP1+2
031550  000004            $EOP1:  SCOPE
031552  005037  001102            CLR     $TSTNM          ;;ZERO THE TEST NUMBER
031556  005037  001174            CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
031562  005237  001216            INC     $PASS           ;;INCREMENT THE PASS NUMBER
031566  042737  100000  001216    BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
031574  005327                    DEC     (PC)+           ;;LOOP?
031576  000001            $EOPCT: .WORD   1
031600  003022                    BGT     $DOAGN          ;;YES
031602  012737                    MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
031604  000001            $ENDCT: .WORD   1
031606  031576                    $EOPCT
031610  104401  031655            TYPE    ,$ENDMG         ;;TYPE 'END PASS #''
031614  013746  001216            MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
031620  104405                    TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
031622  104401  031652            TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
031626  013700  000042    $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
031632  001405                    BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
031634  000005                    RESET                   ;;CLEAR THE WORLD
031636  004710            $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
031640  000240                    NOP                     ;;SAVE ROOM
031642  000240                    NOP                     ;;FOR
031644  000240                    NOP                     ;;ACT11
031646                  $DOAGN:
031646  000137                    JMP     @(PC)+          ;;RETURN
031650  031672            $RTNAD: .WORD   STAPT
031652     377     377     000    $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
031655     015  042412  042116    $ENDMG: .ASCIZ  <15><12>/END PASS #/
031662  050040  051501  020123
031670  000043

                        ;ADD WAITING LOOP FOR THE APT MODE
031672  122737  000001  001230    STAPT:  CMPB    #APTENV,$ENV
031700  001007                    BNE     2$
031702  023727  001216  000002            CMP     $PASS,#2        ;TWO PASS DONE ?,AS REQUIRED BY
031710  103403                    BLO     2$                      ;NOT YET
031712  005237  001102    1$:     INC     $TSTNM  ;INCREMENT THE TEST NUMBER
```

```
 5997  031716  000775                      BR      1$              ;WAITING LOOP FOR APT TO DUMP NEXT PROG
 5998  031720  000137  013372        2$:   JMP     ST5             ;RETURN TO MAIN LOOP
```

```
 5999                                   .SBTTL  SUBROUTINES
 6000
 6001                                   ;SUBROUTINE TO CLEAR ALL FLAGS FROM DDUMP THRU DOTIM
 6002                                   ;
 6003
 6004  031724  012700  007464           CLRFLG: MOV     #DDUMP,R0
 6005  031730  012701  177757                   MOV     #-17.,R1
 6006  031734  005020                   1$:     CLR     (R0)+
 6007  031736  005201                           INC     R1
 6008  031740  001375                           BNE     1$
 6009  031742  000207                           RTS     PC
 6010
 6011                                   ;
 6012                                   ;TYPE PROGRAM ID IF FTITLE=0
 6013                                   ;
 6014
 6015  031744  005737  001340           TITLE:  TST     FTITLE
 6016  031750  001024                           BNE     1$
 6017  031752  005237  001340                   INC     FTITLE
 6018  031756  104401  043542                   TYPE    ,MSG1                   ;PROGRAM ID
 6019                                   .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
 6020  031762  005737  000042                   TST     @#42                    ;;ARE WE RUNNING UNDER XXDP/ACT?
 6021  031766  001012                           BNE     64$                     ;;BRANCH IF YES
 6022  031770  123727  001230  000001           CMPB    $ENV,#1                 ;;ARE WE RUNNING UNDER APT?
 6023  031776  001406                           BEQ     64$                     ;;BRANCH IF YES
 6024  032000  023727  001140  000176           CMP     SWR,#SWREG              ;;SOFTWARE SWITCH REG SELECTED?
 6025  032006  001005                           BNE     65$                     ;;BRANCH IF NO
 6026  032010  104406                           GTSWR                           ;;GET SOFT-SWR SETTINGS
 6027  032012  000403                           BR      65$
 6028  032014  112737  000001  001134   64$:    MOVB    #1,$AUTOB               ;;SET AUTO-MODE INDICATOR
 6029  032022                           65$:
 6030  032022  000207                   1$:     RTS     PC
 6031
 6032                                   ;
 6033                                   ;ROUTINE TO INPUT DRIVE NOS. TYPED IN & SET
 6034                                   ;DRIVS, DRIV0-DRIV7 REGISTERS APPROPRIATELY
 6035                                   ;
 6036
 6037  032024  104411                   GDRVS:  RDLIN
 6038  032026  012600                           MOV     (SP)+,R0                ;GET STARTING ADDR OF ASCII STRING
 6039  032030  012701  177770                   MOV     #-8.,R1                 ;SET UP COUNT
 6040  032034  112002                   1$:     MOVB    (R0)+,R2                ;GET ASCII CHAR
 6041  032036  042702  177400                   BIC     #177400,R2              ;MASK HI BYTE
 6042  032042  012703  007476                   MOV     #DRIV0,R3               ;DRIVE FLAG ADDR
 6043  032046  012704  000060                   MOV     #60,R4
 6044
 6045  032052  020402                   2$:     CMP     R4,R2                   ;WAS TYPED CHAR 0 THRU 7?
 6046  032054  001415                           BEQ     3$                      ;BRANCH IF YES
 6047  032056  005723                           TST     (R3)+                   ;NO, INCREMENT DR FLAG ADDR
 6048  032060  005204                           INC     R4
 6049  032062  020427  000070                   CMP     R4,#70
 6050  032066  001371                           BNE     2$                      ;S/B 0-7 OR TERMINATOR
 6051  032070  005702                           TST     R2
 6052  032072  001022                           BNE     4$
 6053  032074  020127  177770                   CMP     R1,#-8.
 6054  032100  001426                           BEQ     6$                      ;DEFAULT ALL DRIVES
```

```
                                7$:     CLR     SIZFLG          ;BYPASS TEST 1 (SIZING)
                                        RTS     PC              ;FOUND TERMINATOR, EXIT

                                3$:     INC     @R3             ;SET UP FLAG FOR THE DRIVE
                                        INC     DRIVS           ;INCREMENT TOTAL # DRIVES TO BE TESTED
                                        MOVB    (R0)+,R2        ;GET NEXT ASCII CHAR.
                                        BIC     #177400,R2      ;MASK
                                        CMP     #54,R2          ;IS IT A COMMA?
                                        BEQ     5$              ;YES, GO TO NEXT WORD.
                                        TST     R2              ;NO, IS IT A TERMINATOR?
                                        BNE     4$              ;IF NOT, SOMETHING WRONG.
                                        BR      7$              ;FOUND TERMINATOR, EXIT

                                4$:     TYPE    ,EM1            ;ONLY 0-7 ALLOWED.
                                        JMP     PRGSRT          ;START ALL OVER

                                5$:     INC     R1              ;S/B NO MORE THAN 8 DIFF
                                        BNE     1$              ;DRIVES TYPED IN.
                                        BR      4$              ;IF NORE, HAVE ERROR.

                                6$:     INC     SIZFLG          ;DO TEST 1 (SIZING)
                                        RTS     PC              ;EXIT.

                                ;
                                ;ROUTINE TO INPUT RKBAS OR DEFAULT.
                                ;
                                GBA:    RDOCT
                                        MOV     (SP)+,R0        ;GET LOW ORDER FROM STACK
                                        TST     R0
                                        BEQ     1$              ;BRANCH IF DEFAULT.
                                        MOV     R0,$BASE
                                        RTS     PC
                                1$:     MOV     #177440,$BASE   ;DEFAULT VALUE
                                        RTS     PC
                                ;
                                ;ROUTINE TO INPUT RKVEC OR DEFAULT
                                ;
                                GINT:   RDOCT
                                        MOV     (SP)+,R0        ;GET LOW ORDER FROM STACK
                                        TST     R0
                                        BEQ     1$              ;BRANCH IF DEFAULT
                                        MOV     R0,RKVEC
                                2$:     JSR     PC,SETINT
                                        RTS     PC
                                1$:     MOV     #210,RKVEC      ;DEFAULT VALUE
                                        BR      2$
                                ;
                                ;ROUTINE TO SETUP INTERRUPT VECTOR & PRIORITY
                                ;
                                SETINT: MOV     RKVEC,R0
                                        MOV     #INTER,(R0)+    ;INTER ADDR TO RKVEC
```

```
                                        MOV     RKPRI,(R0)      ;PR5 TO RKVEC+2
                                        RTS     PC

                                ;
                                ;THIS ROUTINE CDT IN RKCS1 IF DRIVE UNDER TEST IS AN RK07.
                                ;ENTER WITH COMMAND IN HCS1
                                ;
DOCMD:  BIS     $TMP4,HCS1      ;SET CDT IF RK07
        MOV     HCS1,RKCS1(R5)  ;DO COMMAND
        MOV     T10,TEMP1
        JSR     PC,FRDY         ;FIND CONTR READY
        RTS     PC              ;SET HERE IF NOT RDY
        ADD     #2,(SP)         ;ELSE SKIP OVER ERROR
        RTS     PC
                                ;
                                ;THIS ROUTINE IS SIMILAR TO THE ABOVE BUT IS USED FOR DATA TRANSFERS
                                ;& REQUIRES A LONGER TIMEOUT
                                ;
DATCMD: BIS     $TMP4,HCS1      ;SET CDT IF RK07
        MOV     HCS1,RKCS1(R5)  ;DO CMD
        MOV     T50000,TEMP1
        JSR     PC,FRDY         ;FIND CONTR RDY
        RTS     PC
        ADD     #2,(SP)
        RTS     PC

                                ;
                                ;ROUTINE TO FIND CONTROLLER READY (RDY) DURING A DELAY
                                ;ENTER WITH A COUNT IN TEMP1
                                ;RETURN IF RDY NOT PRESENT (ERROR CONDITION)
                                ;RETURN +2 IF RDY PRESENT (SKIP OVER ERROR)
                                ;STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
                                ;
FRDY:   BIT     #RDY,RKCS1(R5)
        BNE     1$
        DEC     TEMP1
        BNE     FRDY
        JSR     PC,HOLD         ;STORE ALL RK611 REGS IN HOLDING REGS.
        JSR     PC,CKCERR       ;CHECK FOR SPECIAL CERR
        RTS     PC              ;NO RDY, EXIT
1$:     ADD     #2,(SP)         ;SKIP OVER ERROR
        JSR     PC,HOLD
        JSR     PC,CKCERR       ;CHECK FOR SPECIAL CERR
        RTS     PC
                                ;
                                ;ROUTINE TO FIND CONTROLLER READY AND STORE DRIVE REGS ONLY
                                ;
FRDY1:  BIT     #RDY,RKCS1(R5)
        BNE     1$
        DEC     TEMP1
        BNE     FRDY1
        MOV     RKMR2(R5),HMR2
        MOV     RKMR3(R5),HMR3
        JSR     PC,CKCERR       ;CHECK FOR SPECIAL CERR CONDITIONS
        RTS     PC              ;NO RDY, EXIT
1$:     ADD     #2,(SP)         ;SKIP OVER ERROR
```

```
                                        ;
                                        ;STORE ALL RK611 REGISTERS IN HOLDING REGS
                                        ;
032512  016537  000000  007354  HOLD:   MOV     RKCS1(R5),HCS1
032520  016537  000010  007356          MOV     RKCS2(R5),HCS2
032526  016537  000002  007360          MOV     RKWC(R5),HWC
032534  016537  000004  007362          MOV     RKBA(R5),HBA
032542  016537  000006  007364          MOV     RKDA(R5),HDA
032550  016537  000012  007366          MOV     RKDS(R5),HDS
032556  016537  000014  007370          MOV     RKER(R5),HER
032564  016537  000016  007372          MOV     RKASOF(R5),HASOF
032572  016537  000020  007374          MOV     RKDC(R5),HDC
032600  016537  000026  007400          MOV     RKMR1(R5),HMR1
032606  016537  000034  007402          MOV     RKMR2(R5),HMR2
032614  016537  000036  007404          MOV     RKMR3(R5),HMR3
032622  016537  000030  007406          MOV     RKECPS(R5),HPOS
032630  016537  000032  007410          MOV     RKECPT(R5),HPAT
032636  000207                          RTS     PC
                                        ;
                                        ;ROUTINE TO CHECK FOR CORRECT ATTN
                                        ;RETURN IF ATTN NOT PRESENT (ERROR CONDITION)
                                        ;RETURN +2 IF ATTN PRESENT (SKIP OVER ERROR)
                                        ;
032640  010446                  TSTATN: MOV     R4,-(SP)        ;SAV R4
032642  013704  001222                  MOV     $UNIT,R4
032646  136437  007344  007373          BITB    ATTN(R4),HASOF+1
032654  001404                          BEQ     1$              ;BRANCH IF ATTN NOT PRESENT
032656  012604                          MOV     (SP)+,R4        ;RESTOR R4
032660  062716  000002                  ADD     #2,(SP)         ;INCR RET ADDR TO JUMP OVER ERROR.
032664  000207                          RTS     PC
032666  012604                  1$:     MOV     (SP)+,R4        ;RESTOR R4
032670  000207                          RTS     PC
                                        ;
                                        ;ROUTINE TO FIND ATTN WITHIN TIMES GREATER THAN 1 SEC
                                        ;ENTER WITH TIME IN SECONDS IN TEMP2
                                        ;RETURN IF NO ATTN (ERROR CONDITION)
                                        ;RETURN +2 IF ATTN FOUND
                                        ;STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
                                        ;
032672  010446                  FATT1:  MOV     R4,-(SP)        ;SAV R4
032674  012737  177777  007412  3$:     MOV     #-1,TEMP1
032702  013704  001222                  MOV     $UNIT,R4
032706  136465  007344  000017  1$:     BITB    ATTN(R4),RKASOF+1(R5) ;FIND CORRECT ATTN
032714  001014                          BNE     2$
032716  005337  007412                  DEC     TEMP1
032722  001371                          BNE     1$
032724  005337  007414                  DEC     TEMP2
```

```
6223  032730  001361                    BNE     3$
6224  032732  005065  000026            CLR     RKMR1(R5)       ;SELECT WORD 0
6225  032736  004737  033722            JSR     PC,GSTAT        ;GET LATEST STATUS
6226  032742  012604                    MOV     (SP)+,R4        ;RESTOR R4
6227  032744  000207                    RTS     PC
6228  032746  005065  000026    2$:     CLR     RKMR1(R5)
6229  032752  004737  033722            JSR     PC,GSTAT        ;GET STATUS AFTER ATTN SEEN
6230  032756  012604                    MOV     (SP)+,R4        ;RESTOR R4
6231  032760  062716  000002            ADD     #2,(SP)         ;SKIP OVER ERROR
6232  032764  000207                    RTS     PC
6233
6234                            ;
6235                            ;ROUTINE TO FIND ATTN WITHIN 1 SEC
6236                            ;ENTER WITH COUNT IN TEMP1
6237                            ;RETURN IF NO ATTN (ERROR)
6238                            ;RETURN +2 IF ATTN FOUND
6239                            ;STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
6240                            ;
6241
6242  032766  010446            FATT2:  MOV     R4,-(SP)        ;SAV R4
6243  032770  013704  001222    2$:     MOV     $UNIT,R4
6244  032774  136465  007344  000017    BITB    ATTN(R4),RKASOF+1(R5)  ;FIND CORRECT ATTN
6245  033002  001011                    BNE     1$
6246  033004  005337  007412            DEC     TEMP1
6247  033010  001367                    BNE     2$
6248  033012  005065  000026            CLR     RKMR1(R5)       ;SELECT WORD 0
6249  033016  004737  033722            JSR     PC,GSTAT        ;GET LATEST STATUS.
6250  033022  012604                    MOV     (SP)+,R4        ;RESTOR R4
6251  033024  000207                    RTS     PC
6252  033026  005065  000026    1$:     CLR     RKMR1(R5)
6253  033032  004737  033722            JSR     PC,GSTAT
6254  033036  012604                    MOV     (SP)+,R4        ;RESTOR R4
6255  033040  062716  000002            ADD     #2,(SP)         ;SKIP OVER ERROR
6256  033044  000207                    RTS     PC
6257                            ;
6258                            ;ENTER WITH A COUNT IN TEMP1
6259                            ;THE DELAY IS APPROX 17 US/ITERATION + 12 US TO EXIT
6260                            ;WHEN COUNT IS 0. BASED ON AN 11/05
6261                            ;
6262  033046  005737  007412    DLY:    TST     TEMP1           ;5.6 US
6263  033052  001403                    BEQ     1$              ;1.9 US
6264  033054  005337  007412            DEC     TEMP1           ;6.8 US
6265  033060  000772                    BR      DLY             ;2.5 US
6266  033062  000207            1$:     RTS     PC              ;3.8 US
6267                            ;
6268                            ;THIS ROUTINE TYPES BYPASSED DRIVE#.  ENTER WITH DRIVE# IN R0
6269                            ;
6270
6271  033064  104401  045435    BYP:    TYPE    ,MSG14          ;BYPASS DRIVE
6272  033070  010046                    MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
6273                                                            ;;TYPE DR#
6274  033072  104403                    TYPOS                   ;;GO TYPE--OCTAL ASCII
6275  033074     001                    .BYTE   1               ;;TYPE 1 DIGIT(S)
6276  033075     000                    .BYTE   0               ;;SUPPRESS LEADING ZEROS
6277  033076  000207                    RTS     PC
6278                            ;
```

```
                                        ;THIS ROUTINE READS ALL MSG A & B WORDS & CHECKS THEM AS REQ'D.
                                        ;
033100  017637  000000  001520  CHKMSG: MOV     @(SP),CHKFLG    ;PASS MSGS TO BE TESTED
033106  062716  000002                  ADD     #2,(SP)         ;BUMP RETURN ADDR TO 1ST ERROR
033112  004737  033756                  JSR     PC,GSTAT1       ;GET ALL ACTUAL DRIVE & CONTR STATUS

033116  053737  001222  007444          BIS     $UNIT,E.A0      ;SET UNIT #
033124  053737  001222  007450          BIS     $UNIT,E.A1
033132  053737  001222  007454          BIS     $UNIT,E.A2
033140  053737  001222  007460          BIS     $UNIT,E.A3
033146  053737  015470  007444          BIS     E.DDT,E.A0      ;ADD DRIVE TYPE

033154  013746  007412                  MOV     TEMP1,-(SP)     ;SAVE TEMP1

033160  013737  007444  007412          MOV     E.A0,TEMP1
033166  004737  036214                  JSR     PC,SBPAR        ;GET PARITY FOR MSG A0
033172  013737  007412  007444          MOV     TEMP1,E.A0

033200  013737  007450  007412          MOV     E.A1,TEMP1
033206  004737  036214                  JSR     PC,SBPAR        ;GET PARITY FOR MSG A1
033212  013737  007412  007450          MOV     TEMP1,E.A1

033220  013737  007454  007412          MOV     E.A2,TEMP1
033226  004737  036214                  JSR     PC,SBPAR        ;GET PARITY FOR MSG A2
033232  013737  007412  007454          MOV     TEMP1,E.A2

033240  013737  007446  007412          MOV     E.B0,TEMP1
033246  004737  036214                  JSR     PC,SBPAR        ;GET PARITY FOR MSG B0
033252  013737  007412  007446          MOV     TEMP1,E.B0

033260  013737  007452  007412          MOV     E.B1,TEMP1
033266  004737  036214                  JSR     PC,SBPAR        ;GET PARITY FOR MSG B1
033272  013737  007412  007452          MOV     TEMP1,E.B1

033300  013737  007456  007412          MOV     E.B2,TEMP1
033306  004737  036214                  JSR     PC,SBPAR        ;GET PARITY FOR MSG B2
033312  013737  007412  007456          MOV     TEMP1,E.B2

033320  013737  007462  007412          MOV     E.B3,TEMP1
033326  004737  036214                  JSR     PC,SBPAR        ;GET PARITY FOR MSG B3
033332  013737  007412  007462          MOV     TEMP1,E.B3

033340  012637  007412                  MOV     (SP)+,TEMP1     ;RESTORE TEMP1
033344  013737  001176  001172          MOV     $ESCAPE,$TMP5   ;SAVE ESCAPE

033352  023737  007424  007444          CMP     H.A0,E.A0       ;TEST MSG A0
033360  001411                          BEQ     2$              ;BR IF OK
033362  012737  033374  001176          MOV     #1$,$ESCAPE     ;ELSE SETUP ESCAPE
033370  011646                          MOV     (SP),-(SP)      ;COPY RET ADDR.
033372  000207                          RTS     PC              ;& RETURN TO MAINLINE ERROR

033374  032777  001000  145536  1$:     BIT     #SW9,@SWR       ;RET HERE FROM MAINLINE ERROR
033402  001107                          BNE     20$             ;& BR IF LOOP ON ERROR
033404  062716  000002          2$:     ADD     #2,(SP)         ;BUMP RET ADDR TO NEXT ERROR

033410  023737  007426  007446          CMP     H.B0,E.B0       ;TEST MSG B0
```

```
 6335  033416  001411                          BEQ     5$                ;BR IF OK
 6336  033420  012737  033432  001176          MOV     #4$,$ESCAPE       ;ELSE SETUP ESCAPE
 6337  033426  011646                          MOV     (SP),-(SP)        ;COPY RET ADDR
 6338  033430  000207                          RTS     PC                ;& RETURN TO MAINLINE ERROR
 6339
 6340  033432  032777  001000  145500  4$:     BIT     #SW9,@SWR         ;RETURN HERE FROM MAINLINE ERROR
 6341  033440  001070                          BNE     20$               ;& BR IF LOOP ON ERROR
 6342  033442  062716  000002          5$:     ADD     #2,(SP)           ;BUMP RET ADDR TO NEXT ERROR
 6343
 6344  033446  023737  007430  007450          CMP     H.A1,E.A1         ;TEST MSG A1
 6345  033454  001411                          BEQ     8$                ;BR IF OK
 6346  033456  012737  033470  001176          MOV     #7$,$ESCAPE
 6347  033464  011646                          MOV     (SP),-(SP)
 6348  033466  000207                          RTS     PC
 6349
 6350  033470  032777  001000  145442  7$:     BIT     #SW9,@SWR
 6351  033476  001051                          BNE     20$
 6352  033500  062716  000002          8$:     ADD     #2,(SP)
 6353
 6354  033504  023737  007432  007452          CMP     H.B1,E.B1         ;TEST MSG B1
 6355  033512  001411                          BEQ     11$               ;BR IF OK
 6356  033514  012737  033526  001176          MOV     #10$,$ESCAPE
 6357  033522  011646                          MOV     (SP),-(SP)
 6358  033524  000207                          RTS     PC
 6359
 6360  033526  032777  001000  145404  10$:    BIT     #SW9,@SWR
 6361  033534  001032                          BNE     20$
 6362  033536  062716  000002          11$:    ADD     #2,(SP)
 6363
 6364  033542  032737  000001  001520  12$:    BIT     #T.A2,CHKFLG      ;TEST MSG A2?
 6365  033550  001402                          BEQ     13$               ;BR IF NO
 6366  033552  004737  034642                  JSR     PC,RCYLD          ;PUT INFO CYLDIF, DO NOT CHECK
 6367  033556  032737  000002  001520  13$:    BIT     #T.B2,CHKFLG      ;TEST MSG B2?
 6368  033564  001402                          BEQ     14$               ;BR IF NO
 6369  033566  004737  034714                  JSR     PC,RCYLA          ;PUT INFO IN CYLADD, DO NOT CHECK
 6370
 6371  033572  032737  000004  001520  14$:    BIT     #T.B3,CHKFLG      ;TEST MSG B3?
 6372  033600  001404                          BEQ     15$
 6373  033602  004737  034752                  JSR     PC,RSEC           ;PUT INFO IN SECTOR, DO NOT CHECK
 6374  033606  004737  035010                  JSR     PC,RHEAD          ;PUT INFO IN HEADA, DO NOT CHECK
 6375
 6376  033612  013737  001172  001176  15$:    MOV     $TMP5,$ESCAPE     ;RESTORE ESCAPE
 6377  033620  000207                          RTS     PC
 6378
 6379  033622  012706  001100          20$:    MOV     #STACK,SP         ;RESET STACK PTR
 6380  033626  013737  001172  001176          MOV     $TMP5,$ESCAPE     ;RESTORE ESCAPE
 6381  033634  000177  145250                  JMP     @$LPERR
 6382                                  ;
 6383                                  ;THIS ROUTINE CHECKS FOR CERTAIN ERROR CONDITIONS ONLY
 6384                                  ;I.E.: IF NED, CTO OR MDS SET MESSAGE A & B ARE INVALID
 6385                                  ;
 6386  033640  005737  001516          CKCERR: TST     BYPCERR
 6387  033644  001025                          BNE     4$
 6388  033646  032737  100000  007354          BIT     #CERR,HCS1
 6389  033654  001001                          BNE     1$                ;BR IF CERR
 6390  033656  000207                          RTS     PC
```

```
6391
6392  033660  032737  004000  007354  1$:     BIT     #CTO,HCS1
6393  033666  001402                          BEQ     2$              ;BR IF NOT CTO
6394  033670  104125                          ERROR   125             ;CTO ERROR, MSG A & B INVALID
6395  033672  000207                          RTS     PC
6396
6397  033674  032737  010000  007356  2$:     BIT     #NED,HCS2
6398  033702  001401                          BEQ     3$              ;BR IF NOT NED
6399  033704  104126                          ERROR   126             ;NED ERROR, MSG A & B INVALID
6400
6401  033706  032737  001000  007356  3$:     BIT     #MDS,HCS2
6402  033714  001401                          BEQ     4$
6403  033716  104127                          ERROR   127             ;MDS ERROR, MSG A & B INVALID
6404
6405  033720  000207                  4$:     RTS     PC
6406
6407                                  ;
6408                                  ;THIS ROUTINE DOES THE SELECT DRIVE COMMAND TO GET STATUS
6409                                  ;IT THEN WAITS FOR CONTROLLER READY
6410                                  ;IF RDY NOT RECEIVED BY THE TIMEOUT, AN ERROR IS FLAGGED
6411                                  ;
6412
6413  033722  013746  007412          GSTAT:  MOV     TEMP1,-(SP)     ;SAVE TEMP1
6414  033726  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;CURRENT DRIVE #
6415  033734  012737  000001  007354          MOV     #SELDRV,HCS1
6416  033742  004737  032262                  JSR     PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6417  033746  104117                          ERROR   117             ;RDY NOT SET BY END OF SELECT DRIVE CMD
6418  033750  012637  007412                  MOV     (SP)+,TEMP1     ;RESTOR TEMP1.
6419  033754  000207                          RTS     PC
6420                                  ;
6421                                  ;THIS ROUTINE GETS STATUS OF ALL DRIVE REGISTERS (MSG A0-A3, B0-B3)
6422                                  ;& ALL CONTROLLER REGISTERS.
6423                                  ;
6424  033756  013746  007412          GSTAT1: MOV     TEMP1,-(SP)     ;SAVE TEMP1
6425  033762  004737  032512                  JSR     PC,HOLD         ;GET ALL CONTR REG
6426  033766  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CLEAR CONTR
6427  033774  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;CURRENT DRIVE #
6428  034002  012765  000003  000026          MOV     #3,RKMR1(R5)    ;SELECT WORD 3
6429  034010  004737  034206                  JSR     PC,GSTAT2
6430  034014  104117                          ERROR   117             ;RDY NOT SET BY END OF SELECT DRV CMD
6431  034016  013737  007402  007440          MOV     HMR2,H.A3       ;STORE MSG A3
6432  034024  013737  007404  007442          MOV     HMR3,H.B3       ;STORE MSG B3
6433
6434  034032  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
6435  034040  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
6436  034046  012765  000002  000026          MOV     #2,RKMR1(R5)    ;SELECT WORD 2
6437  034054  004737  034206                  JSR     PC,GSTAT2
6438  034060  104117                          ERROR   117             ;RDY NOT SET BY END OF SELECT DRV CMD
6439  034062  013737  007402  007434          MOV     HMR2,H.A2       ;STORE MSG A2
6440  034070  013737  007404  007436          MOV     HMR3,H.B2       ;STORE MSG B2
6441
6442  034076  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
6443  034104  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
6444  034112  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
6445  034120  004737  034206                  JSR     PC,GSTAT2
6446  034124  104117                          ERROR   117             ;RDY NOT SET BY END OF SELECT DRV CMD
```

```
6447  034126  013737  007402  007430          MOV     HMR2,H.A1       ;STORE MSG A1
6448  034134  013737  007404  007432          MOV     HMR3,H.B1       ;STORE MSG B1
6449
6450  034142  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
6451  034150  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
6452  034156  004737  034206                  JSR     PC,GSTAT2
6453  034162  104117                          ERROR   117             ;RDY NOT SET BY END OF SEL DRV CMD
6454  034164  013737  007402  007424          MOV     HMR2,H.A0       ;STORE MSG A0
6455  034172  013737  007404  007426          MOV     HMR3,H.B0       ;STORE MSG B0
6456
6457  034200  012637  007412                  MOV     (SP)+,TEMP1     ;RESTORE TEMP1
6458  034204  000207                          RTS     PC
6459
6460  034206  012737  000001  007354  GSTAT2: MOV     #SELDRV,HCS1
6461  034214  053737  001170  007354          BIS     $TMP4,HCS1      ;RET CDT IF RK07
6462  034222  013765  007354  000000          MOV     HCS1,RKCS1(R5)  ;GET STATUS
6463  034230  013737  001406  007412          MOV     T10,TEMP1
6464  034236  004737  032424                  JSR     PC,FRDY1        ;FIND CONTR RDY & STORE DRIVE REGS ONLY
6465  034242  000207                          RTS     PC              ;RET HERE IF NOT RDY
6466  034244  062716  000002                  ADD     #2,(SP)         ;RET HERE IF OK
6467  034250  000207                          RTS     PC
6468
6469                                  ;
6470                                  ;THIS ROUTINE DOES A SUBSYSTEM CLEAR & WAITS FOR CONTROLLER READY
6471                                  ;IF RDY IS NOT RECEIVED BY THE END OF THE TIMEOUT, AN ERROR IS FLAGGED.
6472                                  ;THE ROUTINE THEN GETS CURRENT STATUS & CHECKS FOR CONTROLLER ERROR (CERR)
6473                                  ;RETURN IF CERR SET
6474                                  ;RETURN +2 IF CERR CLEAR
6475                                  ;
6476  034252  012765  000040  000010  SUBCLR: MOV     #SCLR,RKCS2(R5) ;SUBSYS CLEAR
6477  034260  013737  001406  007412          MOV     T10,TEMP1
6478  034266  004737  032356                  JSR     PC,FRDY         ;FIND RDY
6479  034272  104120                          ERROR   120             ;RDY NOT SET BY END OF SCLR
6480  034274  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;CURRENT DRIVE #
6481  034302  005065  000026                  CLR     RKMR1(R5)       ;SELECT WORD 0
6482  034306  004737  033722                  JSR     PC,GSTAT        ;GET STATUS
6483  034312  032737  100000  007354          BIT     #CERR,HCS1      ;CHECK FOR CONT ERROR
6484  034320  001401                          BEQ     1$
6485  034322  000207                          RTS     PC
6486  034324  062716  000002          1$:     ADD     #2,(SP)         ;SKIP OVER ERROR
6487  034330  000207                          RTS     PC
6488
6489                                  ;
6490                                  ;READ THE SECTOR COUNT IN RKMR3, RIGHT JUSTIFY IT & STORE IT IN 'SECTOR'
6491                                  ;
6492  034332  012765  000003  000026  RDSEC:  MOV     #3,RKMR1(R5)    ;WORD 3
6493  034340  004737  033722                  JSR     PC,GSTAT
6494  034344  013737  007404  001402          MOV     HMR3,SECTOR
6495  034352  042737  177017  001402          BIC     #^C<M.SECT>,SECTOR
6496  034360  006237  001402                  ASR     SECTOR          ;RIGHT JUSTIFY
6497  034364  006237  001402                  ASR     SECTOR          ;SECTOR
6498  034370  006237  001402                  ASR     SECTOR          ;INFO
6499  034374  006237  001402                  ASR     SECTOR
6500  034400  000207                          RTS     PC
6501                                  ;
6502                                  ;READ THE CYL DIFF/OFFSET IN RKMR2, RIGHT JUSTIFY IT & STORE IT IN 'CYLDIF'
```

```
 6503                                    ;
 6504  034402  012765  000002  000026  RDCYLD: MOV     #2,RKMR1(R5)    ;WORD 2
 6505  034410  004737  033722                  JSR     PC,GSTAT
 6506  034414  013737  007402  001356          MOV     HMR2,CYLDIF
 6507  034422  043737  015466  001356          BIC     MASK1,CYLDIF
 6508  034430  006237  001356                  ASR     CYLDIF          ;RIGHT JUSTIFY
 6509  034434  006237  001356                  ASR     CYLDIF          ;CYL DIFF/OFFSET
 6510  034440  006237  001356                  ASR     CYLDIF          ;INFO
 6511  034444  006237  001356                  ASR     CYLDIF
 6512  034450  023737  001356  015464          CMP     CYLDIF,MASK     ;CHK TO SEE IF RET IN COMPL. FORM
 6513  034456  001002                          BNE     1$              ;BR IF NOT
 6514  034460  005037  001356                  CLR     CYLDIF          ;CLR IF YES
 6515  034464  000207                  1$:     RTS     PC
 6516
 6517                                    ;
 6518                                    ;QUICK SELECT DRIVE COMMAND TO OBTAIN CYL DIFF
 6519                                    ;
 6520  034466  013746  007412          QKCYLD: MOV     TEMP1,-(SP)     ;SAVE TEMP1
 6521  034472  012765  000002  000026          MOV     #2,RKMR1(R5)    ;SELECT WORD 2
 6522  034500  012737  000001  007354          MOV     #SELDRV,HCS1    ;SELECT DRIVE CMD
 6523  034506  053737  001170  007354          BIS     $TMP4,HCS1
 6524  034514  013765  007354  000000          MOV     HCS1,RKCS1(R5)
 6525  034522  013737  001406  007412          MOV     T10,TEMP1
 6526  034530  032765  000200  000000  1$:     BIT     #RDY,RKCS1(R5)  ;TEST FOR CONT RDY
 6527  034536  001004                          BNE     2$              ;BR IF THERE
 6528  034540  005337  007412                  DEC     TEMP1
 6529  034544  001371                          BNE     1$
 6530  034546  104117                          ERROR   117             ;NO RDY AFTER SEL DRV CMD
 6531
 6532  034550  016537  000034  001356  2$:     MOV     RKMR2(R5),CYLDIF
 6533  034556  043737  015466  001356          BIC     MASK1,CYLDIF    ;GET CYL DIFF ONLY (NO SHIFTING)
 6534  034564  012637  007412                  MOV     (SP)+,TEMP1     ;RESTORE TEMP1
 6535  034570  000207                          RTS     PC
 6536                                    ;
 6537                                    ;READ THE CYL ADDR IN RKMR3, RIGHT JUSTIFY IT & STORE IT IN  'CYLADD'
 6538                                    ;
 6539  034572  012765  000002  000026  RDCYLA: MOV     #2,RKMR1(R5)    ;WORD 2
 6540  034600  004737  033722                  JSR     PC,GSTAT
 6541  034604  013737  007404  001360          MOV     HMR3,CYLADD
 6542  034612  043737  015466  001360          BIC     MASK1,CYLADD
 6543  034620  006237  001360                  ASR     CYLADD          ;RIGHT JUSTIFY
 6544  034624  006237  001360                  ASR     CYLADD          ;CYL ADDR
 6545  034630  006237  001360                  ASR     CYLADD          ;INFO
 6546  034634  006237  001360                  ASR     CYLADD
 6547  034640  000207                          RTS     PC
 6548                                    ;
 6549                                    ;READ THE CYL DIFF/OFFSET IN H.A2, RIGHT JUSTIFY IT & STORE IT IN 'CYLDIF'
 6550                                    ;
 6551  034642  013737  007434  001356  RCYLD:  MOV     H.A2,CYLDIF
 6552  034650  043737  015466  001356          BIC     MASK1,CYLDIF  ;CLEAR UNWANTED INFO
 6553  034656  006237  001356                  ASR     CYLDIF  ;RIGHT JUSTIFY
 6554  034662  006237  001356                  ASR     CYLDIF
 6555  034666  006237  001356                  ASR     CYLDIF
 6556  034672  006237  001356                  ASR     CYLDIF
 6557  034676  023737  001356  015464          CMP     CYLDIF,MASK     ;CHK TO SEE IF RET IN COMPL. FORM
 6558  034704  001002                          BNE     1$              ;BR IF NO
```

```
                                        CLR     CYLDIF          ;ELSE CLEAR
                                1$:     RTS     PC
                                        ;
                                        ;READ THE CYL ADDR IN H.B2, RIGHT JUSTIFY IT & STORE IT IN 'CYLADD'
                                        ;
                                RCYLA:  MOV     H.B2,CYLADD
                                        BIC     MASK1,CYLADD    ;CLEAR UNWANTED INFO
                                        ASR     CYLADD          ;RIGHT JUSTIFY
                                        ASR     CYLADD
                                        ASR     CYLADD
                                        ASR     CYLADD
                                        RTS     PC
                                        ;
                                        ;READ THE SECTOR COUNT IN H.B3, RIGHT JUSTIFY IT & STORE IT IN 'SECTOR'
                                        ;
                                RSEC:   MOV     H.B3,SECTOR
                                        BIC     #^C<M.SECT>,SECTOR  ;CLEAR UNWANTED INFO
                                        ASR     SECTOR          ;RIGHT JUSTIFY
                                        ASR     SECTOR
                                        ASR     SECTOR
                                        ASR     SECTOR
                                        RTS     PC
                                        ;
                                        ;READ THE HEAD ADDR IN H.B3, RIGHT IT & STORE IT IN 'HEADA'
                                        ;
                                RHEAD:  MOV     H.B3,HEADA
                                        BIC     #^C<M.HEAD>,HEADA  ;CLEAR UNWANTED INFO
                                        ASR     HEADA           ;RIGHT JUSTIFY IT
                                        SWAB    HEADA
                                        RTS     PC
                                        ;
                                        ;FIND SECTOR 23
                                        ;RETURN IF NOT FOUND
                                        ;RETURN +4 IF FOUND
                                        ;
                                FSEC23: MOV     T5000,TEMP1     ;SETUP TIMEOUT
                                1$:     JSR     PC,RDSEC        ;READ SECTOR
                                        CMP     SECTOR,#23      ;TEST FOR SECTOR 23(8)
                                        BNE     2$              ;BR IF NOT 23(8)

                                        JSR     PC,RDSEC
                                        CMP     SECTOR,#23
                                        BEQ     3$              ;BR IF READ SAME TWICE
                                        JSR     PC,RDSEC        ;ELSE TRY 1 MORE TIME
                                        CMP     SECTOR,#23
                                        BEQ     3$              ;BR IF 23(8)

                                2$:     DEC     TEMP1
                                        BNE     1$              ;TRY AGAIN
                                        RTS     PC

                                3$:     ADD     #4,(SP)         ;SKIP OVER ERROR
                                        RTS     PC
                                        ;
                                        ;ROUTINE TO FIND HEADS HOME IN RKMR2 WORD 1 BEFORE SPECIFIED DELAY
                                        ;ENTER WITH TIME IN SECONDS IN TEMP2
```

```
                                        ;RETURN IF NOT FOUND
                                        ;RETURN+2 IF FOUND - SKIP OVER ERROR
                                        ;
035126  012737  177777  007412  FHDHM:  MOV     #-1,TEMP1       ;ALL 1'S
035134  012765  000001  000026          MOV     #1,RKMR1(R5)    ;WORD 1
035142  004737  033722          1$:     JSR     PC,GSTAT
035146  032737  000040  007402          BIT     #D.HDHM,HMR2
035154  001007                          BNE     2$
035156  005337  007412                  DEC     TEMP1
035162  001367                          BNE     1$
035164  005337  007414                  DEC     TEMP2
035170  001356                          BNE     FHDHM
035172  000207                          RTS     PC
035174  062716  000002          2$:     ADD     #2,(SP)         ;SKIP OVER ERROR
035200  000207                          RTS     PC
                                        ;
                                        ;ROUTINE TO FIND LOAD HEADS IN RKMR2 WORD 1 BEFORE THE TIMEOUT
                                        ;RETURN IF NOT FOUND
                                        ;RETURN+2 IF FOUND: SKIP OVER ERROR
                                        ;
035202  012737  000372  007412  FLOAD:  MOV     #250.,TEMP1
035210  012765  000001  000026          MOV     #1,RKMR1(R5)    ;WORD 1
035216  004737  033722          1$:     JSR     PC,GSTAT
035222  032737  010000  007402          BIT     #D.LOAD,HMR2
035230  001004                          BNE     2$
035232  005337  007412                  DEC     TEMP1
035236  001367                          BNE     1$
035240  000207                          RTS     PC
035242  062716  000002          2$:     ADD     #2,(SP)         ;SKIP OVER ERROR
035246  000207                          RTS     PC
                                        ;
                                        ;FILL HEADER TABLE WITH 66 WORDS OF VALID HEADERS
                                        ;ENTER WITH CYL # IN 'CALADD'
                                        ;ENTER WITH HEAD # IN 'HEAD'
                                        ;ENTER WITH FORMAT IN 'FORMAT'
                                        ;
035250  010046          FHDTAB: MOV     R0,-(SP)        ;SAV R0
035252  010146                  MOV     R1,-(SP)        ;SAV R1
035254  012700  001522          MOV     #HDTAB,R0       ;HEADER WORD TABLE ADDR
035260  005001                  CLR     R1              ;SECTOR COUNTER
035262  013737  001460  001464  MOV     HEAD,HD1
035270  006337  001464          ASL     HD1
035274  006337  001464          ASL     HD1
035300  006337  001464          ASL     HD1
035304  006337  001464          ASL     HD1
035310  006337  001464          ASL     HD1             ;SETUP HEAD # FOR WORD 2 OF HEADER
035314  013737  001466  001470  MOV     FORMAT,FMT1
035322  000337  001470          SWAB    FMT1
035326  006337  001470          ASL     FMT1            ;SETUP FORMAT FOR WORD 2 OF HEADER
                                        ;
035332  013720  001362  1$:     MOV     CALADD,(R0)+    ;HEADER WORD 1-CYL ADDR
035336  010110                  MOV     R1,(R0)         ;HEADER WORD 2-SECTOR NO
035340  053710  001464          BIS     HD1,(R0)        ;              -HEAD NO
035344  053710  001470          BIS     FMT1,(R0)       ;              -FORMAT
035350  004737  035430          JSR     PC,SECFLG
                                        ;
```

```
6671  035354  013737  001362  007412          MOV     CALADD,TEMP1
6672  035362  011037  007414                  MOV     (R0),TEMP2
6673  035366  043737  001362  007414          BIC     CALADD,TEMP2
6674  035374  042037  007412                  BIC     (R0)+,TEMP1
6675  035400  053737  007412  007414          BIS     TEMP1,TEMP2
6676  035406  013720  007414                  MOV     TEMP2,(R0)+     ;HEADER WORD 3-HEADER CHECK
6677
6678  035412  005201                          INC     R1              ;SECTOR CTR
6679  035414  020127  000026                  CMP     R1,#22.         ;ALL 22 SECTORS DONE? (66 WORDS)
6680  035420  001344                          BNE     1$              ;BR IF NO
6681
6682  035422  012601                          MOV     (SP)+,R1        ;RESTOR R1
6683  035424  012600                          MOV     (SP)+,R0        ;RESTOR R0
6684  035426  000207                          RTS     PC
6685                                  ;
6686                                  ;THIS ROUTINE GETS INFORMATION FROM THE BAD SECTOR TABLE FILLED BY A PREVIOUS
6687                                  ;TEST & SETS BITS 14 & 15 APPROPRIATLY.
6688                                  ;
6689  035430  010246                  SECFLG: MOV     R2,-(SP)        ;SAVE R2
6690  035432  005737  001466                  TST     FORMAT
6691  035436  001016                          BNE     1$              ;BR IF 20 SECTOR FORMAT
6692  035440  012702  003346                  MOV     #BSE22H+8.,R2
6693  035444  004737  035530                  JSR     PC,FLGTST       ;GET HARDWARE DETECTED FLAG
6694  035450  052710  100000                  BIS     #BIT15,(R0)     ;RETURN HERE IF GOOD SECTOR
6695
6696  035454  012702  005346                  MOV     #BSE22S+8.,R2   ;ELSE RETURN HERE
6697  035460  004737  035530                  JSR     PC,FLGTST       ;GET SOFTWARE DETECTED FLAG
6698  035464  052710  040000                  BIS     #BIT14,(R0)     ;RETURN HERE IF GOOD SECTOR
6699
6700  035470  012602                          MOV     (SP)+,R2        ;ELSE RETURN HERE
6701  035472  000207                          RTS     PC
6702
6703  035474  012702  002346          1$:     MOV     #BSE20H+8.,R2
6704  035500  004737  035530                  JSR     PC,FLGTST       ;GET HARDWARE DETECTED FLAG
6705  035504  052710  100000                  BIS     #BIT15,(R0)     ;RETURN HERE IF GOOD SECTOR
6706
6707  035510  012702  004346                  MOV     #BSE20S+8.,R2
6708  035514  004737  035530                  JSR     PC,FLGTST       ;GET SOFTWARE DETECTED FLAG
6709  035520  052710  040000                  BIS     #BIT14,(R0)     ;RETURN HERE IF GOOD SECTOR
6710
6711  035524  012602                          MOV     (SP)+,R2        ;RESTORE R2
6712  035526  000207                          RTS     PC
6713
6714                                  ;
6715                                  ;THIS ROUTINE DOES THE ACTUAL SCANNING OF THE BAD SECTOR TABLES
6716                                  ;ENTER WITH THE ADDRESS OF TABLE (BSE22H, BSE22S, ETC.) IN TEMP1
6717                                  ;RETURN IF NO COMPARE
6718                                  ;RETURN+4 IF COMPARE
6719                                  ;
6720  035530  010346                  FLGTST: MOV     R3,-(SP)        ;SAVE R3
6721
6722  035532  021227  177777          1$:     CMP     (R2),#-1        ;SEE IF ALL 1'S
6723  035536  001002                          BNE     2$              ;BR IF NO
6724  035540  012603                          MOV     (SP)+,R3        ;RESTORE R3
6725  035542  000207                          RTS     PC
6726
```

```
6727  035544  022237  001362          2$:     CMP     (R2)+,CALADD    ;SEE IF=CYL # & ADR PTR TO TRK/SECTOR WORD
6728  035550  001403                          BEQ     3$
6729  035552  062702  000002                  ADD     #2,R2           ;GO TO NEXT CYL WORD IN TABLE
6730  035556  000765                          BR      1$
6731
6732  035560  013703  001460          3$:     MOV     HEAD,R3         ;GET HEAD # FROM FHDTAB ROUTINE
6733  035564  000303                          SWAB    R3
6734  035566  050103                          BIS     R1,R3           ;ADD SECTOR # FROM FHDTAB ROUTINE
6735  035570  022203                          CMP     (R2)+,R3        ;SEE IF SECTOR/HEAD COMPARE
6736                                                                  ;& INCR PTR TO NEXT CYL WORD
6737  035572  001401                          BEQ     4$              ;BR IF COMPARE
6738  035574  000756                          BR      1$              ;ELSE TRY NEXT CYL
6739
6740  035576  012603          4$:     MOV     (SP)+,R3        ;RESTORE R3
6741  035600  062716  000004                  ADD     #4,(SP)         ;INCREMENT RET ADDR
6742  035604  000207                          RTS     PC
6743
6744                                  ;
6745                                  ;THIS ROUTINE SORTS THE RHTAB TABLE FROM WHATEVER SECTOR IT BEGINS
6746                                  ;WITH AND RE-WRITES THE INFO IN SRTTAB TABLE TO BEGIN WITH SECTOR 0
6747                                  ;
6748  035606  010046          SORT:   MOV     R0,-(SP)        ;SAVE R0
6749  035610  010146                  MOV     R1,-(SP)        ;SAVE R1
6750  035612  004737  034332                  JSR     PC,RDSEC
6751  035616  062737  000001  001402          ADD     #1,SECTOR
6752  035624  004737  035714                  JSR     PC,MULT6        ;MULT SECTOR BY 6
6753
6754  035630  012700  000204                  MOV     #132.,R0
6755  035634  163700  001402                  SUB     SECTOR,R0       ;R0-SECTOR TO R0 = INDEX
6756  035640  010037  001402                  MOV     R0,SECTOR
6757  035644  062737  001726  001402          ADD     #RHTAB,SECTOR   ;SAVE INDEX
6758
6759  035652  062700  001726                  ADD     #RHTAB,R0       ;INDEX TO BOT HALF OF RHTAB
6760  035656  012701  002132                  MOV     #SRTTAB,R1      ;INDEX TO TOP HALF OF SRTTAB
6761
6762  035662  012021          1$:     MOV     (R0)+,(R1)+     ;PUT BOTTOM OF RHTAB TO TOP OF SRTTAB
6763  035664  020027  002132                  CMP     R0,#RHTAB+132.
6764  035670  001374                          BNE     1$
6765
6766  035672  012700  001726                  MOV     #RHTAB,R0       ;PUT TOP OF RHTAB TO BOT OF SRTTAB
6767  035676  012021          2$:     MOV     (R0)+,(R1)+
6768  035700  020037  001402                  CMP     R0,SECTOR
6769  035704  001374                          BNE     2$
6770
6771  035706  012601                  MOV     (SP)+,R1        ;RESTOR R1
6772  035710  012600                  MOV     (SP)+,R0        ;RESTOR R0
6773  035712  000207                  RTS     PC
6774
6775                                  ;
6776                                  ;MULT BY 6. ENTER WITH DESIRED # IN 'SECTOR'
6777                                  ;
6778  035714  006337  001402          MULT6:  ASL     SECTOR          ;2 X SECTOR
6779  035720  013746  001402                  MOV     SECTOR,-(SP)
6780  035724  006337  001402                  ASL     SECTOR          ;4 X SECTOR
6781  035730  062637  001402                  ADD     (SP)+,SECTOR    ;(4 X S)+(2 X S) = 6 X SECTOR
6782  035734  000207                          RTS     PC
```

```
6783
6784                                    ;
6785                                    ;THIS ROUTINE IS ENTERED ONLY IF THERE IS A BSE ERROR AFTER A WRITE DATA
6786                                    ;CMD.  IT VERIFIES THAT THE BAD SECTOR IS LISTED IN THE BSE INFORMATION
6787                                    ;CYLINDER AT CYL 410, TRACK 2.
6788                                    ;
6789                                    ;RETURN IF SECTOR NOT LISTED IN BSE TABLE, ERROR CONDITION.
6790                                    ;RETURN+2 IF LISTED, SKIP OVER ERROR
6791                                    ;
6792  035736  010446                    TRUERR: MOV     R4,-(SP)                ;SAVE R4
6793
6794  035740  032737  010000  007354            BIT     #CFMT,HCS1      ;CHECK FORMAT
6795  035746  001014                            BNE     2$              ;BR FOR 20 SECTOR FORMAT
6796
6797  035750  012704  003346                    MOV     #BSE22H+8.,R4
6798  035754  004737  036036                    JSR     PC,TERR1        ;SEE IF ON HARDWARE DETECTED TABLE
6799  035760  000422                            BR      3$              ;RETURN HERE IF YES
6800
6801  035762  012704  005346                    MOV     #BSE22S+8.,R4   ;ELSE RETURN HERE
6802  035766  004737  036036                    JSR     PC,TERR1        ;& SEE IF ON SOFTWARE DETECTED TABLE
6803  035772  000415                            BR      3$              ;RETURN HERE IF YES
6804
6805  035774  012604                    1$:     MOV     (SP)+,R4        ;RESTORE R4
6806  035776  000207                            RTS     PC              ;RETURN WITHOUT JUMPING OVER ERROR
6807
6808  036000  012704  002346            2$:     MOV     #BSE20H+8.,R4
6809  036004  004737  036036                    JSR     PC,TERR1        ;SEE IF ON HARDWARE DETECTED TABLE
6810  036010  000406                            BR      3$              ;RETURN HERE IF YES
```

```
 6811
 6812  036012  012704  004346                MOV     #BSE20S+8.,R4   ;ELSE RETURN HERE
 6813  036016  004737  036036                JSR     PC,TERR1        ;SEE IF ON SOFTWARE DETECTED TABLE
 6814  036022  000401                        BR      3$              ;RETURN HERE IF YES
 6815  036024  000763                        BR      1$              ;RETURN HERE IF NO
 6816
 6817  036026  012604              3$:       MOV     (SP)+,R4        ;RESTORE R4
 6818  036030  062716  000002                ADD     #2,(SP)         ;SKIP OVER ERROR ON RETURN
 6819  036034  000207                        RTS     PC
 6820
```

```
                                        ;
                                        ;THIS ROUTINE DOES THE ACTUAL COMPARING OF CYLINDER, HEAD & TRACK AGAINST
                                        ;THE BSE TABLE FOR THE ABOVE SUBROUTINE.
                                        ;RETURN IF FOUND ON TABLE
                                        ;RETURN+2 IF NOT FOUND
                                        ;
036036  021427  177777          TERR1:  CMP     (R4),#-1        ;SEE IF ALL 1'S
```

```
 6828  036042  001405                    BEQ     1$              ;BR IF YES, NOT ON TABLE
 6829  036044  022437  007374            CMP     (R4)+,HDC       ;SEE IF CYL MATCH
 6830  036050  001405                    BEQ     2$              ;BR IF YES
 6831  036052  005724                    TST     (R4)+           ;ELSE ADV TO NEXT CYL WORD
 6832  036054  000770                    BR      TERR1           ;& TRY AGAIN.
 6833
 6834  036056  062716  000002    1$:     ADD     #2,(SP)
 6835  036062  000207                    RTS     PC
 6836
 6837  036064  022437  007364    2$:     CMP     (R4)+,HDA       ;SEE IF SECTOR & TRACK MATCH
 6838  036070  001401                    BEQ     3$              ;BR IF YES
 6839  036072  000761                    BR      TERR1           ;OR TRY AGAIN
 6840
 6841  036074  000207    3$:     RTS     PC
 6842                    ;
 6843                    ;ROUTINE TO TURN L OR P CLOCK INTERRUPT ON
 6844                    ;
 6845  036076  005037  001372    CLKON:  CLR     TIMUP
 6846  036102  005737  007520            TST     PCLKF
```

```
                        BNE     1$              ;BRANCH IF P-CLOCK PRESENT
                        MOV     #100,@LKS       ;L-CLOCK, ENABLE INT
                        RTS     PC
                1$:     MOV     #-1,@PKSB       ;P-CLOCK, ALL 1'S
                        MOV     #135,@PKS       ;ENABLE INT, CT UP, REP INT
                        RTS     PC              ;LINE FREQ & RUN
                ;
                ;KW11-L & KW11-P INTERRUPT HANDLER
                ;
                CLOCK:  CLR     TIMUP
                        DEC     COUNT
                        BNE     1$
                        MOV     HZ,COUNT
                        DEC     SEC
                        BNE     1$
                        INC     TIMUP           ;SORRY, TIME IS UP
                1$:     RTI
                ;
                ;ROUTINE TO TURN L OR P CLOCK INTERRUPT OFF
                ;
                CLKOF:  TST     PCLKF
                        BNE     1$              ;BRACH IF P-CLOCK PRESENT
                        CLR     @LKS            ;L-CLOCK, CLEAR INTERRUPT
                        RTS     PC
                1$:     CLR     @PKS            ;P-CLOCK, CLEAR INTERRUPT
                        RTS     PC
                ;
                ;
                ;THIS ROUTINE GENERATES PARITY FOR THE EXPECTED MESSAGE
                ;ENTER WITH THE EXPECTED WORD IN TEMP1
                ; TEMP1 IS ROTATED LEFT 17 TIMES. EACH TIME THE CARRY BIT IS SET,
                ;R1 IS INCREMENTED. AT THE END OF 17 ROTATES ( TEMP1 BACK TO ORIG),
                ;R1 BIT 0 IS EXAMINED. IF IT IS SET, INDICATING AN ODD # OF 1'S,
                ; THE PARITY BIT IS NOT SE1 IN B .
                ;IF IT IS NOT SET, INDICATING AN EVEN # OF 1'S ,THE PARITY BIT IS
                ;SET IN TEMP1
                ;
                SBPAR:  MOV     R0,-(SP)        ;SAVE R0
                        MOV     R1,-(SP)        ;SAVE R1
                        MOV     #17.,R0         ;SHIFT COUNTER
                        CLR     R1              ;COUNT # OF 1'S IN TEMP1
                        CLC                     ;CLEAR CARRY
                1$:     ROL     TEMP1
                        BCC     2$              ;BR IF CARRY CLEAR
                        INC     R1              ;COUNT # OF 1'S
                2$:     DEC     R0              ;SHIFT COUNTER
                        BNE     1$
                        BIT     #BIT0,R1
                        BNE     3$              ;BR IF ODD # IN R0
                        BIS     #M.PAR,TEMP1    ;SET PARITY BIT
                3$:     MOV     (SP)+,R1        ;RESTORE R1
                        MOV     (SP)+,R0        ;RESTORE R0
                        RTS     PC
```

```
                                        ;
                                        ;ROUTINE TO ENABLE LOOPING ON INTERMITTANT ERRORS
                                        ;WHEN $LPERR SET BY OTHER THAN SCOPE ROUTINE
                                        ; IE: MY LOOP MACRO
                                        ;
036266  032777  001000  142644  $COP1$: BIT     #SW9,@SWR       ;LOOP ON ERROR?
036274  001406                          BEQ     1$              ;BR IF NO
036276  105737  001103                  TSTB    $ERFLG          ;HAD ERROR?
036302  001403                          BEQ     1$              ;BR IF NO
036304  013716  001110                  MOV     $LPERR,(SP)
036310  000002                          RTI
                                        ;
036312  011637  001110          1$:     MOV     (SP),$LPERR     ;SET LOOP ADDR FOR TIGHT SCOPE LOOP
036316  000002                          RTI
                                        ;
                                        ;THIS ROUTINE IS ENTERED BY TYPING A CONTRCL-C.
                                        ;IT IS USED TO ALLOW THE OPERATOR TO HLAT THE CPU WHILE INSURING
                                        ;THAT HEADS ARE LOADED & FORMATTING IS VALID BEFORE ACTUALLY HALTING
                                        ;THE CPU.
                                        ;
036320  022626                  STOP:   CMP     (SP)+,(SP)+     ;RESTORE STACK FROM INTERRUPT
                                        ;
036322  004737  034252                  JSR     PC,SUBCLR
036326  104024                          ERROR   24              ;CERR AFTER
                                        ;
036330  005737  007336                  TST     UNLD            ;SEE IF HEADS UNLOADED
036334  001431                          BEQ     3$              ;BR IF NO
036336  005737  000042                  TST     42              ;SEE IF MANUAL OR AUTO MODE
036342  001403                          BEQ     1$              ;BR IF MANUAL MODE
036344  104401  046167                  TYPE    ,MSG74          ;PGM ABORT PENDING
036350  000402                          BR      2$
036352  104401  046235          1$:     TYPE    ,MSG75          ;HALT PENDING
036356                          2$:
                                        ;
036356  004737  034252                  JSR     PC,SUBCLR
036362  104024                          ERROR   24              ;CERR AFTER SCLR
                                        ;
036364  012737  000011  007354          MOV     #SRTSPL,HCS1
036372  004737  032262                  JSR     PC,DOCMD        ;DO START SPINDLE CMD & GET CONTR RDY
036376  104121                          ERROR   121             ;RDY NOT SET AFTER ST SPIN CMD.
                                        ;
036400  013737  001414  007414          MOV     T100,TEMP2      ;SETUP TIMEOUT
036406  004737  032672                  JSR     PC,FATT1        ;FIND ATTN
036412  104074                          ERROR   74              ;NO ATTN AFTER ST SPIN CMD.
                                        ;
036414  005037  007336                  CLR     UNLD
                                        ;
                                        ;
036420  005737  007340          3$:     TST     BADHDR          ;SEE IF HEADERS VALID
036424  001460                          BEQ     4$              ;BR IF YES
036426  005237  007342                  INC     HPEND
                                        ;
036432  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
036440  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
036446  012737  000013  007354          MOV     #RECAL,HCS1
036454  004737  032262                  JSR     PC,DOCMD        ;DO RECAL CMD & GET CONTR RDY
```

```
 6959  036460  104124                          ERROR   124             ;RDY NOT SET AFTER RECAL CMD
 6960
 6961  036462  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
 6962  036470  004737  033722                  JSR     PC,GSTAT
 6963  036474  032737  020000  007402          BIT     #D.RTZ,HMR2
 6964  036502  001001                          BNE     64$
 6965  036504  104244                          ERROR   244             ;RTZ NOT SET DURING RECAL CMD
 6966  036506  013737  001406  007414  64$:    MOV     T10,TEMP2       ;SETUP TIMEOUT
 6967  036514  004737  032672                  JSR     PC,FATT1        ;FIND ATTN
 6968  036520  104055                          ERROR   55              ;NO ATTN AFTER RECAL CMD
 6969
 6970  036522  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
 6971  036530  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
 6972  036536  012737  000005  007354          MOV     #CLEAR,HCS1
 6973  036544  004737  032262                  JSR     PC,DOCMD        ;DO DRIVE CLEAR CMD & GET CONTR RDY
 6974  036550  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
 6975  036552  004737  032640                  JSR     PC,TSTATN       ;TEST FOR ATTN
 6976  036556  000401                          BR      65$
 6977  036560  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
 6978  036562                          65$:
 6979
 6980
 6981  036562   00137  031220                  JMP     FORM            ;WRITE VALID FORMATS
 6982
 6983  036566  005737  000042          4$:     TST     42              ;SEE IF MANUAL OR AUTO MODE
 6984  036572  001410                          BEQ     5$              ;BR IF MANUAL MODE
 6985  036574  104401  046272                  TYPE    ,MSG76          ;PGM ABORTED
 6986  036600  005037  031576                  CLR     $EOPCT          ;SET UP EOP TO EXIT TO MONITOR
 6987  036604  005037  001176                  CLR     $ESCAPE
 6988  036610  000137  031550                  JMP     $EOP1           ;ABORT PGM
 6989
 6990  036614  104401  046314          5$:     TYPE    ,MSG77          ;CPU HALTED
 6991  036620  000000                          HALT
 6992  036622  000137  013372                  JMP     ST5             ;START OVER IF CONTINUE PRESSED
 6993
 6994
 6995
 6996                                  .SBTTL  UNEXPECTED TIMEOUT HANDLER
 6997
 6998                                  ;
 6999                                  ;THIS ROUTINE IS ENTERED IF THERE IS
 7000                                  ;       A.  NON EXISTANT MEMORY (NO SSYN)
 7001                                  ;       B.  BOUNDRY ERROR
 7002                                  ;       C.  STACK OVERFLOW
 7003                                  ;
 7004
 7005  036626  011600                  BADTMO: MOV     (SP),R0         ;SAVE PC WHERE TIMEOUT OCCURRED.
 7006  036630  005740                          TST     -(R0)           ;GET PC BEFORE UPDATE
 7007  036632  032777  020000  142300          BIT     #SW13,@SWR      ;INHIBIT ERR TYPOUT?
 7008  036640  001005                          BNE     1$              ;YES, DON'T TYPE
 7009  036642  104401  046475                  TYPE    ,EM3            ;ABORT TESTS,UNEXP T.O. @ PC=
 7010  036646  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
 7011                                                                  ;;TYPE PC
 7012  036650  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
 7013  036652     006                          .BYTE   6               ;;TYPE 6 DIGIT(S)
 7014  036653     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
```

```
7015  036654  032777  001000  142256  1$:     BIT     #SW9,@SWR       ;LOOP ON ERROR?
7016  036662  001403                          BEQ     2$              ;NO, BRANCH
7017  036664  022626                          CMP     (SP)+,(SP)+     ;YES, RESTORE STACK
7018  036666  000177  142214                  JMP     @$LPADR         ;GO TO STARTING ADDR OF TEST
7019                                                                  ;THAT GAVE BAD TIMEOUT
7020  036672  032777  040000  142240  2$:     BIT     #SW14,@SWR      ;LOOP ON TEST?
7021  036700  001401                          BEQ     3$              ;NO BRANCH
7022  036702  000002                          RTI                     ;YES
7023
7024  036704  000000                  3$:     HALT                    ;UNEXPECTED TIME OUT OCCURRED
7025                                                                  ;AS INDICATED. YOU CAN LOOP ON
7026                                                                  ;ERROR, LOOP ON TEST OR INHIBIT
7027                                                                  ;ERROR TYPEOUT BY SETTING THOSE
7028                                                                  ;SWITCHES.
7029                                                                  ;
7030  036706  022626                          CMP     (SP)+,(SP)+     ;RESTORE STACK
7031  036710  000137  031550                  JMP     $EOP1           ;ABORT TESTS
7032
7033                          .SBTTL  MEMORY CHECK ENABLE TRAP
7034
7035  036714  012737  036730  001176  MEMERR: MOV     #1$,$ESCAPE
7036  036722  011637  001334                  MOV     (SP),TRAPPC     ;STORE PC
7037  036726  104041                          ERROR   41              ;UNEXP MEM PARITY TRAP
7038  036730  005037  001176          1$:     CLR     $ESCAPE
7039  036734  032777  001000  142176          BIT     #SW9,@SWR       ;CHECK IF LOOP ON ERROR
7040  036742  001001                          BNE     2$              ;YES, FORCE STACK AND TRY AGAIN
7041  036744  000002                          RTI                     ;ELSE RETURN
7042
7043  036746  012706  001100          2$:     MOV     #STACK,SP       ;INIT STACK
7044  036752  000177  142132                  JMP     @$LPERR         ;LOOP ON ERROR
7045
7046
7047                          .SBTTL  RK06 INTERRUPT HANDLER
7048
7049  036756  000240          INTER:  NOP
7050  036760  000240                  NOP
7051  036762  000240                  NOP
7052  036764  011600                  MOV     (SP),R0         ;SAVE PC WHERE INT OCCURRED.
7053  036766  005740                  TST     -(R0)           ;GET PC BEFORE UPDATE.
7054  036770  104401  044742          TYPE    ,MSG6           ;INT AT PC=
7055  036774  010046                  MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
7056                                                          ;;TYPE PC
7057  036776  104403                  TYPOS                   ;;GO TYPE--OCTAL ASCII
7058  037000     006                  .BYTE   6               ;;TYPE 6 DIGIT(S)
7059  037001     000                  .BYTE   0               ;;SUPPRESS LEADING ZEROS
7060  037002  000000                  HALT
7061  037004  000240                  NOP
7062  037006  000240                  NOP
7063  037010  000002                  RTI
7064
7065                          .SBTTL  POWER DOWN AND UP ROUTINES
7066
7067                          ;POWER DOWN ROUTINE
7068
7069  037012  012737  037024  000024  $PWRDN: MOV     #$PWRUP,PWRVEC  ;SET UP VECTOR
7070  037020  000000                  HALT
```

```
7071  037022  000776                        BR      .-2             ;HANG UP.
7072
7073                                  ;POWER UP ROUTINE
7074
7075  037024  005037  037076          $PWRUP: CLR     $PWRCT          ;WAIT LOOP FOR TTY
7076  037030  005237  037076          1$:     INC     $PWRCT          ;WAIT FOR THE INCR
7077  037034  001375                          BNE     1$              ;OF WORD
7078  037036  012737  037012  000024          MOV     #$PWRDN,PWRVEC  ;SET POWER DOWN VECTOR
7079  037044  012737  000340  000026          MOV     #PR7,PWRVEC+2   ;PRIORITY 7
7080  037052  012737  000340  000036          MOV     #PR7,TRAPVEC+2  ;LOCKOUT ALL INTERRUPTS FOR TRAPS
7081  037060  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK
7082  037064  104401  045132                  TYPE    ,MSG11          ;REPORT POWER FAIL
7083  037070  000005                          RESET
7084  037072  000137  015472                  JMP     PFSRT
7085
7086  037076  000000                  $PWRCT: 0                       ;WAIT COUNT FOR TTY
7087
7088                                  ;
7089                                  ;DIVISION UTILITY ROUTINE
7090                                  ;
7091                                  ;R0-R1-R2-R3=DIVIDEND
7092                                  ;R4-R5=DIVISOR
7093                                  ;R0-R1=REMAINDER AFTER DIVISION
7094                                  ;R2-R3=QUOTIENT AFTER DIVISION
7095                                  ;ENTER WITH JSR PC,M.DPID
7096                                  ;
7097  037100  012746  000040          M.DPID: MOV     #40,-(SP)       ;COUNTER FOR DIVISION CYCLES
7098  037104  010446                          MOV     R4,-(SP)        ;HI ORDER
7099  037106  010546                          MOV     R5,-(SP)        ;LO ORDER TO THE STACK
7100  037110  005466  000002                  NEG     2(SP)           ;FORM NEGATIVE
7101  037114  005416                          NEG     @SP             ;VERSION OF DIVISOR
7102  037116  005666  000002                  SBC     2(SP)
7103  037122  061601                          ADD     @SP,R1
7104  037124  005500                          ADC     R0              ;PERFORM INIT SUBT.
7105  037126  066600  000002                  ADD     2(SP),R0
7106  037132  103445                          BCS     M.DP50          ;IF CARRY THEN OVERFLOW HAS OCCURRED
7107  037134  005046                          CLR     -(SP)           ;THIS IS A LONGER LASTING CARRY BIT
7108  037136  006103          M.DP40: ROL     R3
7109  037140  006102                  ROL     R2
7110  037142  006101                  ROL     R1
7111  037144  006100                  ROL     P0
7112  037146  005716                  TST     @SP             ;TEST CARRY INDICATOR
7113  037150  001410                  BEQ     M.DP41          ;IF TO CARRY THEN ADD, ELSE SUBT.
7114  037152  005016                  CLR     @SP             ;CLEAR UP FOR NEXT TIME
7115  037154  066601  000002          ADD     2(SP),R1
7116  037160  005500                  ADC     R0              ;ADD -(DIVISOR)
7117  037162  005516                  ADC     @SP             ;SET CARRY
7118  037164  066600  000004          ADD     4(SP),R0
7119  037170  000404                  BR      M.DP42
7120
7121  037172  060501          M.DP41: ADD     R5,R1
7122  037174  005500                  ADC     R0              ;ADD +(DIVISOR)
7123  037176  005516                  ADC     @SP             ;SET CARRY
7124  037200  060400                  ADD     R4,R0
7125  037202  005516          M.DP42: ADC     @SP             ;SET CARRY
7126  037204  005716                  TST     @SP             ;TEST THE UPDATE INDICATOR
```

```
 7127  037206  001401                      BEQ     .+4             ;IF 0,FORGET IT
 7128  037210  005203                      INC     R3              ;NO CARRY POSSIBLE HERE
 7129  037212  005366  000006              DEC     6(SP)           ;DECREMENT CTR
 7130  037216  003347                      BGT     M.DP40          ;BR IF MORE TO DO
 7131  037220  006003                      ROR     R3
 7132  037222  103404                      BCS     M.DP44
 7133  037224  060501                      ADD     R5,R1
 7134  037226  005500                      ADC     R0
 7135  037230  060400                      ADD     R4,R0
 7136  037232  000241                      CLC
 7137
 7138  037234  006103              M.DP44: ROL     R3
 7139  037236  062706  000010              ADD     #10,SP          ;ADJUST STACK BY 4 WORDS
 7140  037242  000242                      CLV
 7141  037244  000207                      RTS     PC
 7142
 7143  037246  062706  000006      M.DP50: ADD     #6,SP
 7144  037252  000262                      SEV
 7145  037254  000207                      RTS     PC
 7146
```

```
                        SBTTL  SCOPE HANDLER ROUTINE

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

037256                  $SCOPE:
037256  104407                  CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
037260  032777  040000  141652  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
037266  001114                  BNE     $OVER           ;;YES IF SW14=1
                        ;#####START OF CODE FOR THE XOR TESTER#####
037270  000416          $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                        ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
037272  013746  000004          MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
037276  012737  037316  000004  MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
037304  005737  177060          TST     @#177060        ;;TIME OUT ON XOR?
037310  012637  000004          MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
037314  000463                  BR      $SVLAD          ;;GO TO THE NEXT TEST
037316  022626          5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
037320  012637  000004          MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
037324  000423                  BR      7$              ;;LOOP ON THE PRESENT TEST
037326                  6$:;#####END OF CODE FOR THE XOR TESTER#####
037326  032777  000400  141604  BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
037334  001404                  BEQ     2$              ;;BR IF NO
037336  127737  141576  001102  CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
037344  001465                  BEQ     $OVER           ;;BR IF YES
037346  105737  001103  2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
037352  001421                  BEQ     3$              ;;BR IF NO
037354  123737  001115  001103  CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
037362  101015                  BHI     3$              ;;BR IF NO
037364  032777  001000  141546  BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
037372  001404                  BEQ     4$              ;;BR IF NO
037374  013737  001110  001106  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
037402  000446                  BR      $OVER
037404  105037  001103  4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
037410  005037  001174          CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
037414  000415                  BR      1$              ;;ESCAPE TO THE NEXT TEST
037416  032777  004000  141514  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
037424  001011                  BNE     1$              ;;BR IF YES
037426  005737  001216          TST     $PASS           ;;IF FIRST PASS OF PROGRAM
037432  001406                  BEQ     1$              ;;      INHIBIT ITERATIONS
037434  005237  001104          INC     $ICNT           ;;INCREMENT ITERATION COUNT
037440  023737  001174  001104  CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
037446  002024                  BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
037450  012737  000001  001104  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
037456  013737  037534  001174          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
037464  105237  001102  $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
037470  113737  001102  001214          MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
```

```
 7203  037476  011637  001106                 MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
 7204  037502  011637  001110                 MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
 7205  037506  005037  001176                 CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
 7206  037512  112737  000001  001115         MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
 7207  037520  013777  001102  141414 $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
 7208  037526  013716  001106                 MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
 7209  037532  000002                         RTI                     ;;FIXES PS
 7210  037534  003720                 $MXCNT: 2000.                   ;;MAX. NUMBER OF ITERATIONS
 7211                                  .SBTTL  ERROR HANDLER ROUTINE
 7212
 7213                                  ;;**************************************************************************
 7214                                  ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
 7215                                  ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
 7216                                  ;*AND GO TO TYPERR ON ERROR
 7217                                  ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
 7218                                  ;*SW15=1         HALT ON ERROR
 7219                                  ;*SW13=1         INHIBIT ERROR TYPEOUTS
 7220                                  ;*SW10=1         BELL ON ERROR
 7221                                  ;*SW09=1         LOOP ON ERROR
 7222                                  ;*CALL
 7223                                  ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
 7224
 7225  037536                 $ERROR:
 7226  037536  104407                         CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
 7227  037540  105237  001103         7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
 7228  037544  001775                         BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
 7229  037546  013777  001102  141366         MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
 7230  037554  032777  002000  141356         BIT     #BIT10,@SWR     ;;BELL ON ERROR?
 7231  037562  001402                         BEQ     1$              ;;NO - SKIP
 7232  037564  104401  001200                 TYPE    ,$BELL          ;;RING BELL
 7233  037570  005237  001112         1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
 7234  037574  011637  001116                 MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
 7235  037600  162737  000002  001116         SUB     #2,$ERRPC
 7236  037606  117737  141304  001114         MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
 7237  037614  032777  020000  141316         BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
 7238  037622  001004                         BNE     20$             ;;SKIP TYPEOUTS
 7239  037624  004737  055332                 JSR     PC,TYPERR       ;;GO TO USER ERROR ROUTINE
 7240  037630  104401  001205                 TYPE    ,$CRLF
 7241  037634                         20$:
 7242  037634  122737  000001  001230         CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
 7243  037642  001007                         BNE     2$              ;;NO,SKIP APT ERROR REPORT
 7244  037644  113737  001114  037656         MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
 7245  037652  004737  040534                 JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
 7246  037656     000                 21$:    .BYTE   0
 7247  037657     000                         .BYTE   0
 7248  037660  000777                 22$:    BR      22$             ;;APT ERROR LOOP
 7249  037662  005777  141252         2$:     TST     @SWR            ;;HALT ON ERROR
 7250  037666  100002                         BPL     3$              ;;SKIP IF CONTINUE
 7251  037670  000000                         HALT                    ;;HALT ON ERROR!
 7252  037672  104407                         CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
 7253  037674  032777  001000  141236 3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
 7254  037702  001402                         BEQ     4$              ;;BR IF NO
 7255  037704  013716  001110                 MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
 7256  037710  005737  001176         4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
 7257  037714  001402                         BEQ     5$              ;;BR IF NONE
 7258  037716  013716  001176                 MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
```

```
 7259  037722                          5$:
 7260  037722  022737  031636  000042          CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
 7261  037730  001001                          BNE     6$              ;;BRANCH IF NO
 7262  037732  000000                          HALT                    ;;YES
 7263  037734                          6$:
 7264  037734  000002                          RTI                     ;;RETURN
 7265                                  .SBTTL  TYPE ROUTINE
 7266
 7267                                  ;;**********************************************************************
 7268                                  ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
 7269                                  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
 7270                                  ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
 7271                                  ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
 7272                                  ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
 7273                                  ;*
 7274                                  ;*CALL:
 7275                                  ;*1) USING A TRAP INSTRUCTION
 7276                                  ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
 7277                                  ;*OR
 7278                                  ;*      TYPE
 7279                                  ;*      MESADR
 7280                                  ;*
 7281
 7282  037736  105737  001157          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
 7283  037742  100002                          BPL     1$              ;;BR IF YES
 7284  037744  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
 7285  037746  000430                          BR      3$              ;;LEAVE
 7286  037750  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
 7287  037752  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
 7288  037756  122737  000001  001230          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
 7289  037764  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
 7290  037766  132737  000100  001231          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
 7291  037774  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
 7292  037776  010037  040006                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
 7293  040002  004737  040524                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
 7294  040006  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
 7295  040010  132737  000040  001231  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
 7296  040016  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
 7297  040020  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
 7298  040022  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
 7299  040024  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
 7300  040026  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
 7301  040030  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
 7302  040034  000002                          RTI                     ;;RETURN
 7303  040036  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
 7304  040042  001430                          BEQ     8$
 7305  040044  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
 7306  040050  001006                          BNE     5$
 7307  040052  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
 7308  040054  104401                          TYPE                    ;;TYPE A CR AND LF
 7309  040056  001205                          $CRLF
 7310  040060  105037  040266                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
 7311  040064  000755                          BR      2$              ;;GET NEXT CHARACTER
 7312  040066  004737  040150          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
 7313  040072  123726  001156          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
 7314  040076  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
```

```
7315  040100  013746  001154            MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
7316                                                            ;;AND THE NULL CHAR.
7317  040104  105366  000001      7$:   DFCB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
7318  040110  002770                    BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
7319  040112  004737  040150            JSR     PC,$TYPEC       ;;GO TYPE A NULL
7320  040116  105337  040266            DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
7321  040122  000770                    BR      7$              ;;LOOP
7322
7323                                    ;HORIZONTAL TAB PROCESSOR
7324
7325  040124  112716  000040      8$:   MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
7326  040130  004737  040150      9$:   JSR     PC,$TYPEC       ;;TYPE A SPACE
7327  040134  132737  000007  040266    BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
7328  040142  001372                    BNE     9$              ;;TAB STOP
7329  040144  005726                    TST     (SP)+           ;;POP SPACE OFF STACK
7330  040146  000724                    BR      2$              ;;GET NEXT CHARACTER
7331  040150                    $TYPEC:
7332  040150  105777  140770            TSTB    @$TKS           ;;CHAR IN KYBD BUFFER?          ;MJD001
7333  040154  100022                    BPL     10$             ;;BR IF NOT                     ;MJD001
7334  040156  017746  140764            MOV     @$TKB,-(SP)     ;;GET CHAR                      ;MJD001
7335  040162  042716  177600            BIC     #177600,(SP)    ;;STRIP EXTRANEOUS BITS         ;MJD001
7336  040166  122716  000023            CMPB    #$XOFF,(SP)     ;;WAS CHAR XOFF                 ;MJD001
7337  040172  001012                    BNE     102$            ;;BR IF NOT                     ;MJD001
7338  040174                    101$:                                                           ;MJD001
7339  040174  105777  140744            TSTB    @$TKS           ;;WAIT FOR CHAR                 ;MJD001
7340  040200  100375                    BPL     101$                                            ;MJD001
7341  040202  117716  140740            MOVB    @$TKB,(SP)      ;;GET CHAR                      ;MJD001
7342  040206  042716  177600            BIC     #177600,(SP)    ;;STRIP IT                      ;MJD001
7343  040212  122716  000021            CMPB    #$XON,(SP)      ;;WAS IT XON?                   ;MJD001
7344  040216  001366                    BNE     101$            ;;BR IF NOT                     ;MJD001
7345  040220                    102$:                                                           ;MJD001
7346  040220  005726                    TST     (SP)+           ;;FIX STACK                     ;MJD001
7347  040222                    10$:                                                            ;MJD001
7348  040222  105777  140722            TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
7349  040226  100375                    BPL     10$                                             ;MJD001
7350  040230  116677  000002  140714    MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
7351  040236  122766  000015  000002    CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
7352  040244  001003                    BNE     1$              ;;BRANCH IF NO
7353  040246  105037  040266            CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
7354  040252  000406                    BR      $TYPEX          ;;EXIT
7355  040254  122766  000012  000002  1$:   CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
7356  040262  001402                    BEQ     $TYPEX          ;;BRANCH IF YES
7357  040264  105227                    INCB    (PC)+           ;;COUNT THE CHARACTER
7358  040266  000000            $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
7359  040270  000207            $TYPEX: RTS     PC
7360
7361                                    .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
7362
7363                                    ;;**********************************************************************
7364                                    ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
7365                                    ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
7366                                    ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
7367                                    ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
7368                                    ;*REPLACED WITH SPACES.
7369                                    ;*CALL:
7370                                    ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
```

```
                        ;*      TYPDS                   ;;GO TO THE ROUTINE
                        ;
040272                  $TYPDS:
040272  010046                MOV     R0,-(SP)        ;;PUSH R0 ON STACK
040274  010146                MOV     R1,-(SP)        ;;PUSH R1 ON STACK
040276  010246                MOV     R2,-(SP)        ;;PUSH R2 ON STACK
040300  010346                MOV     R3,-(SP)        ;;PUSH R3 ON STACK
040302  010546                MOV     R5,-(SP)        ;;PUSH R5 ON STACK
040304  012746  020200        MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
040310  016605  000020        MOV     20(SP),R5       ;;GET THE INPUT NUMBER
040314  100004                BPL     1$              ;;BR IF INPUT IS POS.
040316  005405                NEG     R5              ;;MAKE THE BINARY NUMBER POS.
040320  112766  000055  000001  MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
040326  005000          1$:   CLR     R0              ;;ZERO THE CONSTANTS INDEX
040330  012703  040506        MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
040334  112723  000040        MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
040340  005002          2$:   CLR     R2              ;;CLEAR THE BCD NUMBER
040342  016001  040476        MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
040346  160105          3$:   SUB     R1,R5           ;;FORM THIS BCD DIGIT
040350  002402                BLT     4$              ;;BR IF DONE
040352  005202                INC     R2              ;;INCREASE THE BCD DIGIT BY 1
040354  000774                BR      3$
040356  060105          4$:   ADD     R1,R5           ;;ADD BACK THE CONSTANT
040360  005702                TST     R2              ;;CHECK IF BCD DIGIT=0
040362  001002                BNE     5$              ;;FALL THROUGH IF 0
040364  105716                TSTB    (SP)            ;;STILL DOING LEADING 0'S?
040366  100407                BMI     7$              ;;BR IF YES
040370  106316          5$:   ASLB    (SP)            ;;MSD?
040372  103003                BCC     6$              ;;BR IF NO
040374  116663  000001  177777  MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
040402  052702  000060  6$:   BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
040406  052702  000040  7$:   BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
040412  110223                MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
040414  005720                TST     (R0)+           ;;JUST INCREMENTING
040416  020027  000010        CMP     R0,#10          ;;CHECK THE TABLE INDEX
040422  002746                BLT     2$              ;;GO DO THE NEXT DIGIT
040424  003002                BGT     8$              ;;GO TO EXIT
040426  010502                MOV     R5,R2           ;;GET THE LSD
040430  000764                BR      6$              ;;GO CHANGE TO ASCII
040432  105726          8$:   TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
040434  100003                BPL     9$              ;;BR IF NO
040436  116663  177777  177776  MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
040444  105013          9$:   CLRB    (R3)            ;;SET THE TERMINATOR
040446  012605                MOV     (SP)+,R5        ;;POP STACK INTO R5
040450  012603                MOV     (SP)+,R3        ;;POP STACK INTO R3
040452  012602                MOV     (SP)+,R2        ;;POP STACK INTO R2
040454  012601                MOV     (SP)+,R1        ;;POP STACK INTO R1
040456  012600                MOV     (SP)+,R0        ;;POP STACK INTO R0
040460  104401  040506        TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
040464  016666  000002  000004  MOV     2(SP),4(SP)     ;;ADJUST THE STACK
040472  012616                MOV     (SP)+,(SP)
040474  000002                RTI                     ;;RETURN TO USER
040476  023420          $DTBL:  10000.
040500  001750                1000.
040502  000144                100.
040504  000012                10.
```

```
                                $DBLK:  .BLKW   4
                                .SBTTL  APT COMMUNICATIONS ROUTINE
                                ;;**************************************************************************
040516  112737  000001  040762  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
040524  112737  000001  040760  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
040532  000403                  BR      $ATYC
040534  112737  000001  040762  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
040542                          $ATYC:
040542  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
040544  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
040546  105737  040760          TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
040552  001450                  BEQ     5$              ;;IF NOT:  BR
040554  122737  000001  001230  CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
040562  001031                  BNE     3$              ;;IF NOT:  BR
040564  132737  000100  001231  BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
040572  001425                  BEQ     3$              ;;IF NOT:  BR
040574  017600  000004          MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
040600  062766  000002  000004  ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
040606  005737  001210          1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
040612  001375                  BNE     1$              ;;IF NOT:  WAIT
040614  010037  001224          MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
040620  105720                  2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
040622  001376                  BNE     2$
040624  163700  001224          SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
040630  006200                  ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
040632  010037  001226          MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
040636  012737  000004  001210  MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
040644  000413                  BR      5$
040646  017637  000004  040672  3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
040654  062766  000002  000004  ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
040662  013746  177776          MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
040666  004737  037736          JSR     PC,$TYPE        ;;CALL TYPE MACRO
040672  000000                  4$:     .WORD   0
040674                          5$:
040674  105737  040762          10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
040700  001416                  BEQ     12$             ;;IF NOT:  BR
040702  005737  001230          TST     $ENV            ;;RUNNING UNDER APT?
040706  001413                  BEQ     12$             ;;IF NOT:  BR
040710  005737  001210          11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
040714  001375                  BNE     11$             ;;IF NOT:  WAIT
040716  017637  000004  001212  MOV     @4(SP),$FATAL   ;;GET ERROR #
040724  062766  000002  000004  ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
040732  005237  001210          INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
040736  105037  040762          12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
040742  105037  040761          CLRB    $LFLG           ;;CLEAR LOG FLAG
040746  105037  040760          CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
040752  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
040754  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
040756  000207                  RTS     PC              ;;RETURN
040760     000                  $MFLG:  .BYTE   0       ;;MESSG. FLAG
040761     000                  $LFLG:  .BYTE   0       ;;LOG FLAG
040762     000                  $FFLG:  .BYTE   0       ;;FATAL FLAG
          040764                          .EVEN
          000200                APTSIZE=200
          000001                APTENV=001
```

```
                        000100          APTSPOOL=100
                        000040          APTCSUP=040
                                        .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE

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

 7510  040764  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
 7511  040770  116637  000001  041207          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
 7512  040776  112637  041211                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
 7513  041002  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
 7514  041006  000406                          BR      $TYPON
 7515  041010  112737  000001  041207  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
 7516  041016  112737  000006  041211          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
 7517  041024  112737  000005  041206  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
 7518  041032  010346                          MOV     R3,-(SP)        ;;SAVE R3
 7519  041034  010446                          MOV     R4,-(SP)        ;;SAVE R4
 7520  041036  010546                          MOV     R5,-(SP)        ;;SAVE R5
 7521  041040  113704  041211                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
 7522  041044  005404                          NEG     R4
 7523  041046  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
 7524  041052  110437  041210                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
 7525  041056  113704  041207                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
 7526  041062  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
 7527  041066  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
 7528  041070  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
 7529  041072  000404                          BR      3$              ;;GO DO MSB
 7530  041074  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
 7531  041076  006105                          ROL     R5
 7532  041100  006105                          ROL     R5
 7533  041102  010503                          MOV     R5,R3
 7534  041104  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
 7535  041106  105337  041210                  DECB    $OMODE          ;;TYPE THIS DIGIT?
 7536  041112  100016                          BPL     7$              ;;BR IF NO
 7537  041114  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
 7538  041120  001002                          BNE     4$              ;;TEST FOR 0
```

```
                                        ;;SUPPRESS THIS 0?
                                BEQ     5$              ;;BR IF YES
                        4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
                                BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
                        5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
                                MOVB    R3,8$           ;;SAVE FOR TYPING
                                TYPE    ,8$             ;;GO TYPE THIS DIGIT
                        7$:     DECB    $OCNT           ;;COUNT BY 1
                                BGT     2$              ;;BR IF MORE TO DO
                                BLT     6$              ;;BR IF DONE
                                INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
                                BR      2$              ;;GO DO THE LAST DIGIT
                        6$:     MOV     (SP)+,R5        ;;RESTORE R5
                                MOV     (SP)+,R4        ;;RESTORE R4
                                MOV     (SP)+,R3        ;;RESTORE R3
                                MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
                                MOV     (SP)+,(SP)
                                RTI                     ;;RETURN
                        8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
                                .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
                        $OCNT:  .BYTE   0               ;;OCTAL DICIT COUNTER
                        $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
                        $OMCDE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
                        .SBTTL  TTY INPUT ROUTINE

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

                        ;*TK INITIALIZE ROUTINE
                        ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
                        ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
                        ;
                        ;*CALL:
                        ;*      JSR     PC,$TKINT
                        ;*      RETURN
                        ;
041222  005037  041212  $TKINT: CLR     $TKCNT          ;;CLEAR COUNT OF ITEMS IN QUEUE
041226  012737  041220  041214   MOV     #$TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
041234  013737  041214  041216   MOV     $TKQIN,$TKQOUT  ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
041242  012737  041272  000060   MOV     #$TKSRV,@#TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
041250  012737  000200  000062   MOV     #200,@#TKVEC+2  ;;'BR' LEVEL 4
041256  005777  137664           TST     @$TKB           ;;CLEAR DONE FLAG
041262  012777  000100  137654   MOV     #100,@$TKS      ;;ENABLE TTY KEYBOARD INTERRUPT
041270  000207                   RTS     PC              ;;RETURN TO CALLER

                        ;*TK SERVICE ROUTINE
                        ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
                        ;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
                        ;*IT IN THE QUEUE.
                        ;*IF THE CHARACTER IS A "CONTROL-C" (^C) $TKINT IS CALLED AND
```

```
                                        ;*UPON RETURN EXIT IS MADE TO THE ''CONTROL-C'' RESTART ADDRESS (STOP)
                                        ;
041272  117746  137650        $TKSRV: MOVB    @$TKB,-(SP)     ;;PICKUP THE CHARACTER
041276  042716  177600                BIC     #^C177,(SP)     ;;STRIP THE JUNK
041302  021627  000021                CMP     (SP),#$XON      ;;IS IT A RANDOM XON?                   ;RAN001
041306  001002                        BNE     30$             ;;BRANCH IF NO                          ;RAN001
041310  005726                        TST     (SP)+           ;;CLEAN RANDOM XON OFF STACK            ;RAN001
041312  000002                        RTI                     ;;RETURN                                ;RAN001
041314                          30$:                                                                    ;RAN001
041314  021627  000003                CMP     (SP),#3         ;;IS IT A CONTROL C?
041320  001007                        BNE     1$              ;;BRANCH IF NO
041322  104401  042432                TYPE    ,$CNTLC         ;;TYPE A CONTROL-C (^C)
041326  004737  041222                JSR     PC,$TKINT       ;;INIT THE KEYBOARD
041332  005726                        TST     (SP)+           ;;CLEAN UP STACK
041334  000137  036320                JMP     STOP            ;;CONTROL C RESTART
041340  021627  000007          1$:   CMP     (SP),#7         ;;IS IT A CONTROL G?
041344  001004                        BNE     2$              ;;BRANCH IF NO
041346  022737  000176  001140        CMP     #SWREG,SWR      ;;IS SOFT-SWR SELECTED?
041354  001500                        BEQ     6$              ;;GO TO SWR CHANGE
                                
041356                          2$:
041356  022737  000001  041212        CMP     #1,$TKCNT       ;;IS THE QUEUE FULL?
041364  001004                        BNE     3$              ;;BRANCH IF NO
041366  104401  001200                TYPE    ,$BELL          ;;RING THE TTY BELL
041372  005726                        TST     (SP)+           ;;CLEAN CHARACTER OFF OF STACK
041374  000451                        BR      5$              ;;EXIT
041376  021627  000023          3$:   CMP     (SP),#23        ;;IS IT A CONTROL-S?
041402  001021                        BNE     32$             ;;BRANCH IF NO
041404  005077  137534                CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
041410  005726                        TST     (SP)+           ;;CLEAN CHAR OFF STACK
041412  105777  137526          31$:  TSTB    @$TKS           ;;WAIT FOR A CHAR
041416  100375                        BPL     31$             ;;LOOP UNTIL ITS THERE
041420  117746  137522                MOVB    @$TKB,-(SP)     ;;GET THE CHARACTER
041424  042716  177600                BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
041430  022627  000021                CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
041434  001366                        BNE     31$             ;;BRANCH IF NO
041436  012777  000100  137500        MOV     #100,@$TKS      ;;REENABLE TTY KEYBOARD INTERRUPTS
041444  000002                        RTI                     ;;RETURN
041446  005237  041212          32$:  INC     $TKCNT          ;;COUNT THIS CHARACTER
041452  021627  000140                CMP     (SP),#140       ;;IS IT UPPER CASE?
041456  002405                        BLT     4$              ;;BRANCH IF YES
041460  021627  000175                CMP     (SP),#175       ;;IS IT A SPECIAL CHAR?
041464  003002                        BGT     4$              ;;BRANCH IF YES
041466  042716  000040                BIC     #40,(SP)        ;;MAKE IT UPPER CASE
041472  112677  177516          4$:   MOVB    (SP)+,@$TKQIN   ;;AND PUT IT IN QUEUE
041476  005237  041214                INC     $TKQIN          ;;UPDATE THE POINTER
041502  023727  041214  041221        CMP     $TKQIN,#$TKQEND ;;GO OFF THE END?
041510  001003                        BNE     5$              ;;BRANCH IF NO
041512  012737  041220  041214        MOV     #$TKQSRT,$TKQIN ;;RESET THE POINTER
041520  000002                  5$:   RTI                     ;;RETURN
                                
                                        ;;**************************************************************************
                                        ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
                                        ;*ROUTINE IS ENTERED  FROM THE TRAP HANDLER, AND WILL
                                        ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
                                        ;*CALL WHEN OPERATING IN TTY INTERRUPT MODE.
```

```
 7651  041522  022737  000176  001140  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED
 7652  041530  001124                          BNE     15$             ;;EXIT IF NOT
 7653  041532  105777  137406                  TSTB    @$TKS           ;;IS A CHAR WAITING?
 7654  041536  100121                          BPL     15$             ;;IF NOT, EXIT
 7655  041540  117746  137402                  MOVB    @$TKB,-(SP)     ;;YES
 7656  041544  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT  7-BIT ASCII
 7657  041550  021627  000007                  CMP     (SP),#7         ;;IS IT A CONTROL-G?
 7658  041554  001300                          BNE     2$              ;;IF NOT, PUT IT IN THE TTY QUEUE
 7659                                                                  ;;AND EXIT
 7660
 7661                                  ;;**********************************************************************
 7662                                  ;*CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
 7663                                  ;*ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
 7664                                  ;*CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
 7665  041556  123727  001134  000001  6$:     CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
 7666  041564  001674                          BEQ     2$              ;;BRANCH IF YES
 7667  041566  005726                          TST     (SP)+           ;;CLEAR CONTROL-G OFF STACK
 7668  041570  004737  041222                  JSR     PC,$TKINT       ;;FLUSH THE TTY INPUT QUEUE
 7669  041574  005077  137344                  CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
 7670  041600  112737  000001  001135          MOVB    #1,$INTAG       ;;SET INTERRUPT MODE INDICATOR
 7671
 7672  041606  104401  042444                  TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
 7673  041612  104401  042451  $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
 7674  041616  013746  000176                  MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
 7675  041622  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 7676  041624  104401  042462                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
 7677  041630  005046                  19$:     CLR     -(SP)           ;;CLEAR COUNTER
 7678  041632  005046                          CLR     -(SP)           ;;THE NEW SWR
 7679  041634  105777  137304          7$:     TSTB    @$TKS           ;;CHAR THERE?
 7680  041640  100375                          BPL     7$              ;;IF NOT TRY AGAIN
 7681
 7682  041642  117746  137300                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
 7683  041646  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
 7684
 7685  041652  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL-C?
 7686  041656  001015                          BNE     9$              ;;BRANCH IF NOT
 7687  041660  104401  042432                  TYPE    ,$CNTLC         ;;YES, ECHO CONTROL-C (^C)
 7688  041664  062706  000006                  ADD     #6,SP           ;;CLEAN UP STACK
 7689  041670  123727  001135  000001          CMPB    $INTAG,#1       ;;REENABLE TTY KEYBOARD INTERRUPTS?
 7690  041676  001003                          BNE     8$              ;;BRANCH IF NO
 7691  041700  012777  000100  137236          MOV     #100,@$TKS      ;;ALLOW TTY KEYBOARD INTERRUPTS
 7692  041706  000137  036320          8$:     JMP     STOP            ;;CONTROL-C RESTART
 7693
 7694
 7695  041712  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
 7696  041716  001005                          BNE     10$             ;;BRANCH IF NOT
 7697  041720  104401  042437                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
 7698  041724  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
 7699  041730  000737                          BR      19$             ;;LET'S TRY IT AGAIN
 7700
 7701
 7702  041732  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
 7703  041736  001022                          BNE     16$             ;;BRANCH IF NO
 7704  041740  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
 7705  041744  001403                          BEQ     11$             ;;BRANCH IF YES
 7706  041746  016677  000002  137164          MOV     2(SP),@SWR      ;;SAVE NEW SWR
```

```
7707  041754  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
7708  041760  104401  001205          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
7709  041764  123727  001135  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
7710  041772  001003                          BNE     15$             ;;BRANCH IF NOT
7711  041774  012777  000100  137142          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
7712  042002  000002                  15$:    RTI                     ;;RETURN
7713  042004  004737  040150          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
7714  042010  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
7715  042014  002420                          BLT     18$             ;;BRANCH IF YES
7716  042016  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
7717  042022  003015                          BGT     18$             ;;BRANCH IF YES
7718  042024  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
7719  042030  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
7720  042034  001403                          BEQ     17$             ;;BRANCH IF YES
7721  042036  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
7722  042040  006316                          ASL     (SP)            ;;   CHAR OVER TO MAKE
7723  042042  006316                          ASL     (SP)            ;;   ROOM FOR NEW ONE.
7724  042044  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
7725  042050  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
7726  042054  000667                          BR      7$              ;;GET THE NEXT ONE
7727  042056  104401  001204          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
7728  042062  000720                          BR      20$             ;;SIMULATE CONTROL-U
7729                                  .DSABL  LSB
7730
7731
7732                                  ;;*******************************************************************
7733                                  ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
7734                                  ;*CALL:
7735                                  ;*      RDCHR                   ;;GET A CHARACTER FROM THE QUEUE
7736                                  ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
7737                                  ;*                              ;;WITH PARITY BIT STRIPPED OFF
7738                                  ;
7739
7740  042064  011646          $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC AND
7741  042066  016666  000004  000002          MOV     4(SP),2(SP)     ;;THE PS
7742  042074  005066  000004                  CLR     4(SP)           ;;GET READY FOR A CHARACTER
7743  042100  005046                          CLR     -(SP)           ;;PUT NEW PS ON STACK
7744  042102  012746  042110                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
7745  042106  000002                          RTI                     ;;POP NEW PC AND PS
7746  042110                          64$:
7747  042110  005737  041212          1$:     TST     $TKCNT          ;;WAIT ON A CHARACTER
7748  042114  001775                          BEQ     1$
7749  042116  005337  041212                  DEC     $TKCNT          ;;DECREMENT THE COUNTER
7750  042122  117766  177070  000004          MOVB    @$TKQOUT,4(SP)  ;;GET ONE CHARACTER
7751  042130  005237  041216                  INC     $TKQOUT         ;;UPDATE THE POINTER
7752  042134  023727  041216  041221          CMP     $TKQOUT,#$TKQEND ;;DID IT GO OFF OF THE END?
7753  042142  001003                          BNE     2$              ;;BRANCH IF NO
7754  042144  012737  041220  041216          MOV     #$TKQSRT,$TKQOUT ;;RESET THE POINTER
7755  042152  000002                  2$:     RTI                     ;;RETURN
7756                                  ;;*******************************************************************
7757                                  ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
7758                                  ;*CALL:
7759                                  ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
7760                                  ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
7761                                  ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
7762
```

```
 7763  042154  010346                  $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
 7764  042156  005046                          CLR     -(SP)           ;;CLEAR THE RUBOUT KEY
 7765  042160  012703  042410            1$:   MOV     #$TTYIN,R3      ;;GET ADDRESS
 7766  042164  022703  042432            2$:   CMP     #$TTYIN+22,R3   ;;BUFFER FULL?
 7767  042170  101456                          BLOS    4$              ;;BR IF YES
 7768  042172  104410                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
 7769  042174  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
 7770  042176  122713  000177           10$:   CMPB    #177,(R3)       ;;IS IT A RUBOUT
 7771  042202  001022                          BNE     5$              ;;BR IF NO
 7772  042204  005716                          TST     (SP)            ;;IS THIS THE FIRST RUBOUT?
 7773  042206  001007                          BNE     6$              ;;BR IF NO
 7774  042210  112737  000134  042406        MOVB    #'\,9$          ;;TYPE A BACK SLASH
 7775  042216  104401  042406                  TYPE    ,9$
 7776  042222  012716  177777                  MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
 7777  042226  005303                  6$:     DEC     R3              ;;BACKUP BY ONE
 7778  042230  020327  042410                  CMP     R3,#$TTYIN      ;;STACK EMPTY?
 7779  042234  103434                          BLO     4$              ;;BR IF YES
 7780  042236  111337  042406                  MOVB    (R3),9$         ;;SETUP TO TYPEOUT THE DELETED CHAR.
 7781  042242  104401  042406                  TYPE    ,9$             ;;GO TYPE
 7782  042246  000746                          BR      2$              ;;GO READ ANOTHER CHAR.
 7783  042250  005716                  5$:     TST     (SP)            ;;RUBOUT KEY SET?
 7784  042252  001406                          BEQ     7$              ;;BR IF NO
 7785  042254  112737  000134  042406        MOVB    #'\,9$          ;;TYPE A BACK SLASH
 7786  042262  104401  042406                  TYPE    ,9$
 7787  042266  005016                          CLR     (SP)            ;;CLEAR THE RUBOUT KEY
 7788  042270  122713  000025            7$:   CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
 7789  042274  001003                          BNE     8$              ;;BR IF NO
 7790  042276  104401  042437                  TYPE    ,$CNTLU         ;;TYPE A CONTROL 'U''
 7791  042302  000726                          BR      1$              ;;GO START OVER
 7792  042304  122713  000022            8$:   CMPB    #22,(R3)        ;;IS CHARACTER A ''^R''?
 7793  042310  001011                          BNE     3$              ;;BRANCH IF NO
 7794  042312  105013                          CLRB    (R3)            ;;CLEAR THE CHARACTER
 7795  042314  104401  001205                  TYPE    ,$CRLF          ;;TYPE A ''CR'' & 'LF''
 7796  042320  104401  042410                  TYPE    ,$TTYIN         ;;TYPE THE INPUT STRING
 7797  042324  000717                          BR      2$              ;;GO PICKUP ANOTHER CHACTER
 7798  042326  104401  001204            4$:   TYPE    ,$QUES          ;;TYPE A '?'
 7799  042332  000712                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
 7800  042334  111337  042406            3$:   MOVB    (R3),9$         ;;ECHO THE CHARACTER
 7801  042340  104401  042406                  TYPE    ,9$
 7802  042344  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
 7803  042350  001305                          BNE     2$              ;;LOOP IF NOT RETURN
 7804  042352  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
 7805  042356  104401  001206                  TYPE    ,$LF            ;;TYPE A LINE FEED
 7806  042362  005726                          TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
 7807  042364  012603                          MOV     (SP)+,R3        ;;RESTORE R3
 7808  042366  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
 7809  042370  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
 7810  042376  012766  042410  000004          MOV     #$TTYIN,4(SP)
 7811  042404  000002                          RTI                     ;;RETURN
 7812  042406     000                  9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
 7813  042407     000                          .BYTE   0               ;;TERMINATOR
 7814  042410  000022                  $TTYIN: .BLKB   22              ;;RESERVE 22 BYTES FOR TTY INPUT
 7815  042432  041536  005015     000  $CNTLC: .ASCIZ  /^C/<15><12>    ;;CONTROL ''C''
 7816  042437     136  006525  000012  $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL 'U''
 7817  042444  043536  005015     000  $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL ''G''
 7818  042451     015  051412  051127  $MSWR:  .ASCIZ  <15><12>/SWR = /
```

```
 7819  042456  036440  000040
 7820  042462  020040  042516  020127  $MNEW:  .ASCIZ  /  NEW = /
 7821  042470  020075     000
 7822          042474                  .EVEN
 7823                                  .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
 7824
 7825                                  ;;*****************************************************************
 7826                                  ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
 7827                                  ;*CHANGE IT TO BINARY.
 7828                                  ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
 7829                                  ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
 7830                                  ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
 7831                                  ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
 7832                                  ;*CALL:
 7833                                  ;*      RDOCT                   ;;READ AN OCTAL NUMBER
 7834                                  ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
 7835                                  ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT
 7836
 7837  042474  011646          $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
 7838  042476  016666  000004  000002  MOV     4(SP),2(SP)     ;;INPUT NUMBER
 7839  042504  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 7840  042506  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 7841  042510  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 7842  042512  104411          1$:     RDLIN                   ;;READ AN ASCIZ LINE
 7843  042514  012600                  MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
 7844  042516  010037  042622          MOV     R0,5$           ;;AND SAVE IT
 7845  042522  005001                  CLR     R1              ;;CLEAR DATA WORD
 7846  042524  005002                  CLR     R2
 7847  042526  112046          2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
 7848  042530  001420                  BEQ     3$              ;;IF ZERO GET OUT
 7849  042532  122716  000060          CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
 7850  042536  003026                  BGT     4$              ;;IS AN OCTAL DIGIT
 7851  042540  122716  000067          CMPB    #'7,(SP)
 7852  042544  002423                  BLT     4$
 7853  042546  006301                  ASL     R1              ;;*2
 7854  042550  006102                  ROL     R2
 7855  042552  006301                  ASL     R1              ;;*4
 7856  042554  006102                  ROL     R2
 7857  042556  006301                  ASL     R1              ;;*8
 7858  042560  006102                  ROL     R2
 7859  042562  042716  177770          BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
 7860  042566  062601                  ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
 7861  042570  000756                  BR      2$              ;;LOOP
 7862  042572  005726          3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
 7863  042574  010166  000012          MOV     R1,12(SP)       ;;SAVE THE RESULT
 7864  042600  010237  042632          MOV     R2,$HIOCT
 7865  042604  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
 7866  042606  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
 7867  042610  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
 7868  042612  000002                  RTI                     ;;RETURN
 7869  042614  005726          4$:     TST     (SP)+           ;;CLEAN PARTIAL FROM STACK
 7870  042616  105010                  CLRB    (R0)            ;;SET A TERMINATOR
 7871  042620  104401                  TYPE                    ;;TYPE UP THRU THE BAD CHAR.
 7872  042622  000000          5$:     .WORD   0
 7873  042624  104401  001204          TYPE    ,$QUES          ;;"?" "CR" & "LF"
 7874  042630  000730                  BR      1$              ;;TRY AGAIN
```

```
                                $HIOCT: .WORD   0                       ;;HIGH ORDER BITS GO HERE
                                .SBTTL  DOUBLE LENGTH BINARY TO OCTAL ASCII CONVERT ROUTINE

                                ;;**********************************************************************
                                ;*THIS ROUTINE WILL CONVERT A 32-BIT UNSIGNED BINARY NUMBER TO AN
                                ;*UNSIGNED OCTAL ASCIZ NUMBER.
                                ;*CALL
                                ;*      MOV     #PNTR,-(SP)     ;;POINTER TO LOW WORD OF BINARY NUMBER
                                ;*      JSR     PC,@#$DB2O      ;;CALL THE ROUTINE
                                ;*      RETURN                  ;;THE ADDRESS OF THE FIRST ASCIZ CHAR. IS ON THE STACK

042634  104413                  $DB2O:  SAVREG                  ;;SAVE ALL REGISTERS
042636  016601  000002                  MOV     2(SP),R1        ;;PICKUP THE POINTER TO LOW WORD
042642  012705  042753                  MOV     #$OCTVL+13.,R5  ;;POINTER TO DATA TABLE
042646  012704  000014                  MOV     #12.,R4         ;;DO ELEVEN CHARACTERS
042652  012703  177770                  MOV     #^C7,R3         ;;MASK
042656  012100                          MOV     (R1)+,R0        ;;LOWER WORD
042660  012101                          MOV     (R1)+,R1        ;;HIGH WORD
042662  005002                          CLR     R2              ;;TERMINATOR
042664  110245                  1$:     MOVB    R2,-(R5)        ;;PUT CHARACTER IN DATA TABLE
042666  010002                          MOV     R0,R2           ;;GET THIS DIGIT
042670  005304                          DEC     R4              ;;COUNT THIS CHARACTER
042672  003007                          BGT     3$              ;;BR IF NOT THE LAST DIGIT
042674  001405                          BEQ     2$              ;;BR IF IT IS THE LAST DIGIT
042676  005205                          INC     R5              ;;ALL DIGITS DONE-ADJUST POINTER FOR FIRST
042700  010566  000002                  MOV     R5,2(SP)        ;;ASCIZ CHAR. & PUT IT ON THE STACK
042704  104414                          RESREG                  ;;RESTORE ALL REGISTERS
042706  000207                          RTS     PC              ;;RETURN TO USER
042710  006203                  2$:     ASR     R3              ;;POSITION THE MASK FOR THE LAST DIGIT
042712  006001                  3$:     ROR     R1              ;;POSITION THE BINARY NUMBER FOR
042714  006000                          ROR     R0              ;;      THE NEXT OCTAL DIGIT
042716  006001                          ROR     R1
042720  006000                          ROR     R0
042722  006001                          ROR     R1
042724  006000                          ROR     R0
042726  040302                          BIC     R3,R2           ;;MASK OUT ALL JUNK
042730  062702  000060                  ADD     #'0,R2          ;;MAKE THIS CHAR. ASCII
042734  000753                          BR      1$              ;;GO PUT IT IN THE DATA TABLE
042736  000016                  $OCTVL: .BLKB   14.             ;;RESERVE DATA TABLE
                                .SBTTL  DOUBLE LENGTH BINARY TO DECIMAL ASCII CONVERT ROUTINE

                                ;;**********************************************************************
                                ;*THIS ROUTINE WILL CONVERT A 32-BIT BINARY NUMBER TO AN UNSIGNED
                                ;*DECIMAL (ASCII) NUMBER. THE SIGN OF THE BINARY NUMBER MUST BE
                                ;*POSITIVE.
                                ;*CALL
                                ;*      MOV     #PNTR,-(SP)     ;;POINTER TO LOW WORD OF BINARY NUMBER
                                ;*      JSR     PC,@#$DB2D
                                ;*      RETURN                  ;;THE FIRST ADDRESS OF ASCIZ
                                                                ;;IS ON THE STACK

042754  104413                  $DB2D:  SAVREG                  ;;SAVE REGISTERS
042756  016602  000002                  MOV     2(SP),R2        ;;PICKUP THE DATA POINTER
042762  012700  043134                  MOV     #$DECVL,R0      ;;GET ADDRESS OF "$DECVL" STRING
```

```
 7931  042766  010066  000002                  MOV     R0,2(SP)        ;;PUT ADDRESS OF ASCIZ STRING ON STACK
 7932  042772  012201                          MOV     (R2)+,R1        ;;PICKUP THE BINARY NUMBER
 7933  042774  012202                          MOV     (R2)+,R2
 7934  042776  012737  000012  043052          MOV     #10.,4$         ;;SET UP TO DO 10 CONVERSIONS
 7935  043004  012704  043064                  MOV     #$TNPWR,R4      ;;ADDRESS OF TEN POWER
 7936  043010  012705  043066                  MOV     #$TNPWR+2,R5
 7937  043014  005003                  1$:     CLR     R3              ;;CLEAR PARTIAL
 7938  043016  161401                  2$:     SUB     (R4),R1         ;;SUBTRACT TEN POWER
 7939  043020  005602                          SBC     R2
 7940  043022  161502                          SUB     (R5),R2
 7941  043024  002402                          BLT     3$              ;;BR IF TEN POWER TO LARGE
 7942  043026  005203                          INC     R3              ;;ADD 1 TO PARTIAL
 7943  043030  000772                          BR      2$              ;;LOOP
 7944  043032  062401                  3$:     ADD     (R4)+,R1        ;;RESTORE SUBTRACTED VALUE
 7945  043034  005502                          ADC     R2
 7946  043036  062402                          ADD     (R4)+,R2
 7947  043040  022525                          CMP     (R5)+,(R5)+     ;;MOVE TO NEXT TEN POWER
 7948  043042  052703  000060                  BIS     #'0,R3          ;;CHANGE PARTIAL TO ASCII
 7949  043046  110320                          MOVB    R3,(R0)+        ;;SAVE IT
 7950  043050  005327                          DEC     (PC)+           ;;DONE?
 7951  043052  000000                  4$:     .WORD   0
 7952  043054  001357                          BNE     1$              ;;BR IF NO
 7953  043056  105020                          CLRB    (R0)+           ;;TERMINATOR
 7954  043060  104414                          RESREG                  ;;RESTORE REGISTERS
 7955  043062  000207                          RTS     PC              ;;RETURN
 7956  043064  145000                  $TNPWR: 145000                  ;;1.0E09
 7957  043066  035632                          35632
 7958  043070  160400                          160400                  ;;1.0E08
 7959  043072  002765                          2765
 7960  043074  113200                          113200                  ;;1.0E07
 7961  043076  000230                          230
 7962  043100  041100                          041100                  ;;1.0E06
 7963  043102  000017                          17
 7964  043104  103240                          103240                  ;;1.0E05
 7965  043106  000001                          1
 7966  043110  023420                          23420                   ;;1.0E04
 7967  043112  000000                          0
 7968  043114  001750                          1750                    ;;1.0E03
 7969  043116  000000                          0
 7970  043120  000144                          144                     ;;1.0E02
 7971  043122  000000                          0
 7972  043124  000012                          12                      ;;1.0E01
 7973  043126  000000                          0
 7974  043130  000001                          1                       ;;1.0E00
 7975  043132  000000                          0
 7976  043134  000014          $DECVL: .BLKB   12.                     ;;RESERVE STORAGE FOR ASCIZ STRING
 7977                          .SBTTL  SINGLE LENGTH BINARY TO DECIMAL ASCIZ ROUTINE
 7978
 7979                          ;;******************************************************************
 7980                          ;*THIS ROUTINE WILL CONVERT A 16-BIT UNSIGNED BINARY NUMBER TO AN
 7981                          ;*UNSIGNED DECIMAL ASCIZ NUMBER.
 7982                          ;*CALL
 7983                          ;*      MOV     NUMBER,-(SP)    ;;PUT BINARY NUMBER ON THE STACK
 7984                          ;*      JSR     PC,@#$SB2D      ;;CALL
 7985                          ;*      RETURN                  ;;ADDRESS OF THE 1ST ASCIZ CHAR.IS ON THE STACK
 7986
```

```
 7987
 7988  043150  016637  000002  043200  $SB2D:  MOV     2(SP),1$        ;;SAVE BINARY NUMBER
 7989  043156  012746  043200                  MOV     #1$,-(SP)       ;;SET POINTER
 7990  043162  004737  042754                  JSR     PC,@#$DB2D      ;;CALL DOUBLE LENGTH CONVERT
 7991  043166  062716  000005                  ADD     #5,(SP)         ;;ONLY ALLOW FIVE CHARACTERS
 7992  043172  012666  000002                  MOV     (SP)+,2(SP)     ;;PICKUP POINTER
 7993  043176  000207                          RTS     PC              ;;RETURN
 7994  043200  000000  000000          1$:     .WORD   0,0
 7995                                  .SBTTL  TYPE NUMERICAL ASCIZ STRING SUPPRESS LEADING ZEROS
 7996
 7997                                  ;;**************************************************************************
 7998                                  ;*THIS ROUTINE IS USED TO TYPE AN ASCIZ NUMBER SUPPRESSING THE
 7999                                  ;*LEADING NUMBERS.
 8000                                  ;*CALL
 8001                                  ;*      MOV     #NUMADR,-(SP)   ;;FIRST ADDRESS OF ASCIZ STRING
 8002                                  ;*      JSR     PC,@#$SUPRS
 8003
 8004
 8005  043204  010046                  $SUPRS: MOV     R0,-(SP)        ;;SAVE R0
 8006  043206  016600  000004                  MOV     4(SP),R0        ;;PICKUP THE POINTER
 8007  043212  105710                  1$:     TSTB    (R0)            ;;TERMINATEOR?
 8008  043214  001403                          BEQ     2$              ;;BR IF YES
 8009  043216  122720  000060                  CMPB    #'0,(R0)+       ;;IS THIS AN ASCII '0' ?
 8010  043222  001773                          BEQ     1$              ;;BR IF YES
 8011  043224  005300                  2$:     DEC     R0              ;;BACKUP BY '1'
 8012  043226  010037  043234                  MOV     R0,3$           ;;SAVE FOR TYPING
 8013  043232  104401                          TYPE                    ;;GO TYPE
 8014  043234  000000                  3$:     .WORD   0               ;;ASCIZ POINTER GOES HERE
 8015  043236  012600                          MOV     (SP)+,R0        ;;RESTORE R0
 8016  043240  012616                          MOV     (SP)+,(SP)      ;;RESTORE THE STACK
 8017  043242  000207                          RTS     PC              ;;RETURN
 8018                                  .SBTTL  INTEGER MULTIPLY ROUTINE
 8019
 8020                                  ;;**************************************************************************
 8021                                  ;*CALL
 8022                                  ;*      MOV     MULTIPLER,-(SP)
 8023                                  ;*      MOV     MULTIPLICAND,-(SP)
 8024                                  ;*      JSR     PC,@#$MULT
 8025                                  ;*      RETURN  ;;PRODUCT IS ON THE STACK
 8026                                  ;*
 8027                                  ;*      STACK   PRODUCT
 8028                                  ;*      -----   -------
 8029                                  ;*      TOP     LSB'S
 8030                                  ;*      +2      MSB'S
 8031
 8032  043244                          $MULT:
 8033  043244  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 8034  043246  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 8035  043250  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 8036  043252  005046                          CLR     -(SP)           ;;CLEAR THE SIGN KEY
 8037  043254  016601  000012                  MOV     12(SP),R1       ;;GET THE MULTIPLICAND
 8038  043260  100002                          BPL     1$              ;;BR IF PLUS
 8039  043262  005216                          INC     (SP)            ;;SET THE SIGN KEY
 8040  043264  005401                          NEG     R1              ;;MAKE THE MULTIPLICAND POSTIVE
 8041  043266  016602  000014          1$:     MOV     14(SP),R2       ;;GET THE MULTIPLIER
 8042  043272  100002                          BPL     2$              ;;BR IF PLUS
```

```
8043  043274  005316                  DEC     (SP)            ;;UPDATE THE SIGN KEY
8044  043276  005402                  NEG     R2              ;;MAKE THE MULTIPLIER POSTIVE
8045  043300  012746  000021    2$:   MOV     #17.,-(SP)      ;;SET THE LOOP COUNT
8046  043304  005000                  CLR     R0              ;;SETUP FOR THE MULTIPLY LOOP
8047  043306  103001            3$:   BCC     4$              ;;DON'T ADD IF MULTIPLICAND = 0
8048  043310  060200                  ADD     R2,R0
8049  043312  006000            4$:   ROR     R0              ;;POSITION THE PARITIAL PRODUCT AND
8050  043314  006001                  ROR     R1              ;;THE MULTIPLICAND
8051  043316  005316                  DEC     (SP)            ;;HAS ALL BITS OF THE MULTIPLICAND BEEN DONE?
8052  043320  001372                  BNE     3$              ;;BR IF NO
8053  043322  022616                  CMP     (SP)+,(SP)      ;;SHOULD PRODUCT BE NEGATIVE?
8054  043324  001403                  BEQ     5$              ;;GO TO EXIT IF NO
8055  043326  005400                  NEG     R0              ;;YES--SO MAKE IT SO
8056  043330  005401                  NEG     R1
8057  043332  005600                  SBC     R0
8058  043334  005726            5$:   TST     (SP)+           ;;CLEAR SIGN INFO. OFF OF STACK
8059  043336  010066  000012        MOV     R0,12(SP)       ;;PUT THE PRODUCT ON THE STACK (MSB'S)
8060  043342  010166  000010        MOV     R1,10(SP)       ;;LSB'S
8061  043346  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
8062  043350  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
8063  043352  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
8064  043354  000207                  RTS     PC
8065                          .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES
8066
8067                          ;;**************************************************************************
8068                          ;*SAVE R0-R5
8069                          ;*CALL:
8070                          ;*      SAVREG
8071                          ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
8072                          ;*
8073                          ;*TOP---(+16)
8074                          ;* +2---(+18)
8075                          ;* +4---R5
8076                          ;* +6---R4
8077                          ;* +8---R3
8078                          ;*+10---R2
8079                          ;*+12---R1
8080                          ;*+14---R0
8081
8082  043356                  $SAVREG:
8083  043356  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
8084  043360  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
8085  043362  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
8086  043364  010346                  MOV     R3,-(SP)        ;;PUSH R3 ON STACK
8087  043366  010446                  MOV     R4,-(SP)        ;;PUSH R4 ON STACK
8088  043370  010546                  MOV     R5,-(SP)        ;;PUSH R5 ON STACK
8089  043372  016646  000022          MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
8090  043376  016646  000022          MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
8091  043402  016646  000022          MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
8092  043406  016646  000022          MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
8093  043412  000002                  RTI
8094
8095                          ;*RESTORE R0-R5
8096                          ;*CALL:
8097                          ;*      RESREG
8098  043414                  $RESREG:
```

```
8099  043414  012666  000022          MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
8100  043420  012666  000022          MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
8101  043424  012666  000022          MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
8102  043430  012666  000022          MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
8103  043434  012605                  MOV     (SP)+,R5        ;;POP STACK INTO R5
8104  043436  012604                  MOV     (SP)+,R4        ;;POP STACK INTO R4
8105  043440  012603                  MOV     (SP)+,R3        ;;POP STACK INTO R3
8106  043442  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
8107  043444  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
8108  043446  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
8109  043450  000002                  RTI
8110                          .SBTTL  TRAP DECODER
8111
8112                          ;;******************************************************************
8113                          ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
8114                          ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
8115                          ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
8116                          ;*GO TO THAT ROUTINE.
8117
8118  043452  010046          $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
8119  043454  016600  000002          MOV     2(SP),R0        ;;GET TRAP ADDRESS
8120  043460  005740                  TST     -(R0)           ;;BACKUP BY 2
8121  043462  111000                  MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
8122  043464  006300                  ASL     R0              ;;POSITION FOR INDEXING
8123  043466  016000  043506          MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
8124  043472  000200                  RTS     R0              ;;GO TO ROUTINE
8125
8126
8127                          ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
8128
8129  043474  011646          $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
8130  043476  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
8131  043504  000002                  RTI                     ;;RESTORE THE PSW
8132
8133                          .SBTTL  TRAP TABLE
8134
8135                          ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
8136                          ;*BY THE "TRAP" INSTRUCTION.
8137
8138                          ;       ROUTINE
8139                          ;       -------
8140  043506  043474          $TRPAD: .WORD   $TRAP2
8141  043510  037736                  $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
8142  043512  041010                  $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
8143  043514  040764                  $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
8144  043516  041024                  $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
8145  043520  040272                  $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
8146
8147  043522  041612                  $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING
8148
8149  043524  041522                  $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
8150  043526  042064                  $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
8151  043530  042154                  $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
8152  043532  042474                  $RDOCT  ;;CALL=RDOCT    TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
8153  043534  043356                  $SAVREG ;;CALL=SAVREG   TRAP+13(104413) SAVE R0-R5 ROUTINE
8154  043536  043414                  $RESREG ;;CALL=RESREG   TRAP+14(104414) RESTORE R0-R5 ROUTINE
```

```
  8155  043540  036266                    SCOP1$  ;;CALL=SCOP1    TRAP+15(104415) INTERNAL LOOP ON ERROR
  8156
```

```
 8157
 8158                                   .SBTTL  SERVICE MESSAGES
 8159
 8160
 8161  043542  005015  047125  041111  MSG1:   .ASCII  <CR><LF>/UNIBUSS RK6 DR PRT2/
 8162  043550  051525  020123  045522
 8163  043556  020066  051104  050040
 8164  043564  052122     062
 8165  043567     015  041412  051132          .ASCII  <CR><LF>/CZR6IF0/<CR><LF>
 8166  043574  044466  030106  005015
 8167  043602  005015  025011  025052          .ASCII  <CR><LF>/       ***** CAUTION *****/<CR><LF>
 8168  043610  025052  041440  052501
 8169  043616  044524  047117  025040
 8170  043624  025052  025052  005015
 8171  043632  005015  044124  051511          .ASCII  <CR><LF>/THIS PROGRAM SHOULD  BE HALTED ONLY BY TYPING CONTROL-C/
 8172  043640  050040  047522  051107
 8173  043646  046501  051440  047510
 8174  043654  046125  020104  041040
 8175  043662  020105  040510  052114
 8176  043670  042105  047440  046116
 8177  043676  020131  054502  052040
 8178  043704  050131  047111  020107
 8179  043712  047503  052116  047522
 8180  043720  026514     103
 8181  043723     015  047412  044124          .ASCII  <CR><LF>/OTHERWISE, CARTRIDGE FORMATTING AND, OR THE DRIVE/
 8182  043730  051105  044527  042523
 8183  043736  020054  040503  052122
 8184  043744  044522  043504  020105
 8185  043752  047505  046522  052101
 8186  043760  044524  043516  040440
 8187  043766  042116  020054  051117
 8188  043774  052040  042510  042040
 8189  044002  044522  042526
 8190  044006  005015  040515  020131          .ASCII  <CR><LF>/MAY BE LEFT IN AN UNDETERMINED STATE/
 8191  044014  042502  046040  043105
 8192  044022  020124  047111  040440
 8193  044030  020116  047125  042504
 8194  044036  042524  046522  047111
 8195  044044  042105  051440  040524
 8196  044052  042524
 8197  044054  005015  047111  052111          .ASCII  <CR><LF>/INITIALLY, DRIVES TO BE TESTED SHOULD HAVE:/<CR><LF>
 8198  044062  040511  046114  026131
 8199  044070  042040  044522  042526
 8200  044076  020123  047524  041040
 8201  044104  020105  042524  052123
 8202  044112  042105  051440  047510
 8203  044120  046125  020104  040510
 8204  044126  042526  006472     012
 8205  044133     015  040412  020056          .ASCII  <CR><LF>/A. HEADS MANUALLY LOADED/
 8206  044140  042510  042101  020123
 8207  044146  040515  052516  046101
 8208  044154  054514  046040  040517
 8209  044162  042504     104
 8210  044165     015  041012  020056          .ASCII  <CR><LF>/B. CORRECT PORT SELECTED/
 8211  044172  047503  051122  041505
 8212  044200  020124  047520  052122
```

```
 8213  044206  051440  046105  041505
 8214  044214  042524     104
 8215  044217     015  041412  020056          .ASCII  <CR><LF>/C. WRITE LOCK DISABLED/
 8216  044224  051127  052111  020105
 8217  044232  047514  045503  042040
 8218  044240  051511  041101  042514
 8219  044246     104
 8220  044247     015  042012  020056          .ASCII  <CR><LF>/D. DRIVE READY INDICATOR ON/<CR><LF>
 8221  044254  051104  053111  020105
 8222  044262  042522  042101  020131
 8223  044270  047111  044504  040503
 8224  044276  047524  020122  047117
 8225  044304  005015
 8226  044306  005015  051104  053111          .ASCII  <CR><LF>/DRIVES NOT TO BE TESTED MUST HAVE/
 8227  044314  051505  047040  052117
 8228  044322  052040  020117  042502
 8229  044330  052040  051505  042524
 8230  044336  020104  052515  052123
 8231  044344  044040  053101     105
 8232  044351     015  041012  052117          .ASCIZ  <CR><LF>/BOTH PORTS DESELECTED/<CR><LF>
 8233  044356  020110  047520  052122
 8234  044364  020123  042504  042523
 8235  044372  042514  052103  042105
 8236  044400  005015     000
 8237
 8238  044403     015  052012  020117  MSG2:   .ASCII  <CR><LF>/TO TEST DRIVE 0, REMOVE XXDP MEDIA,/
 8239  044410  042524  052123  042040
 8240  044416  044522  042526  030040
 8241  044424  020054  042522  047515
 8242  044432  042526  054040  042130
 8243  044440  020120  042515  044504
 0244  044446  026101
 8245  044450  005015  046103  040505          .ASCII  <CR><LF>/CLEAR LOC 40,& HIT CONT. ./
 8246  044456  020122  047514  020103
 8247  044464  030064  023054  044040
 8248  044472  052111  041440  047117
 8249  044500  027124  027040
 8250  044504  005015  043111  042040          .ASCII  <CR><LF>/IF DRIVE 0 ISN'T TO BE TESTED, HIT CONT. ./<CR><LF>
 8251  044512  044522  042526  030040
 8252  044520  044440  047123  052047
 8253  044526  052040  020117  042502
 8254  044534  052040  051505  042524
 8255  044542  026104  044040  052111
 8256  044550  041440  047117  027124
 8257  044556  027040  005015
 8258  044562  005015  051104  053111  MSG3:   .ASCIZ  <CR><LF>/DRIVE(S) TO BE TESTED: /
 8259  044570  024105  024523  052040
 8260  044576  020117  042502  052040
 8261  044604  051505  042524  035104
 8262  044612  000040
 8263  044614  005015  054524  042520  MSG4:   .ASCIZ  <CR><LF>/TYPE BUSS ADDRESS IF NOT 177440:  /
 8264  044622  041040  051525  020123
 8265  044630  042101  051104  051505
 8266  044636  020123  043111  047040
 8267  044644  052117  030440  033467
 8268  044652  032064  035060  020040
```

```
 8269  044660     000
 8270  044661     015  052012  050131  MSG5:   .ASCIZ  <CR><LF>/TYPE CONTROLLER INTERRUPT VECTOR IF NOT 210:  /
 8271  044666  020105  047503  052116
 8272  044674  047522  046114  051105
 8273  044702  044440  052116  051105
 8274  044710  052522  052120  053040
 8275  044716  041505  047524  020122
 8276  044724  043111  047040  052117
 8277  044732  031040  030061  020072
 8278  044740  000040
 8279  044742  005015  047111  042524  MSG6:   .ASCIZ  <CR><LF>/INTERRUPT OCCURRED AT PC=/
 8280  044750  051122  050125  020124
 8281  044756  041517  052503  051122
 8282  044764  042105  040440  020124
 8283  044772  041520  000075
 8284  044776  005015  051104  053111  MSG7:   .ASCIZ  <CR><LF>/DRIVE 0 WILL NOT BE TESTED/
 8285  045004  020105  020060  044527
 8286  045012  046114  047040  052117
 8287  045020  041040  020105  042524
 8288  045026  052123  042105     000
 8289  045033     015  051012  040505  MSG8:   .ASCIZ  <CR><LF>/READ DATA WITH OFFSET TEST/<CR><LF>
 8290  045040  020104  040504  040524
 8291  045046  053440  052111  020110
 8292  045054  043117  051506  052105
 8293  045062  052040  051505  006524
 8294  045070  000012
 8295  045072  005015  042510  042101  MSG9:   .ASCIZ  <CR><LF>/HEAD NO./
 8296  045100  047040  027117     000
 8297  045105     015  005012  044527  MSG10:  .ASCIZ  <CR><LF><LF>/WILL TEST DRIVES:/
 8298  045112  046114  052040  051505
 8299  045120  020124  051104  053111
 8300  045126  051505  000072
 8301  045132  005015  050012  053517  MSG11:  .ASCIZ  <CR><LF><LF>/POWER UP RESTART TO TEST 1/<CR><LF>
 8302  045140  051105  052440  020120
 8303  045146  042522  052123  051101
 8304  045154  020124  047524  052040
 8305  045162  051505  020124  006461
 8306  045170  000012
 8307  045172  005015  052012  042510  MSG12:  .ASCII  <CR><LF><LF>/THE ABOVE OFFSET FAILURES ARE NOT ERRORS/
 8308  045200  040440  047502  042526
 8309  045206  047440  043106  042523
 8310  045214  020124  040506  046111
 8311  045222  051125  051505  040440
 8312  045230  042522  047040  052117
 8313  045236  042440  051122  051117
 8314  045244     123
 8315  045245     015  041012  052125          .ASCIZ  <CR><LF>/BUT INDICATORS OF SURFACE,HEAD, & ELECTRONICS QUALITY/<CR><LF>
 8316  045252  044440  042116  041511
 8317  045260  052101  051117  020123
 8318  045266  043117  051440  051125
 8319  045274  040506  042503  044054
 8320  045302  040505  026104  023040
 8321  045310  042440  042514  052103
 8322  045316  047522  044516  051503
 8323  045324  050440  040525  044514
 8324  045332  054524  005015     000
```

```
 8325  045337     015  047012  020117  MSG13:  .ASCII  <CR><LF>/NO L OR P CLOCKS PRESENT/
 8326  045344  020114  051117  050040
 8327  045352  041440  047514  045503
 8328  045360  020123  051120  051505
 8329  045366  047105     124
 8330  045371     015  044012  040505          .ASCIZ  <CR><LF>/HEAD SWITCHING TIME TEST BYPASSED/
 8331  045376  020104  053523  052111
 8332  045404  044103  047111  020107
 8333  045412  044524  042515  052040
 8334  045420  051505  020124  054502
 8335  045426  040520  051523  042105
 8336  045434     000
 8337  045435     015  041012  050131  MSG14:  .ASCIZ  <CR><LF>/BYPASSING DRIVE /
 8338  045442  051501  044523  043516
 8339  045450  042040  044522  042526
 8340  045456  000040
 8341  045460  005015  042012  044522  MSG15:  .ASCIZ  <CR><LF><LF>/DRIVE /
 8342  045466  042526  000040
 8343  045472  005015  051104  053111  MSG16:  .ASCIZ  <CR><LF>/DRIVE SERIAL NO. /
 8344  045500  020105  042523  044522
 8345  045506  046101  047040  027117
 8346  045514  000040
 8347  045516  005015  040503  052122  MSG17:  .ASCIZ  <CR><LF>/CARTRIDGE SERIAL NO. /
 8348  045524  044522  043504  020105
 8349  045532  042523  044522  046101
 8350  045540  047040  027117  000040
 8351  045546  005015  040412  047502  MSG26:  .ASCIZ  <CR><LF><LF>/ABORTING BALANCE OF TESTS ON THIS DRIVE/<CR><LF><LF>
 8352  045554  052122  027111  020107
 8353  045562  040502  040514  041516
 8354  045570  020105  043117  052040
 8355  045576  051505  051524  047440
 8356  045604  020116  044124  051511
 8357  045612  042040  044522  042526
 8358  045620  005015  000012
 8359  045624  005015  040412  046114  MSG27:  .ASCIZ  <CR><LF><LF>/ALL DRIVES TESTED/<CR><LF><LF>
 8360  045632  042040  044522  042526
 8361  045640  020123  042524  052123
 8362  045646  042105  005015  000012
 8363  045654  005015  047516  053440  MSG37:  .ASCIZ  <CR><LF>/NO WRITE CHECK ERROR AT MAX POSITIVE OFFSET/
 8364  045662  044522  042524  041440
 8365  045670  042510  045503  042440
 8366  045676  051122  051117  040440
 8367  045704  020124  040515  020130
 8368  045712  047520  044523  044524
 8369  045720  042526  047440  043106
 8370  045726  042523  000124
 8371  045732  005015  047516  053440  MSG38:  .ASCIZ  <CR><LF>/NO WRITE CHECK ERROR AT MAX NEGATIVE OFFSET/<CR><LF>
 8372  045740  044522  042524  041440
 8373  045746  042510  045503  042440
 8374  045754  051122  051117  040440
 8375  045762  020124  040515  020130
 8376  045770  042516  040507  044524
 8377  045776  042526  047440  043106
 8378  046004  042523  006524  000012
 8379  046012  005015  051127  052111  MSG39:  .ASCIZ  <CR><LF>/WRITE CHECK FAILURE AT OFFSET =/
 8380  046020  020105  044103  041505
```

```
 8381  046026  020113  040506  046111
 8382  046034  051125  020105  052101
 8383  046042  047440  043106  042523
 8384  046050  020124  000075
 8385  046054  005015  047503  046125  MSG40:  .ASCII  <CR><LF>/COULD NOT READ BAD SECTOR INFO ON CYL 410/
 8386  046062  020104  047516  020124
 8387  046070  042522  042101  041040
 8388  046076  042101  051440  041505
 8389  046104  047524  020122  047111
 8390  046112  047506  047440  020116
 8391  046120  054503  020114  030464
 8392  046126     060
 8393  046127     015  047412  020122          .ASCIZ  <CR><LF>/OR ALIGNMENT CARTRIDGE USED/<CR><LF>
 8394  046134  046101  043511  046516
 8395  046142  047105  020124  040503
 8396  046150  052122  044522  043504
 8397  046156  020105  051525  042105
 8398  046164  005015     000
 8399  046167     015  050012  047522  MSG74:  .ASCIZ  <CR><LF>/PROGRAM ABORT PENDING...PLEASE WAIT/
 8400  046174  051107  046501  040440
 8401  046202  047502  052122  050040
 8402  046210  047105  044504  043516
 8403  046216  027056  050056  042514
 8404  046224  051501  020105  040527
 8405  046232  052111     000
 8406  046235     015  044012  046101  MSG75:  .ASCIZ  <CR><LF>/HALT PENDING...PLEASE WAIT/
 8407  046242  020124  042520  042116
 8408  046250  047111  027107  027056
 8409  046256  046120  040505  042523
 8410  046264  053440  044501  000124
 8411  046272  005015  051120  043517  MSG76:  .ASCIZ  <CR><LF>/PROGRAM ABORTED/
 8412  046300  040522  020115  041101
 8413  046306  051117  042524  000104
 8414  046314  005015  050103  020125  MSG77:  .ASCIZ  <CR><LF>/CPU HALTED/
 8415  046322  040510  052114  042105
 8416  046330     000
 8417
 8418                                  .SBTTL  ERROR MESSAGES
 8419
 8420  046331     015  042412  051122  EM1:    .ASCIZ  <CR><LF>/ERROR, ONLY 0 THRU 7 ALLOWED, TRY AGAIN/<CR><LF>
 8421  046336  051117  020054  047117
 8422  046344  054514  030040  052040
 8423  046352  051110  020125  020067
 8424  046360  046101  047514  042527
 8425  046366  026104  052040  054522
 8426  046374  040440  040507  047111
 8427  046402  005015     000
 8428  046405     104  044522  042526  EM2:    .ASCIZ  /DRIVE # IN RKCS2 CANNOT BE READ BACK CORRECTLY IN RKMR2/
 8429  046412  021440  044440  020116
 8430  046420  045522  051503  020062
 8431  046426  040503  047116  052117
 8432  046434  041040  020105  042522
 8433  046442  042101  041040  041501
 8434  046450  020113  047503  051122
 8435  046456  041505  046124  020131
 8436  046464  047111  051040  046513
```

```
 8437  046472  031122     000
 8438  046475     015  040412  047502  EM3:    .ASCIZ  <CR><LF>/ABORT TESTS...UNEXPECTED TIME OUT AT PC=/
 8439  046502  052122  052040  051505
 8440  046510  051524  027056  052456
 8441  046516  042516  050130  041505
 8442  046524  042524  020104  044524
 8443  046532  042515  047440  052125
 8444  046540  040440  020124  041520
 8445  046546  000075
 8446  046550  005015  041101  051117  EM4:    .ASCIZ  <CR><LF>/ABORT TESTS...UNEXPECTED INTERRUPT AT PC=/
 8447  046556  020124  042524  052123
 8448  046564  027123  027056  047125
 8449  046572  054105  042520  052103
 8450  046600  042105  044440  052116
 8451  046606  051105  052522  052120
 8452  046614  040440  020124  041520
 8453  046622  000075
 8454  046624  042115  020123  042523  EM5:    .ASCIZ  /MDS SET IN RKCS2/
 8455  046632  020124  047111  051040
 8456  046640  041513  031123     000
 8457  046645     125  042506  051440  EM6:    .ASCIZ  /UFE SET IN RKCS2/
 8458  046652  052105  044440  020116
 8459  046660  045522  051503  000062
 8460  046666  051104  020101  047111  EM7:    .ASCIZ  /DRA IN RKDS & NED IN RKCS2 RESET; WRONG PORT SELECTED?/
 8461  046674  051040  042113  020123
 8462  046702  020046  042516  020104
 8463  046710  047111  051040  041513
 8464  046716  031123  051040  051505
 8465  046724  052105  020073  051127
 8466  046732  047117  020107  047520
 8467  046740  052122  051440  046105
 8468  046746  041505  042524  037504
 8469  046754     000
 8470  046755     104  044522  042526  EM8:    .ASCIZ  /DRIVE PRESENT BUT NOT SPECIFIED BY OPERATOR/
 8471  046762  050040  042522  042523
 8472  046770  052116  041040  052125
 8473  046776  047040  052117  051440
 8474  047004  042520  044503  044506
 8475  047012  042105  041040  020131
 8476  047020  050117  051105  052101
 8477  047026  051117     000
 8478  047031     104  044522  042526  EM9:    .ASCIZ  /DRIVE NOT PRESENT BUT SPECIFIED BY OPERATOR/
 8479  047036  047040  052117  050040
 8480  047044  042522  042523  052116
 8481  047052  041040  052125  051440
 8482  047060  042520  044503  044506
 8483  047066  042105  041040  020131
 8484  047074  050117  051105  052101
 8485  047102  051117     000
 8486  047105     101  047502  052122  EM10:   .ASCIZ  /ABORT TESTS...CANNOT REFERENCE CONTROLLER REGISTER/
 8487  047112  052040  051505  051524
 8488  047120  027056  041456  047101
 8489  047126  047516  020124  042522
 8490  047134  042506  042522  041516
 8491  047142  020105  047503  052116
 8492  047150  047522  046114  051105
```

```
 8493  047156  051040  043505  051511
 8494  047164  042524  000122
 8495  047170  051104  020101  047111  EM11:   .ASCIZ  /DRA IN RKDS & NED IN RKCS2 BOTH SET/
 8496  047176  051040  042113  020123
 8497  047204  020046  042516  020104
 8498  047212  047111  051040  041513
 8499  047220  031123  041040  052117
 8500  047226  020110  042523  000124
 8501  047234  047503  052116  047522  EM12:   .ASCIZ  /CONTROLLER NOT READY IN RKCS1/
 8502  047242  046114  051105  047040
 8503  047250  052117  051040  040505
 8504  047256  054504  044440  020116
 8505  047264  045522  051503  000061
 8506  047272  047516  040440  052124  EM13:   .ASCIZ  /NO ATTN IN RKASOF/
 8507  047300  020116  047111  051040
 8508  047306  040513  047523  000106
 8509  047314  047125  054105  042520  EM14:   .ASCIZ  /UNEXPECTED MEMORY PARITY TRAP/
 8510  047322  052103  042105  046440
 8511  047330  046505  051117  020131
 8512  047336  040520  044522  054524
 8513  047344  052040  040522  000120
 8514  047352  045522  047504  023040  EM15:   .ASCII  /RKDC & RKDA INDICATE THAT WCE OCCURRED AT/
 8515  047360  051040  042113  020101
 8516  047366  047111  044504  040503
 8517  047374  042524  052040  040510
 8518  047402  020124  041527  020105
 8519  047410  041517  052503  051122
 8520  047416  042105  040440     124
 8521  047423     015  041412  046131          .ASCIZ  <CR><LF>/CYL 411, TRACK 2, SECTOR 21/
 8522  047430  032040  030461  020054
 8523  047436  051124  041501  020113
 8524  047444  026062  051440  041505
 8525  047452  047524  020122  030462
 8526  047460     000
 8527  047461     103  047101  047516  EM16:   .ASCIZ  /CANNOT READ BAD SECTOR INFORMATION/
 8528  047466  020124  042522  042101
 8529  047474  041040  042101  051440
 8530  047502  041505  047524  020122
 8531  047510  047111  047506  046522
 8532  047516  052101  047511  000116
 8533  047524  042515  051523  043501  EM17:   .ASCIZ  /MESSAGE A0 ERROR/
 8534  047532  020105  030101  042440
 8535  047540  051122  051117     000
 8536  047545     115  051505  040523  EM18:   .ASCIZ  /MESSAGE B0 ERROR/
 8537  047552  042507  041040  020060
 8538  047560  051105  047522  000122
 8539  047566  042515  051523  043501  EM19:   .ASCIZ  /MESSAGE A1 ERROR/
 8540  047574  020105  030501  042440
 8541  047602  051122  051117     000
 8542  047607     115  051505  040523  EM20:   .ASCIZ  /MESSAGE B1 ERROR/
 8543  047614  042507  041040  020061
 8544  047622  051105  047522  000122
 8545  047630  042503  051122  051440  EM21:   .ASCIZ  /CERR SET IN RKCS1/
 8546  047636  052105  044440  020116
 8547  047644  045522  051503  000061
 8548  047652  047516  042040  044522  EM22:   .ASCII  /NO DRIVES FOUND IN DEVICE MAP ($DEVM)/<CR><LF>
```

```
 8549  047660  042526  020123  047506
 8550  047666  047125  020104  047111
 8551  047674  042040  053105  041511
 8552  047702  020105  040515  020120
 8553  047710  022050  042504  046526
 8554  047716  006451     012
 8555  047721     123  052105  050125          .ASCIZ  /SETUP CORRECTLY AND RESTART/<CR><LF>
 8556  047726  041440  051117  042522
 8557  047734  052103  054514  040440
 8558  047742  042116  051040  051505
 8559  047750  040524  052122  005015
 8560  047756     000
 8561  047757     116  020117  051104  EM23:   .ASCII  /NO DRIVES FOUND ON BUSS/<CR><LF>
 8562  047764  053111  051505  043040
 8563  047772  052517  042116  047440
 8564  050000  020116  052502  051523
 8565  050006  005015
 8566  050010  042523  052524  020120          .ASCIZ  /SETUP CORRECTLY AND PRESS 'CONTINUE'/<CR><LF>
 8567  050016  047503  051122  041505
 8568  050024  046124  020131  047101
 8569  050032  020104  051120  051505
 8570  050040  020123  041447  047117
 8571  050046  044524  052516  023505
 8572  050054  005015     000
 8573  050057     126  046117  053040  EM24:   .ASCIZ  /VOL VALID NOT SET IN RKMR2/
 8574  050064  046101  042111  047040
 8575  050072  052117  051440  052105
 8576  050100  044440  020116  045522
 8577  050106  051115  000062
 8578  050112  005015  042504  042524  EM25:   .ASCIZ  <CR><LF>/DETECTED 10 BAD SECTORS...ABORTING TEST/
 8579  050120  052103  042105  030440
 8580  050126  020060  040502  020104
 8581  050134  042523  052103  051117
 8582  050142  027123  027056  041101
 8583  050150  051117  044524  043516
 8584  050156  052040  051505  000124
 8585  050164  042504  042524  052103  EM26:   .ASCIZ  /DETECTED BSE BUT NOT LISTED IN BAD SECTOR FILE/
 8586  050172  042105  041040  042523
 8587  050200  041040  052125  047040
 8588  050206  052117  046040  051511
 8589  050214  042524  020104  047111
 8590  050222  041040  042101  051440
 8591  050230  041505  047524  020122
 8592  050236  044506  042514     000
 8593  050243     104  052105  041505  EM27:   .ASCII  /DETECTED BSE IN READ COMMAND/
 8594  050250  042524  020104  051502
 8595  050256  020105  047111  051040
 8596  050264  040505  020104  047503
 8597  050272  046515  047101     104
 8598  050277     015  041012  052125          .ASCIZ  <CR><LF>/BUT NOT IN PREVIOUS WRITE COMMAND TO SAME SECTOR/
 8599  050304  047040  052117  044440
 8600  050312  020116  051120  053105
 8601  050320  047511  051525  053440
 8602  050326  044522  042524  041440
 8603  050334  046517  040515  042116
 8604  050342  052040  020117  040523
```

```
 8605  050350  042515  051440  041505
 8606  050356  047524  000122
 8607  050362  054503  020114  042101  EM36:   .ASCIZ  /CYL ADDR IN RKMR3 NOT SAME AS RKDC/
 8608  050370  051104  044440  020116
 8609  050376  045522  051115  020063
 8610  050404  047516  020124  040523
 8611  050412  042515  040440  020123
 8612  050420  045522  041504     000
 8613  050423     103  046131  042040  EM39:   .ASCIZ  /CYL DIFF & OFFSET IN RKMR2 NOT CLEARED/
 8614  050432  043111  020106  020046
 8615  050440  043117  051506  052105
 8616  050446  044440  020116  045522
 8617  050454  051115  020062  047516
 8618  050462  020124  046103  040505
 8619  050470  042522  000104
 8620  050474  054503  020114  042101  EM40:   .ASCIZ  /CYL ADDR IN RKMR3 NOT CLEARED/
 8621  050502  051104  044440  020116
 8622  050510  045522  051115  020063
 8623  050516  047516  020124  046103
 8624  050524  040505  042522  000104
 8625  050532  054503  020114  042101  EM41:   .ASCIZ  /CYL ADDR IN B2 DID NOT REMAIN CLEARED/
 8626  050540  051104  044440  020116
 8627  050546  031102  042040  042111
 8628  050554  047040  052117  051040
 8629  050562  046505  044501  020116
 8630  050570  046103  040505  042522
 8631  050576  000104
 8632  050600  052101  047124  047040  EM55:   .ASCIZ  /ATTN NOT CLEARED IN RKASOF/
 8633  050606  052117  041440  042514
 8634  050614  051101  042105  044440
 8635  050622  020116  045522  051501
 8636  050630  043117     000
 8637  050633     104  052114  051440  EM63:   .ASCIZ  /DLT SET IN RKCS2/
 8638  050640  052105  044440  020116
 8639  050646  045522  051503  000062
 8640  050654  042522  042101  044040  EM65:   .ASCIZ  /READ HEADER ERROR/
 8641  050662  040505  042504  020122
 8642  050670  051105  047522  000122
 8643  050676  046101  043511  046516  EM69:   .ASCIZ  /ALIGNMENT CARTRIDGE USED/
 8644  050704  047105  020124  040503
 8645  050712  052122  044522  043504
 8646  050720  020105  051525  042105
 8647  050726     000
 8648  050727     103  047524  051440  EM73:   .ASCIZ  /CTO SET IN RKCS1/
 8649  050734  052105  044440  020116
 8650  050742  045522  051503  000061
 8651  050750  052122  020132  047516  EM74:   .ASCIZ  /RTZ NOT SET IN RKMR2/
 8652  050756  020124  042523  020124
 8653  050764  047111  051040  046513
 8654  050772  031122     000
 8655  050775     116  042105  051440  EM79:   .ASCIZ  /NED SET IN RKCS2/
 8656  051002  052105  044440  020116
 8657  051010  045522  051503  000062
 8658  051016  051127  052111  020105  EM80:   .ASCIZ  /WRITE CHECK ERROR SET IN RKCS2/
 8659  051024  044103  041505  020113
 8660  051032  051105  047522  020122
```

```
 8661  051040  042523  020124  047111
 8662  051046  051040  041513  031123
 8663  051054     000
 8664  051055     127  044522  042524  EM81:   .ASCIZ  /WRITE CHECK COMMAND NOT FUNCTIONING/
 8665  051062  041440  042510  045503
 8666  051070  041440  046517  040515
 8667  051076  042116  047040  052117
 8668  051104  043040  047125  052103
 8669  051112  047511  044516  043516
 8670  051120     000
 8671  051121     122  040505  020104  EM82:   .ASCIZ  /READ DATA DID NOT COMPARE WITH WRITE DATA/
 8672  051126  040504  040524  042040
 8673  051134  042111  047040  052117
 8674  051142  041440  046517  040520
 8675  051150  042522  053440  052111
 8676  051156  020110  051127  052111
 8677  051164  020105  040504  040524
 8678  051172     000
 8679  051173     104  052101  020101  EM83:   .ASCIZ  /DATA CHECK ERROR SET IN RKER/
 8680  051200  044103  041505  020113
 8681  051206  051105  047522  020122
 8682  051214  042523  020124  047111
 8683  051222  051040  042513  000122
 8684  051230  044127  046111  020105  EM84:   .ASCIZ  /WHILE WAITING FOR CONTR READY OR AFTER CONTR READY REC'D/
 8685  051236  040527  052111  047111
 8686  051244  020107  047506  020122
 8687  051252  047503  052116  020122
 8688  051260  042522  042101  020131
 8689  051266  051117  040440  052106
 8690  051274  051105  041440  047117
 8691  051302  051124  051040  040505
 8692  051310  054504  051040  041505
 8693  051316  042047     000
 8694  051321     117  043106  042523  EM85:   .ASCIZ  /OFFSET STATUS BIT IN RKMR2 CLEARED/
 8695  051326  020124  052123  052101
 8696  051334  051525  041040  052111
 8697  051342  044440  020116  045522
 8698  051350  051115  020062  046103
 8699  051356  040505  042522  000104
 8700  051364  043117  051506  052105  EM86:   .ASCIZ  /OFFSET REG IN A2 NOT = RKASOF/
 8701  051372  051040  043505  044440
 8702  051400  020116  031101  047040
 8703  051406  052117  036440  051040
 8704  051414  040513  047523  000106
 8705  051422  044504  020104  047516  EM88:   .ASCIZ  /DID NOT FIND SECTOR 0 FROM INDEX/
 8706  051430  020124  044506  042116
 8707  051436  051440  041505  047524
 8708  051444  020122  020060  051106
 8709  051452  046517  044440  042116
 8710  051460  054105     000
 8711  051463     122  040505  044504  EM93:   .ASCIZ  /READING WRONG CYLINDER # IN HEADER...MISPOSITION/
 8712  051470  043516  053440  047522
 8713  051476  043516  041440  046131
 8714  051504  047111  042504  020122
 8715  051512  020043  047111  044040
 8716  051520  040505  042504  027122
```

```
 8717  051526  027056  044515  050123
 8718  051534  051517  052111  047511
 8719  051542  000116
 8720  051544  043117  051506  052105  EM94:   .ASCIZ  /OFFSET IT IN A2 NOT CLEARED/
 8721  051552  044440  020124  047111
 8722  051560  040440  020062  047516
 8723  051566  020124  046103  040505
 8724  051574  042522  000104
 8725  051600  047506  046522  052101  EM95:   .ASCIZ  /FORMAT BIT NOT SET IN RKMR2/
 8726  051606  041040  052111  047040
 8727  051614  052117  051440  052105
 8728  051622  044440  020116  045522
 8729  051630  051115  000062
 8730  051634  040503  047116  052117  EM96:   .ASCIZ  /CANNOT FIND SECTOR 23(8)/
 8731  051642  043040  047111  020104
 8732  051650  042523  052103  051117
 8733  051656  031040  024063  024470
 8734  051664     000
 8735  051665     110  040505  020104  EM97:   .ASCIZ  /HEAD SWITCHING REQ'D ANOTHER FULL REVOLUTION OF DISK/
 8736  051672  053523  052111  044103
 8737  051700  047111  020107  042522
 8738  051706  023521  020104  047101
 8739  051714  052117  042510  020122
 8740  051722  052506  046114  051040
 8741  051730  053105  046117  052125
 8742  051736  047511  020116  043117
 8743  051744  042040  051511  000113
 8744  051752  051104  053111  020105  EM100:  .ASCIZ  /DRIVE OFF TRACK SET IN RKMR3/
 8745  051760  043117  020106  051124
 8746  051766  041501  020113  042523
 8747  051774  020124  047111  051040
 8748  052002  046513  031522     000
 8749  052007     104  042111  047040  EM101:  .ASCIZ  /DID NOT GO TO CYLINDER 10/
 8750  052014  052117  043440  020117
 8751  052022  047524  041440  046131
 8752  052030  047111  042504  020122
 8753  052036  030061     000
 8754
 8755                                   .SBTTL  DATA HEADERS
 8756
 8757  052041     124  051505  020124  DH1:    .ASCIZ  /TEST NO.  PC/
 8758  052046  047516  020056  050040
 8759  052054  000103
 8760  052056  045522  051115  004461  DH2:    .ASCIZ  /RKMR1  RKMR2   RKMR3   RKER    RKDS    RKCS1   RKCS2/
 8761  052064  045522  051115  004462
 8762  052072  045522  051115  004463
 8763  052100  045522  051105  051011
 8764  052106  042113  004523  045522
 8765  052114  051503  004461  045522
 8766  052122  051503  000062
 8767  052126  045522  041527  051011  DH3:    .ASCIZ  /RKWC   RKBA    RKDA    RKASOF  RKDC    RKECPS  RKECPT/
 8768  052134  041113  004501  045522
 8769  052142  040504  051011  040513
 8770  052150  047523  004506  045522
 8771  052156  041504  051011  042513
 8772  052164  050103  004523  045522
```

```
 8773  052172  041505  052120     000
 8774  052177     106  047522  020115  DH6:    .ASCIZ  /FROM CYL  TO CYL  CYL DIFF/
 8775  052204  054503  020114  052040
 8776  052212  020117  054503  020114
 8777  052220  041440  046131  042040
 8778  052226  043111  000106
 8779  052232  042524  052123  047040  DH8:    .ASCIZ  /TEST NO.       TRAP PC/
 8780  052240  027117  052011  040522
 8781  052246  020120  041520     000
 8782  052253     101  052106  051105  DH9:    .ASCIZ  /AFTER START SPINDLE COMMAND REC'D BY DRIVE/
 8783  052260  051440  040524  052122
 8784  052266  051440  044520  042116
 8785  052274  042514  041440  046517
 8786  052302  040515  042116  051040
 8787  052310  041505  042047  041040
 8788  052316  020131  051104  053111
 8789  052324  000105
 8790  052326  052101  042440  042116  DH10:   .ASCIZ  /AT END OF HEAD LOADING/
 8791  052334  047440  020106  042510
 8792  052342  042101  046040  040517
 8793  052350  044504  043516     000
 8794  052355     105  050130  041505  DH11:   .ASCIZ  /EXPECTED       WAS/
 8795  052362  042524  004504  040527
 8796  052370  000123
 8797  052372  047117  051440  041505  DH13:   .ASCIZ  /ON SECTORS 10, 12, 14, 16, 18 OR 20 CYL 410 TRACK 2/
 8798  052400  047524  051522  030440
 8799  052406  026060  030440  026062
 8800  052414  030440  026064  030440
 8801  052422  026066  030440  020070
 8802  052430  051117  031040  020060
 8803  052436  054503  020114  030464
 8804  052444  020060  051124  041501
 8805  052452  020113  000062
 8806  052456  047117  051440  041505  DH14:   .ASCIZ  /ON SECTORS 11, 13, 15, 17, 19 OR 21 CYL 410 TRACK 2/
 8807  052464  047524  051522  030440
 8808  052472  026061  030440  026063
 8809  052500  030440  026065  030440
 8810  052506  026067  030440  020071
 8811  052514  051117  031040  020061
 8812  052522  054503  020114  030464
 8813  052530  020060  051124  041501
 8814  052536  020113  000062
 8815  052542  043101  042524  020122  DH17:   .ASCIZ  /AFTER RECAL COMMAND/
 8816  052550  042522  040503  020114
 8817  052556  047503  046515  047101
 8818  052564  000104
 8819  052566  043101  042524  020122  DH19:   .ASCIZ  /AFTER PACK COMMAND/
 8820  052574  040520  045503  041440
 8821  052602  046517  040515  042116
 8822  052610     000
 8823  052611     101  052106  051105  DH20:   .ASCIZ  /AFTER SELECT DRIVE COMMAND/
 8824  052616  051440  046105  041505
 8825  052624  020124  051104  053111
 8826  052632  020105  047503  046515
 8827  052640  047101  000104
 8828  052644  043101  042524  020122  DH21:   .ASCIZ  /AFTER SUBSYSTEM CLEAR/
```

```
 8829  052652  052523  051502  051531
 8830  052660  042524  020115  046103
 8831  052666  040505  000122
 8832  052672  043101  042524  020122  DH22:   .ASCIZ  /AFTER DRIVE CLEAR COMMAND/
 8833  052700  051104  053111  020105
 8834  052706  046103  040505  020122
 8835  052714  047503  046515  047101
 8836  052722  000104
 8837  052724  043101  042524  020122  DH24:   .ASCIZ  /AFTER OFFSET COMMAND/
 8838  052732  043117  051506  052105
 8839  052740  041440  046517  040515
 8840  052746  042116     000
 8841  052751     101  052106  051105  DH25:   .ASCIZ  /AFTER SEEK COMMAND/
 8842  052756  051440  042505  020113
 8843  052764  047503  046515  047101
 8844  052772  000104
 8845  052774  043101  042524  020122  DH26:   .ASCIZ  /AFTER READ DATA COMMAND/
 8846  053002  042522  042101  042040
 8847  053010  052101  020101  047503
 8848  053016  046515  047101  000104
 8849  053024  043101  042524  020122  DH27:   .ASCIZ  /AFTER WRITE DATA COMMAND/
 8850  053032  051127  052111  020105
 8851  053040  040504  040524  041440
 8852  053046  046517  040515  042116
 8853  053054     000
 8854  053055     011  054105  042520  DH28:   .ASCIZ  /       EXPECT/
 8855  053062  052103     000
 8856  053065     011  041501  052524  DH29:   .ASCIZ  /       ACTUAL/
 8857  053072  046101     000
 8858  053075     101  052106  051105  DH30:   .ASCIZ  /AFTER READ HEADER COMMAND/
 8859  053102  051040  040505  020104
 8860  053110  042510  042101  051105
 8861  053116  041440  046517  040515
 8862  053124  042116     000
 8863  053127     101  004460  030102  DH31:   .ASCIZ  /A0     B0      A1      B1      A2      B2      B3/
 8864  053134  040411  004461  030502
 8865  053142  040411  004462  031102
 8866  053150  041011  000063
 8867  053154  043101  042524  020122  DH32:   .ASCIZ  /AFTER WRITE CHECK COMMAND/
 8868  053162  051127  052111  020105
 8869  053170  044103  041505  020113
 8870  053176  047503  046515  047101
 8871  053204  000104
 8872  053206  043101  042524  020122  DH39:   .ASCIZ  /AFTER WRITE HEADER COMMAND/
 8873  053214  051127  052111  020105
 8874  053222  042510  042101  051105
 8875  053230  041440  046517  040515
 8876  053236  042116     000
 8877  053241     127  051117  021504  DH40:   .ASCIZ  /WORD#  HEADER WAS  SHOULD BE/
 8878  053246  044011  040505  042504
 8879  053254  020122  040527  020123
 8880  053262  051440  047510  046125
 8881  053270  020104  042502     000
 8882  053275     104  051125  047111  DH41:   .ASCIZ  /DURING RECAL COMMAND/
 8883  053302  020107  042522  040503
 8884  053310  020114  047503  046515
```

```
 8885  053316  047101  000104
 8886  053322  047117  051440  041505  DH42:   .ASCIZ  /ON SECTORS 0,2,4,6 OR 8  CYL 410 TRACK 2/
 8887  053330  047524  051522  030040
 8888  053336  031054  032054  033054
 8889  053344  047440  020122  020070
 8890  053352  041440  046131  032040
 8891  053360  030061  052040  040522
 8892  053366  045503  031040     000
 8893  053373     117  020116  042523  DH43:   .ASCIZ  /ON SECTORS 1,3,5,7 OR 9  CYL 410 TRACK 2/
 8894  053400  052103  051117  020123
 8895  053406  026061  026063  026065
 8896  053414  020067  051117  034440
 8897  053422  020040  054503  020114
 8898  053430  030464  020060  051124
 8899  053436  041501  020113  000062
 8900  053444  047506  046522  052101  DH44:   .ASCIZ  /FORMAT & ALL READ-WRITE TESTS WILL BE BYPASSED/
 8901  053452  023040  040440  046114
 8902  053460  051040  040505  026504
 8903  053466  051127  052111  020105
 8904  053474  042524  052123  020123
 8905  053502  044527  046114  041040
 8906  053510  020105  054502  040520
 8907  053516  051523  042105     000
 8908  053523     101  052106  051105  DH47:   .ASCIZ  /AFTER READ HEADER COMMAND WITH MOVEMENT/
 8909  053530  051040  040505  020104
 8910  053536  042510  042101  051105
 8911  053544  041440  046517  040515
 8912  053552  042116  053440  052111
 8913  053560  020110  047515  042526
 8914  053566  042515  052116     000
 8915  053573     115  043523  040440  DH49:   .ASCIZ  /MSG A & B IN RKMR2 & RKMR3 RESP. ARE INVALID/
 8916  053600  023040  041040  044440
 8917  053606  020116  045522  051115
 8918  053614  020062  020046  045522
 8919  053622  051115  020063  042522
 8920  053630  050123  020056  051101
 8921  053636  020105  047111  040526
 8922  053644  044514  000104
 8923  053650  043101  042524  020122  DH51:   .ASCIZ  /AFTER SEEK TO SELF COMMAND/
 8924  053656  042523  045505  052040
 8925  053664  020117  042523  043114
 8926  053672  041440  046517  040515
 8927  053700  042116     000
 8928  053703     127  052111  020110  DH52:   .ASCIZ  /WITH INTENTIONAL MISCOMPARE/
 8929  053710  047111  042524  052116
 8930  053716  047511  040516  020114
 8931  053724  044515  041523  046517
 8932  053732  040520  042522     000
 8933  053737     104  051125  047111  DH53:   .ASCIZ  /DURING OFFSET COMMAND/
 8934  053744  020107  043117  051506
 8935  053752  052105  041440  046517
 8936  053760  040515  042116     000
 8937  053765     101  052106  051105  DH54:   .ASCIZ  /AFTER FORMAT CHANGE AND CONTR READY REC'D/
 8938  053772  043040  051117  040515
 8939  054000  020124  044103  047101
 8940  054006  042507  040440  042116
```

```
 8941  054014  041440  047117  051124
 8942  054022  051040  040505  054504
 8943  054030  051040  041505  042047
 8944  054036     000
 8945  054037     103  046131  021440  DH56:   .ASCIZ  /CYL #  HEADER WORD 0/
 8946  054044  044011  040505  042504
 8947  054052  020122  047527  042122
 8948  054060  030040     000
 8949  054063     101  052106  051105  DH57:   .ASCIZ  /AFTER WRITE COMMAND WITH OFFSET/
 8950  054070  053440  044522  042524
 8951  054076  041440  046517  040515
 8952  054104  042116  053440  052111
 8953  054112  020110  043117  051506
 8954  054120  052105     000
 8955
 8956                           .SBTTL  ERROR OUTPUT DATA
 8957
 8958          054124                   .EVEN
 8959  054124  001214  001116           DT1:    $TESTN,$ERRPC
 8960  054130  007400  007402  007404           HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
 8961  054136  007370  007366  007354
 8962  054144  007356
 8963  054146  007360  007362  007364           HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
 8964  054154  007372  007374  007406
 8965  054162  007410
 8966  054164  001214  001334           DT3:    $TESTN,TRAPPC
 8967  054170  001214  001116  001344   DT4:    $TESTN,$ERRPC,FRCYL,TOCYL,CALDIF
 8968  054176  001346  001354
 8969  054202  007400  007402  007404           HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
 8970  054210  007370  007366  007354
 8971  054216  007356
 8972  054220  007360  007362  007364           HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
 8973  054226  007372  007374  007406
 8974  054234  007410
 8975  054236  001214  001116  001454   DT6:    $TESTN,$ERRPC,WD2,WD1
 8976  054244  001452
 8977  054246  007400  007402  007404           HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
 8978  054254  007370  007366  007354
 8979  054262  007356
 8980  054264  007360  007362  007364           HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
 8981  054272  007372  007374  007406
 8982  054300  007410
 8983  054302  001214  001116  001472   DT7:    $TESTN,$ERRPC,WDCNT,HDWD,TEMP1
 8984  054310  001510  007412
 8985  054314  007400  007402  007404           HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
 8986  054322  007370  007366  007354
 8987  054330  007356
 8988  054332  007360  007362  007364           HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
 8989  054340  007372  007374  007406
 8990  054346  007410
 8991  054350  001214  001116  001346   DT8:    $TESTN,$ERRPC,TOCYL,FRCYL,CALDIF
 8992  054356  001344  001354
 8993  054362  007400  007402  007404           HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
 8994  054370  007370  007366  007354
 8995  054376  007356
 8996  054400  007360  007362  007364           HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
```

```
 8997  054406  007372  007374  007406
 8998  054414  007410
 8999  054416  001214  001116  001346  DT9:    $TESTN,$ERRPC,TOCYL,RHTAB
 9000  054424  001726
 9001  054426  007400  007402  007404          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
 9002  054434  007370  007366  007354
 9003  054442  007356
 9004  054444  007360  007362  007364          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
 9005  054452  007372  007374  007406
 9006  054460  007410
 9007  054462  001214  001116  007444  DT13:   $TESTN,$ERRPC,E.A0,E.B0,E.A1,E.B1,H.A0,H.B0,H.A1,H.B1
 9008  054470  007446  007450  007452
 9009  054476  007424  007426  007430
 9010  054504  007432
 9011  054506  007400  007402  007404          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
 9012  054514  007370  007366  007354
 9013  054522  007356
 9014  054524  007360  007362  007364          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
 9015  054532  007372  007374  007406
 9016  054540  007410
 9017  054542  001214  001116  007444  DT14:   $TESTN,$ERRPC,E.A0,E.B0,E.A1,E.B1,E.A2,E.B2
 9018  054550  007446  007450  007452
 9019  054556  007454  007456
 9020  054562  007424  007426  007430          H.A0,H.B0,H.A1,H.B1,H.A2,H.B2
 9021  054570  007432  007434  007436
 9022  054576  007400  007402  007404          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
 9023  054604  007370  007366  007354
 9024  054612  007356
 9025  054614  007360  007362  007364          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
 9026  054622  007372  007374  007406
 9027  054630  007410
 9028  054632  001214  001116  007444  DT15:   $TESTN,$ERRPC,E.A0,E.B0,E.A1,E.B1,E.A2,E.B2,E.B3
 9029  054640  007446  007450  007452
 9030  054646  007454  007456  007462
 9031  054654  007424  007426  007430          H.A0,H.B0,H.A1,H.B1,H.A2,H.B2,H.B3
 9032  054662  007432  007434  007436
 9033  054670  007442
 9034  054672  007400  007402  007404          HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
 9035  054700  007370  007366  007354
 9036  054706  007356
 9037  054710  007360  007362  007364          HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
 9038  054716  007372  007374  007406
 9039  054724  007410
 9040
 9041                                  .SBTTL  ERROR DATA FORMATS
 9042
 9043  054726  000003                  DF1:    3
 9044  054730     002     000                  .BYTE 2,0
 9045  054732  052056                          DH2
 9046  054734     007     000                  .BYTE 7,0
 9047  054736  052126                          DH3
 9048  054740     007     000                  .BYTE   7,0
 9049
 9050  054742  000001                  DF2:    1
 9051  054744     002     000                  .BYTE   2,0
 9052
```

```
 9053  054746  000005                  DF3:    5
 9054  054750     000      000                 .BYTE   0,0
 9055  054752  052041                          DH1
 9056  054754     002      000                 .BYTE   2,0
 9057  054756  052355                          DH11
 9058  054760     002      000                 .BYTE   2,0
 9059  054762  052056                          DH2
 9060  054764     007      000                 .BYTE   7,0
 9061  054766  052126                          DH3
 9062  054770     007      000                 .BYTE   7,0
 9063
 9064  054772  000003                  DF4:    3
 9065  054774     002      000                 .BYTE   2,0
 9066  054776  052056                          DH2
 9067  055000     007      000                 .BYTE   7,0
 9068  055002  052126                          DH3
 9069  055004     007      000                 .BYTE   7,0
 9070
 9071  055006  000005                  DF5:    5
 9072  055010     000      000                 .BYTE   0,0
 9073  055012  053573                          DH49
 9074  055014     000      000                 .BYTE   0,0
 9075  055016  052041                          DH1
 9076  055020     002      000                 .BYTE   2,0
 9077  055022  052056                          DH2
 9078  055024     007      000                 .BYTE   7,0
 9079  055026  052126                          DH3
 9080  055030     007      000                 .BYTE   7,0
 9081
 9082  055032  000005                  DF6:    5
 9083  055034     000      000                 .BYTE 0,0
 9084  055036  052041                          DH1
 9085  055040     002      000                 .BYTE 2,0
 9086  055042  052177                          DH6
 9087  055044     003      000                 .BYTE 3,0
 9088  055046  052056                          DH2
 9089  055050     007      000                 .BYTE   7,0
 9090  055052  052126                          DH3
 9091  055054     007      000                 .BYTE   7,0
 9092
 9093
 9094  055056  000004                  DF10:   4
 9095  055060     000      000                 .BYTE 0,0
 9096  055062  052041                          DH1
 9097  055064     002      000                 .BYTE 2,0
 9098  055066  052056                          DH2
 9099  055070     007      000                 .BYTE 7,0
 9100  055072  052126                          DH3
 9101  055074     007      000                 .BYTE   7,0
 9102
 9103  055076  000004                  DF14:   4
 9104  055100     002      000                 .BYTE   2,0
 9105  055102  053241                          DH40
 9106  055104     003      000                 .BYTE   3,0
 9107  055106  052056                          DH2
 9108  055110     007      000                 .BYTE   7,0
```

```
 9109  055112  052126                        DH3
 9110  055114     007     000                .BYTE   7,0
 9111
 9112
 9113  055116  000004                DF15:   4
 9114  055120     000     000                .BYTE   0,0
 9115  055122  052041                        DH1
 9116  055124     002     000                .BYTE   2,0
 9117  055126  052056                        DH2
 9118  055130     007     000                .BYTE   7,0
 9119  055132  052126                        DH3
 9120  055134     007     000                .BYTE   7,0
 9121
 9122  055136  000005                DF17:   5
 9123  055140     000     000                .BYTE   0,0
 9124  055142  053444                        DH44
 9125  055144     000     000                .BYTE   0,0
 9126  055146  052041                        DH1
 9127  055150     002     000                .BYTE   2,0
 9128  055152  052056                        DH2
 9129  055154     007     000                .BYTE   7,0
 9130  055156  052126                        DH3
 9131  055160     007     000                .BYTE   7,0
 9132  055162  000005                DF20:   5
 9133  055164     000     000                .BYTE   0,0
 9134  055166  052041                        DH1
 9135  055170     002     000                .BYTE   2,0
 9136  055172  054037                        DH56
 9137  055174     002     000                .BYTE   2,0
 9138  055176  052056                        DH2
 9139  055200     007     000                .BYTE   7,0
 9140  055202  052126                        DH3
 9141  055204     007     000                .BYTE   7,0
 9142
 9143  055206  000007                DF21:   7
 9144  055210     000     000                .BYTE   0,0
 9145  055212  052041                        DH1
 9146  055214     002     000                .BYTE   2,0
 9147  055216  053055                        DH28
 9148  055220     000     000                .BYTE   0,0
 9149  055222  053127                        DH31
 9150  055224     004     000                .BYTE   4,0
 9151  055226  053065                        DH29
 9152  055230     004     000                .BYTE   4,0
 9153  055232  052056                        DH2
 9154  055234     007     000                .BYTE   7,0
 9155  055236  052126                        DH3
 9156  055240     007     000                .BYTE   7,0
 9157
 9158  055242  000007                DF22:   7
 9159  055244     000     000                .BYTE   0,0
 9160  055246  052041                        DH1
 9161  055250     002     000                .BYTE   2,0
 9162  055252  053055                        DH28
 9163  055254     000     000                .BYTE   0,0
 9164  055256  053127                        DH31
```

```
9165  055260     006     000                .BYTE   6,0
9166  055262  053065                        DH29
9167  055264     006     000                .BYTE   6,0
9168  055266  052056                        DH2
9169  055270     007     000                .BYTE   7,0
9170  055272  052126                        DH3
9171  055274     007     000                .BYTE   7,0
9172
9173  055276  000007             DF23:      7
9174  055300     000     000                .BYTE   0,0
9175  055302  052041                        DH1
9176  055304     002     000                .BYTE   2,0
9177  055306  053055                        DH28
9178  055310     000     000                .BYTE   0,0
9179  055312  053127                        DH31
9180  055314     007     000                .BYTE   7,0
9181  055316  053065                        DH29
9182  055320     007     000                .BYTE   7,0
9183  055322  052056                        DH2
9184  055324     007     000                .BYTE   7,0
9185  055326  052126                        DH3
9186  055330     007     000                .BYTE   7,0
9187                               .EVEN
9188                               ;;*****************************************************************************
9189                               .SBTTL  TYPE ERROR ROUTINE
9190                               ;*ENTRY JSR PC,TYP ERR
9191                               ;*RETURN RTS PC
9192                               ;*
9193                               ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
9194                               ;*ERROR IS TO BE REPORTED. IT THEN USES THE 'ERROR TABLE" ($ERRTB)
9195                               ;*ENTRY TO DEFINE WHAT INFORMATION IS TO BE REPORTED CONCERNING
9196                               ;*THE ERROR.
9197                               ;;*****************************************************************************
9198  055332  104413             TYPERR: SAVREG
9199  055334  113700  001114             MOVB    $ITEMB,R0       ;ENTER ERROR NUMBER
9200  055340  042700  177400             BIC     #177400,R0      ;CLEAR SIGN EXTENSION
9201  055344  005300                      DEC     R0              ;FORM INDEX FOR ERROR TABLE
9202  055346  006300                      ASL     R0
9203  055350  006300                      ASL     R0
9204  055352  006300                      ASL     R0
9205  055354  062700  007526       1$:   ADD     #$ERRTB,R0      ;FORM ADDRESS OF ERROR ENTRY
9206  055360  012037  055374             MOV     (R0)+,2$        ;GET EM POINTER
9207  055364  001404                      BEQ     3$              ;BRANCH IF THERE ISN'T ONE
9208  055366  104401  001205             TYPE    ,$CRLF          ;TYPE CARRIAGE RETURN LINE FEED
9209  055372  104401                      TYPE                    ;TYPE ERROR MESSAGE (EM)
9210  055374  000000       2$:   .WORD   0               ;EM POINTER GOES HERE
9211  055376  012037  055412       3$:   MOV     (R0)+,4$        ;GET DH POINTER
9212  055402  001404                      BEQ     5$              ;BRANCH IF THERE ISN'T ONE
9213  055404  104401  001205             TYPE    ,$CRLF          ;TYPE CR-LF
9214  055410  104401                      TYPE                    ;TYPE DATA HEADER
9215  055412  000000       4$:   .WORD   0               ;DH POINTER GOES HERE
9216  055414  012001       5$:   MOV     (R0)+,R1        ;GET DT POINTER
9217  055416  001455             BEQ     20$             ;BRANCH IF THERE ARE NONE
9218  055420  005004             CLR     R4              ;SET INDENT SWITCH
9219  055422  012000             MOV     (R0)+,R0        ;GET DF POINTER
9220  055424  012002             MOV     (R0)+,R2        ;STORE NUMBER OF DH'S
```

```
                                        BEQ     17$             ;DH NUM IS 0-BRANCH
                                        COM     R4              ;NO INDENT
                                        TYPE    ,$CRLF
                                10$:    MOVB    (R0)+,R3        ;GET & STORE NUMBER OF DATA WORDS
                                        TSTB    (R0)+           ;BUMP PAST FORMAT WORD
                                        TST     R3              ;TEST IF ANY DATA FOR THIS HEADER
                                        BEQ     14$             ;NO - SKIP DATA PRINT
                                11$:    MOV     @(R1)+,-(SP)    ;PUT FIRST DATA WORD ON STACK
                                        TYPOC                   ;TYPE IT
                                        DEC     R3              ;MORE DATA WORDS
                                        BEQ     14$             ;NO-BRANCH
                                        TYPE    ,SPACE2         ;TYPE SEPARATORS
                                        BR      11$             ;LOOP
                                14$:    DEC     R2              ;MORE DH'S?
                                        BLE     20$             ;NO-BRANCH
                                        TYPE    ,$CRLF
                                        TST     2(R0)           ;ONLY A DH IN THIS REQUEST?
                                        BEQ     15$             ;YES-BRANCH BYPASS INDENT
                                        COM     R4              ;INDENT?
                                        BNE     15$             ;NO-BRANCH
                                        TYPE    ,SPACE2         ;YES-TYPE SPACES
                                15$:    MOV     (R0)+,16$       ;GET NEXT DH POINTER
                                        TYPE                    ;TYPE DH
                                16$:    .WORD   0               ;DH POINTER GOES HERE
                                        TSTB    (R0)            ;TYPE A DT?
                                        BNE     21$             ;YES-BRANCH
                                        ADD     #2,R0           ;INCREMENT DF POINTER
                                        BR      14$             ;SEE IF END OF DF BLOCK
                                21$:    TYPE    ,$CRLF
                                        TST     R4              ;INDENT?
                                        BNE     10$             ;NO-BRANCH
                                17$:    TYPE    ,SPACE2         ;YES-TYPE SPACES
                                        BR      10$             ;LOOP
                                20$:    RESREG

                                        BIT     #SW12,@SWR      ;SEE IF ABORT DRV AFTER 20 ERRORS
                                        BEQ     25$             ;BR IF NO
                                        CMP     $ERFLG,#20.     ;ELSE SEE IF HAVE 20 ERRORS
                                        BNE     25$             ;BR IF NO
                                        MOV     #STACK,SP       ;ELSE RESTORE STACK PTR
                                        JMP     $EOP            ;AND GO TO NEXT DRV

                                25$:    RTS     PC
                                SPACE2: .ASCIZ/  /              ;2 SPACES
                                ;  ODT-11 -- V005A

                                ; DEC-11-UODPA-A-LA

                                ; COPYRIGHT 1969,1970,1972
                                ; DIGITAL EQUIPMENT CORPORATION
                                ; MAYNARD, MASSACHUSETTS 01754
                                        .ENABL  ABS,AMA
                                        .EVEN
                                        .=.+60
                                R0      =       %0      ; REGISTER
                                R1      =       %1      ;  NAMING
```

```
 9277            000002                  R2      =       %2      ;   CONVENTIONS
 9278            000003                  R3      =       %3
 9279            000004                  R4      =       %4
 9280            000005                  R5      =       %5
 9281            000006                  SP      =       %6
 9282            000007                  PC      =       %7
 9283            177776                  ST      =       177776          ;STATUS REGISTER
 9284                                    ;
 9285            000014                  0.TVEC  =       14              ;TRT VECTOR LOCATION
 9286            000340                  0.STM   =       340             ;PRIORITY MASK - STATUS REGISTER
 9287            000020                  0.TBT   =       20              ;T-BIT MASK - STATUS REGISTER
 9288            000003                  TRT     =       000003          ;TRT INSTRUCTION
 9289            000006                  RTT     =       000006          ;RTT INSTRUCTION
 9290                                    ;
 9291                                    ; R5 IS USUALLY CONSIDERED SAFE, THE CURRENT ADDRESS WORD
 9292                                    ; RESIDES IN IT.  AFTER A BREAKPOINT, IT IS SET TO ZERO, AND SEARCH
 9293                                    ; OPERATIONS LEAVE IT RANDOMLY FILLED.  OTHERWISE, IT SHOULD NOT
 9294                                    ; BE USED EXCEPT FOR JSR'S AND THE CURRENT ADDRESS POINTER (CAD).
 9295                                    ;
 9296            177562                  0.RDB   =       177562  ;R DATA BUFFER
 9297            177560                  0.RCSR  =       177560  ;R C/SR
 9298            177566                  0.TDB   =       177566  ;T DATA BUFFER
 9299            177564                  0.TCSR  =       177564  ;T C/SR
 9300                                    ;
 9301                                    ;
 9302                                    ; INITIALIZE ODT
 9303                                    ;  USE 0.ODT FOR A NORMAL ENTRY
 9304                                    ;  USE 0.ODT+2 TO RESTART ODT - WIPING OUT ALL BREAKPOINTS
 9305                                    ;  USE 0.ODT+4 TO RE-ENTER (I.E. - FAKE A BREAKPOINT)
 9306                                    ;
 9307  055672  000413                    0.ODT:  BR      C.STRT          ;NORMAL ENTRY
 9308  055674  000417                            BR      0.RST           ;RESTART
 9309  055676  013737  177776  055652    0.ENTR: MOV     ST,0.UST        ;RE-ENTER  --  SAVE STATUS
 9310  055704  013737  000016  177776            MOV     0.TVEC+2,ST     ;SET UP LOCAL STATUS
 9311  055712  010737  055650                    MOV     PC,0.UPC        ;FAKE THE PC
 9312  055716  000137  057050                    JMP     0.BK1
 9313                                    ;
 9314  055722  012706  055632    0.STRT: MOV     #0.UR0,SP       ;SET UP STACK
 9315  055726  010637  055646            MOV     SP,0.USP        ;FAKE THE SAVED STACK
 9316  055732  000414                    BR      0.RST1          ;CLEAR BREAKPOINT TABLES
 9317  055734  004037  057256    0.RST:  JSR     0,0.SVR         ;SAVE REGISTERS
 9318  055740  013777  055670  177716            MOV     0.UIN,@0.ADR1   ;REMOVE THE BREAKPOINT
 9319  055746  113704  055654                    MOVB    0.PRI,R4        ;GET ODT PRIORITY
 9320  055752  106004                            RORB    R4              ;SHIFT
 9321  055754  106004                            RORB    R4              ;  INTO
 9322  055756  106004                            RORB    R4              ;   POSITION
 9323  055760  110437  177776                    MOVB    R4,ST           ;STORE IN STATUS
 9324  055764  000127                    0.RST1: JMP     (PC)+
 9325  055766  000403                            BR      0.45
 9326  055770  012737  000002  056760            MOV     #RTI,0.RTIT     ;SET TO RTI IF 11/20 OR /05
 9327  055776  105037  057677            0.45:   CLRB    0.P             ;DISALLOW PROCEED
 9328  056002  012737  000340  000016            MOV     #0.STM,0.TVEC+2 ;STATUS WORD TO TRT VECTOR + 2
 9329  056010  012737  057040  000014            MOV     #0.BRK,0.TVEC   ;PC TO TRT VECTOR
 9330  056016  000447                            BR      0.RALL          ;CLEAR BREAKPOINT TABLES
 9331                                    ;
 9332                                    ; SPECIAL NAME HANDLER
```

```
 9333                                   ;  DEPENDS UPON THE EXPLICIT ORDER OF THE TWO TABLES O.TL AND O.URO
 9334                                   ;
 9335  056020  004537  057500           O.REGT: JSR     5,O.GET         ;SPECIAL NAME, GET ONE MORE CHARACTER
 9336  056024  012704  057723                   MOV     #O.TL,R4        ;TABLE START ADDRESS
 9337  056030  120024                   O.RSP:  CMPB    R0,(R4)+        ;IS THIS THE CORRECT CHARACTER?
 9338  056032  001413                           BEQ     O.SP            ;JUMP IF YES
 9339  056034  022704  057731                   CMP     #O.TL+O.LG,R4   ;IS THE SEARCH DONE?
 9340  056040  101373                           BHI     O.RSP           ;BRANCH IF NOT
 9341  056042  042700  177770                   BIC     #177770,R0      ;MASK OFF OCTAL
 9342  056046  010004                           MOV     R0,R4
 9343  056050  006304                   O.SP1:  ASL     R4
 9344  056052  062704  055632                   ADD     #O.URO,R4       ;GENERATE ADDRESS
 9345  056056  005202                           INC     R2              ;SET FOUND FLAG
 9346  056060  000444                           BR      O.SCAN          ;GO FIND NEXT CHARACTER
 9347  056062  162704  057714           O.SP:   SUB     #O.TL-7,R4      ;CORRECT CONSTANT
 9348  056066  000770                           BR      O.SP1
 9349                                   ;
 9350                                   ; _ HANDLER - OPEN INDEXED ON THE PC
 9351                                   ;
 9352  056070  004737  057624           O.ORPC: JSR     PC,O.TCLS
 9353  056074  010502                           MOV     R5,R2           ;CURRENT ADDRESS IN R2
 9354  056076  061202                           ADD     @R2,R2          ;COMPUTE
 9355  056100  006202                           ASR     R2              ;MOVE ONE BIT TO CARRY
 9356  056102  103421                           BCS     O.ERR           ;ERROR IF ODD NUMBER
 9357  056104  006302                           ASL     R2              ;RESTORE WORD
 9358  056106  005722                           TST     (R2)+           ; AND INCREMENT BY TWO
 9359  056110  010205                           MOV     R2,R5           ;UPDATE CAD
 9360  056112  000137  056364                   JMP     O.OP2           ;GO FINISH UP
 9361                                   ;
 9362                                   ; B HANDLER - SET AND REMOVE BREAKPOINTS
 9363                                   ;
 9364  056116  005702                   O.BKPT: TST     R2              ;IF NO NUMBER TYPED
 9365  056120  001406                           BEQ     O.RALL          ;  REMOVE BREAKPOINT
 9366  056122  006204                           ASR     R4              ;CHECK IF ODD
 9367  056124  103410                           BCS     O.ERR           ;JUMP IF ODD
 9368  056126  006304                           ASL     R4              ;RESTORE ONE BIT
 9369  056130  010437  055664                   MOV     R4,O.ADR1       ;SET A BREAKPOINT
 9370  056134  000412                           BR      O.DCD
 9371  056136  012737  057740  055664   O.RALL: MOV     #O.TRTC,O.ADR1  ;CLEAR BREAKPOINT
 9372  056144  000406                           BR      O.DCD
 9373                                   ;
 9374                                   ; COMMAND DECODER - ODT11
 9375                                   ;
 9376                                   ;   REGISTERS R0-R4 MAY BE USED,
 9377                                   ;    REGISTER R5 WILL BE CONSIDERED SAFE
 9378                                   ;
 9379  056146  052705  000001           O.ERR:  BIS     #1,R5           ;CLOSE EVERYTHING
 9380  056152  012700  000077                   MOV     #'?,R0          ;  ? TO BE TYPED
 9381  056156  004537  057556                   JSR     5,O.FTYP        ; OUTPUT ?
 9382  056162  004537  057656           O.DCD:  JSR     5,O.CRLS        ;TYPE  <CR><LF>*
 9383  056166  005004                   O.DCD1: CLR     R4              ; R4 CONTAINS THE CONVERTED OCTAL
 9384  056170  005002                           CLR     R2              ; R2 IS THE NUMBER FOUND FLAG
 9385  056172  004537  057500           O.SCAN: JSR     5,O.GET         ;GET A CHAR, RETURN IN R0
 9386  056176  022700  000060                   CMP     #'0,R0          ;COMPARE WITH ASCII 0
 9387  056202  101013                           BHI     O.CLGL          ;CHECK LEGALITY IF NON-NUMERIC
 9388  056204  022700  000067                   CMP     #'7,R0          ;COMPARE WITH ASCII 7
```

```
 9389  056210  103410                  BLO     O.CLGL          ;CHECK LEGALITY IF NOT OCTAL
 9390  056212  042700  177770          BIC     #177770,R0      ;CONVERT TO BCD
 9391  056216  006304                  ASL     R4              ; MAKE ROOM
 9392  056220  006304                  ASL     R4              ;  IN
 9393  056222  006304                  ASL     R4              ;    R4
 9394  056224  060004                  ADD     R0,R4           ;PACK THREE BITS IN R4
 9395  056226  005202                  INC     R2              ;R2 HAS NUMERIC FLAG
 9396  056230  000760                  BR      O.SCAN          ;   AND TRY AGAIN
 9397  056232  005001          O.CLGL: CLR     R1              ;CLEAR INDEX
 9398  056234  120061  057707  O.LGL1: CMPB    R0,O.LGCH(R1)   ;DO THF CODES MATCH?
 9399  056240  001405                  BEQ     O.LGL2          ;JUMP IF YES
 9400  056242  005201                  INC     R1              ; SET INDEX FOR NEXT SEARCH
 9401  056244  020127  000014          CMP     R1,#O.CLGT      ;IS THE SEARCH DONE?
 9402  056250  103336                  BHIS    O.ERR           ;    OOPS!
 9403  056252  000770                  BR      O.LGL1          ;RE-LOOP
 9404  056254  006301          O.LGL2: ASL     R1              ;MULTIPLY BY TWO
 9405  056256  000171  056262          JMP     @O.LGDR(R1)     ;GO TO PROPER ROUTINE
 9406                                  ;
 9407  056262  056312          O.LGDR: O.WRD   ;  /    OPEN WORD
 9408  056264  056344                  O.CRET  ;  CARRIAGE RETURN    CLOSE
 9409  056266  056020                  O.REGT  ;  $  REGISTER OPS
 9410  056270  056654                  O.GO    ;  G  GO TO ADDRESS K
 9411  056272  056356                  O.OP1   ;  <LF>  MODIFY, CLOSE, OPEN NEXT
 9412  056274  056070                  O.ORPC  ;     OPEN RELATED, INDEX - PC
 9413  056276  056410                  O.BACK  ;  ^  OPEN PREVIOUS
 9414  056300  056420                  O.OFST  ;  O  OFFSET
 9415  056302  056476                  O.WSCH  ;  W  SEARCH WORD
 9416  056304  056472                  O.EFF   ;  E  SEARCH EFFECTIVE ADDRESS
 9417  056306  056116                  O.BKPT  ;  B  BREAKPOINTS
 9418  056310  056762                  O.PROC  ;  P  PROCEED
 9419          000030          O.LGL   =       .-O.LGDR        ;LGL MUST EQUAL 2X CHLGT ALWAYS
 9420                                  ;
 9421                                  ; PROCESS / - OPEN WORD
 9422                                  ;
 9423  056312  005702          O.WRD:  TST     R2              ;GET VALUE IF R2 IS NON-ZERO
 9424  056314  001410                  BEQ     O.WRDA          ;SKIP OTHERWISE
 9425  056316  010405                  MOV     R4,R5           ;   PUT VALUE IN CAD
 9426  056320  006205          O.WRD1: ASR     R5              ;MOVE ONE BIT TO CARRY
 9427  056322  103711          O.ERR2: BCS     O.ERR           ;JUMP IF ODD ADDRESS
 9428  056324  006305                  ASL     R5              ;RESTORE THE CARRY BIT
 9429  056326  011500                  MOV     @R5,R0          ;GET CONTENTS OF WORD
 9430  056330  004537  057414          JSR     5,O.CADV        ;GO GET AND TYPE OUT @CAD
 9431  056334  000714                  BR      O.DCD1          ;GO BACK TO DECODER
 9432  056336  042705  000001  O.WRDA: BIC     #1,R5           ;CLEAR CLOSED BIT
 9433  056342  000766                  BR      O.WRD1          ;GO BACK TO MAIN-LINE
 9434                                  ;
 9435                                  ; PROCESS CARRIAGE RETURN
 9436                                  ;
 9437  056344  004737  057624  O.CRET: JSR     PC,O.TCLS       ;CLOSE LOCATION
 9438  056350  052705  000001          BIS     #1,R5           ;CLOSE EVERYTHING
 9439  056354  000702                  BR      O.DCD           ;RETURN TO DECODER
 9440                                  ;
 9441                                  ; PROCESS <LF>, OPEN NEXT WORD
 9442                                  ;
 9443  056356  004737  057624  O.OP1:  JSR     PC,O.TCLS       ;CLOSE PRESENT CELL
 9444  056362  005725                  TST     (R5)+           ;GENERATE NEW ADDRESS
```

```
 9445  056364  004537  057650          0.OP2:  JSR     5,0.CRLF        ;<CR><LF>
 9446  056370  010500                          MOV     R5,R0           ;NUMBER TO TYPE
 9447  056372  004537  057414                  JSR     5,0.CADV        ; TYPE OUT ADDRESS
 9448  056376  012700  000057                  MOV     #'/,R0          ;TYPE A /
 9449  056402  004537  057556                  JSR     5,0.FTYP
 9450  056406  000744                          BR      0.WRD1          ;GO PROCESS IT
 9451                                  ;
 9452                                  ; PROCESS ^, OPEN PREVIOUS WORD
 9453                                  ;
 9454  056410  004737  057624          0.BACK: JSR     PC,0.TCLS
 9455  056414  005745                          TST     -(R5)           ;GENERATE NEW ADDRESS
 9456  056416  000762                          BR      0.OP2           ;GO DO THE REST
 9457                                  ;
 9458                                  ; PROCESS 0, COMPUTE OFFSET
 9459                                  ;
 9460  056420  006205                  0.OFST: ASR     R5              ;GET LOW ORDER BIT
 9461  056422  103737                          BCS     0.ERR2          ;ERROR IF CLOSED
 9462  056424  006305                          ASL     R5              ;RESTORE WORD
 9463  056426  012700  000040                  MOV     #' ,R0          ;TYPE ONE BLANK
 9464  056432  004537  057556                  JSR     5,0.FTYP        ; AS A SEPARATOR
 9465  056436  160504                          SUB     R5,R4           ;COMPUTE
 9466  056440  005304                          DEC     R4
 9467  056442  005304                          DEC     R4              ;    16 BIT OFFSET
 9468  056444  010400                          MOV     R4,R0           ;TYPE A
 9469  056446  010402                          MOV     R4,R2           ;SAVE R4
 9470  056450  004537  057414                  JSR     5,0.CADV        ;NUMBER IN R0 - WORD MODE
 9471  056454  010200                          MOV     R2,R0
 9472  056456  006200                          ASR     R0              ;DIVIDE BY TWO
 9473  056460  103402                          BCS     0.OF1           ;BRANCH IF ODD
 9474  056462  004537  057414                  JSR     5,0.CADV        ;NUMBER IN R0 - BYTE MODE
 947   056466  000137  056166          0.OF1:  JMP     0.DCD1          ;ALL DONE
 9476                                  ;
 9477                                  ; SEARCHES - $MSK   HAS THE MASK
 9478                                  ;            $MSK+2 HAS THE FWA
 9479                                  ;            $MSK+4 HAS THE LWA
```

```
9480                                    ;
9481  056472  005201                    O.EFF:  INC     R1              ;SET EFFECTIVE SEARCH
9482  056474  000401                            BR      O.WDS
9483  056476  005001                    O.WSCH: CLR     R1              ;SET WORD SEARCH
9484  056500  005702                    O.WDS:  TST     R2              ;CHECK FOR OBJECT FOUND
9485  056502  001621                    O.ERR1: BEQ     O.ERR           ;ERROR IF NO OBJECT
9486  056504  013702  055660                    MOV     O.MSK+2,R2      ;SET ORIGIN
9487  056510  013705  055656                    MOV     O.MSK,R5        ;SET MASK
9488  056514  005105                            COM     R5              ;AND COMPLEMENT IT
9489  056516  020237  055662            O.WDS2: CMP     R2,O.MSK+4      ; IS THE SEARCH ALL DONE?
9490  056522  101217                            BHI     O.DCD           ;  YES
9491  056524  011200                            MOV     @R2,R0          ; GET OBJECT
9492  056526  005701                            TST     R1              ;NO
9493  056530  001027                            BNE     O.EFF1          ;BRANCH IF EFFECTIVE SEARCH
9494  056532  010046                            MOV     R0,-(SP)
9495  056534  010403                            MOV     R4,R3           ;EXCLUSIVE OR
9496  056536  040400                            BIC     R4,R0           ; IS DONE
9497  056540  042603                            BIC     (SP)+,R3                ;  IN A VERY
9498  056542  050003                            BIS     R0,R3           ;    FANCY MANNER HERE
9499  056544  040503                            BIC     R5,R3           ;AND RESULT WITH MASK
9500  056546  001016                    O.WDS3: BNE     O.WDS4          ;RE-LOOP IF NO MATCH
9501  056550  010446                            MOV     R4,-(SP)        ;REGISTERS R2,R4, AND R5 ARE SAFE
9502  056552  004537  057650                    JSR     5,O.CRLF        ;TYPE <CR,LF>
9503  056556  010200                            MOV     R2,R0           ;GET READY TO TYPE
9504  056560  004537  057414                    JSR     5,O.CADV        ;  TYPE ADDRESS
9505  056564  012700  000057                    MOV     #'/,R0          ;SLASH TO R0
9506  056570  004537  057556                    JSR     5,O.FTYP        ;TYPE IT
9507  056574  011200                            MOV     @R2,R0          ;GET CONTENTS
9508  056576  004537  057414                    JSR     5,O.CADV        ;TYPE CONTENTS
9509  056602  012604                            MOV     (SP)+,R4        ; RESTORE R4
9510  056604  005722                    O.WDS4: TST     (R2)+           ;INCREMENT TO NEXT CELL AND
9511  056606  000743                            BR      O.WDS2          ;    RETURN
9512  056610  020004                    O.EFF1: CMP     R0,R4           ; IS (X)=K?
9513  056612  001755                            BEQ     O.WDS3          ;TYPE IF EQUAL
9514  056614  010003                            MOV     R0,R3           ;(X) TO R3
9515  056616  060203                            ADD     R2,R3           ;(X)+X
9516  056620  005203                            INC     R3
9517  056622  005203                            INC     R3              ;(X)+X+2
9518  056624  020304                            CMP     R3,R4           ;IS (X)+X+2=K?
9519  056626  001747                            BEQ     O.WDS3          ;BRANCH IF EQUAL
9520  056630  042700  177400                    BIC     #177400,R0      ;WIPE OUT EXTRANEOUS BITS
9521  056634  110000                            MOVB    R0,R0           ;EXTEND SIGN
9522  056636  000257                            CCC
9523  056640  006300                            ASL     R0              ;MULTIPLY BY TWO
9524  056642  005200                            INC     R0              ;ADD TWO
9525  056644  005200                            INC     R0
9526  056646  060200                            ADD     R2,R0           ;ADD PC
9527  056650  020004                            CMP     R0,R4           ;IS THE RESULT A PROPER REL. BRANCH?
9528  056652  000735                            BR      O.WDS3
9529                                    ;
9530                                    ; PROCESS G - GO
9531                                    ;
9532  056654  105037  057677            O.GO:   CLRB    O.P             ;DISALLOW PROCEED
9533  056660  006204                            ASR     R4              ;CHECK LOW ORDER BIT
9534  056662  103617                            BCS     O.ERR2          ;ERROR IF ODD NUMBER
9535  056664  006304                            ASL     R4              ;RESTORE WORD
```

```
 9536  056666  010437  055650                 MOV     R4,0.UPC        ;SET UP NEW PC
 9537  056672  112737  000340  177776         MOVB    #0.STM,ST       ;SET HIGH PRIORITY
 9538  056700  004537  057346                 JSR     5,0.RSTT        ;RESTORE TELETYPE
 9539  056704  105037  057676           0.TBIT: CLRB    0.T             ;CLEAR BOTH
 9540  056710  042737  000020  055652         BIC     #0.TBT,0.UST    ;  T-BIT FLAGS
 9541  056716  017737  176742  055670         MOV     @0.ADR1,0.UIN   ;SAVE INSTRUCTION
 9542  056724  013777  057740  176732         MOV     0.TRTC,@0.ADR1  ;REPLACE WITH TRAP
 9543  056732  012600                 0.GO2:  MOV     (SP)+,R0        ;RESTORE
 9544  056734  012601                         MOV     (SP)+,R1        ; R0
 9545  056736  012602                         MOV     (SP)+,R2        ;   THRU
 9546  056740  012603                         MOV     (SP)+,R3
 9547  056742  012604                         MOV     (SP)+,R4        ;
 9548  056744  012605                         MOV     (SP)+,R5        ;   R5
 9549  056746  012606                         MOV     (SP)+,SP        ; AND SP
 9550  056750  013746  055652                 MOV     0.UST,-(SP)     ;   AND STATUS
 9551  056754  013746  055650                 MOV     0.UPC,-(SP)     ;   AND PC
 9552  056760  000006                 0.RTIT: RTT                     ;CHANGED TO RTI FOR 11/20 AND /05
 9553                                 ;
 9554                                 ; PROCESS P - PROCEED
 9555                                 ;   ONLY ALLOWED AFTER A BREAKPOINT
 9556                                 ;
 9557  056762  105737  057677           0.PROC: TSTB    0.P             ;CHECK LEGALITY OF PROCEED
 9558  056766  001645                         BEQ     0.ERR1          ;NOT LEGAL
 9559  056770  105037  057677                 CLRB    0.P             ;CLEAR PROCEED FLAG
 9560  056774  005702                         TST     R2              ;WAS COUNT SPECIFIED?
 9561  056776  001402                         BEQ     0.PR1           ;NO
 9562  057000  010437  055666                 MOV     R4,0.CT         ;YES, PUT AWAY COUNT
 9563  057004  112737  000340  177776   0.PR1:  MOVB    #0.STM,ST       ;FORCE HIGH PRIORITY
 9564  057012  004537  057346                 JSR     5 0.RSTT        ;RESTORE TTY
 9565  057016  112737  000340  177776   0.C1:   MOVB    #0.STM,ST       ;SET HIGH PRIORITY
 9566  057024  105237  057676                 INCB    0.T             ;SET T-BIT FLAG
 9567  057030  052737  000020  055652         BIS     #0.TBT,0.UST    ;SET T-BIT
 9568  057036  000735                         BR      0.GO2
 9569                                 ;
 9570                                 ; BREAKPOINT HANDLER
 9571                                 ;  A TRT BREAKPOINT CAUSES 0.BRK TO BE ENTERED, WHICH SAVES
 9572                                 ;  VARIOUS ODDS AND ENDS, FINDS OUT IF THE BREAKPOINT WAS LEGAL,
 9573                                 ;  AND GIVES CONTROL TO THE COMMAND DECODER
 9574                                 ;
 9575  057040  012637  055650           0.BRK:  MOV     (SP)+,0.UPC     ;PRIORITY IS 7 UPON ENTRY
 9576  057044  012637  055652                   MOV     (SP)+,0.UST     ;SAVE STATUS AND PC
 9577  057050  004037  057256           0.BK1:  JSR     0,0.SVR         ;SAVE VARIOUS REGISTERS
 9578  057054  105737  057676                   TSTB    0.T             ;CHECK FOR T-BIT SET
 9579  057060  001311                           BNE     0.TBIT          ;JUMP IF SET
 9580  057062  013777  055670  176574           MOV     0.UIN,@0.ADR1   ;REMOVE BREAKPOINTS
 9581  057070  105737  055654                   TSTB    0.PRI           ;CHECK IF PRIORITY
 9582  057074  100003                           BPL     0.BK2           ; IS AS SAME AS USER PGM
 9583  057076  113705  055652                   MOVB    0.UST,R5        ;PICK UP USER UST IF SO
 9584  057102  000407                           BR      0.BK3           ;AND DON'T COMPUTE THE PRIORITY
 9585  057104  113705  055654           0.BK2:  MOVB    0.PRI,R5        ;OTHERWISE PICK UP ACTUAL PRIORITY
 9586  057110  000257                           CCC                     ;CLEAR CARRY
 9587  057112  106005                           RORB    R5              ;SHIFT LOW ORDER BITS
 9588  057114  106005                           RORB    R5              ;  INTO
 9589  057116  106005                           RORB    R5              ;    HIGH ORDER
 9590  057120  106005                           RORB    R5              ;      POSITION
 9591  057122  110537  177776           0.BK3:  MOVB    R5,ST           ;PUT THE STATUS AWAY WHERE IT BELONGS
```

```
 9592  057126  013705  055650                  MOV     0.UPC,R5        ;GET PC, IT POINTS TO THE TRT
 9593  057132  005745                          TST     -(R5)           ;SUBTRACT TWO
 9594  057134  010537  055650                  MOV     R5,0.UPC        ;FROM THE USER'S PC
 9595  057140  020537  055664                  CMP     R5,0.ADR1       ;COMPARE WITH LIST
 9596  057144  001417                          BEQ     0.B2            ;JUMP IF FOUND
 9597  057146  004537  057314                  JSR     5,0.SVTT        ;SAVE TELETYPE STATUS
 9598  057152  004537  057650                  JSR     5,0.CRLF
 9599  057156  012704  057702                  MOV     #0.BD,R4        ;ERROR, NOTHING FOUND
 9600  057162  012703  057703                  MOV     #0.BD+1,R3
 9601  057166  004537  057542                  JSR     5,0.TYPE        ;OUTPUT 'BE'' FOR BAD ENTRY
 9602  057172  010500                          MOV     R5,R0
 9603  057174  042737  000020  055652          BIC     #0.TBT,0.UST    ;CLEAR OUT ANY POSSIBLE FAKE T-BIT
 9604  057202  000420                          BR      0.B3            ; AND CONTINUE
 9605  057204  005337  055666          0.B2:   DEC     0.CT
 9606  057210  003302                          BGT     0.C1            ;JUMP IF REPEAT
 9607  057212  012737  000001  055666          MOV     #1,0.CT         ;RESET COUNT TO 1
 9608  057220  105237  057677                  INCB    0.P             ;ALLOW PROCEED
 9609  057224  004537  057314                  JSR     5,0.SVTT        ;SAVE TELETYPE STATUS, R4 IS SAFE
 9610  057230  012700  000102                  MOV     #'B,R0
 9611  057234  004537  057556                  JSR     5,0.FTYP        ;TYPE 'B''
 9612  057240  013700  055664                  MOV     0.ADR1,R0       ;GET ADDRESS OF BREAK
 9613  057244  004537  057414          0.B3:   JSR     5,0.CADV        ;TYPE ADDRESS
 9614  057250  005005                          CLR     R5              ;CLEAR CAD
 9615  057252  000137  056162                  JMP     0.DCD           ;GO TO DECODER
 9616                                  ;
 9617                                  ; SAVE REGISTERS R0-R6 IN INTERNAL STACK
 9618                                  ;
 9619  057256  012637  057674          0.SVR:  MOV     (SP)+,0.XXX     ;PICK REGISTER FROM STACK AND SAVE
 9620  057262  010637  055646                  MOV     SP,0.USP        ;SAVE USER STACK ADDRESS
 9621  057266  012706  055646                  MOV     #0.USP,SP       ;SET TO INTERNAL STACK
 9622  057272  010546                          MOV     R5,-(SP)        ;SAVE
 9623  057274  010446                          MOV     R4,-(SP)        ; REGISTERS
 9624  057276  010346                          MOV     R3,-(SP)        ;1
 9625  057300  010246                          MOV     R2,-(SP)        ; THRU
 9626  057302  010146                          MOV     R1,-(SP)        ;     5
 9627  057304  013746  057674                  MOV     0.XXX,-(SP)     ;PUT SAVED REGISTER ON STACK
 9628  057310  005746                          TST     -(SP)
 9629  057312  000200                          RTS     R0
 9630                                  ;
 9631                                  ; SAVE TELETYPE STATUS
 9632                                  ;
 9633  057314  113737  177560  057700  0.SVTT: MOVB    0.RCSR,0.CSR1   ;SAVE R C/SR
 9634  057322  113737  177564  057701          MOVB    0.TCSR,0.CSR2   ;SAVE T C/SR
 9635  057330  105037  177560                  CLRB    0.RCSR          ;CLEAR ENABLE AND MAINTENANCE
 9636  057334  105037  177564                  CLRB    0.TCSR          ;  BITS IN BOTH C/SR
 9637  057340  004537  057650                  JSR     5,0.CRLF        ;TYPE <CR,LF>
 9638  057344  000205                          RTS     R5
 9639                                  ;
 9640                                  ; RESTORE TELETYPE STATUS
 9641                                  ;
 9642  057346  004537  057650          0.RSTT: JSR     5,0.CRLF                ;<CR,LF> BEFORE RESTORING
 9643  057352  105737  177564                  TSTB    0.TCSR          ;WAIT READY ON PRINTER
 9644  057356  100375                          BPL     .-4
 9645  057360  032737  004000  177560          BIT     #4000,0.RCSR    ;CHECK BUSY FLAG ON READER
 9646  057366  001403                          BEQ     0.RSE1          ;SKIP READY LOOP IF NOT BUSY
 9647  057370  105737  177560                  TSTB    0.RCSR          ;WAIT READY
```

```
 9648  057374  100375                          BPL     .-4             ;    ON READER
 9649  057376  113737  057700  177560  O.RSE1: MOVB    O.CSR1,O.RCSR   ;RESTORE
 9650  057404  113737  057701  177564          MOVB    O.CSR2,O.TCSR   ;  THE STATUS REGISTERS
 9651  057412  000205                          RTS     R5
 9652                                  ;
 9653                                  ; TYPE OUT CONTENTS OF WORD OR BYTE WITH ONE TRAILING SPACE
 9654                                  ; WORD IS IN R0
 9655                                  ;
 9656  057414  010246                  O.CADV: MOV     R2,-(SP)        ;SAVE R2
 9657  057416  012704  057737                  MOV     #O.BUF+6,R4     ;BUFFER START ADDRESS
 9658  057422  012746  000060                  MOV     #'0,-(SP)               ;CONSTANT ASCII 0
 9659  057426  010002                  O.SPC:  MOV     R0,R2           ; GET
 9660  057430  042702  177770                  BIC     #177770,R2      ;    OCTAL CHARACTER
 9661  057434  061602                          ADD     @SP,R2          ;CONVERT TO ASCII
 9662  057436  110244                          MOVB    R2,-(R4)        ;STORE IN BUFFER
 9663  057440  006200                          ASR     R0              ;SHIFT THIS MESS
 9664  057442  006200                          ASR     R0              ; RIGHT
 9665  057444  006200                          ASR     R0              ;   THREE WHOLE PLACES
 9666  057446  020427  057732                  CMP     R4,#O.BUF+1     ;DONE?
 9667  057452  101365                          BHI     O.SPC           ;    NO
 9668  057454  042700  177776                  BIC     #177776,R0      ;GET LAST BIT
 9669  057460  062600                          ADD     (SP)+,R0                ;CONVERT TO ASCII
 9670  057462  110044                          MOVB    R0,-(R4)        ;AND PUT IT AWAY
 9671  057464  012703  057737                  MOV     #O.BUF+6,R3     ;LWA
 9672  057470  004537  057542                  JSR     5,O.TYPE        ;TYPE WHOLE STRING OF CHARACTERS
 9673  057474  012602                          MOV     (SP)+,R2        ;RESTORE R2
 9674  057476  000205                          RTS     R5
 9675                                  ;
 9676                                  ; GENERAL CHARACTER INPUT ROUTINE
 9677                                  ; CHARACTER INPUT GOES TO R0
 9678                                  ;
 9679  057500  105737  177560          O.GET:  TSTB    O.RCSR          ;WAIT FOR
 9680  057504  100375                          BPL     .-4             ;   INPUT FROM KEYBOARD
 9681  057506  113700  177562                  MOVB    O.RDB,R0        ;GET A CHARACTER
 9682  057512  004537  057556                  JSR     5,O.FTYP        ;ECHO CHARACTER
 9683  057516  042700  177600                  BIC     #177600,R0      ;STRIP OFF PARITY FROM CHARACTER
 9684  057522  001766                          BEQ     O.GET           ;IGNORE NULLS
 9685  057524  122700  000040                  CMPB    #40,R0          ;CHECK FOR SPACES
 9686  057530  001763                          BEQ     O.GET           ;IGNORE NULLS
 9687  057532  122700  000073                  CMPB    #';,R0          ;CHECK FOR SEMI-COLON
 9688  057536  001760                          BEQ     O.GET           ;IGNORE THEM IF FOUND
 9689  057540  000205                          RTS     R5
 9690                                  ;
 9691                                  ; GENERAL CHARACTER OUTPUT ROUTINE
 9692                                  ;  ADDRESS OF FIRST BYTE IN R4,
 9693                                  ;  ADDRESS OF LAST BYTE IN R3, (R3)>(R4)
 9694                                  ;
 9695  057542  020304                  O.TYPE: CMP     R3,R4           ;CHECK FOR COMPLETION
 9696  057544  103426                          BLO     O.TYP1          ; EXIT WHEN DONE
 9697  057546  112400                          MOVB    (R4)+,R0        ;GET A CHARACTER
 9698  057550  004537  057556                  JSR     5,O.FTYP        ;TYPE ONE CHARACTER
 9699  057554  000772                          BR      O.TYPE          ;LOOP UNTIL DONE
 9700                                  ;
 9701                                  ; TYPE ONLY ONE CHARACTER (CONTAINED IN R0)
 9702                                  ;
 9703  057556  105737  177564          O.FTYP: TSTB    O.TCSR          ;CHECK STATUS
```

```
 9704  057562  100375                  BPL     .-4             ;WAIT UNTIL READY
 9705  057564  110037  177566          MOVB    R0,@.TDB        ;TYPE ONE CHARACTER
 9706  057570  120037  000045          CMPB    R0,@#45         ;IS CHAR TO BE FILLED?
 9707  057574  001012                  BNE     0.TYP1          ;NO
 9708  057576  113746  000044          MOVB    @#44,-(SP)      ;YES, INIT THE COUNT
 9709  057602  105737  177564  0.TYP2: TSTB    @.TCSR
 9710  057606  100375                  BPL     0.TYP2
 9711  057610  105037  177566          CLRB    @.TDB           ;GENERATE NULL FILLER
 9712  057614  105316                  DECB    @SP
 9713  057616  003371                  BGT     0.TYP2
 9714  057620  005726                  TST     (SP)+           ;POP STACK
 9715  057622  000205          0.TYP1: RTS     R5
 9716                          ;
 9717                          ; CLOSE WORD OR BYTE AND EXIT,
 9718                          ; UPON ENTERING, R2 HAS NUMERIC FLAG, R4 HAS CONTENTS
 9719                          ;
 9720  057624  006205          0.TCLS: ASR     R5              ;GET LOW ORDER BIT
 9721  057626  103405                  BCS     0.TC            ;JUMP IF ALREADY CLOSED
 9722  057630  006305                  ASL     R5
 9723  057632  005702                  TST     R2              ;IF NO NUMBER WAS TYPED THERE IS
 9724  057634  001401                  BEQ     0.CLS1          ;NO CHANGE TO THE OPEN CELL
 9725  057636  010415                  MOV     R4,@R5          ;STORE WORD
 9726  057640  000207          0.CLS1: RTS     PC
 9727  057642  005746          0.TC:   TST     -(SP)           ;POP EXTRA CELL FROM STACK
 9728  057644  000137  056146          JMP     0.ERR           ;AND SCREAM BLOODY MURDER
 9729                          ;
 9730                          ; 0.CRLF - TYPE <CR,LF>
 9731                          ; 0.CRLS - TYPE <CR,LF>*
 9732                          ;
 9733  057650  012703  057705  0.CRLF: MOV     #0.CR+1,R3      ;LWA <CR,LF>
 9734  057654  000402                  BR      0.CRS
 9735  057656  012703  057706  0.CRLS: MOV     #0.CR+2,R3      ;LWA <CR,LF>*
 9736  057662  012704  057704  0.CRS:  MOV     #0.CR,R4        ;FWA
 9737  057666  004537  057542          JSR     5,@0.TYPE       ;TYPE SOMETHING
 9738  057672  000205                  RTS     R5
 9739                          ;
 9740  057674  000000          0.XXX:  .WORD   0               ;TEMPORARY STORAGE
 9741  057676     000          0.T:    .BYTE   0               ;  T-BIT FLAG
 9742  057677     000          0.P:    .BYTE   0               ;PROCEED FLAG = 0 IF PROCEED NOT ALLOWED
 9743                                                          ;               = 1 IF PROCEED ALLOWED
 9744  057700     000          0.CSR1: .BYTE   0               ;SAVE CELL - R C/SR
 9745  057701     000          0.CSR2: .BYTE   0               ;SAVE CELL - T C/SR
 9746                          ;
 9747                                  .EVEN
 9748  057702  042502          0.BD:   .WORD   'BE
 9749                          ;
 9750  057704     015          0.CR:   .BYTE   015     ;  <CR>
 9751  057705     012                  .BYTE   012     ;  <LF>
 9752  057706     052                  .BYTE   '*      ;  *
 9753                          ;
 9754  057707     057          0.LGCH: .BYTE   '/      ;  /
 9755  057710     015                  .BYTE   015     ;  CARRIAGE RETURN
 9756  057711     044                  .BYTE   '$      ;  $
 9757  057712     107                  .BYTE   'G      ;  G
 9758  057713     012                  .BYTE   012     ;  <LF>
 9759  057714     137                  .BYTE   '_      ;  _
```

```
 9760  057715     136                    .BYTE   '^      ;  ^
 9761  057716     117                    .BYTE   'O      ;  O
 9762  057717     127                    .BYTE   'W      ;  W
 9763  057720     105                    .BYTE   'E      ;  E
 9764  057721     102                    .BYTE   'B      ;  B
 9765  057722     120                    .BYTE   'P      ;  P
 9766             000014          O.CLGT  =       .-O.LGCH                ;TABLE LENGTH
 9767                             ;
 9768  057723     123             O.TL:   .BYTE   'S      ;DO       1
 9769  057724     120                     .BYTE   'P      ;NOT      2
 9770  057725     115                     .BYTE   'M      ;CHANGE   3
 9771  057726     000                     .BYTE   0       ;THE      4
 9772  057727     000                     .BYTE   0       ;ORDER    5
 9773  057730     102                     .BYTE   'B      ;HERE     6
 9774             000006          O.LG    =       .-O.TL
 9775                             ;
 9776  057731                     O.BUF:                          ;6 CHAR. BUFFER WITH
 9777             057737          .       =       .+6
 9778  057737     040                     .BYTE                   ;TRAILING BLANK
 9779                                     .EVEN
 9780                             ;
 9781  057740  000003             O.TRTC: TRT                     ;TRACE TRAP PROTOTYPE
 9782                             ;
 9783                             ;THE ORDER OF THE FOLLOWING ENTRIES IS CRITICAL
 9784                             ;
 9785             055632          .       =       O.ODT-40
 9786  055632  000000             O.UR0:  0       ;USER R0
 9787  055634  000000                     0       ;     R1
 9788  055636  000000                     0       ;     R2
 9789  055640  000000                     0       ;     R3
 9790  055642  000000                     0       ;     R4
 9791  055644  000000                     0       ;     R5
 9792  055646  000000             O.USP:  0       ;USER SP
 9793  055650  000000             O.UPC:  0       ;USER PC
 9794  055652  000000             O.UST:  0       ;USER ST
 9795  055654  000007             O.PRI:  7       ;ODT PRIORITY
 9796  055656  000000             O.MSK:  0       ;MASK
 9797  055660  000000                     0       ;LOW LIMIT
 9798  055662  000000                     0       ;HIGH LIMIT
 9799                             ;
 9800                             ; BREAK POINT LISTS, ADR1 = ADDRESS OF BREAKPOINT,CT = COUNT,
 9801                             ;   UIN = CONTENTS
 9802                             ;
 9803  055664  000000             O.ADR1: 0
 9804  055666  000000             O.CT:   0
 9805  055670  000000             O.UIN:  0
 9806             000001                 .END
```

```
ABASE = 177440          1898    1939    1953#
ACDW1 = 000000          1898    1941
ACDW2 = 000000          1898    1942
ACLO  = 000010          1251#
ACPUOP= 000000          1898    1913
ACT11   007470          2131#   3309*
ADDW0 = 000000          1898    1943
ADDW1 = 000000          1898    1944
ADDW10= 000000          1898
ADDW11= 000000          1898
ADDW12= 000000          1898
ADDW13= 000000          1898
ADDW14= 000000          1898
ADDW15= 000000          1898
ADDW2 = 000000          1898    1945
ADDW3 = 000000          1898    1946
ADDW4 = 000000          1898    1947
ADDW5 = 000000          1898    1948
ADDW6 = 000000          1898    1949
ADDW7 = 000000          1898    1950
ADDW8 = 000000          1898
ADDW9 = 000000          1898
ADEVCT= 000000          1898    1904
ADEVM = 000000          1898    1940
AENV  = 000000          1898    1909
AENVM = 000000          1898    1910
AFATAL= 000000          1898    1901
AMADR1= 000000          1898    1926
AMADR2= 000000          1898    1930
AMADR3= 000000          1898    1933
AMADR4= 000000          1898    1936
AMAMS1= 000000          1898    1920
AMAMS2= 000000          1898    1928
AMAMS3= 000000          1898    1931
AMAMS4= 000000          1898    1934
AMSGAD= 000000          1898    1906
AMSGLG= 000000          1898    1907
AMSGTY= 000000          1898    1900
AMTYP1= 000000          1898    1921
AMTYP2= 000000          1898    1929
AMTYP3= 000000          1898    1932
AMTYP4= 000000          1898    1935
APASS = 000000          1898    1903
APRIOR= 000000          1898
APTCSU= 000040          7295    7484#
APTENV= 000001          5992    7242    7288    7440    7482#
APTSIZ= 000200          3235    7481#
APTSPO= 000100          7290    7442    7483#
ASWREG= 000000          1898    1911
ATESTN= 000000          1898    1902
ATTN    007344          2064#   6199    6218    6244
AUNIT = 000000          1898    1905
AUSWR = 000000          1898    1912
AVECT1= 000000          1898    1937
AVECT2= 000000          1898    1938
BADHDR  007340          2054#   3317*   4033*   4164*   4431*   4621*   5721*   5936*   5939*   6951
```

```
BADTMO  036626          3391    7005#
BAI   = 000020          1214#   4175    4324    4462    4532    4934    4994    5119    5281    5357    5482    5535    5655
                        5789    5834
BA16  = 000400          1200#
BA17  = 001000          1201#
BIT0  = 000001          1143#   1197    1229    1248    2121    3432    6896
BIT00 = 000001          1133#   1143
BIT01 = 000002          1132#   1142
BIT02 = 000004          1131#   1141
BIT03 = 000010          1130#   1140
BIT04 = 000020          1129#   1139
BIT05 = 000040          1128#   1138
BIT06 = 000100          1127#   1137
BIT07 = 000200          1126#   1136
BIT08 = 000400          1125#   1135    7177
BIT09 = 001000          1124#   1134    7185    7253
BIT1  = 000002          1142#   1230    2122
BIT10 = 002000          1123#   1202    1220    1239    1271    1285    1299    1312    1326    7230
BIT11 = 004000          1122#   1203    1221    1240    1257    1272    1286    1300    1313    1327    7192
BIT12 = 010000          1121#   1204    1222    1241    1273    1287    1301    1314    1328
BIT13 = 020000          1120#   1205    1223    1242    1258    1274    1288    1302    1315    1329    7237
BIT14 = 040000          1119#   1206    1224    1243    1259    1275    1289    1303    1316    1330    1340    6698    6709
                        7163
BIT15 = 100000          1118#   1207    1208    1225    1244    1260    1276    1343    5723    5883    6694    6705
BIT2  = 000004          1141#   1231    1250    2123
BIT3  = 000010          1140#   1213    1232    1251
BIT4  = 000020          1139#   1214    1233    1252    1265    1293    1320
BIT5  = 000040          1138#   1215    1234    1253    1266    1280    1294    1307    1321
BIT6  = 000100          1137#   1198    1216    1235    1254    1267    1281    1295    1308    1322
BIT7  = 000200          1136#   1199    1217    1236    1255    1268    1282    1296    1309    1323    3422    4692    4736
                        4788    4885    5143
BIT8  = 000400          1135#   1200    1218    1237    1256    1269    1283    1297    1310    1324
BIT9  = 001000          1134#   1201    1219    1238    1270    1284    1298    1311    1325
BPTVEC= 000014          1150#
BSE   = 000200          1236#   4078    4185    4249    4475    4946    5291    5492
BSERR   001512          2037#   3942*   3947*   4024    5958*   5960*
BSE20H  002336          2046#   3963    6703    6808
BSE20S  004336          2048#   3972    6707    6812
BSE22H  003336          2047#   3883    3944    3981    6692    6797
BSE22S  005336          2049#   3954    6696    6801
BYP     033064          3510    3538    3542    3557    3616    3643    3647    3651    3655    6271#
BYPCER  001516          2039#   3419*   3668*   6386
CALADD  001362          1980#   4147*   4434*   4604*   5892*   6665    6671    6673    6727
CALDIF  001354          1977#   5781*   8967    8991
CCLR  = 100000          1208#   3848    3863    4088    4195    4485    4706    4716    4956    5108    5176    5191    5267
                        5301    5417    5432    5592    5607    6426    6434    6442    6450    6955    6970
CCYL    001350          1975#
CDT   = 002000          1202#   3525    3626    3707    3709
CERR  = 100000          1207#   3487    3600    3894    3999    4051    4075    4157    4182    4247    4333    4446    4472
                        4541    4614    4758    4815    4878    4943    5002    5288    5365    5489    5542    5674    5746
                        5768    5801    5846    5902    5926    6388    6483
CFMT  = 010000          1204#   4440    4443    4466    4469    4507    4535    4538    4570    6794
CHKFLG  001520          2040#   6281*   6364    6367    6371
CHKMSG  033100          3906    4100    4119    4207    4226    4273    4290    4352    4371    4405    4497    4524    4560
                        4587    4657    4678    4727    4779    4842    4968    4987    5021    5040    5079    5100    5238
                        5259    5313    5332    5384    5403    5507    5526    5561    5580    5686    5815    5860    6281#
```

```
CKCERR  033640          6149  6153  6164  6169  6386#
CKSWR = 104407          7162  7226  7252  8149#
CLEAR = 000005          1185# 3865  4708  5110  5193  5269  5434  5609  6972
CLKOF   036172          6867#
CLKON   036076          6845#
CLOCK   036136          3340  6856#
CLRFLG  031724          3258  6004#
COE   = 001000          1238#
COUNT   001366          1984# 6857* 6859*
CR    = 000015          1058# 7351  7361  8161  8165  8167  8171  8181  8190  8197  8205  8210  8215
                        8220  8226  8232  8238  8245  8250  8258  8263  8270  8279  8284  8289  8295
                        8297  8301  8307  8315  8325  8330  8337  8341  8343  8347  8351  8359  8363
                        8371  8379  8385  8393  8399  8406  8411  8414  8420  8438  8446  8521  8548
                        8555  8561  8566  8578  8598
CRLF  = 000200          1059# 7305  7361
CTO   = 004000          1203# 6392
CYL     001422          2001# 5475
CYLADD  001360          1979# 4685  4859  5344  5821  5866  6541* 6542* 6543* 6544* 6545* 6546* 6564*
                        6565* 6566* 6567* 6568* 6569*
CYLDIF  001356          1978# 4689* 4695  4700  4733* 4739  4744  4785* 4791  4796  4863  5341  6506*
                        6507* 6508* 6509* 6510* 6511* 6512  6514* 6532* 6533* 6551* 6552* 6553* 6554*
                        6555* 6556* 6557  6559*
CYL7    001436          2008# 5473
DATA0   001474          2028# 4173  4174  4308  4384  5280  5356  5370
DATA01  001476          2029# 5654
DATA1   001500          2030# 4067  4312  4313  4325  4338  4461  4531  4546  4933  4993  5007  5118
                        5788  5833
DATCMD  032320          3891  4048  4072  4154  4179  4244  4330  4388  4441  4467  4536  4611  4813
                        4940  4999  5123  5285  5362  5486  5539  5743  5793  5838  5899  6129#
DCK   = 100000          1244# 4256
DCLO  = 000020          1252#
DCPAR = 020000          1205#
DDISP = 177570          1065# 1873  3223
DDPCH   007466          2130# 3305* 3495  3700
DDT   = 000400          1256#
DDUMP   007464          2129# 3271* 6004
DF1     054726          2171  2177  2183  2189  2200  2206  2212  2438  2478  2483  2523  9043#
DF10    055056          2217  2222  2227  2232  2237  2247  2257  2262  2272  2288  2308  2313  2348
                        2358  2363  2398  2428  2433  2473  2503  2508  2563  2568  2573  2578  2593
                        2618  2623  2648  2698  2713  2799  2819  2824  2874  2944  2979  3015  9094#
DF14    055076          2844  9103#
DF15    055116          2528  2809  2814  2834  2839  2949  9113#
DF17    055136          2954  2959  2969  2974  9122#
DF2     054742          2338  9050#
DF20    055162          2378  9132#
DF21    055206          2267  2277  2282  2293  2298  2303  2318  2323  2328  2333  2383  2388  2393
                        2403  2408  2413  2418  2423  2653  2658  2663  2668  2919  2924  3075  3080
                        3100  3105  3110  3115  3120  3125  3130  3135  3140  3145  3150  3155  3160
                        3105  9143#
DF22    055242          2343  2368  2373  2498  2513  2518  2553  2558  2583  2588  9158#
DF23    055276          9173#
DF3     054746          2242  2252  9053#
DF4     054772          2353  9064#
DF5     055006          2598  2603  2608  9071#
DF6     055032          2543  2548  2869  3010  9082#
DH1     052041          2169  2175  2181  2187  2198  2204  2210  2351  2436  2476  2481  2521  2842
```

		8757#	9055	9075	9084	9096	9115	9126	9134	9145	9160	9175	
DH10	052326	2471	2817	2822	8790#								
DH11	052355	8794#	9057										
DH13	052372	2952	8797#										
DH14	052456	2957	8806#										
DH17	052542	2366	2371	2396	2591	2917	2922	3138	3143	8815#			
DH19	052566	2286	2561	8819#									
DH2	052056	8760#	9045	9059	9066	9077	9088	9098	9107	9117	9128	9138	9153	9168
		9183											
DH20	052611	2566	8823#										
DH21	052644	2270	2571	8828#									
DH22	052672	2346	2556	2696	2711	3098	3103	3128	3133	8832#			
DH24	052724	2306	2311	2326	2331	2341	2551	3073	3078	8837#			
DH25	052751	2376	2616	2621	2646	2867	2872	3008	8841#				
DH26	052774	2225	2230	2250	2255	2280	2291	2391	2401	2942	2947	8845#	
DH27	053024	2215	2220	2265	2275	2361	2381	2386	2426	2501	2526	2541	2546	8849#
DH28	053055	8854#	9147	9162	9177								
DH29	053065	8856#	9151	9166	9181								
DH3	052126	8767#	9047	9061	9068	9079	9090	9100	9109	9119	9130	9140	9155	9170
		9185											
DH30	053075	2581	2797	2807	2812	3118	3123	3158	3163	8858#			
DH31	053127	8863#	9149	9164	9179								
DH32	053154	2235	2240	2260	2296	2301	2406	2411	2506	8867#			
DH39	053206	2832	2837	3108	3113	3148	3153	8872#					
DH40	053241	8877#	9105										
DH41	053275	3013	8882#										
DH42	053322	2967	8886#										
DH43	053373	2972	8893#										
DH44	053444	2977	8900#	9124									
DH47	053523	2496	8908#										
DH49	053573	8915#	9073										
DH51	053650	2356	2586	2651	2656	2661	2666	8923#					
DH52	053703	2245	8928#										
DH53	053737	2316	2321	2416	2421	8933#							
DH54	053765	2431	8937#										
DH56	054037	8945#	9136										
DH57	054063	2511	2516	8949#									
DH6	052177	8774#	9086										
DH8	052232	2336	8779#										
DH9	052253	2576	8782#										
DI =	040000	1206#											
DISPLA	001142	1873#	3223*	3231*	7207*	7229*							
DISPRE	000174	1352#	3231										
DLT =	100000	1225#	4826										
DLY	033046	3483	3596	5665	6262#	6265							
DMD =	000040	1266#											
DOCMD	032262	3808	3851	3866	3992	4444	4470	4508	4539	4571	4649	4709	4751	4871
		5071	5111	5179	5194	5230	5270	5420	5435	5595	5610	5761	5919	6118#
		6416	6941	6958	6973									
DOTIM	007522	2149#	3339*	3348*	5642									
DPAT1	001502	2031#	5481	5534	5547									
DPAT2	001504	2032#												
DRA =	000001	1248#	3532	3633										
DRDY =	000200	1255#												
DRIVS	007474	2133#	3427*	3434*	3451	3467*	3499*	3518	3620*	3687	5956	6059*		
DRIVO	007476	2138#	3321	3429	3469	3582	6042							

```
DRIV1   007500          2139#
DRIV2   007502          2140#
DRIV3   007504          2141#
DRIV4   007506          2142#
DRIV5   007510          2143#
DRIV6   007512          2144#
DRIV7   007514          2145#
DROT  = 000040          1253#
DRPAR = 000010          1232#
DRVMSK= 000007          1212#   3490    3603
DRVPTR  001342          1971#   3321*   3692    3710*
DSC   = 040000          1259#
DSWR  = 177570          1064#   1872    3222
DTE   = 010000          1241#
DTYE  = 000040          1234#   3523    3624
DT1     054124          2170    2176    2182    2188    2199    2205    2211    2216    2221    2226    2231    2236    2246
                        2256    2261    2271    2287    2307    2312    2347    2352    2357    2362    2397    2427    2432
                        2437    2472    2477    2482    2502    2507    2522    2527    2562    2567    2572    2577    2592
                        2597    2602    2607    2617    2622    2647    2697    2712    2798    2808    2813    2818    2823
                        2833    2838    2873    2943    2948    2953    2958    2968    2973    2978    3014    8959#
DT13    054462          2266    2276    2281    2292    2297    2302    2317    2322    2327    2332    2382    2387    2392
                        2402    2407    2412    2417    2422    2652    2657    2662    2667    2918    2923    3074    3079
                        3099    3104    3109    3114    3119    3124    3129    3134    3139    3144    3149    3154    3159
                        3164    9007#
DT14    054542          2342    2367    2372    2497    2512    2517    2552    2557    2582    2587    9017#
DT15    054632          9028#
DT3     054164          2337    8966#
DT4     054170          2542    2547    2868    8967#
DT6     054236          2241    2251    8975#
DT7     054302          2843    8983#
DT8     054350          3009    8991#
DT9     054416          2377    8999#
D.ACLO= 000100          1308#
D.BRHM= 000100          1295#   3900    4094    4113    4201    4220    4267    4284    4346    4365    4399    4491    4518
                        4554    4581    4654    4672    4724    4773    4836    4962    4981    5015    5034    5076    5094
                        5235    5253    5307    5326    5378    5397    5501    5520    5555    5574    5680    5809    5854
D.CART= 000400          1297#   3900    4094    4113    4201    4220    4267    4284    4346    4365    4399    4491    4518
                        4554    4581    4654    4672    4724    4773    4836    4962    4981    5015    5034    5076    5094
                        5235    5253    5307    5326    5378    5397    5501    5520    5555    5574    5680    5809    5854
D.DDT = 000400          1283#   3731
D.DOOR= 000200          1296#   3900    4094    4113    4201    4220    4267    4284    4346    4365    4399    4491    4518
                        4554    4581    4654    4672    4724    4773    4836    4962    4981    5015    5034    5076    5094
                        5235    5253    5307    5326    5378    5397    5501    5520    5555    5574    5680    5809    5854
D.DRA = 000040          1280#   3898    4092    4111    4199    4218    4265    4282    4344    4363    4397    4489    4516
                        4552    4579    4652    4670    4722    4771    4834    4960    4979    5013    5032    5074    5092
                        5233    5251    5305    5324    5376    5395    5499    5518    5553    5572    5678    5807    5852
D.DRDY= 000200          1282#   3898    4092    4111    4199    4218    4265    4282    4344    4363    4397    4489    4516
                        4552    4579    4670    4722    4771    4834    4960    4979    5013    5032    5092    5251    5305
                        5324    5376    5395    5499    5518    5553    5572    5678    5807    5852
D.DROT= 020000          1315#   5797    5842
D.DSC = 040000          1289#   4670    4771    5092    5251
D.FLT = 000200          1309#
D.FORM= 001000          1284#   4510    4573
D.FWD = 002000          1299#
D.HDFL= 000200          1323#
D.HDHM= 000040          1294#   6621
```

```
D.IDAE= 000040          1307#
D.ILF = 000400          1310#
D.LIMD= 020000          1329#
D.LOAD= 010000          1301#   6638
D.MHD = 000400          1324#
D.NMOV= 010000          1328#
D.OFF = 002000          1285#   4652    4670    4718    4722    4765    4771    4855    5074    5092    5233    5251
D.PAR = 001000          1311#
D.PIP = 020000          1288#   4652    5074    5233
D.PLO = 004000          1327#
D.REV = 004000          1300#
D.RTZ = 020000          1302#   3856    5184    5425    5600    6963
D.SECT= 000020          1320#
D.SKI = 002000          1312#
D.SPIN= 010000          1287#   3898    4092    4111    4199    4218    4265    4282    4344    4363    4397    4489    4516
                        4552    4579    4652    4670    4722    4771    4834    4960    4979    5013    5032    5074    5092
                        5233    5251    5305    5324    5376    5395    5499    5518    5553    5572    5678    5807    5852
D.SPLS= 010000          1314#
D.SPOK= 001000          1298#   3900    4094    4113    4201    4220    4267    4284    4346    4365    4399    4491    4518
                        4554    4581    4654    4672    4724    4773    4836    4962    4981    5015    5034    5076    5094
                        5235    5253    5307    5326    5378    5397    5501    5520    5555    5574    5680    5809    5854
D.SSP = 000020          1293#   3900    4094    4113    4201    4220    4267    4284    4346    4365    4399    4491    4518
                        4554    4581    4654    4672    4724    4773    4836    4962    4981    5015    5034    5076    5094
                        5235    5253    5307    5326    5378    5397    5501    5520    5555    5574    5680    5809    5854
D.SUNS= 040000          1330#
D.TIB = 002000          1326#
D.UNLD= 040000          1303#
D.UNS = 040000          1316#
D.VV  = 000100          1281#   3798    3811    3898    4092    4111    4199    4218    4265    4282    4344    4363    4397
                        4489    4516    4552    4579    4652    4670    4722    4771    4834    4960    4979    5013    5032
                        5074    5092    5233    5251    5305    5324    5376    5395    5499    5518    5553    5572    5678
                        5807    5852
D.WCUR= 000040          1321#
D.WGAT= 000100          1322#
D.WLE = 004000          1313#
D.WRL = 004000          1286#
D.XERR= 001000          1325#
ECCW  = 020000          1274#
ECH   = 000100          1235#
EMTVEC= 000030          1153#   3206*   3207*   3239*
EM1     046331          6068    8420#
EM10    047105          2203    8486#
EM100   051752          2540    8744#
EM101   052007          2580    8749#
EM11    047170          2209    8495#
EM12    047234          2214    2224    2234    2305    2560    2565    2570    2575    2590    2615    2695    2796    2831
                        2941    8501#
EM13    047272          2310    2395    2470    2620    8506#
EM14    047314          2335    8509#
EM15    047352          2350    8514#
EM16    047461          2951    2956    2966    2971    8527#
EM17    047524          2315    238C    2390    2405    2650    2916    3072    3127    3147    3157    8533#
EM18    047545          2264    2279    2295    2415    2655    3077    3097    3107    3117    3137    8536#
EM19    047566          2320    2325    2385    2400    2410    2660    2921    3132    3152    3162    8539#
EM2     046405          2168    8428#
EM20    047607          2274    2290    2300    2330    2420    2665    3102    3112    3122    3142    8542#
```

```
EM21    047630  2219    2229    2259    2269    2811    2836    2871    2946    8545#
EM22    047652  2475    8548#
EM23    047757  2480    8561#
EM24    050057  2285    8573#
EM25    050112  2360    8578#
EM26    050164  2425    8585#
EM27    050243  2435    8593#
EM3     046475  7009    8438#
EM36    050362  2545    2866    3007    8607#
EM39    050425  2365    2510    2645    2816    8613#
EM4     046550  8446#
EM40    050474  2370    2515    2821    8620#
EM41    050532  2340    8625#
EM5     046624  2174    2605    8454#
EM55    050600  2710    8632#
EM6     046645  2180    8457#
EM63    050633  2806    8637#
EM65    050654  2841    8640#
EM69    050676  2976    8643#
EM7     046666  2186    8460#
EM73    050727  2595    8648#
EM74    050750  3012    8651#
EM79    050775  2600    8655#
EM8     046755  8470#
EM80    051016  2239    8658#
EM81    051055  2244    8664#
EM82    051121  2249    8671#
EM83    051173  2254    8679#
EM84    051230  2596    2601    2606    8684#
EM85    051321  2345    2355    8694#
EM86    051364  2550    2555    2585    8700#
EM88    051422  2430    8705#
EM9     047031  2197    8478#
EM93    051463  2375    8711#
EM94    051544  2495    8720#
EM95    051600  2500    2505    8725#
EM96    051634  2520    8730#
EM97    051665  2525    8735#
ERRVEC= 000004  1146#   3220    3221*   3232*   3241*   3242*   3255*   3256*   3324*   3328*   3335*   3373*   3391*
                3392*   7168    7169*   7171*   7174*
ESEC    001400  1990#
F.A0    007444  2110#   3898*   4092*   4111*   4199*   4218*   4265*   4282*   4344*   4363*   4397*   4489*   4516*
                4552*   4579*   4652*   4670*   4722*   4771*   4834*   4960*   4979*   5013*   5032*   5074*   5092*
                5233*   5251*   5305*   5324*   5376*   5395*   5499*   5518*   5553*   5572*   5678*   5807*   5852*
                6285*   6289*   6293    6295*   6324    9007    9017    9028
E.A1    007450  2112#   3900*   4094*   4113*   4201*   4220*   4267*   4284*   4346*   4365*   4399*   4491*   4518*
                4554*   4581*   4654*   4672*   4724*   4773*   4836*   4962*   4981*   5015*   5034*   5076*   5094*
                5235*   5253*   5307*   5326*   5378*   5397*   5501*   5520*   5555*   5574*   5680*   5809*   5854*
                6286*   6297    6299*   6344    9007    9017    9028
E.A2    007454  2114#   3902*   4096*   4115*   4203*   4222*   4269*   4286*   4348*   4367*   4401*   4493*   4520*
                4556*   4583*   4674*   4775*   4838*   4964*   4983*   5017*   5036*   5096*   5255*   5309*   5328*
                5380*   5399*   5503*   5522*   5557*   5576*   5682*   5811*   5856*   6287*   6301    6303*   9017
                9028
E.A3    007460  2116#   6288*
E.B0    007446  2111#   3899*   4093*   4112*   4200*   4219*   4266*   4283*   4345*   4364*   4398*   4420*   4517*
                4553*   4580*   4653*   4671*   4723*   4772*   4835*   4961*   4980*   5014*   5033*   5075*   5093*
```

```
                    5234*  5252*  5306*  5325*  5377*  5396*  5500*  5519*  5554*  5573*  5679*  5808*  5853*
                    6305   6307*  6334   9007   9017   9028
E.B1     007452     2113#  3901*  4095*  4114*  4202*  4221*  4268*  4285*  4347*  4366*  4400*  4492*  4519*
                    4555*  4582*  4655*  4673*  4725*  4774*  4837*  4963*  4982*  5016*  5035*  5077*  5095*
                    5236*  5254*  5308*  5327*  5379*  5398*  5502*  5521*  5556*  5575*  5681*  5810*  5855*
                    6309   6311*  6354   9007   9017   9028
E.B2     007456     2115#  3903*  4097*  4116*  4204*  4223*  4270*  4287*  4349*  4368*  4402*  4494*  4521*
                    4557*  4584*  4675*  4776*  4839*  4965*  4984*  5018*  5037*  5097*  5256*  5310*  5329*
                    5381*  5400*  5504*  5523*  5558*  5577*  5683*  5812*  5857*  6313   6315*  9017   9028
E.B3     007462     2117#  3904*  4098*  4117*  4205*  4224*  4271*  4288*  4350*  4369*  4403*  4495*  4522*
                    4558*  4585*  4676*  4777*  4840*  4966*  4985*  5019*  5038*  5098*  5257*  5311*  5330*
                    5382*  5401*  5505*  5524*  5559*  5578*  5684*  5813*  5858*  6317   6319*  9028
E.DDT    015470     3724*  3731*  3741#  6289
FATT1    032672     3860   5188   5429   5604   6215#  6945   6967
FATT2    032766     3996   4666   4755   4875   5088   5247   5765   5923   6242#
FHDHM    035126     6618#  6626
FHDTAB   035250     4150   4437   4607   5895   6651#
FLGTST   035530     6693   6697   6704   6708   6720#
FLOAD    035202     6635#
FMTE  =  000020     1233#
FMT1     001470     2025#  6661*  6662*  6663*  6668
FORM     031220     5786   5878   5881#  6981
FORMAT   001466     2024#  4149*  4436*  4606*  5894*  6661   6690
FRCYL    001344     1973#  5776*  8967   8991
FRDY     032356     3462   3476   3485   3589   3598   5669   6121   6132   6144#  6147   6478
FRDY1    032424     6158#  6161   6464
FSEC23   035036     5650   6594#
FTITLE   001340     1969#  6015   6017*
GBA      032164     3291   6082#
GDRVS    032024     3289   6037#
GINT     032212     3293   6095#
GNS   =  ****** U   1351   8141   8142   8143   8144   8145   8147   8149   8150   8151   8152   8153   8154
                    8155
GO    =  000001     1197#
GSTAT    033722     3762   3855   3893   4050   4074   4156   4181   4246   4332   4390   4613   4717   4828
                    4942   5001   5125   5183   5287   5340   5364   5424   5488   5541   5599   5671   5745
                    5796   5841   5901   6225   6229   6249   6253   6413#  6482   6493   6505   6540   6620
                    6637   6962
GSTAT1   033756     6283   6424#
GSTAT2   034206     6429   6437   6445   6452   6460#
GTSWR =  104406     6026   8147#
HASOF    007372     2081#  6183*  6199   8963   8972   8980   8988   8996   9004   9014   9025   9037
HBA      007362     2077#  6179*  8963   8972   8980   8988   8996   9004   9014   9025   9037
HCS1     007354     2074#  3479*  3480*  3481   3487   3592*  3593*  3594   3600   3807*  3850*  3865*  3890*
                    3894   3991*  3999   4047*  4051   4071*  4075   4153*  4157   4178*  4182   4243*  4247
                    4329*  4333   4387*  4440*  4443*  4446   4466*  4469*  4472   4507*  4535*  4538*  4541
                    4570*  4610*  4614   4648*  4708*  4750*  4758   4812*  4815   4870*  4878   4939*  4943
                    4998*  5002   5070*  5110*  5122*  5178*  5193*  5229*  5269*  5284*  5288   5361*  5365
                    5419*  5434*  5485*  5489   5538*  5542   5594*  5609*  5658*  5659*  5660   5674   5742*
                    5746   5760*  5768   5792*  5801   5837*  5846   5898*  5902   5918*  5926   6118*  6119
                    6129*  6130   6176*  6388   6392   6415*  6460*  6461*  6462   6483   6522*  6523*  6524
                    6794   6940*  6957*  6972*  8960   8969   8977   8985   8993   9001   9011   9022   9034
HCS2     007356     2075#  3528   3530   3534   3546   3629   3631   3635   3638   4335   4391   4543   5004
                    5126   5367   5544   6177*  6397   6401   8960   8969   8977   8985   8993   9001   9011
                    9022   9034
HDA      007364     2078#  6180*  6837   8963   8972   8980   8988   8996   9004   9014   9025   9037
```

```
HDB     007376          2083#
HDC     007374          2082#  6184*  6829   8963   8972   8980   8988   8996   9004   9014   9025   9037
HDS     007366          2079#  3532   3633   6181*  8960   8969   8977   8985   8993   9001   9011   9022   9034
HDTAB   001522          2042#  4035   4041   4044   4143   4428   4599   5728   5735   5738   5888   6653
HDWD    001510          2035#  8983
HD1     001464          2023#  6655*  6656*  6657*  6658*  6659*  6660*  6667
HEAD    001460          2021#  4148*  4435*  4605*  5893*  6655   6732
HEADA   001462          2022#  6584*  6585*  6586*  6587*
HER     007370          2080#  3523   3624   4078   4185   4249   4256   4475   4946   5291   5492   6182*  8960
                        8969   8977   8985   8993   9001   9011   9022   9034
HMR1    007400          2084#  6185*  8960   8969   8977   8985   8993   9001   9011   9022   9034
HMR2    007402          2085#  3489   3602   3763   3798   3811   3856   4510   4573   4718   4765   4855   5184
                        5425   5600   6162*  6167*  6186*  6431   6439   6447   6454   6506   6621   6638   6963
                        8960   8969   8977   8985   8993   9001   9011   9022   9034
HMR3    007404          2086#  5797   5842   6163*  6168*  6187*  6432   6440   6448   6455   6494   6541   8960
                        8969   8977   8985   8993   9001   9011   9022   9034
HOLD    032512          6148   6152   6176#  6425
HPAT    007410          2088#  6189*  8963   8972   8980   8988   8996   9004   9014   9025   9037
HPEND   007342          2056#  3318*  5933   5935*  6953*
HPOS    007406          2087#  6188*  8963   8972   8980   8988   8996   9004   9014   9025   9037
HT    = 000011          1056#  7303   7361
HVRC  = 000400          1237#
HWC     007360          2076#  6178*  8963   8972   8980   8988   8996   9004   9014   9025   9037
HZ      001364          1982#  6859
H.A0    007424          2099#  6324   6454*  9007   9020   9031
H.A1    007430          2101#  6344   6447*  9007   9020   9031
H.A2    007434          2103#  6439*  6551   9020   9031
H.A3    007440          2105#  6431*
H.B0    007426          2100#  6334   6455*  9007   9020   9031
H.B1    007432          2102#  6354   6448*  9007   9020   9031
H.B2    007436          2104#  6440*  6564   9020   9031
H.B3    007442          2106#  6432*  6574   6584   9031
IDAE  = 002000          1239#
IE    = 000100          1198#
ILF   = 000001          1229#
INTER   036756          6110   7049#
IOTVEC= 000020          1151#  3204*  3205*
IR    = 000100          1216#
LC      015460          3723*  3730*  3737#  3888
LCLKF   007516          2147#  3326*
LCVEC   001330          1961#  3327
LF    = 000012          1057#  7355   7361   8161   8165   8167   8171   8181   8190   8197   8205   8210   8215
                        8220   8226   8232   8238   8245   8250   8258   8263   8270   8279   8284   8289   8295
                        8297   8301   8307   8315   8325   8330   8337   8341   8343   8347   8351   8359   8363
                        8371   8379   8385   8393   8399   8406   8411   8414   8420   8438   8446   8521   8548
                        8555   8561   8566   8578   8598
LIMERR  001514          2038#
LKS     001326          1959#  3325   6848*  6869*
MASK    015464          3726*  3733*  3739#  6512   6557
MASK1   015466          3727*  3734*  3740#  6507   6533   6542   6552   6565
MCLK  = 000400          1269#
MC1     015462          3725*  3732*  3738#  5756   5877   5912
MDS   = 001000          1219#  3528   3629   6401
MEMBAS= 172100          1965#  3244   3253
MEMERR  036714          3247   7035#
MEMPAR  013072          3241#
```

```
MEMVEC= 000114           1964#   3247*   3248*
MERD  = 001000           1270#
MESMSK= 000017           1264#
MEWD  = 002000           1271#
MIND  = 000200           1268#
MSG1    043542           6018    8161#
MSG10   045105           3426    3466    8297#
MSG11   045132           7082    8301#
MSG12   045172           5206    8307#
MSG13   045337           3349    8325#
MSG14   045435           6271    8337#
MSG15   045460           3548    3711    8341#
MSG16   045472           3760    8343#
MSG17   045516           3980    8347#
MSG2    044403           3272    8238#
MSG26   045546           4027    4054    4106    4160    4213    4253    4358    4449    4503    4566    4617    4818    4830
                         4974    5027    5319    5390    5513    5567    5749    5905    8351#
MSG27   045624           3689    8359#
MSG3    044562           3288    8258#
MSG37   045654           5158    8363#
MSG38   045732           5171    8371#
MSG39   046012           5134    8379#
MSG4    044614           3290    8263#
MSG40   046054           4026    8385#
MSG5    044661           3292    8270#
MSG6    044742           7054    8279#
MSG7    044776           3307    8284#
MSG74   046167           6932    8399#
MSG75   046235           6934    8406#
MSG76   046272           6985    8411#
MSG77   046314           6990    8414#
MSG8    045033           5047    8289#
MSG9    045072           5051    8295#
MSP   = 000100           1267#
MULT6   035714           6752    6778#
M.ALGN= 040000           1340#
M.DPID  037100           7097#
M.DP40  037136           7108#   7130
M.DP41  037172           7113    7121#
M.DP42  037202           7119    7125#
M.DP44  037234           7132    7138#
M.DP50  037246           7106    7143#
M.DRV = 000007           1334#
M.HEAD= 007000           1342#   6585
M.ID  = 000003           1339#
M.PAR = 100000           1343#   6898
M.SECT= 000760           1341#   6495    6575
M.SER = 077770           1335#
NED   = 010000           1222#   3534    3546    3635    3638    6397
NEM   = 004000           1221#
NUDRV   015172           3560    3668#   5959
NXF   = 000004           1231#
OFFERR  001456           2018#   5059*   5130*   5132    5148    5156    5165    5169
OFFSET= 000015           1189#   4648    5070    5229
OFST  = 000004           1250#
OPI   = 020000           1242#
```

```
OR     = 000200  1217#
O.ADR1   055664  9318* 9369* 9371* 9541  9542* 9580* 9595  9612  9803#
O.BACK   056410  9413  9454#
O.BD     057702  9599  9600  9748#
O.BKPT   056116  9364# 9417
O.Bk1    057050  9312  9577#
O.BK2    057104  9582  9585#
O.BK3    057122  9584  9591#
O.BRK    057040  9329  9575#
O.BUF    057731  9657  9666  9671  9776#
O.B2     057204  9596  9605#
O.B3     057244  9604  9613#
O.CADV   057414  9430  9447  9470  9474  9504  9508  9613  9656#
O.CLGL   056232  9387  9389  9397#
O.CLGT=  000014  9401  9766#
O.CLS1   057640  9724  9726#
O.CR     057704  9733  9735  9736  9750#
O.CRET   056344  9408  9437#
O.CRLF   057650  9445  9502  9598  9637  9642  9733#
O.CRLS   057656  9382  9735#
O.CRS    057662  9734  9736#
O.CSR1   057700  9633* 9649  9744#
O.CSR2   057701  9634* 9650  9745#
O.CT     055666  9562* 9605* 9607* 9804#
O.C1     057016  9565# 9606
O.DCD    056162  9370  9372  9382# 9439  9490  9615
O.DCD1   056166  9383# 9431  9475
O.EFF    056472  9416  9481#
O.EFF1   056610  9493  9512#
O.ENTR   055676  9309#
O.ERR    056146  9356  9367  9379# 9402  9427  9485  9728
O.ERR1   056502  9485# 9558
O.ERR2   056322  9427# 9461  9534
O.FTYP   057556  9381  9449  9464  9506  9611  9682  9698  9703#
O.GET    057500  9335  9385  9679# 9684  9686  9688
O.GO     056654  9410  9532#
O.GO2    056732  9543# 9568
O.LG   = 000006  9339  9774#
O.LGCH   057707  9398  9754# 9766
O.LGDR   056262  9405  9407# 9419
O.LGL  = 000030  9419#
O.LGL1   056234  9398# 9403
O.LGL2   056254  9399  9404#
O.MSK    055656  9486  9487  9489  979* #
O.ODT    055672  1360  9307# 9785
O.OFST   056420  9414  9460#
O.OF1    056466  9473  9475#
O.OP1    056356  9411  9443#
O.OP2    056364  9360  9445# 9456
O.ORPC   056070  9352# 9412
O.P      057677  9327* 9532* 9557  9559* 9608* 9742#
O.PRI    055654  9319  9581  9585  9795#
O.PROC   056762  9418  9557#
O.PR1    057004  9561  9563#
O.RALL   056136  9330  9365  9371#
O.RCSR=  177560  9297# 9633  9635* 9645  9647  9649* 9679
```

```
O.RDB = 177562          9296#   9681
O.REGT  056020          9335#   9409
O.RSE1  057376          9646    9649#
O.RSP   056030          9337#   9340
O.RST   055734          9308    9317#
O.RSTT  057346          9538    9564    9642#
O.RST1  055764          9316    9324#
O.RTIT  056760          9325*   9552#
O.SCAN  056172          9346    9385#   9396
O.SP    056062          9338    9347#
O.SPC   057426          9654#   9667
O.SP1   056050          9343#   9348
O.STM = 000340          9286#   9328    9537    9563    9565
O.STRT  055722          9307    9314#
O.SVR   057256          9317    9577    9619#
O.SVTT  057314          9597    9609    9633#
O.T     057676          9539*   9566*   9578    9741#
O.TBIT  056704          9539#   9579
O.TBT = 000020          9287#   9540    9567    9603
O.TC    057642          9721    9727#
O.TCLS  057624          9352    9437    9443    9454    9720#
O.TCSR= 177564          9299#   9634    9636*   9643    9650*   9703    9709
O.TDB = 177566          9298#   9705*   9711*
O.TL    057723          9336    9339    9347    9768#   9774
O.TRTC  057740          9371    9542    9781#
O.TVEC= 000014          9285#   9310    9328*   9329*
O.TYPE  057542          9601    9672    9695#   9699    9737
O.TYP1  057622          9696    9707    9715#
O.TYP2  057602          9709#   9710    9713
O.UIN   055670          9318    9541*   9580    9805#
O.UPC   055650          9311*   9536*   9551    9575*   9592    9594*   9793#
O.URO   055632          9314    9344    9786#
O.USP   055646          9315*   9620*   9621    9792#
O.UST   055652          9309*   9540*   9550    9567*   9576*   9583    9603*   9794#
O.WDS   056500          9482    9484#
O.WDS2  056516          9489#   9511
O.WDS3  056546          9500#   9513    9519    9528
O.WDS4  056604          9500    9510#
O.WRD   056312          9407    9423#
O.WRDA  056336          9424    9432#
O.WRD1  056320          9426#   9433    9450
O.WSCH  056476          9415    9483#
O.XXX   057674          9619*   9627    9740#
O.45    055776          9325    9327#
PACK  = 000003          1184#   3807
PARAM   001336          1968#   3170*   3173*   3285
PARSRT  012546          1357    3170#
PAT   = 000020          1265#
PCA   = 004000          1272#
PCD   = 010000          1273#
PCLKF   007520          2148#   3330*   3337*   6846    6867
PCVEC   001332          1962#   3331    3338
PCYL    001352          1976#
PFSRT   015472          3743#   7084
PGE   = 002000          1220#
PIP   = 020000          1258#
```

```
PIRQ   = 177772          1063#
PIRQVE= 000240           1157#
PKRB    001324           1958#
PKS     001320           1956#   3329    3336    6851*   6871*
PKSB    001322           1957#   6850*
PPTP    007472           2132#   3276*
PRGSRT  012562           3171    3174#   6069
PR0    = 000000          1080#   3176    3239    3240    3368
PR1    = 000040          1081#
PR2    = 000100          1082#
PR3    = 000140          1083#
PR4    = 000200          1084#
PR5    = 000240          1085#   1955
PR6    = 000300          1086#   3341
PR7    = 000340          1087#   3242    3248    3392    7079    7080
PS     = 177776          1060#   1061
PSEC    001376           1989#
PSW    = 177776          1061#
PWRVEC= 000024           1152#   3210*   3211*   7069*   7078*   7079*
QKCYLD  034466           6520#
RCYLA   034714           6369    6564#
RCYLD   034642           6366    6551#
RDCHR  = 104410          7768    8150#
RDCYLA  034572           6539#
RDCYLD  034402           6504#
RDDATA= 000021           1191#   3890    4243
RDGATE= 100000           1276#
RDHEAD= 000025           1193#   4812
RDLIN  = 104411          6037    7842    8151#
RDOCT  = 104412          6082    6095    8152#
RDSEC   034332           6492#   6595    6599    6602    6750
RDTAB   006336           2050#   4239    4297    4305
RDY    = 000200          1199#   5667    6144    6158    6526
RECAL  = 000013          1188#   3850    5178    5419    5594    6957
RESREG= 104414           7902    7954    8154#   9254
RESVEC= 000010           1147#
RHEAD   035010           6374    6584#
RHTAB   001726           2043#   4811    4849    6757    6759    6763    6766    8999
RKASOF= 000016           1170#   3382    4645*   4690*   4734*   4786*   5064*   5226*   6183    6218    6244
RKBA   = 000004          1165#   3378    3884*   3937*   3955*   3964*   3973*   4044*   4067*   4143*   4173*   4239*   4312*
                         4325*   4384*   4428*   4461*   4531*   4599*   4933*   4993*   5118*   5280*   5356*   5481*   5534*
                         5654*   5738*   5788*   5833*   5888*   6179
RKCS1  = 000000          1163#   3375    3481*   3594*   3848*   3863*   4088*   4195*   4485*   4706*   4716*   4956*   5108*
                         5176*   5191*   5267*   5301*   5417*   5432*   5592*   5607*   5660*   5667    6119*   6130*   6144
                         6158    6176    6426*   6434*   6442*   6450*   6462*   6524*   6526    6955*   6970*
RKCS2  = 000010          1167#   3376    3460*   3474*   3478*   3587*   3591*   3849*   3864*   4175*   4324*   4462*   4532*
                         4707*   4826    4934*   4994*   5109*   5119*   5177*   5192*   5268*   5281*   5357*   5418*   5433*
                         5482*   5535*   5593*   5608*   5655*   5789*   5834*   6177    6414*   6427*   6435*   6443*   6451*
                         6476*   6480*   6956*   6971*
RKDA   = 000006          1166#   3379    3886*   3939*   3957*   3966*   3975*   4066*   4172*   4238*   4314*   4327*   4386*
                         4460*   4533*   4932*   4996*   5066*   5121*   5279*   5359*   5657*   6180
RKDB   = 000024          1172#   3384    4337    4545    4821    4822    4823    5006    5129    5369    5546
RKDC   = 000020          1171#   3383    3888*   4806*   5480*   5533*   5740*   5758*   5787*   5890*   5915*   6184
RKDS   = 000012          1168#   3380    6181
RKECPS= 000030           1176#   3388    6188
RKECPT= 000032           1177#   3389    6189
```

```
RKER   = 000014  1169# 3381  6182
RKMR1  = 000026  1173# 3385  3761* 3854* 5182* 5339* 5423* 5598* 6185  6224* 6228* 6248* 6252*
                 6428* 6436* 6444* 6481* 6492* 6504* 6521* 6539* 6619* 6636* 6961*
RKMR2  = 000034  1174# 3386  6162  6167  6186  6532
RKMR3  = 000036  1175# 3387  6163  6168  6187
RKPRI    001316  1955# 6111
RKVEC    001314  1954# 3312* 6099* 6102* 6109
RKWC   = 000002  1164# 3377  3889* 4045* 4068* 4144* 4176* 4240* 4326* 4385* 4429* 4463* 4530*
                 4600* 4935* 4995* 5120* 5282* 5358* 5483* 5536* 5656* 5739* 5790* 5835* 5889*
                 6178
RLS    = 000010  1213#
RSEC     034752  6373  6574#
RTT    = 000006  9289#
SAVREG= 104413   7887  7928  8153#  9198
SBPAR    036214  6294  6298  6302  6306  6310  6314  6318  6884#
SCLR   = 000040  1215# 3460  3474  3587  6476
SCOP1  = 104415  3471  3584  3801  4059  4165  4232  4318  4378  4453  4593  5349  5778  5826
                 8155#
SCOP1$   036266  6908# 8155
SEC      001370  1986# 6860*
SECFLG   035430  6669  6689#
SECNT    001374  1988#
SECTOR   001402  1991# 4065* 4066  4083* 4084  4171* 4172  4190* 4191  4238  4314  4327  4386
                 4459* 4460  4480* 4481  4533  4931* 4932  4951* 4952  4996  5121  5278* 5279
                 5296* 5297  5359  6494* 6495* 6496* 6497* 6498* 6499* 6574* 6575* 6576* 6577*
                 6578* 6579* 6596  6600  6603  6751* 6755  6756* 6757* 6768  6778* 6779  6780*
                 6781*
SEEK   = 000017  1190# 3991  4750  4870  5760  5918
SELDRV= 000001   1183# 3479  3592  4443  4469  4507  4538  4570  6415  6460  6522
SETINT   032244  3313  6100  6109#
SIZFLG   007524  2150# 3314* 3464  6055* 6075*
SKI    = 000002  1230#
SORT     035606  6748#
SPACE2   055606  9232  9241  9252  9264#
SRTSPL= 000011   1187# 6940
SRTTAB   002132  2044# 6760
ST     = 177776  9283# 9309  9310* 9323* 9537* 9563* 9565* 9591*
STACK  = 001100  1051# 3175  3202  3366  3417  3472  3580  3585  3683  3753  3793  3802  3842
                 4022  4060  4138  4166  4233  4319  4379  4423  4454  4594  4638  4926  5219
                 5350  5469  5640  5720  5779  5827  5954  6379  7043  7081  9260
STAPT    031672  5986  5992#
START    012556  1355  3173#
START1   013200  3266#
STKLMT= 177774   1062#
STOP     036320  5937  6923#  7609  7692
ST2      013250  3274  3285#
ST3      013314  3275  3303#
ST4      013346  3287  3308  3311#
ST5      013372  3294  3316#  3457  3522  5998  6992
SUBCLR   034252  3757  3795  3804  3844  3875  3913  3987  4030  4062  4140  4168  4235  4321
                 4381  4425  4456  4596  4642  4802  4867  4928  5061  5223  5352  5477  5647
                 5725  5783  5829  5885  6476#  6925  6937
SVAL   = 100000  1260#
SWR      001140  1872# 3200  3222* 3224  3230* 3237* 6024  6330  6340  6350  6360  6908  7007
                 7015  7020  7039  7163  7177  7179  7185  7192  7230  7237  7249  7253  7612
                 7651  7706* 9256
```

```
SWREG   000176          1353#  3230   6024   7612   7651   7674
SW0   = 000001          1115#
SW00  = 000001          1105#  1115
SW01  = 000002          1104#  1114
SW02  = 000004          1103#  1113
SW03  = 000010          1102#  1112
SW04  = 000020          1101#  1111
SW05  = 000040          1100#  1110
SW06  = 000100          1099#  1109
SW07  = 000200          1098#  1108
SW08  = 000400          1097#  1107
SW09  = 001000          1096#  1106
SW1   = 000002          1114#
SW10  = 002000          1095#
SW11  = 004000          1094#
SW12  = 010000          1093#  9256
SW13  = 020000          1092#  7007
SW14  = 040000          1091#  7020
SW15  = 100000          1090#
SW2   = 000004          1113#
SW3   = 000010          1112#
SW4   = 000020          1111#
SW5   = 000040          1110#
SW6   = 000100          1109#
SW7   = 000200          1108#
SW8   = 000400          1107#
SW9   = 001000          1106#  6330   6340   6350   6360   6908   7015   7039
TBITVE= 000014          1148#
TEMP1   007412          2091#  3461*  3475*  3482*  3484*  3489*  3490*  3491   3588*  3595*  3597*  3602*  3603*
                        3604   3995*  4665*  4754*  4874*  5087*  5246*  5662*  5764*  5922*  6120*  6131*  6146*
                        6160*  6216*  6220*  6246*  6262   6264*  6291   6293*  6295   6297*  6299   6301*  6303
                        6305*  6307   6309*  6311   6313*  6315   6317*  6319   6321*  6413   6418*  6424   6457*
                        6463*  6477*  6520   6525*  6528*  6534*  6594*  6606*  6618*  6623*  6635*  6640*  6671*
                        6674*  6675   6890*  6898*  8983
TEMP2   007414          2092#  3859*  3878*  3916*  3917   3953*  3962*  3971*  5187*  5228*  5603*  6222*  6625*
                        6672*  6673*  6675*  6676   6944*  6966*
TEMP3   007416          2093#  3879*  3919   3924   3929   3950*  3951   3960   3969
TEMP4   007420          2094#  3883*  3884   3937   3954*  3955   3963*  3964   3972*  3973
TEMP5   007422          2095#  3885*  3886   3938*  3939   3956*  3957   3965*  3966   3974*  3975   5646*  5657
                        5693   5695*  5723*  5754*  5883*  5910*
TERR1   036036          6798   6802   6809   6813   6827#  6832   6839
TIMUP   001372          1987#  6845*  6856*  6862*
TITLE   031744          3268   3306   6015#
TKVEC = 000060          1155#  7584*  7585*
TOCYL   001346          1974#  4145*  4147   4430*  4434   4601*  4604   4805*  4849   5722*  5730   5733   5740
                        5755*  5756   5775*  5781   5787   5821   5876*  5877   5882*  5890   5892   5911*  5912
                        8967   8991   8999
TPVEC = 000064          1156#
TRAPPC  001334          1966#  7036*  8966
TRAPVE= 000034          1154#  3208*  3209*  3240*  7080*
TRT   = 000003          9288#  9781
TRTVEC= 000014          1149#
TRUERR  035736          4080   4187   4477   4948   5293   5494   6792#
TSTATN  032640          3868   4711   5113   5196   5272   5437   5612   6197#  6975
TST1    013562          3342   3364#
TST10   016730          3943   3979   4020#
```

```
TST11   017432          4136#
TST12   021202          4421#
TST13   022424          4636#
TST14   023722          4900    4924#
TST15   025466          5166    5217#
TST16   027002          5467#
TST17   027762          5638#
TST2    013744          3393    3415#
TST20   030262          5644    5652    5694    5718#
TST3    014600          3578#
TST4    015202          3614    3681#
TST5    015472          3728    3735    3751#
TST6    015620          3756    3791#
TST7    015714          3799    3812    3840#
TYPDS = 104405          5975    8145#
TYPE  = 104401          3272    3288    3290    3292    3307    3349    3426    3436    3466    3502    3548    3689    3711
                        3760    3980    3984    3985    4026    4027    4054    4106    4160    4213    4253    4358    4449
                        4503    4566    4617    4818    4830    4974    5027    5047    5051    5057    5134    5140    5141
                        5158    5171    5206    5319    5390    5513    5567    5749    5905    5973    5976    6018    6068
                        6271    6932    6934    6985    6990    7009    7054    7082    7232    7240    7308    7419    7545
                        7606    7618    7672    7673    7676    7687    7697    7708    7727    7775    7781    7786    7790
                        7795    7796    7798    7801    7805    7871    7873    8013    8141#   9208    9209    9213    9214
                        9223    9232    9236    9241    9243    9249    9252
TYPERR  055332          7239    9198#
TYPOC = 104402          7675    8142#   9229
TYPON = 104404          8144#
TYPOS = 104403          3439    3505    3551    3715    5054    5137    6274    7012    7057    8143#
T.A2  = 000001          2121#   4101    4120    4208    4227    4274    4291    4353    4372    4498    4561    4679    4728
                        4780    4843    4969    4988    5022    5041    5101    5260    5314    5333    5385    5404    5508
                        5527    5562    5581    5816    5861    6364
T.B2  = 000002          2122#   4101    4120    4208    4227    4274    4291    4353    4372    4498    4561    4679    4728
                        4780    4843    4969    4988    5022    5041    5101    5260    5314    5333    5385    5404    5508
                        5527    5562    5581    5816    5861    6367
T.B3  = 000004          2123#   6371
T1      001404          1993#
T10     001406          1994#   3461    3475    3484    3588    3597    3859    5187    5428    5603    6120    6463    6477
                        6525    6966
T100    001414          1997#   6944
T50     001410          1995#
T500    001412          1996#
T5000   001416          1998#   4665    5087    5246    6594
T50000  001420          1999#   3482    3595    3995    4754    4874    5764    5922    6131
UFE   = 000400          1218#   3530    3631
UNLD    007336          2052#   3316*   6928    6948*
UNLOAD= 000007          1186#
UNS   = 040000          1243#
UPE   = 020000          1223#
VERIFY  014600          3453    3458    3562#
VV    = 000100          1254#
WCE   = 040000          1224#   4335    4391    4543    5004    5126    5367    5544
WDCNT   001472          2026#   8983
WD1     001452          2015#   4298*   4337*   4545*   5006*   5129*   5369*   5546*   8975
WD2     001454          2016#   4299*   4338*   4546*   5007*   5370*   5547*   8975
WLE   = 004000          1240#
WORD    001506          2034#
WRDATA= 000023          1192#   4071    4178    4466    4939    5284    5485    5658    5792    5837
```

```
WRHEAD= 000027          1194#  4047   4153   4440   4610   5742   5898
WRL   = 004000          1257#
WRTCHK= 000031          1195#  4329   4387   4535   4998   5122   5361   5538
WRTGAT= 040000          1275#
$APTHD  001000          1382   1388#
$ASTAT= ****** U        7462   7477
$ATYC   040542          7433   7435#
$ATY1   040516          7431#
$ATY3   040524          7293   7432#
$ATY4   040534          7245   7434#
$AUTOB  001134          1869#  6028*  7665   7822
$BASE   001264          1939#  3311*  3374   6086*  6088*
$BDADR  001122          1864#
$BDDAT  001126          1866#
$BELL   001200          1890#  7232   7265   7618   7815
$CDW1   001270          1941#
$CDW2   001272          1942#
$CHARC  040266          7310*  7320*  7327   7353*  7358#
$CKSWR  041522          7651#  8149
$CMTAG  001100          1852#  3197   3198   3206   3212   3213   3214
$CM3  = 000000          1882#
$CM4  = 000006          1882#  1883#  1884#  1885#  1886#  1887#  1888#
$CNTLC  042432          7606   7687   7815#
$CNTLG  042444          7672   7817#
$CNTLU  042437          7697   7790   7816#
$CPUOP  001236          1913#
$CRLF   001205          1892#  3436   3502   3984   3985   5057   5140   5141   7240   7265   7309   7361   7708
                        7795   7815   7876   9208   9213   9223   9236   9249
$DBLK   040506          7385   7419   7427#
$DB2D   042754          7928#  7990
$DB2O   042634          3982   7887#
$DDW0   001274          1943#
$DDW1   001276          1944#
$DDW2   001300          1945#
$DDW3   001302          1946#
$DDW4   001304          1947#
$DDW5   001306          1948#
$DDW6   001310          1949#
$DDW7   001312          1950#
$DECVL  043134          7930   7976#
$DEVCT  001220          1904#  3259*  3322*  3693   5955*  5956
$DEVM   001266          1940#  3430
$DOAGN  031646          5969   5978   5984#
$DTBL   040476          7388   7423#
$ENDAD  031636          1368   5980#  7260
$ENDCT  031604          3212   5971#
$ENDMG  031655          5973   5988#
$ENULL  031652          5976   5987#
$ENV    001230          1909#  5992   6022   7242   7288   7440   7464
$ENVM   001231          1910#  3235   3422   7290   7295   7442
$EOP    031476          4028   4055   4107   4161   4214   4254   4359   4450   4504   4567   4618   4819   4831
                        4975   5028   5320   5391   5514   5568   5750   5906   5949#  9261
$EOPCT  031576          3212*  5968#  5972   6986*
$EOP1   031550          3397   3690   5961   5962#  6988   7031
$ERFLG  001103          1855#  6910   7152   7181   7183   7189*  7211   7227*  7265   9258
$ERMAX  001115          1861#  3215*  7183   7206*  7211
```

```
$ERROR  037536          3206    7225#
$ERRPC  001116          1862#   7234*   7235*   7236    7265    8959    8967    8975    8983    8991    8999    9007    9017
                        9028
$ERRTB  007526          2165#   9205
$ERTTL  001112          1859#   7233*   7265
$ESCAP  001176          1889#   3214*   3319*   4647*   4664*   5069*   5086*   5228*   5245*   5786*   5881*   5916*   5952*
                        6322    6326*   6336*   6346*   6356*   6376*   6380*   6987*   7035*   7038*   7205*   7256    7258
                        7265
$ETABL  001230          1908#
$ETEND  001314          1394    1951#
$FATAL  001212          1901#   7468*
$FFLG   040762          7431*   7434*   7462    7471*   7479#
$FILLC  001156          1880#   7313    7361
$FILLS  001155          1879#   7361
$GDADR  001120          1863#
$GDDAT  001124          1865#
$GET42  031626          5977#
$GTSWR  041612          7673#   8147
$HD   = 000000          1022
$HIBTS  001000          1389#
$HIOCT  042632          7864*   7875#
$ICNT   001104          1856#   7196*   7197    7199*   7210
$INTAG  001135          1870#   7670*   7689    7709    7822
$ITEMB  001114          1860#   7236*   7244    7265    9199
$LF     001206          1893#   7265    7361    7805    7815    7876
$LFLG   040761          7472*   7478#
$LPADR  001106          1857#   3216*   7018    7187*   7203*   7208    7210
$LPERR  001110          1858#   3217*   6381    6912    6915*   7044    7187    7204*   7210    7255
$MADR1  001242          1926#
$MADR2  001246          1930#
$MADR3  001252          1933#
$MADR4  001256          1936#
$MAIL   001210          1390    1394    1899#   3234    6022    7202    7242    7288
$MAMS1  001240          1920#
$MAMS2  001244          1928#
$MAMS3  001250          1931#
$MAMS4  001254          1934#
$MBADR  001002          1390#
$MFLG   040760          7432*   7438    7473*   7477#
$MNEW   042462          7676    7820#
$MSGAD  001224          1906#   7448*   7451
$MSGLG  001226          1907#   7453*
$MSGTY  001210          1900#   7446    7454*   7466    7470*
$MSWR   042451          7673    7818#
$MTYP1  001241          1921#
$MTYP2  001245          1929#
$MTYP3  001251          1932#
$MTYP4  001255          1935#
$MULT   043244          8032#
$MXCNT  037534          7200    7210#
$NULL   001154          1878#   7315    7361
$NWTST= 000001          3354#   3356    3400#   3402    3563#   3565    3672#   3674    3744#   3746    3785#   3787    3817#
                        3819    4010#   4012    4126#   4128    4413#   4415    4623#   4625    4903#   4905    5207#   5209
                        5445#   5447    5620#   5622    5699#   5701
$OCNT   041206          7517*   7546*   7559#
$OCTVL  042736          3764    7889    7914#
```

```
$OMODE  041210          7512* 7516* 7521  7524* 7535* 7561#
$OVER   037520          7164  7180  7188  7198  7207#
$PASS   001216          1903# 3234* 3755  3978  5965* 5966* 5974  5987  5994  7194  7211
$PASTM  001006          1392#
$PWRCT  037076          7075* 7076* 7086#
$PWRDN  037012          3210  7069# 7078
$PWRUP  037024          7069  7075#
$QUES   001204          1891# 7265  7361  7727  7798  7815  7873  7876
$RDCHR  042064          7740# 8150
$RDDEC= ****** U        8153
$RDLIN  042154          7763# 8151
$RDOCT  042474          7837# 8152
$RDSZ = 000022          7756#
$RESRE  043414          8098# 815↳
$RTNAD  031650          5986#
$R2A  = ****** U        8155
$SAVRE  043356          8082# 8153
$SB2D   043150          7988#
$SCOPE  037256          3204  7161#
$SETUP= 000137          3167# 3203  3204  3206  3208  3210  3212  3213  3214  3216  5963  6019  7162
                        7226  7252  7260  7610  7615  7616  7646  7822
$STUP = 177777          3167#
$SUPRS  043204          3781  3983  8005#
$SVLAD  037464          7172  7201#
$SVPC = 000244          1366# 1371
$SWR  = 167400          1007# 1022  1027  1028  1029  1030  1031  1032  1033  1034  1888  1889  1890
                        3213  3214  3216  3217  3365  3416  3579  3682  3752  3792  3841  4021  4137
                        4422  4637  4925  5218  5468  5639  5719  5946  5964  5979  5985  5987  7153
                        7154  7155  7156  7157  7163  7175  7177  7178  7181  7182  7183  7190  7191
                        7192  7204  7207  7210  7217  7218  7219  7220  7221  7230  7237  7249  7253
                        7265
$SWREG  001232          1911# 3237
$SWRMK= 000000          1034  1035  7157  7158  7179
$TESTN  001214          1902# 3684* 7202* 8959  8966  8967  8975  8983  8991  8999  9007  9017  9028
$TIMES  001174          1888# 3213* 3365* 3416* 3579* 3682* 3752* 3792* 3841* 4021* 4137* 4422* 4637*
                        4925* 5218* 5468* 5639* 5719* 5953* 5964* 7190* 7197  7200* 7210
$TKB    001146          1875# 7334  7341  7361  7565  7586  7597  7627  7655  7682
$TKCNT  041212          7566# 7581* 7616  7633* 7747  7749*
$TKINT  041222          3180  7581# 7607  7668
$TKQEN= 041221          7570# 7641  7752
$TKQIN  041214          7567# 7582* 7583  7639* 7640* 7641  7643*
$TKQOU  041216          7568# 7583* 7750  7751* 7752  7754*
$TKQSR  041220          7569# 7582  7643  7754
$TKS    001144          1874# 7332  7339  7361  7565  7587* 7623* 7625  7631* 7653  7669* 7679  7691*
                        7711*
$TKSRV  041272          7584  7597#
$TMP0   001160          1882#
$TMP1   001162          1883#
$TMP2   001164          1884#
$TMP3   001166          1885#
$TMP4   001170          1886# 3320* 3480  3501  3515* 3525* 3593  3608  3611* 3626* 3670* 3709* 3721
                        5471  5659  6118  6129  6461  6523
$TMP5   001172          1887# 6322* 6376  6380
$TN   = 000021          1008# 1022  3342  3354  3365# 3393  3400  3416# 3563  3579# 3614  3672  3682#
                        3684  3685  3728  3735  3744  3752# 3756  3785  3792# 3799  3812  3817  3841#
                        3943  3979  4010  4021# 4126  4137# 4413  4422# 4623  4637# 4900  4903  4925#
```

```
                        5166    5207    5218#   5445    5468#   5620    5639#   5644    5652    5694    5699    5719#
$TNPWR  043064          7935    7936    7956#
$TPB    001152          1877#   7350*   7361
$TPFLG  001157          1881#   7282    7361
$TPS    001150          1876#   7348    7361
$TRAP   043452          3208    8118#
$TRAP2  043474          8129#   8140
$TRP  = 000016          8133#   8142#   8143#   8144#   8145#   8146#   8147    8148#   8149    8150#   8151#   8152#   8153#
                        8154#   8155#   8156#
$TRPAD  043506          8123    8140#
$TSTM   001004          1391#
$TSTNM  001102          1854#   3685*   5963*   5996*   7152    7179    7201*   7202    7207    7211    7229    7265
$TTYIN  042410          7765    7766    7778    7796    7810    7814#
$TYPBN= ****** U        8146
$TYPDS  040272          7373#   8145
$TYPE   037736          7282#   7459    8133    8141
$TYPEC  040150          7312    7319    7326    7331#   7713
$TYPEX  040270          7354    7356    7359#
$TYPOC  041010          7515#   8142
$TYPON  041024          7514    7517#   8144
$TYPOS  040764          7510#   8143
$UNIT   001222          1905#   3260*   3323*   3695*   3702    3712    3849    3864    4707    5109    5177    5192    5268
                        5418    5433    5593    5608    6198    6217    6243    6285    6286    6287    6288    6414    6427
                        6435    6443    6451    6480    6956    6971
$UNITM  001010          1393#
$USWR   001234          1912#
$VECT1  001260          1937#
$VECT2  001262          1938#
$XOFF = 000023          7336    7361
$XON  = 000021          7343    7361    7599
$XTSTR  037270          7166#
$$GET4= 000000          5979#
$0FILL  041207          7511*   7515*   7525    7560#
$40CAT= ****** U        7163    7239
.     = 055672          1347#   1351#   1356#   1359#   1366    1367#   1369#   1371#   1372#   1378    1379#   1381#   1383#
                        1851#   1894    2042#   2043#   2044#   2046#   2047#   2048#   2049#   2050#   3201    3216    3217
                        5987    5991    7071    7127    7210    7211    7265    7361    7427#   7480#   7565    7569#   7570
                        7571#   7814#   7815    7822#   7876    7914*   7976#   8958#   9273#   9274#   9419    9644    9648
                        9680    9704    9766    9774    9777#   9785#
.$ASTA= ****** U        7432    7435
.$X   = 001000          1378#   1383
```

CALIB 1485# 3847 5174 5415 5590 6954
CHECK 1439# 3905 4099 4118 4206 4225 4272 4289 4351 4370 4404 4496 4523 4559 4586
4656 4677 4726 4778 4841 4967 4986 5020 5039 5078 5099 5237 5258 5312 5331
5383 5402 5506 5525 5560 5579 5685 5814 5859
COMMEN 1# 1158#
CWD2 1454# 5338
DRCLR 1467# 3862 4705 5107 5190 5266 5431 5606 6969
ENDCOM 1# 1158#
EOPGM 1806# 5950
ERROR 1052# 3396 3455 3463 3477 3486 3511 3520 3539 3543 3554 3558 3590 3599 3617
3644 3648 3652 3656 3758 3796 3805 3809 3813 3845 3852 3858 3861 3867 3870
3876 3892 3896 3908 3909 3910 3911 3914 3921 3926 3931 3934 3946 3988 3993
3997 4001 4031 4049 4053 4063 4073 4086 4090 4102 4103 4104 4105 4108 4121
4122 4123 4124 4141 4155 4159 4169 4180 4193 4197 4209 4210 4211 4212 4215
4228 4229 4230 4231 4236 4245 4251 4258 4261 4275 4276 4277 4278 4292 4293
4294 4295 4303 4322 4331 4339 4342 4354 4355 4356 4357 4373 4374 4375 4376
4382 4389 4393 4407 4408 4409 4410 4426 4442 4445 4448 4457 4468 4471 4483
4487 4499 4500 4501 4502 4505 4509 4512 4526 4527 4528 4529 4537 4540 4547
4550 4562 4563 4564 4565 4572 4575 4589 4590 4591 4592 4597 4612 4616 4643
4650 4659 4660 4661 4662 4667 4680 4681 4682 4683 4687 4697 4702 4710 4713
4720 4729 4730 4731 4732 4741 4746 4752 4756 4760 4767 4781 4782 4783 4784
4793 4798 4803 4814 4817 4829 4844 4845 4846 4847 4851 4857 4861 4865 4868
4872 4876 4880 4929 4941 4954 4958 4970 4971 4972 4973 4976 4989 4990 4991
4992 5000 5008 5011 5023 5024 5025 5026 5042 5043 5044 5045 5062 5072 5081
5082 5083 5084 5089 5102 5103 5104 5105 5112 5115 5124 5180 5186 5189 5195
5198 5224 5231 5240 5241 5242 5243 5248 5261 5262 5263 5264 5271 5274 5286
5299 5303 5315 5316 5317 5318 5321 5334 5335 5336 5337 5343 5346 5353 5363
5371 5374 5386 5387 5388 5389 5405 5406 5407 5408 5421 5427 5430 5436 5439
5478 5487 5497 5509 5510 5511 5512 5515 5528 5529 5530 5531 5540 5548 5551
5563 5564 5565 5566 5582 5583 5584 5585 5596 5602 5605 5611 5614 5648 5651
5670 5672 5676 5688 5689 5690 5691 5726 5744 5748 5762 5766 5770 5784 5794
5799 5803 5817 5818 5819 5820 5823 5830 5839 5844 5848 5862 5863 5864 5865
5868 5886 5900 5904 5920 5924 5928 6394 6399 6403 6417 6430 6438 6446 6453
6479 6530 6926 6938 6942 6946 6959 6965 6968 6974 6977 7037
ESCAPE 1# 1158#
F.EAB 1419# 3897 4091 4110 4198 4217 4263 4281 4343 4362 4395 4488 4514 4551 4577
4669 4769 4833 4959 4978 5012 5031 5091 5250 5304 5323 5375 5394 5498 5517
5552 5571 5677 5805 5850
GETPRI 1# 1158#
GETSWR 1# 1158# 6019
HDCHKS 1596# 4809
HDTBL 1609# 4146 4433 4603 5891
LOOP 1403# 3470 3583 3801 4059 4165 4232 4317 4378 4453 4593 5349 5778 5825
MSG 3354# 3356 3400# 3402 3563# 3565 3672# 3674 3744# 3746 3785# 3787 3817# 3819 4010#
4012 4126# 4128 4413# 4415 4622# 4625 4903# 4905 5207# 5209 5445# 5447 5620# 5622
5699# 5701
MULT 1# 1158#
NEWTST 1# 1158# 3354 3400 3563 3672 3744 3785 3817 4010 4126 4413 4623 4903 5207
5445 5620 5699
OFFDIR 1825# 4690 4734 4786
OFSET 1782# 4646 5068 5227
OWNTAG 1396# 1953
POP 1# 1158# 7414 7474 7475 7865 8061 8103
PUSH 1# 1158# 7373 7435 7437 7458 7839 8032 8083
QKSEEK 1621# 3990 4749 4869 5759 5917
QKSRT 1513# 6935

```
RDATA   1700#  4242
RDHDR   1564#  4810
REPORT     1#  1158#
SCOPE   1053#  3364  3415  3578  3681  3751  3791  3840  4020  4136  4421  4636  4924  5217  5467
        5638   5718  5951  5962
SETPRI     1#  1158#  7743
SETTRA  8133#  8142  8143  8144  8145  8147  8149  8150  8151  8152  8153  8154  8155
SETUP      1#  1158#  3196
SKIP       1#  1158#  3342  3393  3614  3728  3735  3756  3799  3812  3943  3979  4900  5166  5644
        5652   5694
SLASH      1#  1158#
SPACE   1158#
STARS      1#  1158#  1364  1375  1377  1384  1847  1894  1897  3354  3363  3400  3414  3563  3577
        3672   3680  3744  3750  3785  3790  3817  3839  4010  4019  4126  4135  4413  4420  4623
        4635   4903  4923  5207  5216  5445  5466  5620  5637  5699  5717  5943  7149  7213  7267
        7363   7430  7487  7564  7646  7661  7732  7756  7825  7878  7917  7979  7997  8020  8067
        8112   9188  9197
SWRSU      1#  1156#  3218#
TRMTRP  8133#
TYPBIN     1#  1158#
TYPDEC     1#  1158#  5974
TYPNAM     1#  1158#
TYPNUM     1#  1158#
TYPOCS     1#  1158#  3437  3503  3549  3713  5052  5135  6272  7010  7055
TYPOCT     1#  1158#  7674
TYPTXT     1#  1158#
WDATA   1648#  4070  4177  4465  4938  5283  5484
WRCHK   1746#  4328  4534  4997  5360  5537
WRHDR   1534#  4046  4152  4439  4609  5741  5897
$$CMRE  1845#
$$CMTM  1845#  1882  1883  1884  1885  1886  1887
$$ESCA     1#  1158#
$$NEWT     1#  1158#  3354  3400  3563  3672  3744  3785  3817  4010  4126  4413  4623  4903  5207
        5445   5620  5699
$$SET   8133#  8142  8143  8144  8145  8147  8149  8150  8151  8152  8153  8154  8155
$$SETM  3234#
$$SKIP     1#  1158#  3342  3393  3614  3728  3735  3756  3799  3812  3943  3979  4900  5166  5644
        5652   5694
.EQUAT     1#  1007#  1048
.HEADE     1#  1007#  1012
.KT11      1#
.SETUP     1#  1007#  3167
.SWRHI     1#  1007#  1023
.SWRLO  1007#  1035#
.$ACT1     1#  1007#  1362
.$APTB     1#  1895#
.$APTH     1#  1007#  1373
.$APTY     1#  1007#  7428
.$ASTA     1#
.$CATC     1#  1007#  1345
.$CMTA     1#  1007#  1845
.$DB2D     1#  1007#  7915
.$DB2O     1#  1007#  7876
.$DIV      1#
.$EOP      1#  1007#  5941
.$ERRO     1#  1007#  7211
```

```
.$ERRT      1#
.$MULT      1#    1007#    8018
.$POWE      1#
.$RAND      1#
.$RDDE      1#
.$RDOC      1#    1007#    7823
.$READ      1#    1007#    7562
.$R2AZ      1#
.$SAVE      1#    1007#    8065
.$SB2D      1#    1007#    7977
.$SB2O      1#
.$SCOP      1#    1007#    7147
.$SIZE      1#
.$SUPR      1#    1007#    7995
.$TRAP      1#    1007#    8110
.$TYPB      1#
.$TYPD      1#    1007#    7361
.$TYPE      1#    1007#    7265
.$TYPO      1#    1007#    7485
.$40CA      1#
.1170       1#


. ABS.  057742     000

 ERRORS DETECTED:  0

 CZR6IF,CZR6IF.LST/SOL/CRF/NL:TOC=SYSMAC.SML,CZR6IF.P11
 RUN-TIME: 21 25 2 SECONDS
 RUN-TIME RATIO: 124/49=2.5
 CORE USED:  43K  (86 PAGES)
```