RP04/5/6

DISKLESS 2
CZRJHC0

AH-9215C-MC

FICHE 1 OF 2

DEC 1978
digital
MADE IN USA

RP04/5/6

DISKLESS 2

CZRJHC0

AH-9215C-MC

FICHE 2 OF 2

DEC 1978

digital

MADE IN USA

B 1

.REM @

IDENTIFICATION

PRODUCT CODE: AC-9213C-MC

PRODUCT NAME: CZRJHC0 RP04/5/6 DISKLESS CONTROLLER TEST-PART II

DATE CREATED: MAY 1976

MAINTAINER: DIAGNOSTIC ENGINEERING

AUTHOR: PETE BLACKSTONE

DIGITAL
DEC

PDP
DECUS

UNIBUS
DECTAPE

MASSBUS

CONTENTS

1.0 ABSTRACT

THIS DIAGNOSTIC TESTS THE RH11 AND DCL OF AN RP04/5/6 SUBSYSTEM.
IT DOES NOT USE THE DISK SURFACE OR ANY SIGNALS FROM THE MDLI.
IT REQUIRES THAT THE DCL CABLE BE PLUGGED INTO THE MDLI OR
BE APPROPIATELY TERMINATED. IF THE DISK IS POWERED UP, IT
IS REQUIRED TO GET THE DISK TO THE "HEADS UNLOADED" POSITION.
AFTER A SUCCESSFUL RUN (WITH NO ERRORS) OF THIS DIAGNOSTIC
IT CAN BE ASSERTED THAT, "THAT PART OF THE DCL THAT HANDLES
DATA OR DATA ASSOCIATED LOGIC IS WORKING PROPERLY". THIS
IMPLIES THAT, THAT PART OF THE LOGIC WHICH HANDLES MECHANICAL
COMMANDS OR ITS ASSOCIATED LOGIC IS NOT TESTED IN THIS
DIAGNOSTIC. ALL DATA COMMANDS USE THE MAINTENANCE REGISTER
IN THE WRAPAROUND MODE.

THE DIAGNOSTIC DOES NOT DO ANY TESTING OF THE RH70 CONTROLLER
WHEN IT IS USED TO TEST RP04/5/6 DISK DRIVES CONNECTED TO
THAT TYPE OF CONTROLLER. IT IS ASSUMED THAT THE RH70
SPECIFIC CONTROLLER DIAGNOSTICS HAVE BEEN SUCCESSFULLY RUN TO
COMPLETION BEFORE THIS PROGRAM IS RUN.

2.0 REQUIREMENTS

2.1 EQUIPMENT

PDP-11 COMPUTER WITH CONSOLE TELETYPE, AND AN RP04/5/6 DISK SYSTEM.
THE RP04/5/6 DISK SYSTEM WILL CONSIST OF AN RH11/RH70 CONTROLLER,
AND DISK CONTROL LOGIC (DCL). THE CABLE FROM THE DCL CAN BE
CONNECTED TO THE MDLI, BUT IF NOT THAT CABLE MUST BE PROPERLY
TERMINATED.

2.2 STORAGE

THIS PROGRAM REQUIRES 16K WORDS OF MEMORY.

2.3 PRELIMINARY PROGRAMS

THIS PROGRAM ASSUMES THAT MAINDEC-11-DZRJG-
(LATEST REV) HAS BEEN RUN WITHOUT ERRORS

3.0 LOADING PROCEDURE

USE STANDARD PROCEDURE FOR LOADING .ABS TAPES

4.0 STARTING PROCEDURE

SWITCH 12 MUST BE SET WHEN THIS PROGRAM IS TO BE RUN USING AN
RH70 CONTROLLER. IT CAN BE SET AT THE FRONT PANEL, OR IN THE
SOFWARE SWITCH REGISTER IF THE OPERATOR SO DESIRES. SEE
PARAGRAPH 5.1 FOR A DESCRIPTION OF SOFTWARE SWITCH REGISTER
OPERATION.

4.1 CONTROL SWITCH SETTINGS

SEE SECTION 5.1

4.2 STARTING ADDRESS

START AT ADDRESS 200---FOR NORMAL RUN
START AT ADDRESS 204---TO SELECT NON-DEFAULT PARAMETERS
START AT ADDRESS 210---FOR UNIT SELECTION

200 START
ALL SWITCHES MUST BE DOWN FOR WORST CASE RUN. WITH THIS STARTING ADDRESS ALL THE RP04/5/6'S ON THE SYSTEM WILL BE TESTED ONE AT A TIME BEFORE "END PASS" IS PRINTED OUT. TESTING WILL START WITH THE LOWEST UNIT NUMBER DRIVE THAT IS POWERED UP (THAT IS THE LOWEST UNIT NUMBER RHAS REGISTER THAT RESPONDS)THEN GO ON TO THE NEXT HIGHER UNIT NUMBER THAT IS POWERED UP.

204 RESTART
SAME AS 200 START WITH FOLLOWING EXCEPTIONS: PROGRAM WILL QUERY OPERATOR FOR THE CORRECT CSR AND VECTOR ADDRESS OF THE RHXX CONTROLLER. WHEN THIS ACTION HAS BEEN COMPLETED, THE PROGRAM WILL AUTOMATICALLY RESTART FROM ADDRESS 200, WITH THE SAME CONVENTIONS AS DESCRIBED FOR A 200 START.

210 START
ALL SWITCHES MUST BE DOWN FOR WORST CASE RUN. WITH THIS STARTING ADDRESS THE CONSOLE TELETYPE WILL ASK FOR THE UNIT NUMBER TO BE TESTED. THEN ONLY THAT UNIT WILL BE TESTED FOR EACH PASS OF THE PROGRAM.

4.3 PROGRAM AND/OR OPERATOR ACTION

1. LOAD THE PROGRAM INTO MEMORY.
2. SET STARTING ADDRESS ON THE SWITCH REGISTER
3. PRESS "LOAD ADDRESS".
4. SET "OPERATIONAL SWITCH SETTINGS" (SEE SECTION 5.1) WORST CASE IS ALL SWITCHES DOWN.
5. PRESS "START".
6. FOR THE FIRST PASS EACH TEST WILL BE EXECUTED ONCE ON THE DRIVES PRESENT OR DRIVE SELECTED BEFORE "END PASS" IS PRINTED. THE FIRST PASS WILL REQUIRE OPERATOR INTERVENTION IF THE PROGRAM IS NOT RUN UNDER AN "ACT-11" MONITOR. THE SECOND AND SUBSEQUENT PASSES WILL EXECUTE EACH TEST FOUR TIMES ON EACH DRIVES PRESENT OR DRIVE SELECTED BEFORE "END PASS" IS PRINTED. THE SECOND AND SUBSEQUENT PASSED DO NOT NEED ANY OPERATOR INTERVENTION.

5.0 OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

IF THE PROGRAM IS BEING RUN ON A SWITCHLESS PROCESSOR (I.E. AN 11/34) THE PROGRAM WILL DETERMINE THAT THE HARDWARE SWITCH

REGISTER IS NOT PRESENT AND WILL USE AN 'SOFTWARE' SWITCH
REGISTER. 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 RP04/5/6 INTERRUPT. THE
'SOFTWARE' SWITCH VALUES ARE ENTERED AS AN OCTAL NUMBER IN
RESPONSE TIO THE PROMPT FROM THE SWITCH ENTRY ROUTINE:

'SWR = NNNNNN NEW ='

EACH TIME SWITCH SETTINGS 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 REGISTER, 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.

SWITCH DEFINITIONS ARE GIVEN IN SECTION 9 ''OPERATIONAL
SWITCH SETTINGS'' HOWEVER THE DETAIL DESCRIPTIONS ARE GIVEN
HERE.

SWITCH 15 - HALT ON ERROR
WHEN THIS SWITCH IS SET, IF THE PROGRAM FINDS AN ERROR
THEN THE APPROPIATE INFORMATION WILL BE PRINTED OUT
AND THEN THE PROGRAM WILL HALT. AFTER THIS HALT, PRESSING
''CONTINUE'' WILL CONTINUE WITH THE PROGRAM TILL THE NEXT
ERROR IS FOUND WHEN THE SAME THING WILL HAPPEN.

SWITCH 14 - LOOP ON TEST
WHEN THIS SWITCH IS SET THE PROGRAM WILL BEGIN TO LOOP
ON THE CURRENT TEST BEING EXECUTED. FOR EXAMPLE IF THIS
SWITCH IS SET WHEN THE PROGRAM IS IN TEST 10 THEN THE
PROGRAM WILL KEEP EXECUTING ALL OF TEST 10 REPEATEDLY.
ONE WAY TO BE SURE THAT THE PROGRAM IS IN THE EXPECTED
TEST IS TO SET THIS SWITCH DURING AN ERROR PRINTOUT OR
DURING A PROGRAM HALT.

SWITCH 13 - INHIBIT ERROR TYPEOUTS
WHEN THIS SWITCH IS SET FURTHER ERROR PRINTOUTS WILL
CEASE, HOWEVER OPERATOR INSTRUCTIONS SUCH AS ''STOP DRIVE X''
WILL CONTINUE. AT THE END OF PASS ''TOTAL NUMBER OF ERRORS
ON THIS PASS ON DRIVE X'' WILL BE TRUE, THAT IS, ALTHOUGH
PRINTOUTS WERE INHIBITED IF THAT PASS FOUND 6 ERRORS,
IT WILL SAY SO.

SWITCH 12 - RH70 CONTROLLER SELECT
THIS SWITCH MUST BE SET AT THE START OF THE PROGRAM WHEN THE
DISK DRIVES TO BE TESTED ARE CONNECTED TO AN RH70
CONTROLLER. IT MUST NOT BE SET WHEN DISK DRIVES TO BE TESTED

ARE CONNECTED TO AN RH11 CONTROLLER.

SWITCH 11 - INHIBIT ITERATIONS
WHEN THIS SWITCH IS SET THE PROGRAM ON SECOND PASS WILL
NOT REPEAT EACH TEST FOUR TIMES BUT WILL DO EACH TEST
ONCE ONLY.

SWITCH 10 - BELL ON ERROR
WHEN THIS SWITCH IS SET, IF THE PROGRAM FINDS AN ERROR
THE ''BELL'' OR ''ALARM'' WILL BE SOUNDED. THIS SWITCH IS USEFUL
WHEN SWITCH 11 IS SET YET INFORMATION IS NEEDED WHEN ANY ERROR
IS DETECTED. TAKE THE EXAMPLE OF A PROGRAM LOOPING ON A TEST WITH
SWITCH 11 SET TO HELP SCOPING. THEN IF THIS SWITCH IS
SET AND THE BELL OR ALARM SOUNDS IT MEANS THAT THE ERROR
IS PRESENT BUT IF THE BELL OR ALARM STOPS IT MEANS THAT
THE ERROR IS NOT PRESENT.

SWITCH 9 - LOOP ON ERROR
WHEN THIS SWITCH IS SET, IF THE PROGRAM FINDS AN ERROR
THEN GENERALLY THE PROGRAM WILL LOOP BACK TO THE LAST
EXECUTED ''SCOPE'' STATEMENT. IF ON THE SECOND TIME
THROUGH AN ERROR IS FOUND IT WILL AGAIN LOOP BACK TO
THAT ''SCOPE'' STATEMENT. THIS LOOPING WILL CONTINUE AS LONG
AS THE ERROR IS PRESENT AND THIS SWITCH IS SET. HOWEVER
IF THE ERROR IS NOT PRESENT AT ANY TIME THEN IT WILL
CONTINUE NORMALLY WITH THE PROGRAM. EACH TIME THE ERROR
IS ENCOUNTERED PRINTOUT WILL TAKE PLACE UNLESS SWITCH 11
IS ALSO SET. DURING BEGUG, USING A SCOPE, IT IS RECOMMENDED
THAT SWITCH 11 IS ALSO SET.

NOTE: ALSO SEE SECTION 8.3

SWITCH 8 - LOOP ON TEST IN SWR <7:0>
THIS IS A SPECIAL SWITCH. WHEN SET SWITCHES 0 THRU 7
HAVE ONE MEANING AND WHEN RESET SWITCHES 0 THRU 7 HAVE
ANOTHER MEANING. THIS MEANS THAT ANY SETTING OF SWITCH
0 THRU 7 MUST BE DONE WITH SWITCH 8 IN THE APPROPIATE
POSITION. WHEN THIS SWITCH IS SET THEN SWITCHES 0 THRU
7 GIVE THE TEST NUMBER TO BE LOOPED ON. FOR EXAMPLE
WITH SWITCH 8 SET AND SWITCH 3 SET THE PROGRAM WILL LOOP
ON TEST 10. HOWEVER THIS SETTING MUST BE DONE AT THE
BEGINNING OF THE PROGRAM THEN ALL THE TESTS FROM 1 TO 10
WILL BE EXECUTED AND THEN TEST 10 WILL BE REPEATED OVER
AND OVER AGAIN. WHEN THIS SWITCH IS NOT SET THEN SWITCHES
0 THRU 7 HAVE THE MEAING ITS NAME INDICATES.
FOR EXAMPLE SWITCH 7 IS ''STOP FURTHER COMPARES: THAT IS
IF SWITCH 8 IS NOT SET AND SWITCH 7 IS SET THEN WHEN A
DATA ERROR IS DETECTED NO FURTHER COMPARES WILL BE DONE.
FOR EXAMPLE IN A 256 WORD BUFFER IF ALL THE WORDS ARE IN
ERROR THEN AFTER SEEING THE PRINTOUT FOR THE FIRST FEW
WORDS SETTING SWITCH 7 ONLY WILL STOP FURTHER PRINTOUTS
OF THIS ERROR AND GO ON WITH THE TEST RATHER THAN PRINT
ALL THE 256 WORDS. HOWEVER IF THIS WAS DONE WITH SWITCH 11

THEN THE NEXT ERROR THAT THE PROGRAM DETECTS IN A SUB-
SEQUENT TEST WILL ALSO BE LOST. BUT WITH SWITCH 7, ONLY
THIS GROUP OF DATA ERRORS ARE NOT PRINTED OUT. ANOTHER
EXAMPLE OF SWITCH 8 BEING LOW IS WITH SWITCH 6, WHICH
IS 'ECC TEST-COMPARE END RESULT ONLY''. THAT IS IF SWITCH
8 IS NOT SET AND SWITCH 6 IS SET THEN ON ECC TESTS (TEST
120 THRU TEST 134) INSTEAD OF COMPARING CONTENTS OF THE
POSITION REGISTER AND PATTERN REGISTER AFTER EVERY CLOCK,
COMPARES WILL ONLY BE DONE AT THE END OF ALL THE CLOCKS.

NOTE: ALSO SEE SECTION 8.3

SWITCH 7 - STOP FURTHER COMPARES IF SW08 IS LOW.
IF SWITCH 8 IS SET AND THIS SWITCH IS ALSO SET THEN THIS
SWITCH GIVES THE TEST NUMBER TO BE LOOPED ON AS INDICATED
IN THE DESCRIPTION OF SWITCH 8. IF SWITCH 8 IS NOT SET
AND THIS SWITCH IS SET THEN THE PROGRAM WILL DO AS THE
NAME INDICATES. FOR EXAMPLE IN A 256 WORD BUFFER IF ALL
THE WORDS ARE IN ERROR THEN AFTER SEEING THE ERROR PRINTOUTS
FOR THE FIRST FEW WORDS THEN SETTING SWITCH 7 WITH SWITCH 8
NOT SET WILL STOP THE PRINTOUT OF ALL 256 WORDS BUT WILL
NOT STOP THE PRINTOUT OF ANOTHER ERROR IN ANY SUBSEQUENT
TEST. IT IS EXPECTED THAT SWITCH 7 AFTER BEING SET FOR
A WHILE TO STOP PRINTING ALL THE 256 WORDS WILL BE RESET
AGAIN TO ENABLE THE PRINTING OF OTHER DATA ERRORS.

SWITCH 6 - ECC TEST-COMPARE END RESULTS ONLY IF SW08 IS LOW
IF SWITCH 8 IS SET AND THIS SWITCH IS ALSO SET THEN THIS
SWITCH GIVES THE TEST NUMBER TO BE LOOPED ON AS INDICATED
IN THE DESCRIPTION OF SWITCH 8. IF SWITCH 8 IS NOT SET
AND THIS SWITCH IS SET THEN ON ECC TESTS (TEST 120 THRU
TEST 134) INSTEAD OF COMPARING CONTENTS OF THE POSITION AND
PATTERN REGISTERS AFTER EVERY CLOCK, COMPARES WILL BE DONE
ONLY AT THE END OF ALL THE CLOCKS.

5.2 SUB-ROUTINE ABSTRACTS

SEE SECTION 9 ''SUBROUTINES''

6.0 ERRORS

ERROR PRINTOUTS CONTAIN THE ERROR ADDRESS AND OTHER
PERTINENT INFORMATION CONCERNING THE PARTICULAR FAILURE.
THIS INFORMATION MAY BE THE CONTENTS OF RELEVANT RP04
REGISTERS OR GOOD/RECEIVED DATA. IF THE ERROR OCCURRED IN A
SUBROUTINE, THE ADDRESS OF THE SUBROUTINE CALL IS ALSO
GIVEN. REFER TO THE PROGRAM LISTING AT THE STATED
ADDRESS TO DETERMINE THE CAUSE OF THE ERROR.

6.1 'FATAL' ERRORS

IN THE EVENT THAT THE DISK DRIVE BECOMES UNAVAILABLE TO THE

CONTROLLER, POWERS DOWN, OR CERTAIN CRITICAL STATUS BITS
CANNOT BE CLEARED PRIOR TO THE START OF A T4ST SEQUENCE -
THIS INFORMATION WILL BE COMMUNICATED TO THE OPERATOR. IN
ADDITION, THE TTY BELL WILL RING AND THE PROGRAM WILL HALT.
IT IS SUGGESTED THAT IF THIS HAPPENS THE OPERATOR LOAD
ADDRESS 200 (210) AND RESTART THE PROGRAM AS A FIRST ATTEMPT
TO SOLVE THE PROBLEM. IF THE FAILURE CONTINUES TO OCCUR,
THERE ARE TWO OPTIONS FOR THE OPERATOR:

1. LOOK IN THE TEST LISTING FOR THE 'HALT'
INSTRUCTION AND REPLACE IT PLUS THE TWO WORDS
('TYPE ,CPHALT') ABOVE WITH 'NOP'S. WITH TTY ERROR
PRINTOUTS INHIBITED, A SCOPE LOOP CAN BE INITIATED
FOR THE TEST IN QUESTION.

2. GO BACK AND RERUN DZRPS AS IT IS QUITE POSSIBLE
THAT A HARD FAILURE HAS OCCURRED IN ONE OF THE
HARDWARE REGISTERS.

IT IS ALSO POSSIBLE TO CONTINUE FROM THE HALT POINT, BUT THIS
IS NOT RECOMMENDED AS ALL FOLLOWING TESTS WILL EXHIBIT THE
SAME SYMPTOMS AND GIVE MISLEADING ERROR PRINTOUTS.

7.0 RESTRICTIONS

IF THERE IS A DRIVE CONNECTED THEN THE OPERATOR MUST HAVE
THE DRIVE PORT SWITCH LOCKED EITHER ON PORT A OR PORT B
BUT NEVER LEAVE IT IN THE PROGRAMMABLE STATE. IF THERE
IS NO DRIVE CONNECTED THEN THE CABLE NORMALLY GOING FROM
THE DCL TO THE MDLI MUST BE PROPERLY TERMINATED.

SWITCH 12 MUST BE SET WHEN RUNNING ON AN RH70 CONTROLLER AND
IT MUST NOT BE SET WHEN RUNNING ON AN RH11 CONTROLLER.
BECAUSE OF THIS FACT, THE PROGRAM CANNOT BE RUN IN CHAIN MODE
WHEN USING THE SOFTWARE SWITCH REGISTER AS THE ROUTINE WHICH
ASKS FOR THE SWTICH REGISTER SETTINGS IS NOT OPERABLE WHEN IN
CHAIN MODE.

8.0 MISCELLANEOUS

8.1 EXECUTION TIME

THE FIRST PASS OF THE PROGRAM WILL TAKE 1.75 MINUTES PER
DRIVE. SUBSEQUENT PASSES WILL TAKE 7 MINUTE.

8.2 STACK POINTER

THE STACK IS INITIALLY SET TO 1000

8.3 OPERATOR SELECTABLE SCOPE LOOPS

HERE IS A DETAILED EXPLAINATION OF HOW THE LOOP ON ERROR WORKS.
FOR INSTRUCTIONS REGARDING USAGE OF THESE LOOPS, HIT CONTROL
C ANY TIME WHILE THE PROGRAM IS RUNNING. ON HITTING AN ERROR
IF THE LOOP ON ERROR SWITCH IS SET, THE PROGRAM GOES BACK
- USUALLY BACK TO THE BEGINNING OF THE TEST.

WHEN THIS OPERATOR SELECTABLE SCOPE LOOP IS USED THEN THE POINT
THE PROGRAM GOES BACK TO CAN BE CHANGED.
THE RESTRICTIONS TO THE POINT WHERE THE PROGRAM CAN GO ARE: -
1. IT MUST BE WITHIN THE TEST UNDER CONSIDERATION
2. LOOP ON ERROR SWITCH MUST BE SET
3. THE ERROR MUST OCCUR WITHIN THE TEST UNDER CONSIDERATION
IF THE ERROR DOES NOT OCCUR WITHIN THE TEST UNDER CONSIDERATION
THE PROGRAM WILL REVERT TO NORMAL OPERATION. HOWEVER, IF LOOP ON
TEST SWITCH IS SET AND THIS OPERATOR SELECTABLE SCOPE LOOP IS USED
THEN THE PROGRAM WILL LOOP BACK TO THE SELECTED POINT WHEN IT
COMES TO THE END OF THE TEST UNDER CONSIDERATION.

AFTER LOOPING FOR SOME TIME IF THE LOOP SWITCH IS PUT DOWN THEN
NORMAL OPERATION WILL CONTINUE.

8.4 PROGRAM REVISION HISTORY

9.0 PROGRAM DESCRIPTION

THE FOLLOWING SECTIONS DESCRIBE EACH TEST AND SUBROUTINES
IN DETAIL AND CAN ALSO BE USED AS AN INDEX TO THE LISTING.
THE LEFT MOST COLUMN IS THE LINE NUMBER WITHIN THE LISTING
WHERE THAT ITEM WILL BE FOUND.
@

```
;DRIVE MUST BE LOCKED ON PORT A OR PORT B

                        ;INTERNAL PROGRAM MACROS BEGIN HERE
                        ;**********************************

                ;*
                ;*NOTE: ALL MACRO CALLS BEGINNING WITH ".$" ARE SUPPLIED FROM AN
                ;*      EXTERNAL SYSMAC.SML PACKAGE WHICH MUST BE MADE AVAILABLE
                ;*      TO THE SOURCE PROGRAM AT ASSEMBLY TIME.
                ;*
```

494

```
                                DISPREG: .WORD  0               ;;SOFTWARE DISPLAY REGISTER
                                SWREG:   .WORD  0               ;;SOFTWARE SWITCH REGISTER

                                         $ENDAD                 ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
                                         .WORD  20000           ;;2)SET LOC.52 TO 20000

                                .SBTTL  STARTING ADDRESSES

                                         .=200
000200  000137  017356          RA:      JMP     @#BEGIN                ;NORMAL START

000204  000137  051414          ADDMOD:  JMP     @#BASECH               ;MODIFY DEVICE PARAMETERS

000210  000137  017346                   JMP     @#BEGIN2               ;SELECT DRIVE START


                                ;*STARTING ADDRESS 200 FOR NORMAL STARTS
                                ;*THIS WILL TEST ALL RP04'S ON THE SYSTEM A SINGLE DRIVE AT A TIME

                                ;*STARTING ADDRESS 204 WILL LOAD NON-DEFAULT PARAMETERS
                                ;*AND AUTO RESTART AT 200 AFTER LOADING PARAMETERS.

                                ;*STARTING ADDRESS 210 WILL TEST ONLY ONE SPECIFIED DRIVE
```

```
520
521          001110                          .=1110                          ;
```

```
  522  001100  000000                $PASS:  .WORD   0               ;;CONTAINS PASS COUNT
  523  001102     000                $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
  524  001103     000                $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
  525  001104  000000                $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
  526  001106  000000                $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
  527  001110  000000                $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
  528  001112  000000                $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
  529  001114     000                $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
  530  001115     001                $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
  531  001116  000000                $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
  532  001120  000000                $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
  533  001122  000000                $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
  534  001124  000000                $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
  535  001126  000000                $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
  536  001130  000000                        .WORD   0               ;;RESERVED--NOT TO BE USED
  537  001132  000000                        .WORD   0
  538  001134     000                $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
  539  001135     000                $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
  540  001136  000000                        .WORD   0
  541  001140  177570                SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
  542  001142  177570                DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
  543  001144  177560                $TKS:   177560                  ;;TTY KBD STATUS
  544  001146  177562                $TKB:   177562                  ;;TTY KBD BUFFER
  545  001150  177564                $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
  546  001152  177566                $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
  547  001154     000                $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
  548  001155     002                $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
  549  001156     012                $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A 'LINE FEED''
  550  001157     000                $TPFLG: .BYTE   0               ;;'TERMINAL AVAILABLE'' FLAG (BIT<07>=0=YES)
  551  001160  000000                $REGAD: .WORD   0               ;;CONTAINS THE ADDRESS FROM
  552  001162  000000                $REG0:  .WORD   0               ;;CONTAINS (($REGAD)+0)
  553  001164  000000                $REG1:  .WORD   0               ;;CONTAINS (($REGAD)+2)
  554  001166  000000                $REG2:  .WORD   0               ;;CONTAINS (($REGAD)+4)
  555  001170  000000                $REG3:  .WORD   0               ;;CONTAINS (($REGAD)+6)
  556  001172  000000                $REG4:  .WORD   0               ;;CONTAINS (($REGAD)+10)
  557  001174  000000                $REG5:  .WORD   0               ;;CONTAINS (($REGAD)+12)
  558  001176  000000                $TMP0:  .WORD   0               ;;USER DEFINED
  559  001200  000000                $TMP1:  .WORD   0               ;;USER DEFINED
  560  001202  000000                $TMP2:  .WORD   0               ;;USER DEFINED
  561  001204  000000                $TMP3:  .WORD   0               ;;USER DEFINED
  562  001206  000000                $TMP4:  .WORD   0               ;;USER DEFINED
  563  001210  000000                $TMP5:  .WORD   0               ;;USER DEFINED
  564  001212  000000                $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
  565  001214  000000                $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
  566  001216  177607  000377        $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
  567  001222     077                $QUES:  .ASCII  /?/             ;;QUESTION MARK
  568  001223     015                $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
  569  001224  000012                $LF:    .ASCIZ  <12>            ;;LINE FEED
```

```
570
571
572
573
574
575                                     ;ITEM1
576  001226  002136                             EM1             ;WRONG DATA IN READING OR WRITING HARDWARE REGISTER
577  001230  005451                             DH1             ;PC
578                                                             ;REG. ADDR.
579                                                             ;GOOD DATA
580                                                             ;RECEIVED DATA
581  001232  011764                             DT1             ;$ERRPC,$TSTNM,REGADR,$GDDAT,$BDDAT
582  001234  012516                             DF1             ;0,0,0,0,0
583
584
585
586
587
588                                     ;ITEM2
589  001236  002221                             EM2             ;ERROR ON  DATA COMMAND
590
591  001240  010463                             DH33            ;PC
592                                                             ;PC OF JSR
593                                                             ;TEST NO
594                                                             ;WORD NO.
595                                                             ;GOOD DATA
596                                                             ;CONTENTS OF RHCS1
597                                                             ;CONTENTS OF RHDS1
598                                                             ;CONTENTS OF RHER1
599  001242  012350                             DT33            ;$ERRPC,PCJSR,$TSTNM,ERWORD,$GDDAT,CS1,DS1,ER1
600  001244  012666                             DF33            ;0,0,0,1,0,0,0,0
601
602
603                                     ;ITEM3
604  001246  002221                             EM2             ;ERROR ON  DATA COMMAND
605
606  001250  010246                             DH32            ;PC
607                                                             ;PC OF JSR
608                                                             ;TEST NO
609                                                             ;WORD NO.
610                                                             ;GOOD DATA
611                                                             ;BAD DATA
612                                                             ;CONTENTS OF RHCS1
613                                                             ;CONTENTS OF RHDS1
614                                                             ;CONTENTS OF RHER1
615
616  001252  012324                             DT32            ;$ERRPC,PCJSR,$TSTNM,ERWORD,$GDDAT,$BDDAT,CS1,DS1,ER1
617  001254  012655                             DF32            ;0,0,0,1,0,0,0,0,0,
618
619
620                                     ;ITEM4
621  001256  002221                             EM2             ;ERROR ON  DATA COMMAND
622
623  001260  010050                             DH31            ;PC
624                                                             ;TEST NO
625                                                             ;WORD NO.
```

```
                                                        ;GOOD DATA
                                                        ;BAD DATA
                                                        ;CONTENTS OF RHCS1
                                                        ;CONTENTS OF RHDS1
                                                        ;CONTENTS OF RHER1

001262  012302                      DT31                ;$ERRPC,$TSTNM,ERWORD,$GDDAT,$BDDAT,CS1,DS1,ER1
001264  012645                      DF31                ;0,0,1,0,0,0,0,0,


                            ;ITEM5
001266  000000                      0                   ;
001270  000000                      0                   ;
001272  012302                      DT31                ;$ERRPC,$TSTNM,ERWORD,$GDDAT,$BDDAT,CS1,DS1,ER1
001274  012645                      DF31                ;0,0,1,0,0,0,0,0,


                            ;ITEM6
001276  002250                      EM6                 ;ERROR ON WRITE HEADER AND DATA

001300  010246                      DH32                ;PC
                                                        ;PC OF JSR
                                                        ;TEST NO
                                                        ;WORD NO.
                                                        ;GOOD DATA
                                                        ;BAD DATA
                                                        ;CONTENTS OF RHCS1
                                                        ;CONTENTS OF RHDS1
                                                        ;CONTENTS OF RHER1

001302  012324                      DT32                ;$ERRPC,PCJSR,$TSTNM,ERWORD,$GDDAT,$BDDAT,CS1,DS1,ER1
001304  012655                      DF32                ;0,0,0,1,0,0,0,0,0,


                            ;ITEM7
001306  002250                      EM6                 ;ERROR ON WRITE HEADER AND DATA
001310  005567                      DH2                 ;PC
                                                        ;TEST NO
                                                        ;WORD NO.
                                                        ;GOOD DATA
                                                        ;BAD DATA
001312  012012                      DT3                 ;$ERRPC,$TSTNM,ERWORD,$GDDAT,$BDDAT
001314  012527                      DF3                 ;0,0,1,0,0,


                            ;ITEM10
001316  000000                      0                   ;
001320  000000                      0                   ;
001322  012012                      DT3                 ;$ERRPC,$TSTNM,ERWORD,$GDDAT,$BDDAT
001324  012527                      DF3                 ;0,0,1,0,0,


                            ;ITEM11
001326  002307                      EM11                ;CONTROLLER OR DRIVE STATUS
```

```
001330  005701                          DH11            ;PC
                                                        ;TEST NO
                                                        ;FAILING REG. ADDR
                                                        ;CONTENTS OF RHCS1
                                                        ;CONTENTS OF RHCS2
                                                        ;CONTENTS OF RHDS1
                                                        ;CONTENTS OF RHER1
001332  012026                          DT11            ;$ERRPC,$TSTNM,$BDADR,CS1,CS2,DS1,ER1
001334  012534                          DF11            ;0,0,0,0,0,0

                                ;ITEM12
001336  002307                          EM11            ;WRONG DATA FROM SILO

001340  005451                          DH1             ;PC
                                                        ;REG.ADDR
                                                        ;GOOD DATA
                                                        ;RECEIVED DATA
001342  011764                          DT1             ;$ERRPC,REGADR,$GDDAT,$BDDAT
001344  012516                          DF1             ;0,0,0,0

                                ;ITEM13
001346  000000                          0
001350  000000                          0
001352  011764                          DT1             ;$ERRPC,TSTNM,REGADR,$GDDAT,$BDDAT
001354  012516                          DF1             ;0,0,0,0,0

                                ;ITEM14
001356  002342                          EM14            ;REGISTER FAILED
001360  006056                          DH14            ;PC
                                                        ;FAILING REG. ADDR
                                                        ;CONTENTS OF FAILING REG.
                                                        ;CONTENTS OF RHCS1
                                                        ;CONTENTS OF RHCS2
                                                        ;CONTENTS OF RHDS1
                                                        ;CONTENTS OF RHER1
001362  012046                          DT14            ;$ERRPC,$BDADR,$BDDAT,CS1,CS2,DS1,ER1
001364  012543                          DF14            ;0,0,0,0,0,0,0

                                ;ITEM15
001366  002362                          EM15            ;SPECIFIED REG. NON EXISTANT SO ABORT
                                                        ;PROGRAM
001370  006255                          DH15            ;PC
                                                        ;ADDR. OF REG
001372  012070                          DT15            ;$ERRPC,TEMP1
001374  012553                          DF15            ;0,0

                                ;ITEM16
001376  002432                          EM16            ;WAIT LOOP FAILED
001400  006305                          DH16            ;PC
                                                        ;WAT PC
                                                        ;BIT WANTED
```

```
                                                                ;REG. ADR.
                                                                ;REG. CONT.
001402  012100                          DT16                    ;$ERRPC,$TMP3,$TMP1,$TMP0,$BDDAT
001404  012556                          DF16                    ;0,0,0,0

                                ;ITEM17
001406  002453                          EM17                    ;WRITE CHECK FAILING
001410  006443                          DH17                    ;PC
                                                                ;TEST NO
                                                                ;CONTENTS OF RHBA
                                                                ;CONTENTS OF RHDB
                                                                ;CONTENTS OF RHWC
                                                                ;CONTENTS OF RHCS1
                                                                ;CONTENTS OF RHCS2
001412  012116                          DT17                    ;$ERRPC,$TSTNM,$BA,DB,WC,CS1,CS2
001414  012563                          DF17                    ;0,0,0,0,0,0,0

                                ;ITEM20
001416  002477                          EM20                    ;REGISTER FAILING
001420  006620                          DH20                    ;PC
                                                                ;TST NO
                                                                ;CONTENTS OF RHER1
                                                                ;CONTENTS OF RHER2
                                                                ;CONTENTS OF RHER3
                                                                ;CONTENTS OF RHAS
                                                                ;CONTENTS OF RHDS1
001422  012136                          DT20                    ;$ERRPC,TSTNM ER1,ER2,ER3,AS,DS1
001424  012572                          DF20                    ;0,0,0,0,0,0,0

                                ;ITEM21
001426  002520                          EM21                    ;INTERRUPT FAILING
001430  006774                          DH21                    ;PC
                                                                ;TEST NO
                                                                ;CONTENTS OF RHCS1
                                                                ;CONTENTS OF RHAS
                                                                ;CONTENTS OF RHDS1
001432  012156                          DT21                    ;$ERRPC,TSTNM,CS1,AS,DS1
001434  012601                          DF21                    ;0,0,0,0,0

                                ;ITEM22
001436  002542                          EM22                    ;MISSMATCH IN DRIVE PRESENT
                                                                ;LOOKING AT RHAS AND RHCS2-NED(BIT#12)
                                                                ;DRIVE PRESENT DO NOT AGREE
                                                                ;NOTE: ON DUAL PORT SYSTEM
                                                                ;DRIVE ON OTHER PORT WILL NOT GIVE NED
                                                                ;HENCE THERE WILL BE A MISSMATCH
                                                                ;177777-MEANS NOT PRESENT
001440  007110                          DH22                    ;PC
                                                                ;TEST NO
                                                                ;RHAS UNIT
                                                                ;RHCS2 UNIT
                                                                ;
```

```
001442  012172                  DT22            ;$ERRPC,TSTNM$,$GDDAT,$BDDAT
001444  012606                  DF22            ;0,0,0,0

                        ;ITEM23
001446  000000                  0               ;MISSMATCH IN DRIVE PRESENT
                                                ;LOOKING AT RHAS AND RHCS2-NED(BIT#12)
                                                ;DRIVE PRESENT DO NOT AGREE
                                                ;177777-MEANS NOT PRESENT
001450  000000                  0               ;PC
                                                ;TEST NO
                                                ;RHAS UNIT
                                                ;RHCS2 UNIT
                                                ;
001452  012172                  DT22            ;$ERRPC,TSTNM$,$GDDAT,$BDDAT
001454  012606                  DF22            ;0,0,0,0

                        ;ITEM 24
001456  003143                  EM24            ;LOOK AHEAD REGISTER AT THE
                                                ;BEGINNING OF A SECTOR IS IN
                                                ;ERROR
001460  007204                  DH24            ;PC
                                                ;RHDST
                                                ;BAD RHLA
                                                ;GOOD RHLA
                                                ;SECTOR NO
                                                ;SECTOR CLOCK
001462  012204                  DT24            ;$ERRPC,DST,$BDDAT,$TMP1,$TMP2,$TMP3
001464  012612                  DF24            ;0,0,0,0,0

                        ;ITEM 25
001466  003236                  EM25            ;LOOK AHEAD REGISTER IS
                                                ;IN ERROR
001470  007204                  DH24            ;PC
                                                ;RHDST
                                                ;BAD RHLA
                                                ;GOOD RHLA
                                                ;SECTOR NO
                                                ;SECTOR CLOCK
001472  012204                  DT24            ;$ERRPC,DST,$BDDAT,$TMP1,$TMP2,$TMP3
001474  012612                  DF24            ;0,0,0,0,0
                        ;ITEM26
001476  002307                  EM11            ;CONTROLLER OR DRIVE STATUS

001500  007362                  DH26            ;PC
                                                ;PC OF JSR
                                                ;FAILING REGISTER ADDRESS
                                                ;CONTENTS OF RHCS1
                                                ;CONTENTS OF RHCS2
                                                ;CONTENTS OF RHDS1
                                                ;CONTENTS OF RHER1

001502  012224                  DT26            ;$ERRPC,PCJSR,$BDADR,CS1,CS2,DS1,ER1
```

```
001504  012621                          DF26            ;0,0,0,0,0,0,
                                
                                
                                
                                
                                ;ITEM27
001506  002136                          EM1             ;ERROR IN READING OR WRITING HARDWARE REGISTER
                                
001510  007560                          DH27            ;PC
                                                        ;PC OF JSR
                                                        ;TEST NUMBER
                                                        ;FAILING REGISTER
                                                        ;GOOD DATA
                                                        ;RECEIVED DATA
                                
001512  012246                          DT27            ;$ERRPC,PCJSR,TSTNM,REGADR,$GDDAT,$BDDAT
001514  012631                          DF27            ;0,0,0,0,0,0
                                
                                
                                
                                ;ITEM30
001516  003276                          EM30            ;CURRENT CYLINDER DOES NOT REFLECT DESIRED CYLINDER REG.
001520  007716                          DH30            ;PC
                                                        ;PC OF JSR
                                                        ;REGISTER ADDRESS
                                                        ;GOOD DATA
                                                        ;BAD DATA
                                
001522  012264                          DT30            ;$ERRPC,PCJSR,REGADR,$GDDAT,$BDDAT
001524  012637                          DF30            ;0,0,0,0,0
                                
                                
                                
                                ;ITEM31
001526  003420                          EM31            ;ECC GENERATED IS INCORRECT
                                                        ;EVERY WORD IN THIS SECTOR IS GIVEN IN 'DATA USED''
                                
001530  010662                          DH34            ;PC
                                                        ;TEST NUMBER
                                                        ;GOOD ECC1
                                                        ;GOOD EC2C
                                                        ;WRITTEN ECC1
                                                        ;WRITTEN ECC2
                                                        ;DATA USED
                                
001532  012372                          DT34            ;$ERRPC,TSTNM,GECC1,GECC2,WECC1,WECC2,DISK
                                
001534  012676                          DF34            ;0,0,0,0,0,0,0
                                
                                
                                ;ITEM32
001536  003543                          EM32            ;ON READ COMMAND AFTER DATA AND ECC HAVE BEEN READ
                                                        ;ECC REGISTER OR RHER1 IS IN ERROR
                                                        ;ONLY LOWER 11 BITS OF PATTERN REGISTER
                                                        ;CAN BE READ
                                                        ;THIS SHUOLD MATCH LOWER 11 BITS OF ECC1
```

```
001540  011035                          DH35            ;PC
                                                        ;TEST NUMBER
                                                        ;GOOD ECC1
                                                        ;GOOD ECC2
                                                        ;PATTERN REGISTER
                                                        ;RHER1

001542  012412                          DT35            ;$ERRPC,TSTNM,GECC1,GECC2,EC2,ER1

001544  012705                          DF35            ;0,0,0,0,0,0



                        ;ITEM33
001546  004032                          EM33            ;HIGH COUNT BIT NOT HIGH AFTER 38859 CLOCKS
001550  011232                          DH36            ;PC
                                                        ;PC OF JSR
                                                        ;TEST NUMBER
                                                        ;RHMR
                                                        ;POSITION REG.
                                                        ;PATTERN REGISTER

001552  012434                          DT36            ;$ERRPC,PCJSR,TSTNM,MR,EC1,EC2

001554  012715                          DF36            ;0,0,0,0,0,0

                        ;ITEM34
001556  004104                          EM34            ;ZERO DETECT BIT NOT HIGH WHEN THE
                                                        ;32 BIT ECC REGISTER HAS ITS 21 BITS
                                                        ;OF ZEROS
                                                        ;ERROR PRINTOUT WILL CONTINUE TILL
                                                        ;ZERO DETECT BIT IS HIGH
001560  011232                          DH36            ;PC
                                                        ;PC OF JSR
                                                        ;TEST NUMBER
                                                        ;RHMR
                                                        ;POSITION REG.
                                                        ;PATTERN REGISTER

001562  012434                          DT36            ;$ERRPC,PCJSR,TSTNM,MR,EC1,EC2

001564  012715                          DF36            ;0,0,0,0,0,0



                        ;ITEM35
001566  004177                          EM35            ;POSITION REGISTER OR 11 BITS OF
                                                        ;PATTERN REGISTER INCORRECT
                                                        ;LOWER 11 BITS OF PATTERN REGISTER
                                                        ;SHOULD MATCH LOWER 11 BITS OF GOOD ECC1
                                                        ;DATA ENVELOPE AND N-CODE ZEROS ARE IN DECIMAL

001570  011370                          DH37            ;PC
                                                        ;TEST NUMBER
                                                        ;ECC POSITION
                                                        ;GOOD POSITION
```

```
                                                        ;GOOD ECC1
                                                        ;GOOD ECC2
                                                        ;ECC PATTERN
                                                        ;DATA ENVELOPE
                                                        ;N-CODE ZEROS

001572  012452                      DT37                ;$ERRPC,TSTNM,EC1,POSITI,GECC1,GECC2,EC2,DATENV,ZCODE

001574  012723                      DF37                ;0,0,0,0,0,0,0,0,0



                        ;ITEM36
001576  004476                      EM36                ;ON A READ COMMAND WITH NON CORRECTABLE
                                                        ;ERROR INSERTED DCK AND ECH SHOULD BE SET
001600  011035                      DH35                ;PC
                                                        ;TEST NUMBER
                                                        ;GOOD ECC1
                                                        ;GOOD ECC2
                                                        ;PATTERN REGISTER
                                                        ;POSITION REGISTER
                                                        ;RHER1

001602  012412                      DT35                ;$ERRPC,TSTNM,GECC1,GECC2,EC2,EC1,ER1

001604  012705                      DF35                ;0,0,0,0,0,0,0



                        ;ITEM37
001606  004664                      EM37                ;ERROR ON  DATA COMMAND
                                                        ;WITH A16 A17 USED

001610  010050                      DH31                ;PC
                                                        ;TEST NO
                                                        ;WORD NO.
                                                        ;GOOD DATA
                                                        ;BAD DATA
                                                        ;CONTENTS OF RHCS1
                                                        ;CONTENTS OF RHDS1
                                                        ;CONTENTS OF RHER1

001612  012302                      DT31                ;$ERRPC,$TSTNM,ERWORD,$GDDAT,$BDDAT,CS1,DS1,ER1
001614  012645                      DF31                ;0,0,1,0,0,0,0,0,



                        ;ITEM40
001616  000000                      0                   ;
001620  000000                      0                   ;
001622  012302                      DT31                ;$ERRPC,$TSTNM,ERWORD,$GDDAT,$BDDAT,CS1,DS1,ER1
001624  012645                      DF31                ;0,0,1,0,0,0,0,0,
```

```
1018
1019
1020
1021                                    ;ITEM41
1022  001626  004752                          EM40            ;THERE WAS A READ/WRITE HEADER & DATA
1023                                                          ;ERROR DURING 'DTE' TEST SETUP - THE
1024                                                          ;TEST IS ABORTED AT THAT POINT
1025  001630  011607                          DH40            ;PC
1026                                                          ;TEST NO
1027                                                          ;FAILING REGISTER ADDRESS
1028                                                          ;CONTENTS OF RHCS1
1029                                                          ;CONTENTS OF RHCS2
1030                                                          ;CONTENTS OF RHDS1
1031                                                          ;CONTENTS OF RHER1
1032  001632  012476                          DT40            ;$ERRPC,$TSTNM,$BDADR,CS1,CS2,DS1,ER1
1033  001634  012734                          DF40            ;0,0,0,0,0,0
```

```
1034
1035                                    ;ITEM 42
1036  001636  012744                            EM42
1037  001640  014410                            DH42
1038  001642  014672                            DT42    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS.
1039  001644  014730                            DF42
1040
1041                                    ;ITEM 43
1042
1043  001646  013027                            EM43
1044  001650  014410                            DH42
1045  001652  014672                            DT42    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS.
1046  001654  014730                            DF42
1047
1048                                    ;ITEM 44
1049
1050  001656  013073                            EM44
1051  001660  014471                            DH44
1052  001662  014702                            DT44    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS CORRECT, ACTUAL.
1053  001664  014733                            DF44
1054
1055                                    ;ITEM 45
1056  001666  013130                            EM45
1057  001670  014471                            DH44
1058  001672  014702                            DT44    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS CORRECT, ACTUAL.
1059  001674  014733                            DF44
1060
1061                                    ;ITEM 46
1062
1063  001676  013157                            EM46
1064  001700  014471                            DH44
1065  001702  014702                            DT44    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS CORRECT, ACTUAL.
1066  001704  014733                            DF44
1067
1068                                    ;ITEM 47
1069
1070  001706  013206                            EM47
1071  001710  014410                            DH42
1072  001712  014672                            DT42    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS.
1073  001714  014730                            DF42
1074
1075                                    ;ITEM 50
1076
1077  001716  013243                            EM50
1078  001720  014471                            DH44
1079  001722  014702                            DT44    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS CORRECT, ACTUAL.
1080  001724  014733                            DF44
1081
1082                                    ;ITEM 51
1083
1084  001726  013301                            EM51
1085  001730  014471                            DH44
1086  001732  014702                            DT44    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS CORRECT, ACTUAL.
1087  001734  014733                            DF44
1088
1089                                    ;ITEM 52
```

```
1090
1091  001736  013322                    EM52
1092  001740  014471                    DH44
1093  001742  014702                    DT44
1094  001744  014733                    DF44
1095
1096                              ;ITEM 53
1097
1098  001746  013362                    EM53
1099  001750  014471                    DH44
1100  001752  014702                    DT44    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS CORRECT, ACTUAL.
1101  001754  014733                    DF44
1102
1103                              ;ITEM 54
1104
1105  001756  013422                    EM54
1106  001760  014603                    DH54
1107  001762  014716                    DT54    ;ERROR PC, TEST NUMBER.
1108  001764  014740                    DF54
1109
1110                              ;ITEM 55
1111
1112  001766  013461                    EM55
1113  001770  014471                    DH44
1114  001772  014702                    DT44    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS CORRECT, ACTUAL.
1115  001774  014733                    DF44
1116
1117                              ;ITEM 56
1118
1119  001776  013474                    EM56
1120  002000  014471                    DH44
1121  002002  014702                    DT44    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS CORRECT, ACTUAL.
1122  002004  014733                    DF44
1123
1124                              ;ITEM 57
1125
1126  002006  013511                    EM57
1127  002010  014471                    DH44
1128  002012  014702                    DT44    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS CORRECT, ACTUAL.
1129  002014  014733                    DF44
1130
1131                              ;ITEM 60
1132
1133  002016  013540                    EM60
1134  002020  014471                    DH44
1135  002022  014702                    DT44    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS CORRECT, ACTUAL.
1136  002024  014733                    DF44
1137
1138                              ;ITEM 61
1139
1140  002026  013627                    EM61
1141  002030  014471                    DH44
1142  002032  014702                    DT44    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS CORRECT, ACTUAL.
1143  002034  014733                    DF44
1144
1145                              ;ITEM 62
```

```
1146
1147  002036  013677                    EM62
1148  002040  014471                    DH44
1149  002042  014702                    DT44    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS CORRECT, ACTUAL.
1150  002044  014733                    DF44
1151
1152                                ;ITEM 63
1153
1154  002046  013754                    EM63
1155  002050  014471                    DH44
1156  002052  014702                    DT44    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS CORRECT, ACTUAL.
1157  002054  014733                    DF44
1158
1159                                ;ITEM 64
1160
1161  002056  014032                    EM64
1162  002060  014471                    DH44
1163  002062  014702                    DT44    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS CORRECT, ACTUAL.
1164  002064  014733                    DF44
1165
1166                                ;ITEM 65
1167
1168  002066  014066                    EM65
1169  002070  014471                    DH44
1170  002072  014702                    DT44    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS CORRECT, ACTUAL.
1171  002074  014733                    DF44
1172
1173                                ;ITEM 66
1174
1175  002076  014150                    EM66
1176  002100  014471                    DH44
1177  002102  014702                    DT44    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS CORRECT, ACTUAL.
1178  002104  014733                    DF44
1179
1180                                ;ITEM 67
1181
1182  002106  014222                    EM67
1183  002110  014603                    DH54
1184  002112  014716                    DT54    ;ERROR PC, TEST NUMBER.
1185  002114  014740                    DF54
1186
1187                                ;ITEM 70
1188
1189  002116  014307                    EM70
1190  002120  014603                    DH54
1191  002122  014716                    DT54    ;ERROR PC, TEST NUMBER.
1192  002124  014740                    DF54
1193
1194                                ;ITEM 71
1195
1196  002126  014361                    EM71
1197  002130  014471                    DH44
1198  002132  014702                    DT44    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS CORRECT, ACTUAL.
1199  002134  014733                    DF44
```

```
1200
1201                                    ;********************* ************************************************
1202                                    ;
1203                                    ;ERROR AND MESSAGE TABLE CONDIMENTS
1204                                    ;
1205                                    ;*******************************************************************************
1206
1207
1208
1209
1210  002136  051127  047117  020107  EM1:    .ASCIZ  /WRONG DATA IN READING OR WRITING HARDWARE REGISTER/
1211  002144  040504  040524  044440
1212  002152  020116  042522  042101
1213  002160  047111  020107  051117
1214  002166  053440  044522  044524
1215  002174  043516  044040  051101
1216  002202  053504  051101  020105
1217  002210  042522  044507  052123
1218  002216  051105     000
1219  002221     105  051122  051117  EM2:    .ASCIZ  /ERROR ON  DATA COMMAND/
1220  002226  047440  020116  042040
1221  002234  052101  020101  047503
1222  002242  046515  047101  000104
1223  002250  051105  047522  020122  EM6:    .ASCIZ  /ERROR ON WRITE HEADER AND DATA/
1224  002256  047117  053440  044522
1225  002264  042524  044040  040505
1226  002272  042504  020122  047101
1227  002300  020104  040504  040524
1228  002306     000
1229  002307     103  047117  051124  EM11:   .ASCIZ  /CONTROLLER OR DRIVE STATUS/
1230  002314  046117  042514  020122
1231  002322  051117  042040  044522
1232  002330  042526  051440  040524
1233  002336  052524  000123
1234  002342  042522  044507  052123  EM14:   .ASCIZ  /REGISTER FAILED/
1235  002350  051105  043040  044501
1236  002356  042514  000104
1237  002362  047516  020116  054105  EM15:   .ASCIZ  /NON EXISTENT REGISTER, PROGRAM ABORTED./
1238  002370  051511  042524  052116
1239  002376  051040  043505  051511
1240  002404  042524  026122  050040
1241  002412  047522  051107  046501
1242  002420  040440  047502  052122
1243  002426  042105  000056
1244  002432  040527  052111  046040  EM16:   .ASCIZ  /WAIT LOOP FAILED/
1245  002440  047517  020120  040506
1246  002446  046111  042105     000
1247  002453     127  044522  042524  EM17:   .ASCIZ  /WRITE CHECK FAILING/
1248  002460  041440  042510  045503
1249  002466  043040  044501  044514
1250  002474  043516     000
1251  002477     122  043505  051511  EM20:   .ASCIZ  /REGISTER FAILING/
1252  002504  042524  020122  040506
1253  002512  046111  047111  000107
1254  002520  047111  042524  051122  EM21:   .ASCIZ  /INTERRUPT FAILING/
1255  002526  050125  020124  040506
```

```
1256  002534  046111  047111  000107
1257  002542  051105  047522  020122  EM22:   .ASCII  /ERROR ON DRIVE PRESENT/<15><12>
1258  002550  047117  042040  044522
1259  002556  042526  050040  042522
1260  002564  042523  052116  005015
1261  002572  044124  020105  047125          .ASCII  /THE UNIT NO'S FOUND BY SETTING RHAS/<15><12>
1262  002600  052111  047040  023517
1263  002606  020123  047506  047125
1264  002614  020104  054502  051440
1265  002622  052105  044524  043516
1266  002630  051040  040510  006523
1267  002636     012
1268  002637     104  020117  047516          .ASCII  /DO NOT AGREE WITH THE UNIT NO. FOUND FROM/<15><12>
1269  002644  020124  043501  042522
1270  002652  020105  044527  044124
1271  002660  052040  042510  052440
1272  002666  044516  020124  047516
1273  002674  020056  047506  047125
1274  002702  020104  051106  046517
1275  002710  005015
1276  002712  044122  051503  026462          .ASCII  /RHCS2-'NED' BIT #12/<15><12>
1277  002720  047047  042105  020047
1278  002726  044502  020124  030443
1279  002734  006462     012
1280  002737     061  033467  033467          .ASCII  /177777-MEANS NO UNIT FOUND/<15><12>
1281  002744  026467  042515  047101
1282  002752  020123  047516  052440
1283  002760  044516  020124  047506
1284  002766  047125  006504     012
1285  002773     116  052117  035105          .ASCII  /NOTE: ON DUAL PORT SYSTEM, DRIVE ON OTHER PORT WILL NOT GIVE/<15><12>
1286  003000  047440  020116  052504
1287  003006  046101  050040  051117
1288  003014  020124  054523  052123
1289  003022  046505  020054  051104
1290  003030  053111  020105  047117
1291  003036  047440  044124  051105
1292  003044  050040  051117  020124
1293  003052  044527  046114  047040
1294  003060  052117  043440  053111
1295  003066  006505     012
1296  003071     047  042516  023504          .ASCIZ  /'NED', HENCE THERE WILL BE AN EXTRA DRIVE/
1297  003076  020054  042510  041516
1298  003104  020105  044124  051105
1299  003112  020105  044527  046114
1300  003120  041040  020105  047101
1301  003126  042440  052130  040522
1302  003134  042040  044522  042526
1303  003142     000
1304  003143     114  047517  020113  EM24:   .ASCIZ  /LOOK AHEAD REGISTER AT THE BEGINNING OF SECTOR IS IN ERROR/
1305  003150  044101  040505  020104
1306  003156  042522  044507  052123
1307  003164  051105  040440  020124
1308  003172  044124  020105  042502
1309  003200  044507  047116  047111
1310  003206  020107  043117  051440
1311  003214  041505  047524  020122
```

```
1312  003222  051511  044440  020116
1313  003230  051105  047522  000122
1314  003236  047514  045517  040440  EM25:   .ASCIZ  /LOOK AHEAD REGISTER IS IN ERROR/
1315  003244  042510  042101  051040
1316  003252  043505  051511  042524
1317  003260  020122  051511  044440
1318  003266  020116  051105  047522
1319  003274  000122
1320  003276  052503  051122  047105  EM30:   .ASCII  /CURRENT CYLINDER DOES NOT MATCH DESIRED CYLINDER REGIHSTER/<15><12>
1321  003304  020124  054503  044514
1322  003312  042116  051105  042040
1323  003320  042517  020123  047516
1324  003326  020124  040515  041524
1325  003334  020110  042504  044523
1326  003342  042522  020104  054503
1327  003350  044514  042116  051105
1328  003356  051040  043505  044111
1329  003364  052123  051105  005015
1330  003372  043101  042524  020122          .ASCIZ  /AFTER A SEEK AND INIT/
1331  003400  020101  042523  045505
1332  003406  040440  042116  044440
1333  003414  044516  000124
1334  003420  041505  020103  042507  EM31:   .ASCII  /ECC GENERATED IS INCORRECT/<15><12>
1335  003426  042516  040522  042524
1336  003434  020104  051511  044440
1337  003442  041516  051117  042522
1338  003450  052103  005015
1339  003454  053105  051105  020131          .ASCIZ  /EVERY WORD ON THIS SECTOR IS THAT GIVEN IN 'DATA USED''/
1340  003462  047527  042122  047440
1341  003470  020116  044124  051511
1342  003476  051440  041505  047524
1343  003504  020122  051511  052040
1344  003512  040510  020124  044507
1345  003520  042526  020116  047111
1346  003526  021040  040504  040524
1347  003534  052440  042523  021104
1348  003542     000
1349  003543     117  020116  042522  EM32:   .ASCII  /ON READ COMMAND, AFTER DATA AND ECC HAVE BEEN READ,/<15><12>
1350  003550  042101  041440  046517
1351  003556  040515  042116  020054
1352  003564  043101  042524  020122
1353  003572  040504  040524  040440
1354  003600  042116  042440  041503
1355  003606  044040  053101  020105
1356  003614  042502  047105  051040
1357  003622  040505  026104  005015
1358  003630  041505  020103  042522          .ASCII  /ECC REGISTERS OR RHER1 ARE IN ERROR/<15><12>
1359  003636  044507  052123  051105
1360  003644  020123  051117  051040
1361  003652  042510  030522  040440
1362  003660  042522  044440  020116
1363  003666  051105  047522  006522
1364  003674     012
1365  003675     117  046116  020131          .ASCII  /ONLY LOWER 11 BITS OF PATTERN REG. CAN BE READ/<15><12>
1366  003702  047514  042527  020122
1367  003710  030461  041040  052111
```

```
1368  003716  020123  043117  050040
1369  003724  052101  042524  047122
1370  003732  051040  043505  020056
1371  003740  040503  020116  042502
1372  003746  051040  040505  006504
1373  003754     012
1374  003755     124  044510  020123          .ASCIZ  /THIS SHOULD MATCH LOWER 11 BITS OF GOOD ECC1/
1375  003762  044123  052517  042114
1376  003770  046440  052101  044103
1377  003776  046040  053517  051105
1378  004004  030440  020061  044502
1379  004012  051524  047440  020106
1380  004020  047507  042117  042440
1381  004026  041503  000061
1382  004032  044510  044107  041440  EM33:   .ASCIZ  /HIGH COUNT BIT NOT SET AFTER 38859 CLOCKS/
1383  004040  052517  052116  041040
1384  004046  052111  047040  052117
1385  004054  051440  052105  040440
1386  004062  052106  051105  031440
1387  004070  034070  034465  041440
1388  004076  047514  045503  000123
1389  004104  042532  047522  042040  EM34:   .ASCIZ  /ZERO DETECT BIT NOT HIGH WHEN 32 BIT ECC REG. HAS 21 ZEROS/
1390  004112  052105  041505  020124
1391  004120  044502  020124  047516
1392  004126  020124  044510  044107
1393  004134  053440  042510  020116
1394  004142  031063  041040  052111
1395  004150  042440  041503  051040
1396  004156  043505  020056  040510
1397  004164  020123  030462  055040
1398  004172  051105  051517     000
1399  004177     120  051517  052111  EM35:   .ASCII  /POSITION REGISTER OR 11 BITS OF PATTERN REGISTER INCORRECT/<15><12>
1400  004204  047511  020116  042522
1401  004212  044507  052123  051105
1402  004220  047440  020122  030461
1403  004226  041040  052111  020123
1404  004234  043117  050040  052101
1405  004242  042524  047122  051040
1406  004250  043505  051511  042524
1407  004256  020122  047111  047503
1408  004264  051122  041505  006524
1409  004272     012
1410  004273     114  053517  051105          .ASCII  /LOWER 11 BITS OF PATTERN REGISTER SHOULD MATCH LOWER/<15><12>
1411  004300  030440  020061  044502
1412  004306  051524  047440  020106
1413  004314  040520  052124  051105
1414  004322  020116  042522  044507
1415  004330  052123  051105  051440
1416  004336  047510  046125  020104
1417  004344  040515  041524  020110
1418  004352  047514  042527  006522
1419  004360     012
1420  004361     061  020061  044502          .ASCII  /11 BITS OF GOOD ECC1/<15><12>
1421  004366  051524  047440  020106
1422  004374  047507  042117  042440
1423  004402  041503  006461     012
```

```
1424  004407    104  052101  042440          .ASCIZ  /DAT ENVLOP GOOD POSITION AND N-CODE ZEROS ARE IN OCTAL/
1425  004414  053116  047514  020120
1426  004422  047507  042117  050040
1427  004430  051517  052111  047511
1428  004436  020116  047101  020104
1429  004444  026516  047503  042504
1430  004452  055040  051105  051517
1431  004460  040440  042522  044440
1432  004466  020116  041517  040524
1433  004474  000114
1434  004476  047117  051040  040505  EM36:   .ASCII  /ON READ COMMAND WITH NON-CORRECTABLE ERROR DCK AND ECH SHOULD BE SET/<1
1435  004504  020104  047503  046515
1436  004512  047101  020104  044527
1437  004520  044124  047040  047117
1438  004526  041455  051117  042522
1439  004534  052103  041101  042514
1440  004542  042440  051122  051117
1441  004550  042040  045503  040440
1442  004556  042116  042440  044103
1443  004564  051440  047510  046125
1444  004572  020104  042502  051440
1445  004600  052105  005015
1446  004604  043111  050040  051517          .ASCIZ  /IF POSITION REGISTER =10040 OR 10041 IT IS GOOD/
1447  004612  052111  047511  020116
1448  004620  042522  044507  052123
1449  004626  051105  036440  030061
1450  004634  032060  020060  051117
1451  004642  030440  030060  030464
1452  004650  044440  020124  051511
1453  004656  043440  047517  000104
1454  004664  051127  052111  047111  EM37:   .ASCIZ  /WRITING WITH BUS ADDRESS HIGHER THAN 28K CAUSED ERROR/
1455  004672  020107  044527  044124
1456  004700  041040  051525  040440
1457  004706  042104  042522  051523
1458  004714  044040  043511  042510
1459  004722  020122  044124  047101
1460  004730  031040  045470  041440
1461  004736  052501  042523  020104
1462  004744  051105  047522  000122
1463  004752  044124  051105  020105  EM40:   .ASCII  /THERE WAS A READ-WRITE HEADER & DATA ERROR DURING 'DTE'/<15><12>
1464  004760  040527  020123  020101
1465  004766  042522  042101  053455
1466  004774  044522  042524  044040
1467  005002  040505  042504  020122
1468  005010  020046  040504  040524
1469  005016  042440  051122  051117
1470  005024  042040  051125  047111
1471  005032  020107  042047  042524
1472  005040  006447    012
1473  005043     124  051505  020124          .ASCIZ  /TEST SETUP - THE TEST WAS ABORTED AT THAT POINT/
1474  005050  042523  052524  020120
1475  005056  020055  044124  020105
1476  005064  042524  052123  053440
1477  005072  051501  040440  047502
1478  005100  052122  042105  040440
1479  005106  020124  044124  052101
```

ERROR POINTER TABLE

```
1480  005114  050040  044517  052116
1481  005122     000
```

```
1482  005123     106  052101  046101  CPHALT: .ASCII  /FATAL ERROR - SEE DOCUMENT FOR BEST COURSE OF ACTION/<15><12>
1483  005130  042440  051122  051117
1484  005136  026440  051440  042505
1485  005144  042040  041517  046525
1486  005152  047105  020124  047506
1487  005160  020122  042502  052123
1488  005166  041440  052517  051522
1489  005174  020105  043117  040440
1490  005202  052103  047511  006516
1491  005210     012
1492  005211     040  005015  177607          .ASCII  / /<15><12><207><377><377><207><377><377><207><377><377>
1493  005216  103777  177777  177607
1494  005224     377
1495  005225     124  042510  050040          .ASCII  /THE PROGRAM HAS HALTED DURING A TEST BECAUSE CONTROLLER/<15><12>
1496  005232  047522  051107  046501
1497  005240  044040  051501  044040
1498  005246  046101  042524  020104
1499  005254  052504  044522  043516
1500  005262  040440  052040  051505
1501  005270  020124  042502  040503
1502  005276  051525  020105  047503
1503  005304  052116  047522  046114
1504  005312  051105  005015
1505  005316  051117  042040  053105          .ASCII  /OR DEVICE HAS LOST 'READY', BECOME UNAVAILABLE./<15><12>
1506  005324  041511  020105  040510
1507  005332  020123  047514  052123
1508  005340  023440  042522  042101
1509  005346  023531  020054  042502
1510  005354  047503  042515  052440
1511  005362  040516  040526  046111
1512  005370  041101  042514  006454
1513  005376     012
1514  005377     107  047117  020105          .ASCIZ  /GONE OFFLINE, OR CANNOT CLEAR STATUS BITS/
1515  005404  043117  046106  047111
1516  005412  026105  047440  020122
1517  005420  040503  047116  052117
1518  005426  041440  042514  051101
1519  005434  051440  040524  052524
1520  005442  020123  044502  051524
1521  005450     000
1522
1523  005451     120  020103  020040  DH1:    .ASCII  /PC      TEST    REG.    GOOD    ACTUAL /<15><12>
1524  005456  020040  052040  051505
1525  005464  020124  020040  051040
1526  005472  043505  020056  020040
1527  005500  043440  047517  020104
1528  005506  020040  040440  052103
1529  005514  040525  004514  005015
1530  005522  020040  020040  020040          .ASCIZ  /       NO      ADDR.   DATA    DATA/
1531  005530  020040  047516  020040
1532  005536  020040  020040  042101
1533  005544  051104  020056  020040
1534  005552  040504  040524  020040
1535  005560  020040  040504  040524
1536  005566     000
1537  005567     120  020103  020040  DH2:    .ASCII  /PC      TEST    WORD    GOOD    BAD/<15><12>
```

```
1538  005574  020040  052040  051505
1539  005602  020124  020040  053440
1540  005610  051117  020104  020040
1541  005616  043440  047517  020104
1542  005624  020040  041040  042101
1543  005632  005015
1544  005634  020040  020040  020040          .ASCIZ  /       NO      NO      DATA    DATA/
1545  005642  020040  047516  020040
1546  005650  020040  020040  047516
1547  005656  020040  020040  020040
1548  005664  040504  040524  020040
1549  005672  020040  040504  040524
1550  005700     000
1551  005701     120  020103  020040  DH11:   .ASCII  /PC     TEST    FAILING CONT.   CONT.   CONT.   CONT./<15><12>
1552  005706  020040  052040  051505
1553  005714  020124  020040  043040
1554  005722  044501  044514  043516
1555  005730  041440  047117  027124
1556  005736  020040  041440  047117
1557  005744  027124  020040  041440
1558  005752  047117  027124  020040
1559  005760  041440  047117  027124
1560  005766  005015
1561  005770  020040  020040  020040          .ASCIZ  /       NO      REG ADR RHCS1   RHCS2   RHDS1   RHER1/
1562  005776  020040  047516  020040
1563  006004  020040  020040  042522
1564  006012  020107  042101  020122
1565  006020  044122  051503  020061
1566  006026  020040  044122  051503
1567  006034  020062  020040  044122
1568  006042  051504  020061  020040
1569  006050  044122  051105  000061
1570  006056  041520  020040  020040  DH14:   .ASCII  /PC     TEST    FAILING CONT.   CONT.   CONT.   CONT.   CONT.   /<15><12
1571  006064  020040  042524  052123
1572  006072  020040  020040  040506
1573  006100  046111  047111  020107
1574  006106  047503  052116  020056
1575  006114  020040  047503  052116
1576  006122  020056  020040  047503
1577  006130  052116  020056  020040
1578  006136  047503  052116  020056
1579  006144  020040  047503  052116
1580  006152  004456  005015
1581  006156  020040  020040  020040          .ASCIZ  /       NO      REG ADR BAD REG RHCS1   RHCS2   RHDS1   RHER1   /
1582  006164  020040  047516  020040
1583  006172  020040  020040  042522
1584  006200  020107  042101  020122
1585  006206  040502  020104  042522
1586  006214  020107  044122  051503
1587  006222  020061  020040  044122
1588  006230  051503  020062  020040
1589  006236  044122  051504  020061
1590  006244  020040  044122  051105
1591  006252  004461     000
1592  006255     120  020103  020040  DH15:   .ASCIZ  /PC     TEST    REG ADR/
1593  006262  020040  052040  051505
```

```
1594  006270  020124  020040  051040
1595  006276  043505  040440  051104
1596  006304     000
1597  006305     120  020103  020040  DH16:   .ASCII  /PC      TEST    WAT     BIT     REG     REG/<15><12>
1598  006312  020040  052040  051505
1599  006320  020124  020040  053440
1600  006326  052101  020040  020040
1601  006334  041040  052111  020040
1602  006342  020040  051040  043505
1603  006350  020040  020040  051040
1604  006356  043505  005015
1605  006362  020040  020040  020040          .ASCIZ  /       NO      PC      WANTED  ADDRESS CONTENTS/
1606  006370  020040  047516  020040
1607  006376  020040  020040  041520
1608  006404  020040  020040  020040
1609  006412  040527  052116  042105
1610  006420  020040  042101  051104
1611  006426  051505  020123  047503
1612  006434  052116  047105  051524
1613  006442     000
1614  006443     120  020103  020040  DH17:   .ASCII  /PC      TEST    CONT.   CONT.   CONT.   CONT.   CONT./<15><12>
1615  006450  020040  052040  051505
1616  006456  020124  020040  041440
1617  006464  047117  027124  020040
1618  006472  041440  047117  027124
1619  006500  020040  041440  047117
1620  006506  027124  020040  041440
1621  006514  047117  027124  020040
1622  006522  041440  047117  027124
1623  006530  005015
1624  006532  020040  020040  020040          .ASCIZ  /       NO      RHBA    RHDB    RHWC    RHCS1   RHCS2/
1625  006540  020040  047516  020040
1626  006546  020040  020040  044122
1627  006554  040502  020040  020040
1628  006562  044122  041104  020040
1629  006570  020040  044122  041527
1630  006576  020040  020040  044122
1631  006604  051503  020061  020040
1632  006612  044122  051503  000062
1633  006620  041520  020040  020040  DH20:   .ASCII  /PC      TEST    CONT    CONT    CONT    CONT    CONT/<15><12>
1634  006626  020040  042524  052123
1635  006634  020040  020040  047503
1636  006642  052116  020040  020040
1637  006650  047503  052116  020040
1638  006656  020040  047503  052116
1639  006664  020040  020040  047503
1640  006672  052116  020040  020040
1641  006700  047503  052116  005015
1642  006706  020040  020040  020040          .ASCIZ  /       NO      RHER1   RHER2   RHER3   RHAS    RHDS1/
1643  006714  020040  047516  020040
1644  006722  020040  020040  044122
1645  006730  051105  020061  020040
1646  006736  044122  051105  020062
1647  006744  020040  044122  051105
1648  006752  020063  020040  044122
1649  006760  051501  020040  020040
```

```
1650  006766  044122  051504  000061
1651  006774  041520  020040  020040  DH21:  .ASCII  /PC      TEST    CONT    CONT    CONT/<15><12>
1652  007002  020040  042524  052123
1653  007010  020040  020040  047503
1654  007016  052116  020040  020040
1655  007024  047503  052116  020040
1656  007032  020040  047503  052116
1657  007040  005015
1658  007042  020040  020040  020040          .ASCIZ  /       NO      RHCS1   RHAS    RHDS1/
1659  007050  020040  047516  020040
1660  007056  020040  020040  044122
1661  007064  051503  020061  020040
1662  007072  044122  051501  020040
1663  007100  020040  044122  051504
1664  007106  000061
1665  007110  041520  020040  020040  DH22:  .ASCII  /PC      TEST    RHAS    RHCS2/<15><12>
1666  007116  020040  042524  052123
1667  007124  020040  020040  044122
1668  007132  051501  020040  020040
1669  007140  044122  051503  006462
1670  007146     012
1671  007147     040  020040  020040          .ASCIZ  /       NO      UNIT    UNIT/
1672  007154  020040  047040  020117
1673  007162  020040  020040  052440
1674  007170  044516  020124  020040
1675  007176  052440  044516  000124
1676  007204  041520  020040  020040  DH24:  .ASCII  /PC      TEST    RHDST   BAD     GOOD    SECTOR  SECTOR/<15><12>
1677  007212  020040  042524  052123
1678  007220  020040  020040  044122
1679  007226  051504  020124  020040
1680  007234  040502  020104  020040
1681  007242  020040  047507  042117
1682  007250  020040  020040  042523
1683  007256  052103  051117  020040
1684  007264  042523  052103  051117
1685  007272  005015
1686  007274  020040  020040  020040          .ASCIZ  /       NO      CONT.   RHLA    RHLA    NO      CLOCK/
1687  007302  020040  047516  020040
1688  007310  020040  020040  047503
1689  007316  052116  020056  020040
1690  007324  044122  040514  020040
1691  007332  020040  044122  040514
1692  007340  020040  020040  047516
1693  007346  020040  020040  020040
1694  007354  046103  041517  000113
1695  007362  041520  020040  020040  DH26:  .ASCII  /PC      TEST    PC OF   FAILING CONT.   CONT.   CONT.   CONT./<15><12>
1696  007370  020040  042524  052123
1697  007376  020040  020040  041520
1698  007404  047440  020106  020040
1699  007412  040506  046111  047111
1700  007420  020107  047503  052116
1701  007426  020056  020040  047503
1702  007434  052116  020056  020040
1703  007442  047503  052116  020056
1704  007450  020040  047503  052116
1705  007456  006456     012
```

```
1706  007461     040  020040  020040          .ASCIZ  /       NO      JSR     REG ADD RHCS1   RHCS2   RHDS1   RHER1  /
1707  007466  020040  047040  020117
1708  007474  020040  020040  045040
1709  007502  051123  020040  020040
1710  007510  051040  043505  040440
1711  007516  042104  051040  041510
1712  007524  030523  020040  051040
1713  007532  041510  031123  020040
1714  007540  051040  042110  030523
1715  007546  020040  051040  042510
1716  007554  030522  000011
1717  007560  041520  020040  020040  DH27:   .ASCII  /PC      TEST    PC OF   FAILING GOOD    ACTUAL /<15><12>
1718  007566  020040  042524  052123
1719  007574  020040  020040  041520
1720  007602  047440  020106  020040
1721  007610  040506  046111  047111
1722  007616  020107  047507  042117
1723  007624  020040  020040  041501
1724  007632  052524  046101  006411
1725  007640     012
1726  007641     040  020040  020040          .ASCIZ  /       NO      JSR     REG.    DATA    DATA/
1727  007646  020040  047040  020117
1728  007654  020040  020040  045040
1729  007662  051123  020040  020040
1730  007670  051040  043505  020056
1731  007676  020040  042040  052101
1732  007704  020101  020040  042040
1733  007712  052101  000101
1734  007716  041520  020040  020040  DH30:   .ASCII  /PC      TEST    PC OF   REG.    GOOD    BAD/<15><12>
1735  007724  020040  042524  052123
1736  007732  020040  020040  041520
1737  007740  047440  020106  020040
1738  007746  042522  027107  020040
1739  007754  020040  047507  042117
1740  007762  020040  020040  040502
1741  007770  006504     012
1742  007773     040  020040  020040          .ASCIZ  /       NO      JSR     ADDR    DATA    DATA/
1743  010000  020040  047040  020117
1744  010006  020040  020040  045040
1745  010014  051123  020040  020040
1746  010022  040440  042104  020122
1747  010030  020040  042040  052101
1748  010036  020101  020040  042040
1749  010044  052101  000101
1750  010050  041520  020040  020040  DH31:   .ASCII  /PC      TEST    WORD    GOOD    BAD     CONT.   CONT.   CONT./<15><12>
1751  010056  020040  042524  052123
1752  010064  020040  020040  047527
1753  010072  042122  020040  020040
1754  010100  047507  042117  020040
1755  010106  020040  040502  020104
1756  010114  020040  020040  047503
1757  010122  052116  020056  020040
1758  010130  047503  052116  020056
1759  010136  020040  047503  052116
1760  010144  006456     012
1761  010147     040  020040  020040          .ASCIZ  /       NO      NO      DATA    DATA    RHCS1   RHDS1   RHER1  /
```

```
1762  010154  020040  047040  020117
1763  010162  020040  020040  047040
1764  010170  020117  020040  020040
1765  010176  042040  052101  020101
1766  010204  020040  042040  052101
1767  010212  020101  020040  051040
1768  010220  041510  030523  020040
1769  010226  051040  042110  030523
1770  010234  020040  051040  042510
1771  010242  030522  000011
1772  010246  041520  020040  020040  DH32:   .ASCII  /PC      TEST    PC OF   WORD    GOOD    BAD     CONT.   CONT.   CONT./
1773  010254  020040  042524  052123
1774  010262  020040  020040  041520
1775  010270  047440  020106  020040
1776  010276  047527  042122  020040
1777  010304  020040  047507  042117
1778  010312  020040  020040  040502
1779  010320  020104  020040  020040
1780  010326  047503  052116  020056
1781  010334  020040  047503  052116
1782  010342  004456  020040  047503
1783  010350  052116  006456     012
1784  010355     040  020040  020040          .ASCIZ  /       NO      JSR     NO      DATA    DATA    RHCS1   RHDS1   RHER1/
1785  010362  020040  047040  020117
1786  010370  020040  020040  045040
1787  010376  051123  020040  020040
1788  010404  047040  020117  020040
1789  010412  020040  042040  052101
1790  010420  020101  020040  042040
1791  010426  052101  020101  020040
1792  010434  051040  041510  030523
1793  010442  020040  051040  042110
1794  010450  030523  020040  051040
1795  010456  042510  030522     000
1796  010463     120  020103  020040  DH33:   .ASCII  /PC      TEST    PC OF   WORD    GOOD    CONT.   CONT.   CONT.   /<15><12
1797  010470  020040  052040  051505
1798  010476  020124  020040  050040
1799  010504  020103  043117  020040
1800  010512  053440  051117  020104
1801  010520  020040  043440  047517
1802  010526  020104  020040  041440
1803  010534  047117  027124  020040
1804  010542  041440  047117  027124
1805  010550  020040  041440  047117
1806  010556  027124  006411     012
1807  010563     040  020040  020040          .ASCIZ  /       NO      JSR     NO      DATA    RHCS1   RHDS1   RHER1   /
1808  010570  020040  047040  020117
1809  010576  020040  020040  045040
1810  010604  051123  020040  020040
1811  010612  047040  020117  020040
1812  010620  020040  042040  052101
1813  010626  020101  020040  051040
1814  010634  041510  030523  020040
1815  010642  051040  042110  030523
1816  010650  020040  051040  042510
1817  010656  030522  000011
```

```
1818  010662  041520  020040  020040  DH34:   .ASCII  /PC     TEST   GOOD   GOOD   WRITTEN WRITTEN DATA/<15><12>
1819  010670  020040  042524  052123
1820  010676  020040  020040  047507
1821  010704  042117  020040  020040
1822  010712  047507  042117  020040
1823  010720  020040  051127  052111
1824  010726  042524  020116  051127
1825  010734  052111  042524  020116
1826  010742  040504  040524  005015
1827  010750  020040  020040  020040          .ASCIZ  /      NO     ECC1   ECC2   ECC1   ECC2   USED/
1828  010756  020040  047516  020040
1829  010764  020040  020040  041505
1830  010772  030503  020040  020040
1831  011000  041505  031103  020040
1832  011006  020040  041505  030503
1833  011014  020040  020040  041505
1834  011022  031103  020040  020040
1835  011030  051525  042105     000
1836  011035     120  020103  020040  DH35:   .ASCII  /PC     TEST   GOOD   GOOD   PATTERN POSITON GOOD    RHER1  /<15><12
1837  011042  020040  052040  051505
1838  011050  020124  020040  043440
1839  011056  047517  020104  020040
1840  011064  043440  047517  020104
1841  011072  020040  050040  052101
1842  011100  042524  047122  050040
1843  011106  051517  052111  047117
1844  011114  043440  047517  020104
1845  011122  020040  051040  042510
1846  011130  030522  006411     012
1847  011135     040  020040  020040          .ASCIZ  /      NO     ECC1   ECC2   REG.   REG.   POSITON REG./
1848  011142  020040  047040  020117
1849  011150  020040  020040  042440
1850  011156  041503  020061  020040
1851  011164  042440  041503  020062
1852  011172  020040  051040  043505
1853  011200  020056  020040  051040
1854  011206  043505  020056  020040
1855  011214  050040  051517  052111
1856  011222  047117  051040  043505
1857  011230  000056
1858  011232  041520  020040  020040  DH36:   .ASCII  /PC     TEST   PC OF  RHMR   POSITON PATTERN/<15><12>
1859  011240  020040  042524  052123
1860  011246  020040  020040  041520
1861  011254  047440  020106  020040
1862  011262  044122  051115  020040
1863  011270  020040  047520  044523
1864  011276  047524  020116  040520
1865  011304  052124  051105  006516
1866  011312     012
1867  011313     040  020040  020040          .ASCIZ  /      NO     JSR    CONT.  REG.   REG./
1868  011320  020040  047040  020117
1869  011326  020040  020040  045040
1870  011334  051123  020040  020040
1871  011342  041440  047117  027124
1872  011350  020040  051040  043505
1873  011356  020056  020040  051040
```

```
1874  011364  043505  000056
1875  011370  041520  020040  020040  DH37:   .ASCII  /PC     TEST   POSITON POSITON GOOD   GOOD   PATTERN DATA   N-CODE/
1876  011376  020040  042524  052123
1877  011404  020040  020040  047520
1878  011412  044523  047524  020116
1879  011420  047520  044523  047524
1880  011426  020116  047507  042117
1881  011434  020040  020040  047507
1882  011442  042117  020040  020040
1883  011450  040520  052124  051105
1884  011456  020116  040504  040524
1885  011464  020040  020040  026516
1886  011472  047503  042504  005015
1887  011500  020040  020040  020040          .ASCIZ  /      NO     ECC    GOOD   ECC1   ECC2   ECC    ENVLOPE ZEROS
1888  011506  020040  047516  020040
1889  011514  020040  020040  041505
1890  011522  020103  020040  020040
1891  011530  047507  042117  020040
1892  011536  020040  041505  030503
1893  011544  020040  020040  041505
1894  011552  031103  020040  020040
1895  011560  041505  020103  020040
1896  011566  020040  047105  046126
1897  011574  050117  020105  042532
1898  011602  047522  004523     000
1899
1900  011607     120  020103  020040  DH40:   .ASCII  /PC     TEST   FAILING CONT.  CONT.  CONT.  CONT./<15><12>
1901  011614  020040  052040  051505
1902  011622  020124  020040  043040
1903  011630  044501  044514  043516
1904  011636  041440  047117  027124
1905  011644  020040  041440  047117
1906  011652  027124  020040  041440
1907  011660  047117  027124  020040
1908  011666  041440  047117  027124
1909  011674  005015
1910  011676  020040  020040  020040          .ASCIZ  /      NO     REG ADR RHCS1  RHCS2  RHDS1  RHER1/
1911  011704  020040  047516  020040
1912  011712  020040  020040  042522
1913  011720  020107  042101  020122
1914  011726  044122  051503  020061
1915  011734  020040  044122  051503
1916  011742  020062  020040  044122
1917  011750  051504  020061  020040
1918  011756  044122  051105  000061
1919
1920                                    .EVEN
1921
1922  011764  001116  017330  045534  DT1:    .WORD   $ERRPC,TSTNM,REGADR,$GDDAT,$BDDAT,0
1923  011772  001124  001126  000000
1924  012000  001116  017330  052774  DT2:    .WORD   $ERRPC,TSTNM,ERWORD,$GDDAT,0
1925  012006  001124  000000
1926  012012  001116  017330  052774  DT3:    .WORD   $ERRPC,TSTNM,ERWORD,$GDDAT,$BDDAT,0
1927  012020  001124  001126  000000
1928
1929  012026  001116  017330  001122  DT11:   .WORD   $ERRPC,TSTNM,$BDADR,CS1,CS2,DS1,ER1,0
```

```
1930  012034  015032  015030  015054
1931  012042  015034  000000
1932  012046  001116  017330  001122  DT14:   .WORD   $ERRPC,TSTNM,$BDADR,$BDDAT,CS1,CS2,DS1,ER1,0
1933  012054  001126  015032  015030
1934  012062  015054  015034  000000
1935  012070  001116  017330  001200  DT15:   .WORD   $ERRPC,TSTNM,$TMP1,0
1936  012076  000000
1937  012100  001116  017330  001204  DT16:   .WORD   $ERRPC,TSTNM,$TMP3,$TMP1,$TMP0,$BDDAT,0
1938  012106  001200  001176  001126
1939  012114  000000
1940  012116  001116  017330  015026  DT17:   .WORD   $ERRPC,TSTNM,BA,DB,WC,CS1,CS2,0
1941  012124  015022  015024  015032
1942  012132  015030  000000
1943
1944  012136  001116  017330  015034  DT20:   .WORD   $ERRPC,TSTNM,ER1,ER2,ER3,AS,DS1,0
1945  012144  015040  015046  015050
1946  012152  015054  000000
1947  012156  001116  017330  015032  DT21:   .WORD   $ERRPC,TSTNM,CS1,AS,DS1,0
1948  012164  015050  015054  000000
1949  012172  001116  017330  001124  DT22:   .WORD   $ERRPC,TSTNM,$GDDAT,$BDDAT,0
1950  012200  001126  000000
1951  012204  001116  017330  015036  DT24:   .WORD   $ERRPC,TSTNM,DST,$BDDAT,$TMP1,$TMP2,$TMP3,0
1952  012212  001126  001200  001202
1953  012220  001204  000000
1954  012224  001116  017330  015132  DT26:   .WORD   $ERRPC,TSTNM,PCJSR,$BDADR,CS1,CS2,DS1,ER1,0
1955  012232  001122  015032  015030
1956  012240  015054  015034  000000
1957  012246  001116  017330  015132  DT27:   .WORD   $ERRPC,TSTNM,PCJSR,REGADR,$GDDAT,$BDDAT,0
1958  012254  045534  001124  001126
1959  012262  000000
1960  012264  001116  017330  015132  DT30:   .WORD   $ERRPC,TSTNM,PCJSR,REGADR,$GDDAT,$BDDAT,0
1961  012272  045534  001124  001126
1962  012300  000000
1963
1964  012302  001116  017330  052774  DT31:   .WORD   $ERRPC,TSTNM,ERWORD,$GDDAT,$BDDAT,CS1,DS1,ER1,0
1965  012310  001124  001126  015032
1966  012316  015054  015034  000000
1967  012324  001116  017330  015132  DT32:   .WORD   $ERRPC,TSTNM,PCJSR,ERWORD,$GDDAT,$BDDAT,CS1,DS1,ER1,0
1968  012332  052774  001124  001126
1969  012340  015032  015054  015034
1970  012346  000000
1971  012350  001116  017330  015132  DT33:   .WORD   $ERRPC,TSTNM,PCJSR,ERWORD,$GDDAT,CS1,DS1,ER1,0
1972  012356  052774  001124  015032
1973  012364  015054  015034  000000
1974  012372  001116  017330  050464  DT34:   .WORD   $ERRPC,TSTNM,GECC1,GECC2,WECC1,WECC2,DISK,0
1975  012400  050466  055572  055574
1976  012406  054572  000000
1977  012412  001116  017330  050464  DT35:   .WORD   $ERRPC,TSTNM,GECC1,GECC2,EC2,EC1,POSITI,ER1,0
1978  012420  050466  015064  015062
1979  012426  050476  015034  000000
1980  012434  001116  017330  015132  DT36:   .WORD   $ERRPC,TSTNM,PCJSR,MR,EC1,EC2,0
1981  012442  015052  015062  015064
1982  012450  000000
1983  012452  001116  017330  015062  DT37:   .WORD   $ERRPC,TSTNM,EC1,POSITI,GECC1,GECC2,EC2,DATENV,ZCODE,0
1984  012460  050476  050464  050466
1985  012466  015064  050502  050504
```

```
1986  012474  000000
1987  012476  001116  017330  001122  DT40:   .WORD   $ERRPC,TSTNM,$BDADR,CS1,CS2,DS1,ER1,0
1988  012504  015032  015030  015054
1989  012512  015034  000000
1990
1991
1992
1993
1994  012516      000     000     000  DF1:    .BYTE   0,0,0,0,0
1995  012521      000     000
1996  012523      000     000     001  DF2:    .BYTE   0,0,1,0
1997  012526      000
1998  012527      000     000     001  DF3:    .BYTE   0,0,1,0,0
1999  012532      000     000
2000
2001  012534      000     000     000  DF11:   .BYTE   0,0,0,0,0,0,0
2002  012537      000     000     000
2003  012542      000
2004  012543      000     000     000  DF14:   .BYTE   0,0,0,0,0,0,0,0
2005  012546      000     000     000
2006  012551      000     000
2007  012553      000     000     000  DF15:   .BYTE   0,0,0
2008  012556      000     000     000  DF16:   .BYTE   0,0,0,0,0
2009  012561      000     000
2010  012563      000     000     000  DF17:   .BYTE   0,0,0,0,0,0,0
2011  012566      000     000     000
2012  012571      000
2013  012572      000     000     000  DF20:   .BYTE   0,0,0,0,0,0,0
2014  012575      000     000     000
2015  012600      000
2016
2017  012601      000     000     000  DF21:   .BYTE   0,0,0,0,0
2018  012604      000     000
2019  012606      000     000     000  DF22:   .BYTE   0,0,0,0
2020  012611      000
2021  012612      000     000     000  DF24:   .BYTE   0,0,0,0,0,0,0
2022  012615      000     000     000
2023  012620      000
2024  012621      000     000     000  DF26:   .BYTE   0,0,0,0,0,0,0,0
2025  012624      000     000     000
2026  012627      000     000
2027  012631      000     000     000  DF27:   .BYTE   0,0,0,0,0,0
2028  012634      000     000     000
2029  012637      000     000     000  DF30:   .BYTE   0,0,0,0,0,0
2030  012642      000     000     000
2031
2032  012645      000     000     001  DF31:   .BYTE   0,0,1,0,0,0,0,0
2033  012650      000     000     000
2034  012653      000     000
2035  012655      000     000     000  DF32:   .BYTE   0,0,0,1,0,0,0,0,0
2036  012660      001     000     000
2037  012663      000     000     000
2038  012666      000     000     000  DF33:   .BYTE   0,0,0,1,0,0,0,0
2039  012671      001     000     000
2040  012674      000     000
2041  012676      000     000     000  DF34:   .BYTE   0,0,0,0,0,0,0
```

```
2042  012701  000  000  000
2043  012704  000
2044  012705  000  000  000  DF35:  .BYTE  0,0,0,0,0,0,0,0
2045  012710  000  000  000
2046  012713  000  000
2047  012715  000  000  000  DF36:  .BYTE  0,0,0,0,0,0
2048  012720  000  000  000
2049  012723  000  000  000  DF37:  .BYTE  0,0,0,0,0,0,0,0,0
2050  012726  000  000  000
2051  012731  000  000  000
2052  012734  000  000  000  DF40:  .BYTE  0,0,0,0,0,0,0
2053  012737  000  000  000
2054  012742  000
2055
2056        012744                       .EVEN
2057  012744  044122  030461  051040  EM42:  .ASCIZ  /RH11 REGISTER FAILED TO RESPOND TO A ''TST'' COMMAND/
2058  012752  043505  051511  042524
2059  012760  020122  040506  046111
2060  012766  042105  052040  020117
2061  012774  042522  050123  047117
2062  013002  020104  047524  040440
2063  013010  021040  051524  021124
2064  013016  041440  046517  040515
2065  013024  042116     000
2066
2067  013027     122  030510  020061  EM43:  .ASCIZ  /RH11 ILLEGAL REGISTER RESPONSE TEST/
2068  013034  046111  042514  040507
2069  013042  020114  042522  044507
2070  013050  052123  051105  051040
2071  013056  051505  047520  051516
2072  013064  020105  042524  052123
2073  013072     000
2074
2075  013073     115  053117  020105  EM44:  .ASCIZ  /MOVE BYTE TO WORD COUNT TEST/
2076  013100  054502  042524  052040
2077  013106  020117  047527  042122
2078  013114  041440  052517  052116
2079  013122  052040  051505  000124
2080
2081  013130  044502  020123  054502  EM45:  .ASCIZ  /BIS BYTE TO WORD COUNT/
2082  013136  042524  052040  020117
2083  013144  047527  042122  041440
2084  013152  052517  052116     000
2085
2086  013157     102  041511  041040  EM46:  .ASCIZ  /BIC BYTE TO WORD COUNT/
2087  013164  052131  020105  047524
2088  013172  053440  051117  020104
2089  013200  047503  047125  000124
2090
2091  013206  042522  044507  052123  EM47:  .ASCIZ  /REGISTER ADDRESS SELECT TEST/
2092  013214  051105  040440  042104
2093  013222  042522  051523  051440
2094  013230  046105  041505  020124
2095  013236  042524  052123     000
2096
2097  013243     116  047117  042440  EM50:  .ASCIZ  /NON EXISTANT DRIVE (NED) TEST/
```

```
2098  013250  044530  052123  047101
2099  013256  020124  051104  053111
2100  013264  020105  047050  042105
2101  013272  020051  042524  052123
2102  013300     000
2103
2104  013301     101  020123  042522  EM51:   .ASCIZ  /AS REGISTER TEST/
2105  013306  044507  052123  051105
2106  013314  052040  051505  000124
2107
2108  013322  052502  051523  040440  EM52:   .ASCIZ  /BUSS ADDRESS REGISTER DATA TEST/
2109  013330  042104  042522  051523
2110  013336  051040  043505  051511
2111  013344  042524  020122  040504
2112  013352  040524  052040  051505
2113  013360  000124
2114
2115  013362  052502  051523  040440  EM53:   .ASCIZ  /BUSS ADDRESS REGISTER MOVE BYTE/
2116  013370  042104  042522  051523
2117  013376  051040  043505  051511
2118  013404  042524  020122  047515
2119  013412  042526  041040  052131
2120  013420  000105
2121
2122  013422  044122  030461  044440  EM54:   .ASCIZ  /RH11 INTERRUPT FAILED TO OCCUR/
2123  013430  052116  051105  052522
2124  013436  052120  043040  044501
2125  013444  042514  020104  047524
2126  013452  047440  041503  051125
2127  013460     000
2128
2129  013461     122  051505  052105  EM55:   .ASCIZ  /RESET TEST/
2130  013466  052040  051505  000124
2131
2132  013474  054115  020106  044502  EM56:   .ASCIZ  /MXF BIT TEST/
2133  013502  020124  042524  052123
2134  013510     000
2135
2136  013511     125  044516  052502  EM57:   .ASCIZ  /UNIBUS PARITY BIT TEST/
2137  013516  020123  040520  044522
2138  013524  054524  041040  052111
2139  013532  052040  051505  000124
2140
2141  013540  047504  051505  051440  EM60:   .ASCIZ  /DOES SELECTING THE RH11 CLEAR THE UNIT SELECT REGISTER/
2142  013546  046105  041505  044524
2143  013554  043516  052040  042510
2144  013562  051040  030510  020061
2145  013570  046103  040505  020122
2146  013576  044124  020105  047125
2147  013604  052111  051440  046105
2148  013612  041505  020124  042522
2149  013620  044507  052123  051105
2150  013626     000
2151
2152  013627     104  042517  020123  EM61:   .ASCIZ  /DOES TRE GET SET BY UNIBUS PARITY ERROR/
2153  013634  051124  020105  042507
```

```
2154  013642  020124  042523  020124
2155  013650  054502  052440  044516
2156  013656  052502  020123  040520
2157  013664  044522  054524  042440
2158  013672  051122  051117     000
2159
2160  013677     116  047117  042440  EM62:   .ASCIZ  /NON EXISTANT DRIVE (NED) FAILED TO SET (TRE)/
2161  013704  044530  052123  047101
2162  013712  020124  051104  053111
2163  013720  020105  047050  042105
2164  013726  020051  040506  046111
2165  013734  042105  052040  020117
2166  013742  042523  020124  052050
2167  013750  042522  000051
2168
2169  013754  047516  020116  054105  EM63:   .ASCIZ  /NON EXISTANT MEMORY (NEM) FAILED TO SET (TRE)/
2170  013762  051511  040524  052116
2171  013770  046440  046505  051117
2172  013776  020131  047050  046505
2173  014004  020051  040506  046111
2174  014012  042105  052040  020117
2175  014020  042523  020124  052050
2176  014026  042522  000051
2177
2178  014032  047520  052122  051440  EM64:   .ASCIZ  /PORT SELECT FAILED TO CLEAR/
2179  014040  046105  041505  020124
2180  014046  040506  046111  042105
2181  014054  052040  020117  046103
2182  014062  040505  000122
2183
2184  014066  047503  052116  047522  EM65:   .ASCIZ  /CONTROLLER CLEAR FAILED TO CLEAR COMMAND REGISTER/
2185  014074  046114  051105  041440
2186  014102  042514  051101  043040
2187  014110  044501  042514  020104
2188  014116  047524  041440  042514
2189  014124  051101  041440  046517
2190  014132  040515  042116  051040
2191  014140  043505  051511  042524
2192  014146  000122
2193
2194  014150  042522  042523  042514  EM66:   .ASCIZ  /RESELECT FAILED TO CLEAR COMMAND REGISTER/
2195  014156  052103  043040  044501
2196  014164  042514  020104  047524
2197  014172  041440  042514  051101
2198  014200  041440  046517  040515
2199  014206  042116  051040  043505
2200  014214  051511  042524  000122
2201
2202  014222  047111  042524  051122  EM67:   .ASCIZ  /INTERRUPT CAUSED BY  RDY!IE BITS SET FAILED TO OCCUR/
2203  014230  050125  020124  040503
2204  014236  051525  042105  041040
2205  014244  020131  051040  054504
2206  014252  044441  020105  044502
2207  014260  051524  051440  052105
2208  014266  043040  044501  042514
2209  014274  020104  047524  047440
```

```
2210  014302  041503  051125     000
2211
2212  014307     111  020105  040506  EM70:   .ASCIZ  /IE FAILED TO CLEAR FOLLOWING AN INTERRUPT/
2213  014314  046111  042105  052040
2214  014322  020117  046103  040505
2215  014330  020122  047506  046114
2216  014336  053517  047111  020107
2217  014344  047101  044440  052116
2218  014352  051105  052522  052120
2219  014360     000
2220
2221  014361     101  020123  042522  EM71:   .ASCIZ  /AS REGISTER WRITE TEST/
2222  014366  044507  052123  051105
2223  014374  053440  044522  042524
2224  014402  052040  051505  000124
2225
2226
2227
2228  014410  041520  020040  020040  DH42:   .ASCII  /PC      TEST    ILLEGAL/<15><12>
2229  014416  020040  042524  052123
2230  014424  020040  020040  046111
2231  014432  042514  040507  006514
2232  014440     012
2233  014441     040  020040  020040          .ASCIZ  /       NO      ADDRESS/
2234  014446  020040  047040  020117
2235  014454  020040  020040  040440
2236  014462  042104  042522  051523
2237  014470     000
2238  014471     120  020103  020040  DH44:   .ASCII  /PC      TEST    REGISTR CONT    CONT/<15><12>
2239  014476  020040  052040  051505
2240  014504  020124  020040  051040
2241  014512  043505  051511  051124
2242  014520  041440  047117  020124
2243  014526  020040  041440  047117
2244  014534  006524     012
2245  014537     040  020040  020040          .ASCIZ  /       NO      ADDRESS GOOD    BAD/
2246  014544  020040  047040  020117
2247  014552  020040  020040  040440
2248  014560  042104  042522  051523
2249  014566  043440  047517  020104
2250  014574  020040  041040  042101
2251  014602     000
2252  014603     120  020103  020040  DH54:   .ASCII  /PC      TEST    REGISTR CONT/<15><12>
2253  014610  020040  052040  051505
2254  014616  020124  020040  051040
2255  014624  043505  051511  051124
2256  014632  041440  047117  006524
2257  014640     012
2258  014641     040  020040  020040          .ASCIZ  /       NO      ADDRESS/
2259  014646  020040  047040  020117
2260  014654  020040  020040  040440
2261  014662  042104  042522  051523
2262  014670     000
2263
2264          014672                  .EVEN
2265
```

ERROR POINTER TABLE

```
 2266  014672  001116  017330  045534  DT42:   .WORD   $ERRPC,TSTNM,REGADR,0
 2267  014700  000000
 2268  014702  001116  017330  045534  DT44:   .WORD   $ERRPC,TSTNM,REGADR,$GDDAT,$BDDAT,0
 2269  014710  001124  001126  000000
 2270  014716  001116  017330  045534  DT54:   .WORD   $ERRPC,TSTNM,REGADR,$GDDAT,0
 2271  014724  001124  000000
```

```
2272  014730  000  000  000  DF42:  .BYTE  0,0,0
2273  014733  000  000  000  DF44:  .BYTE  0,0,0,0,0
2274  014736  000  000
2275  014740  000  000  000  DF54:  .BYTE  0,0,0,0
2276  014743  000
```

```
                        .SBTTL  HARDWARE REGISTER BIT DEFINITIONS


                        ;RH11 REGISTERS


                        ;WORD COUNT REGISTER (RHWC)
                        ;EACH BIT IS CALLED BY BIT NUMBER



                        ;BUS ADDRESS REGISTER (RHBA)
                        ;EACH BIT IS CALLED BY BIT NUMBER



                        ;CONTROL AND STATUS REGISTER 2 (RHCS2)

000001                  US1=    1               ;UNIT SELECT (BIT #0)
000002                  US2=    2               ;UNIT SELECT (BIT #1)
000004                  US4=    4               ;UNIT SELECT (BIT #2)
000010                  BAI=    10              ;BUS ADDRESS INCREMENT INHIBIT (BIT #3)
000020                  PAT=    20              ;INVERT PARITY ON MASS BUS TO EVEN (BIT #4)
000040                  CLR=    40              ;CLEAR (BIT #5)
000100                  IR=     100             ;INPUT READY (BIT #6)
000200                  OR=     200             ;OUTPUT READY (BIT #7)
000400                  MPE=    400             ;MASS BUS PARITY ERROR (BIT #8)
001000                  MXF=    1000            ;MISSED TRANSFER ERROR (BIT #9)
002000                  PGE=    2000            ;PROGRAM ERROR (BIT #10)
004000                  NEM=    4000            ;NON EXISTANT MEMORY (BIT #11)
010000                  NED=    10000           ;NON EXISTANT DRIVE (BIT #12)
020000                  UPE=    20000           ;UNIBUS PARITY ERROR (BIT #13)
040000                  WCE=    40000           ;WRITE CHECK ERROR (BIT #14)
100000                  DLT=    100000          ;DATA LATE (BIT #15)

                        ;DATA BUFFER REGISTER (RHDB)
                        ;EACH BIT IS CALLED BY BIT NUMBER
```

```
                        ;RP04 REGISTERS


                        ;CONTROL AND STATUS 1 REGISTER. (#00)

000001                  GO=     1               ;GO (BIT #0)
000100                  IE=     100             ;INTERRUPT ENABLE (BIT #6)
000200                  RDY=    200             ;READY (BIT #7)
000400                  A16=    400             ;HIGH ORDER UNIBUS BITS (BIT #8)
001000                  A17=    1000            ;HIGH ORDER UNIBUS BITS (BIT #9)
002000                  PSEL=   2000            ;PORT SELECT (BIT #10)
004000                  DVA=    4000            ;DEVICE AVAILABLE (BIT #11)
020000                  MCPE=   20000           ;MASSBUSS PARITY ERROR (BIT #13)
040000                  TRE=    40000           ;TRANSFER ERROR (BIT #14)
100000                  SC=     100000          ;SPECIAL CONDITION (BIT #15)

                        ;STATUS REGISTER (RHDS1) (#01)

000001                  DF5=    1               ;DRIVE FORWARD 5''/SEC.  (BIT #0)
000002                  DFF20=  2               ;DRIVE FORWARD 20''/SEC.  (BIT #1)
000004                  DIGB=   4               ;DRIVE TO INNER GAVRD BAND (BIT #2)
000010                  GRV=    10              ;GO REVERSE (BIT #3)
000020                  DL64=   20              ;DIFFERENCE LESS THAN 64 (BIT #4)
000040                  DE1=    40              ;DIFFERENCE EQUALS 1 (BIT #5)
000100                  VV=     100             ;VOLUME VALID (BIT #6)
000200                  DRY=    200             ;DRIVE READY (BIT #7)
000400                  DPR=    400             ;DRIVE PRESENT (BIT #8)
001000                  PROG=   1000            ;PROGRAMABLE (BIT #9)
002000                  LST=    2000            ;LAST SECTOR TRANSFERRED (BIT #10)
004000                  WRL=    4000            ;WRITE LOCK (BIT #11)
010000                  MOL=    10000           ;MEDIUM ON-LINE (BIT #12)
020000                  PIP=    20000           ;POSITIONING OPERATION IN PROGRESS (BIT #13)
040000                  ERR=    40000           ;COMPOSIT ERROR. (BIT #14)
100000                  ATA=    100000          ;ATTENTION ACTIVE (BIT #15)

                        ;ERROR REGISTER #01 (RHER1) (#02)
000001                  ILF=    1               ;ILLEGAL FUNCTION (BIT #0)
000002                  ILR=    2               ;ILLEGAL REGISTER (BIT #1)
000004                  RMR=    4               ;REGISTER MODIFICATION REFUSED (BIT #2)
000010                  PAR=    10              ;PARITY ERROR (BIT #3)
000020                  FER=    20              ;FORMAT ERROR (BIT #4)
000040                  WCF=    40              ;WRITE CLOCK FAIL (BIT #5)
000100                  ECH=    100             ;ECC HARD ERROR (BIT #6)
000200                  HCE=    200             ;HEADER COMPARE ERROR (BIT #7)
000400                  HCRC=   400             ;HEADER CRC ERROR (BIT #8)
001000                  AOE=    1000            ;ADDRESS OVERFLOW ERROR (BIT #9)
002000                  IAE=    2000            ;INVALID ADDRESS ERROR (BIT #10)
004000                  WLE=    4000            ;WRITE LOCK ERROR (BIT #11)
010000                  DTE=    10000           ;DRIVE TIMING ERROR (BIT #12)
020000                  OPI=    20000           ;OPERATION INCOMPLETE (BIT #13)
040000                  UNS=    40000           ;DRIVE UNSAFE (BIT #14)
100000                  DCK=    100000          ;DATA CHECK ERROR (BIT 15)

                        ;MAINTAINABILITY REGISTER (RHMR)(#03)
```

```
                        DMD=    1               ;DIAGINOSTIC MODE (BIT #0)
                        MCLK=   2               ;MAINTAINABILITY CLOCK (BIT #1)
                        MINX=   4               ;MAINTAINABILITY INDEX (BIT #2)
                        MSTCK=  10              ;MAINTAINABILITY SECTOR CLOCK (BIT #3)
                        MRD=    20              ;MAINTAINABILITY READ (BIT #4)
                        MWR=    40              ;MAINTAINABILITY WRITE (BIT #5)
                        DENVL=  200             ;DATA ENVELOPE (BIT #7)
                        ZER=    400             ;ZERO DETECT (BIT #8)
                        DTSY=   1000            ;MAINTAINABILITY SYNC DETECTED (BIT #9)

                        ;ATTENTION SUMMARY PSEUDO-REGISTER (RHAS) (#04)

                        AT0=    1               ;DEVICE 0 (BIT #0)
                        AT1=    2               ;DEVICE 1 (BIT #1)
                        AT2=    4               ;DEVICE 2 (BIT #2)
                        AT3=    10              ;DEVICE 3 (BIT #3)
                        AT4=    20              ;DEVICE 4 (BIT #4)
                        AT5=    40              ;DEVICE 5 (BIT #5)
                        AT6=    100             ;DEVICE 6 (BIT #6)
                        AT7=    200             ;DEVICE 7 (BIT #7)



                        ;DESIRED SECTOR/TRACK ADDRESS REGISTER (RHDST) (#1)
                        ;EACH BIT IS CALLED BY BIT NUMBER



                        ;DRIVE TYPE REGISTER (RHDT) (#06)
                        ;EACH BIT IS CALLED BY BIT NUMBER



                        ;LOOK-AHEAD REGISTER (RHLA) (#07)

000001                  EXT1=   1               ;EXTENSION 1 (BIT #0)
000002                  EXT2=   2               ;EXTENSION 2 (BIT #1)
000004                  EXT4=   4               ;EXTENSION 3 (BIT #2)
000010                  EXT10=  10              ;EXTENSION 4 (BIT #3)
000020                  EXT20=  20              ;EXTENSION 5 (BIT #4)
000040                  EXT40=  40              ;EXTENSION 6 (BIT #5)
000100                  SC1=    100             ;SECTOR COUNT FIELD 0 (BIT #6)
000200                  SC2=    200             ;SECTOR COUNT FIELD 1 (BIT #7)
000400                  SC4=    400             ;SECTOR COUNT FIELD 2 (BIT #8)
001000                  SC10=   1000            ;SECTOR COUNT FIELD 3 (BIT #9)
002000                  SC20=   2000            ;SECTOR COUNT FIELD 4 (BIT #10)
004000                  TRK1=   4000            ;TRACK FIELD 1 (BIT #11)
010000                  TRK2=   10000           ;TRACK FIELD 2 (BIT #12)
020000                  TRK4=   20000           ;TRACK FIELD 3 (BIT #13)
040000                  TRK10=  40000           ;TRACK FIELD 4 (BIT #14)
```

```
2426    100000          TRK20=  100000                  ;TRACK FIELD 5 (BIT #15)
2427
2428                    ;ERROR REGISTER #2 (RHER2) (#10)
2429
2430    000001          WCU=    1                       ;WRITE CURRENT UNSAFE (BIT #0)
2431    000002          CSF=    2                       ;CURRENT SINK FAILURE (BIT #1)
2432    000004          WSU=    4                       ;WRITE SELECT UNSAFE (BIT #2)
2433    000010          CSU=    10                      ;CURRENT SWITCH UNSAFE (BIT #3)
2434    000020          MSE=    20                      ;MOTOR SEQUENCE ERROR (BIT #4)
2435    000040          TDF=    40                      ;TRANSITIONS DETECTOR FAILURE (BIT #5)
2436    000100          TUF=    100                     ;TRANSITIONS UNSAFE (BIT #6)
2437    000200          FEN=    200                     ;FAILSAFE ENABLED (BIT #7)
2438    000400          WRU=    400                     ;WRITE READY UNSAFE (BIT #8)
2439    001000          MHS=    1000                    ;MULTIPLE HEAD SELECT (BIT #9)
2440    002000          NHS=    2000                    ;NO HEAD SELECTION (BIT #10)
2441    004000          IXE=    4000                    ;INDEX ERROR (BIT #11)
2442    010000          VU30=   10000                   ;30VOLT UNSAFE (BIT #12)
2443    020000          PLU=    20000                   ;PLO UNSAFE (BIT #13)
2444    100000          ACU=    100000                  ;ACUNSAFE (BIT #15)
2445
2446                    ;OFFSET REGISTER (RHOF) (#11)
2447
2448    000001          OF25=   1                       ;OFFSET 25 MICRO INCHES (BIT #0)
2449    000002          OF50=   2                       ;OFFSET 50 MICRO INCHES (BIT #1)
2450    000004          OF100=  4                       ;OFFSET 100 MICRO INCHES (BIT #2)
2451    000010          OF200=  10                      ;OFFSET 200 MICRO INCHES (BIT #3)
2452    000020          OF400=  20                      ;OFFSET 400 MICRO INCHES (BIT #4)
2453    000040          OF800=  40                      ;OFFSET 800 MICRO INCHES (BIT #5)
2454
2455    000200          OFREV=  200                     ;OFFSET NEGATIVE (REVERSE) (BIT #7)
2456    002000          HCI=    2000                    ;HEADER COMPARE INHIBIT (BIT #10)
2457    004000          ECI=    4000                    ;ERROR CORRECTION CODE INHIBIT (BIT #11)
2458    010000          FMT22=  10000                   ;FORMAT BIT (BIT #12
2459
2460
2461
2462
2463
2464                    ;DESIRED CYLINDER ADDRESS (RHCA) (#12)
2465                    ;EACH BIT IS CALLED BY BIT NUMBER.
2466
2467
2468
2469
2470
2471                    ;CURRENT CYLINDER ADDRESS (RHCC) (#13)
2472                    ;EACH BIT IS CALLED BY BIT NUMBER
2473
2474
2475
2476
2477
2478                    ;SERIAL NUMBER REGISTER (RHSN) (#14)
2479                    ;EACH IS CALLED BY BIT NUMBER
2480
2481
```

```
2482
2483
2484
2485                            ;FRROR REGISTER #03 (RHER3) (#15)
2486
2487    000001                  PSU=    1               ;PACK SPEED UNSAFE (BIT #0)
2488    000002                  VUF=    2               ;VELOCITY UNSAFE (BIT #1)
2489    000010                  UWR=    10              ;ANY UNSAFE EXCEPT READ/WRITE (BIT #3)
2490    000020                  PRE=    20              ;DISK PACK ROTATION ERROR (BIT #4)
2491    000040                  ACL=    40              ;AC LOW (BIT #5)
2492    000100                  DCL=    100             ;DC LOW (BIT #6)
2493    040000                  SKI=    40000           ;SEEK INCOMPLETE (BIT #14)
2494    100000                  OCYL=   100000          ;OFF CYLINDER (BIT #15)
2495
2496                            ;ECC POSITION REGISTER (RHEC1) (#16)
2497                            ;EACH BIT IS CALLED BY BIT NUMBER
2498
2499
2500
2501
2502
2503                            ;ECC PATTERN REGISTER (RHEC2) (#17)
2504                            ;EACH BIT IS CALLED BY BIT NUMBER
2505
```

```
2506
2507
2508
2509                                    .SBTTL  REGISTER ADDRESSES
2510
2511
2512
2513
2514                                    ;RP04/5/6 VECTOR ADDRESS
2515
2516  014744  000254                    RPVEC:  254                     ;RP04/5/6 VECTOR ADDRESS
2517
2518
2519
2520
2521                                    ;NOTE:  THE CONTENTS OF THESE LOCATIONS WILL BE DIFFRENT
2522                                    ;       IF THE ''CHANGE BASE ADDRESS'' ROUTINE IS USED.
2523                                    ;       THIS ROUTINE STARTS AT LOCATION TAGGED 'BASECH''
2524
2525  014746  176722                    RHDB:   176722                  ;DATA BUFFER
2526  014750  176702                    RHWC:   176702                  ;WORD COUNT
2527  014752  176704                    RHBA:   176704                  ;BUS ADDRESS
2528  014754  176710                    RHCS2:  176710                  ;CONTROL AND STATUS
2529  014756  176700                    RHCS1:  176700                  ;CONTROL AND STATUS 1 SEE NOTE ABOVE
2530  014760  176714                    RHER1:  176714                  ;ERROR #1 SEE NOTE ABOVE
2531  014762  176706                    RHDST:  176706                  ;DESIRED SECTOR / TRACK ADDRESS
2532  014764  176740                    RHER2:  176740                  ;ERROR #2
2533  014766  176732                    RHOF:   176732                  ;OFFSET
2534  014770  176734                    RHCA:   176734                  ;DESIRED CYLINDER ADDRESS
2535  014772  176742                    RHER3:  176742                  ;ERROR #3
2536  014774  176716                    RHAS:   176716                  ;ATTENTION SUMMARY SEE NOTE ABOVE
2537  014776  176724                    RHMR:   176724                  ;MAINTAINABILITY
2538  015000  176712                    RHDS1:  176712                  ;DRIVE STATUS
2539  015002  176726                    RHDT:   176726                  ;DRIVE TYPE
2540  015004  176730                    RHSN:   176730                  ;SERIAL NUMBER SEE NOTE ABOVE
2541  015006  176744                    RHEC1:  176744                  ;ECC POSITION
2542  015010  176746                    RHEC2:  176746                  ;ECC PATTERN
2543  015012  176720                    RHLA:   176720                  ;LOOK AHEAD
2544  015014  176736                    RHCC:   176736                  ;CURRENT CYLINDER ADDRESS
2545
2546                                    ;ADDITIONAL REGISTERS LOCATED IN THE RH70 CONTROLLER
2547
2548  015016  176752                    RHCS3:  176752                  ;CONTROL AND STATUS REG #3
2549  015020  176750                    RHBAE:  176750                  ;BUS ADDRESS EXTENSION REGISTER
```

```
2550
2551                            ;THE FOLLOWING LOCATIONS ARE RESERVED FOR REGISTER SAVES
2552                            ;ANY TIME THERE IS AN ERROR ALL THESE WILL BE FILLED
2553                            ;ONLY SOME MAY BE PRINTED BUT ALL WILL BE FILLED TRUE
2554                            ;FOR THE TIME JUST AFTER THE 'ERROR'' ERROR COMMAND
2555
2556  015022  000000            DB:     0               ;DATA BUFFER
2557  015024  000000            WC:     0               ;WORD COUNT
2558  015026  000000            BA:     0               ;BUS ADDRESS
2559  015030  000000            CS2:    0               ;CONTROL AND STATUS 2
2560
2561
2562  015032  000000            CS1:    0               ;CONTROL AND STATUS 1
2563  015034  000000            ER1:    0               ;ERROR #1
2564  015036  000000            DST:    0               ;DESIRED SECTOR/TRACK ADDRESS
2565  015040  000000            ER2:    0               ;ERROR #2
2566  015042  000000            OF:     0               ;OFFSET
2567  015044  000000            CA:     0               ;DESIRED CYLINDER ADDRESS
2568  015046  000000            ER3:    0               ;ERROR #3
2569  015050  000000            AS:     0               ;ATTENTION SUMMARY
2570  015052  000000            MR:     0               ;MAINTAINABILITY
2571  015054  000000            DS1:    0               ;DRIVE STATUS
2572  015056  000000            DT:     0               ;DRIVE TYPE
2573  015060  000000            SN:     0               ;SERIAL NUMBER
2574  015062  000000            EC1:    0               ;ECC POSITION
2575  015064  000000            EC2:    0               ;ECC PATTERN
2576  015066  000000            LA:     0               ;LOOK-AHEAD
2577  015070  000000            CC:     0               ;CURRENT CYLINDER ADDRESS
2578
2579
```

```
2580                                    ;FLAGS AND INTERNAL PROGRAM CONTROL WORDS
2581
2582
2583
2584  015072  000010                    UNITS:  .BLKW   8.              ;THIS IS FILLED WITH -1
2585  015112  000000                    UNIT:   .WORD   0               ;UNIT UNDER TEST
2586  015114  000000                    NOUNIT: .WORD   0               ;NUMBER OF UNITS PRESENT
2587                                                                    ;USED TO KEEP TRACK OF UNIT UNDER TEST
2588  015116  000000                    NUNIT:  .WORD   0               ;USED TO DETERMIN IF THERE ARE MORE
2589                                                                    ;THAN ONE UNIT
2590  015120  000000                    SELECT: .WORD   0               ;ALL ONES INDICATE UNIT TO BE SELECTED
2591  015122  000000                    UNITSL: .WORD   0               ;UNIT NO. SELECTED
2592
2593  015124  000000                    ERFLG$: 0                       ;ERROR FLAG
2594
2595  015126  000000                    SAVDT:  0                       ;SAVE DRIVE TYPE REGISTER
2596                                                                    ;FOR COMPARISON IN DRIVE CLEAR TEST
2597                                                                    ;AND RH INIT TEST
2598  015130  000000                    SAVSN:  0                       ;SAVE SERIAL NUMBER REGISTER
2599                                                                    ;FOR COMPARISON IN DRIVE CLEAR TEST
2600                                                                    ;AND RH INIT TEST
2601
2602  015132  000000                    PCJSR:  0                       ;SAVE PC OF JSR WHICH GAVE THE ERROR
2603
2604  015134  000000                    ATTENT: 0                       ;ATTENTION BIT FOR PRESENT UNIT
2605  015136  000000                    TOTALAT:        0               ;TOTAL ATTENTION BITS
2606
2607  015140  000000                    TMPILL: 0                       ;TEMPORARY ILLEGAL FUNCTION
2608
2609  015142  000000                    TSECC:  0                       ;FLAG TO SAY IF ECC TEST OR NOT
2610                                                                    ;WHEN =177777 IT IS AN ECC TEST
2611                                                                    ;WHEN =0IT IS NOT AN ECC TEST
2612
2613  015144  000000                    TESDTE: 0                       ;FLAG TO SAY IF DRIVE TIMING ERROR OR NOT
2614                                                                    ;WHEN = 177777 IT IS A DTE TEST
2615                                                                    ;WHEN = 0      IT IS NOT A DTE TEST
2616
2617  015146  000000                    TAGDTE: 0                       ;TEMPORARY TAG USED IN DRIVE TIMING
2618                                                                    ;ERROR TEST
2619
```

```
2620
2621
2622                                    ;FUNCTION EQUATES
2623
2624                                    ;TABLE OF COMMAND FUNCTIONS FOR RHCS1
2625                                    ;THEN ''GO'' BIT HAS TO BE SET
2626
2627  015150                            FUTABL:
2628  015150  000000                    NOPERA: 0                       ;NO OPERATION
2629  015152  000002                    UNLOAD: 2                       ;UNLOAD (STAND BY)
2630  015154  000006                    RECALI: 6                       ;RECALIBRATE
2631  015156  000010                    DCLEAR: 10                      ;DRIVE CLEAR
2632  015160  000012                    RELEAS: 12                      ;RELEASE (DUAL-PORT OPERATION)
2633  015162  000030                    SERCH:  30                      ;SEARCH COMMAND
2634  015164  000050                    WRCHEK: 50                      ;WRITE CHECK DATA
2635  015166  000052                    WRCHDT: 52                      ;WRITE CHECK HEADER AND DATA
2636  015170  000060                    WRIDAT: 60                      ;WRITE DATA
2637  015172  000062                    WRIFOR: 62                      ;WRITE HEADER AND DATA (FORMAT)
2638  015174  000070                    READAT: 70                      ;READ DATA
2639  015176  000072                    REFOR:  72                      ;READ HEADER AND DATA
2640  015200  000004                    SEECOM: 4                       ;SEEK COMMAND
2641  015202  000014                    OFSETC: 14                      ;OFFSET COMMAND
2642  015204  000016                    RETCL:  16                      ;RETURNTO CENTERLINE
2643  015206  000022                    PKACK:  22                      ;PACK ACKNOWLEDGE
2644  015210  000020                    READIN: 20                      ;READ IN
2645  015212  000000                    ILLEGL: .WORD                   ;COMPUTED ILLEGAL FUNCTION
2646
2647
2648
2649                                    ;DATA BUFFER FOR READ WRITE
2650
2651
2652  015220  000422                    WRFROM: .BLKW   274.            ;WRITE FROM THIS BUFFER
2653  016264  000422                    REINTO: .BLKW   274.            ;READ INTO THIS BUFFER
2654
2655
2656  017330  000000                    TSTNM:  0                       ;TEST NUMBER
2657  017332  000000                    FIRST:  0                       ;IF ZERO WILL TYPE HEADER
2658                                                                    ;IF ONES WILL NOT TYPE HEADER
2659
2660  017334  000000                    RH70:   0                       ;FLAG = 1 FOR RH70 CONTROLLER
2661                                                                    ;FLAG = 0 FOR RH11
2662
2663
2664
2665
2666                                    ;TABLE FOR ATTENTION BITS
2667                                    ;ATTENTION TABLE
2668
2669  017336    001     002     004   ATABLE: .BYTE   1,2,4,10,20,40,100,200
2670  017341    010     020     040
2671  017344    100     200
```

```
                                                .SBTTL
                                                .SBTTL  ***PROGRAM SETUP & SETUP TESTS***
                                                .SBTTL

017346  012737  177777  015120  BEGIN2: MOV     #-1,@#SELECT    ;SELECT UNIT
017354  000402                  BR      START
017356  005037  015120          BEGIN:  CLR     @#SELECT        ;DO NOT SELECT UNIT
                                                                ;NORMAL RUN

017362                          START:
017362  000005                          RESET
017364  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
017370  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
017372  022706  001140                  CMP     #SWR,R6 ;;DONE?
017376  001374                          BNE     .-6             ;;LOOP BACK IF NO
017400  012706  001000                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
017404  012737  057374  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
017412  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
017420  012737  061612  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
017426  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
017434  012737  062362  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
017442  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
017450  012737  062452  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
017456  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
017464  005037  001212                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
017470  005037  001214                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
017474  112737  000001  001115          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
017502  012737  017502  001106          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
017510  012737  017510  001110          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
017516  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
017522  012737  017556  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
017530  012737  177570  001140          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
017536  012737  177570  001142          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
017544  022777  177777  161366          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
017552  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
017554  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
017556  012716  017564          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
017562  000002                          RTI
017564  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
017572  012737  000174  001142          MOV     #DISPREG,DISPLAY
017600  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR


017604  012737  000000  177776  STARTA: MOV     #0,PS           ;SET PROCESSOR STATUS TO 0
017612  012777  057306  175124          MOV     #RPVECT,@RPVEC  ;THIS IS FOR UNTIMELY DRIVE INTERRUPTS
017620  004737  060352                  JSR     PC,@#$TKINT     ;INITIALIZE TTY KEYBOARD
017624  005737  017332                  TST     @#FIRST         ;IS THIS FIRST TIME ROUND ?
017630  001001                          BNE     1$              ;DON'T TYPE HEADER IF NOT
017632  000402                          BR      2$              ;TYPE HEADER IF SO

017634  000137  020662          1$:     JMP     @#SND1

017640                          2$:
017640  104401  017646                  TYPE    ,65$            ;;TYPE ASCIZ STRING
```

```
2728  017644  000435                          BR      64$             ;;GET OVER THE ASCIZ
2729
2730  017740  104401  017746                  TYPE    ,67$            ;;TYPE ASCIZ STRING
2731  017744  000417                          BR      66$             ;;GET OVER THE ASCIZ
2732
2733  020004  104401  020012                  TYPE    ,69$            ;;TYPE ASCIZ STRING
2734  020010  000433                          BR      68$             ;;GET OVER THE ASCIZ
2735  020100  104401  020106                  TYPE    ,71$            ;;TYPE ASCIZ STRING
2736  020104  000433                          BR      70$             ;;GET OVER THE ASCIZ
2737  020174  104401  020202                  TYPE    ,73$            ;;TYPE ASCIZ STRING
2738  020200  000436                          BR      72$             ;;GET OVER THE ASCIZ
2739  020276  104401  020304                  TYPE    ,75$            ;;TYPE ASCIZ STRING
2740  020302  000431                          BR      74$             ;;GET OVER THE ASCIZ
2741  020366  104401  020374                  TYPE    ,77$            ;;TYPE ASCIZ STRING
2742  020372  000427                          BR      76$             ;;GET OVER THE ASCIZ
2743  020452  104401  020460                  TYPE    ,79$            ;;TYPE ASCIZ STRING
2744  020456  000427                          BR      78$             ;;GET OVER THE ASCIZ
2745  020536  104401  020544                  TYPE    ,81$            ;;TYPE ASCIZ STRING
2746  020542  000415                          BR      80$             ;;GET OVER THE ASCIZ
2747  020576  104401  020604                  TYPE    ,83$            ;;TYPE ASCIZ STRING
2748  020602  000427                          BR      82$             ;;GET OVER THE ASCIZ
2749
2750  020662  012737  177777  017332  SND1:   MOV     #-1,@#FIRST             ;NEXT TIME DO NOT GIVE HEADER
2751
2752  020670  005737  000042                  TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
2753  020674  001006                          BNE     64$             ;;BRANCH IF YES
2754  020676  023727  001140  000176          CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
2755  020704  001005                          BNE     65$             ;;BRANCH IF NO
2756  020706  104406                          GTSWR                   ;;GET SOFT-SWR SETTINGS
2757  020710  000403                          BR      65$
2758  020712  112737  000001  001134  64$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
2759
2760  020720  032777  010000  160212  RH70CK: BIT     #SW12,@SWR              ;LOOK TO SEE IF USING RH70
2761  020726  001403                          BEQ     3$              ;IF SW12 = 0, SKIP NEXT
2762  020730  012737  000001  017334          MOV     #1,@#RH70               ;IF SW12 = 1, CU IS AN RH70
2763
2764  020736  005737  015120          3$:     TST     @#SELECT                ;200 START?
2765  020742  001434                          BEQ     TST1    ;GO TO FIRST TEST IF STARTING FROM 200 ------)
2766  020744  104401  020752                  TYPE    ,65$            ;;TYPE ASCIZ STRING
2767  020750  000422                          BR      64$             ;;GET OVER THE ASCIZ
2768  021016  104412                          RDOCT
2769  021020  042716  177770                  BIC     #177770,(SP)            ;ONLY KEEP LAST 3 BITS
2770  021024  011637  015112                  MOV     (SP),@#UNIT             ;SAVE UNIT TO BE TESTED
2771  021030  012637  015122                  MOV     (SP)+,@#UNITSL          ;SAVE UNIT TO BE TESTED
2772
2773
2774
```

```
2775
2776  021034  000004                    TST1:   SCOPE
2777  021036  012737  000001  001212            MOV     #1,$TIMES       ;;DO 1 ITERATION
2778  021044  012706  001000                    MOV     #STACK, SP      ;SET UP STACK POINTER
2779  021050  012737  000001  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
2780  021056  012737  061622  000030            MOV     #REGSA1,@#EMTVEC;ERROR VECTOR SO THAT
2781                                                                    ;NO REGISTERS ARE SAVED
2782  021064  012737  021112  000004            MOV     #2$,@#ERRVEC    ;SET UP FOR BUS TIMEOUT
2783  021072  012700  000024                    MOV     #24,R0          ;THERE ARE 24 REG TO TEST
2784  021076  012701  014746                    MOV     #RHDB,R1        ;R1 NOW HAS ADDR OF ADDR OF FIRST REG.
2785  021102  013102                    1$:     MOV     @(R1)+,R2       ;READ HARDWARE REG.
2786  021104  005300                            DEC     R0              ;COUNT DOWN
2787  021106  001375                            BNE     1$              ;BRANCH IF 24 NOT DONE
2788  021110  000454                            BR      3$              ;BRANCH IF 24 DONE
2789  021112  012737  000006  000004    2$:     MOV     #ERRVEC+2,@#ERRVEC ;RESTORE TRAP CATCHER
2790  021120  022626                            CMP     (SP)+,(SP)+     ;CLEAN STACK
2791  021122  016137  177776  001200            MOV     -2(R1), $TMP1   ;STORE FAILING REG ADDR
2792  021130  104015                            ERROR   15              ;REGISTER NON EXISTANT
2793  021132  032777  020000  160000            BIT     #SW13,@SWR      ;INHIBIT ERROR PRINTOUT ?
2794  021140  001036                            BNE     4$              ;BRANCH IF YES
2795  021142  104401  021150                    TYPE    ,65$            ;;TYPE ASCIZ STRING
2796  021146  000427                            BR      64$             ;;GET OVER THE ASCIZ
2797  021226  012746  000204                    MOV     #ADDMOD,-(SP)   ;GET READY TO TYPE STARTING ADDRESS
2798
2799                                                                    ;OF ''CHANGE OF BASE ADDRESS'' ROUTINE
2800  021232  104402                            TYPOC
2801  021234  000000                            HALT                    ;STOP TO FORCE THE RESTART!
2802  021236  000137  044564            4$:     JMP     @#$EOP          ;GO TO END OF PROGRAM ---------------->
2803
2804  021242  012737  021322  000004    3$:     MOV     #TRP,@#4        ;INITIALIZE VECTOR
2805  021250  005037  017334                    CLR     RH70            ;INIT RH INDICATOR ++ C.W
2806  021254  005777  173540                    TST     @RHBAE          ;ADDRESS RPBAE (RH11/RH70?)
2807  021260  005237  017334                    INC     RH70            ;FOUND AN RH70
2808  021264  104401  021272                    TYPE    ,67$            ;;TYPE ASCIZ STRING
2809  021270  000413                            BR      66$             ;;GET OVER THE ASCIZ
2810  021320  000420                            BR      RTN             ;SET MASK AND GET OUT
2811  021322  005726                    TRP:    TST     (SP)+           ;ADJUST THE STACK
2812  021324  005726                            TST     (SP)+
2813  021326  104401  021334                    TYPE    ,65$            ;;TYPE ASCIZ STRING
2814  021332  000413                            BR      64$             ;;GET OVER THE ASCIZ
2815  021362  012737  061612  000030    RTN:    MOV     #$ERROR,@#EMTVEC;RESTORE ERROR VECTOR
2816                                                                    ;SO THAT REGISTERS ARE SAVED
2817  021370  012737  000006  000004            MOV     #ERRVEC+2,@#ERRVEC ;RESTORE TRAP CATCHER
2818
2819
2820
```

```
2821
2822  021376  000004                    TST2:   SCOPE
2823  021400  012737  000001  001212            MOV     #1,$TIMES       ;;DO 1 ITERATION
2824  021406  012706  001000                    MOV     #STACK,SP       ;RESET STACK
2825  021412  012737  000002  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
2826  021420  005737  017334                    TST     @#RH70          ;TEST FOR RH70 CONTROLLER
2827  021424  001402                            BEQ     30$             ;IF FLAG = 1, SKIP THIS TEST
2828  021426  000137  021454                    JMP     TST3            ;JUMP TO NEXT TEST ----------)
2829
2830  021432  013737  014754  021446            MOV     @#RHCS2,@#UN+2
2831  021440  004537  045536                    JSR     R5,@#BITST      ;TEST BITS IN REGISTER
2832  021444  020017                    UN:     20017                   ;ONLY THESE BITS ARE TEST READ/WRITE
2833  021446  000000                            .WORD   0               ;ADDRESS OF REG. BEING TESTED
2834  021450  104001                            ERROR   1               ;IN CORRECT DATA RECEIVED
2835  021452  000207                            RTS     PC              ;RETURN TO BLT3 ROUTINE
2836
2837
2838
2839
2840
2841
2842  021454  000004                    TST3:   SCOPE
2843  021456  012737  000001  001212            MOV     #1,$TIMES       ;;DO 1 ITERATION
2844  021464  012737  000003  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
2845
2846  021472  013701  014774                    MOV     @#RHAS,R1       ;R1 HAS ADDRESS OF RHAS
2847  021476  012711  177777                    MOV     #-1,@R1         ;THIS CLEARS RHAS (SURPRISED!)
2848  021502  011137  001126                    MOV     @R1,@#$BDDAT    ;TEST DATA
2849  021506  105737  001126                    TSTB    @#$BDDAT
2850  021512  001405                            BEQ     TST4            ;BRANCH IF GOOD
2851  021514  005037  001124                    CLR     @#$GDDAT        ;GOOD DATA
2852  021520  010137  045534                    MOV     R1,@#REGADR     ;FAILING REG. RHAS
2853  021524  104001                            ERROR   1               ;RHAS DOES NOT CLEAR
2854                                                                    ;WITH ONES MOVED INTO IT
2855
```

```
2856
2857  021526  000004                    TST4:   SCOPE
2858  021530  012737  000001  001212            MOV     #1,$TIMES       ;;DO 1 ITERATION
2859  021536  000005                            RESET                   ;START WITH AN INIT
2860  021540  004737  060352                    JSR     PC,@#$TKINT     ;INITILIZE TTY KEYBOARD
2861
2862  021544  032777  020000  157366            BIT     #SW13,@SWR      ;INHIBIT ERROR TYPEOUT ?
2863  021552  001026                            BNE     4$              ;SKIP NEXT IF SO
2864  021554  104401  021562                    TYPE    ,65$            ;;TYPE ASCIZ STRING
2865  021560  000423                            BR      64$             ;;GET OVER THE ASCIZ
2866  021630  013701  014774            4$:     MOV     @#RHAS,R1       ;R1 HAS ADDR. OF RHAS
2867  021634  013702  014754                    MOV     @#RHCS2,R2      ;R2 HAS ADDR. OF RHCS2
2868  021640  005012                            CLR     @R2             ;CLEAR RHCS2
2869  021642  012700  000010                    MOV     #8.,R0          ;COUNT
2870  021646  013704  014760                    MOV     @#RHER1,R4      ;R4 HAS ADDR. OF RHER1
2871
2872  021652  012714  177777            1$:     MOV     #-1,@R4         ;MOVE ERRORS INTO RHER1
2873  021656  005212                            INC     @R2             ;INCREMENT UNIT NO.
2874  021660  005300                            DEC     R0              ;COUNT
2875  021662  001373                            BNE     1$              ;BRANCH IF 8 NOT DONE
2876  021664  111137  015136                    MOVB    @R1,@#TOTALAT   ;SAVE TOTAL ATTENTION
2877                                                                    ;USED IN DRIVE CLEAR TEST
2878  021670  105037  015137                    CLRB    @#TOTALAT+1     ;CLEAR UPPER BYTE
2879  021674  105711                            TSTB    @R1             ;TEST FOR ANY DRIVES PRESENT
2880  021676  001402                            BEQ     2$              ;NONE RESPONDING - TYPE THE MESSAGE
2881  021700  000137  022270                    JMP     XE2             ;SOME THERE - GO FILL 'UNITS'' TABLE
2882
2883  021704  032777  020000  157226    2$:     BIT     #SW13,@SWR      ;INHIBIT ERROR TYPE OUT?
2884  021712  001402                            BEQ     3$              ;'NO DRIVES'' MESSAGE IF NO
2885  021714  000137  022626                    JMP     SELTST          ;CHECK FOR SELECTED UNIT START AND LOAD
2886                                                                    ;'UNITS'' TABLE WITH DESIRED DRIVE IF SO
2887
2888  021720                            3$:
2889  021720  104401  021726                    TYPE    ,67$            ;;TYPE ASCIZ STRING
2890  021724  000412                            BR      66$             ;;GET OVER THE ASCIZ
2891  021752  104401  021760                    TYPE    ,69$            ;;TYPE ASCIZ STRING
2892  021756  000436                            BR      68$             ;;GET OVER THE ASCIZ
2893  022054  104401  022062                    TYPE    ,71$            ;;TYPE ASCIZ STRING
2894  022060  000441                            BR      70$             ;;GET OVER THE ASCIZ
2895  022164  104401  022172                    TYPE    ,73$            ;;TYPE ASCIZ STRING
2896  022170  000435                            BR      72$             ;;GET OVER THE ASCIZ
2897
2898  022264  000137  044564                    JMP     @#$EOP          ;GO OUT---------------------------->
2899
2900
2901                                            ;*SET UP UNITS TABLE
2902
2903  022270                            XE2:
2904  022270  012700  000010            2$:     MOV     #8.,R0          ;COUNTER
2905  022274  012703  015072                    MOV     #UNITS,R3       ;POINTER
2906  022300  012723  177777            3$:     MOV     #-1,(R3)+       ;PRESET BLOCK TO ALL ONES
2907  022304  005300                            DEC     R0              ;COUNT
2908  022306  001374                            BNE     3$              ;BRANCH IF 8 NOT DONE
2909  022310  012703  015072                    MOV     #UNITS,R3       ;POINTER
2910  022314  005005                            CLR     R5
2911  022316  005037  015114                    CLR     @#NOUNIT        ;NO. OF UNITS PRESENT
```

```
2912  022322  012700  000010              MOV     #8.,R0              ;COUNTER
2913  022326  011137  001176              MOV     @R1,@#$TMP0         ;TEMPORARY STORAGE
2914  022332  006037  001176        4$:   ROR     @#$TMP0             ;SET CARRY IF ONE IN  0 BIT
2915
2916  022336  103120                    BCC     5$
2917  022340  010577  172410              MOV     R5,@RHCS2           ;INSERT UNIT NUMBER
2918  022344  022777  024020  172430      CMP     #24020,@RHDT        ;IS THIS A DUAL PORT RP04 ?
2919  022352  001503                      BEQ     6$                  ;TYPE DRIVE NO. IF SO
2920  022354  022777  020020  172420      CMP     #20020,@RHDT        ;IS THIS A SINGLE PORT RP04 ?
2921  022362  001477                      BEQ     6$                  ;TYPE DRIVE NO.IF YES
2922
2923
2924  022364  022777  024021  172410      CMP     #24021,@RHDT        ;IS THIS A DUAL PORT RP05 ?
2925  022372  001473                      BEQ     6$                  ;TYPE UNIT NO. OUT
2926  022374  022777  020021  172400      CMP     #20021,@RHDT        ;IS THIS A SINGLE PORT RP05 ?
2927  022402  001467                      BEQ      6$                 ;TYPE UNIT NO. IF SO
2928
2929  022404  022777  024022  172370      CMP     #24022,@RHDT        ;IS THIS A DUAL PORT RP06 ?
2930  022412  001463                      BEQ     6$                  ;TYPE THE NO IF SO
2931  022414  022777  020022  172360      CMP     #20022,@RHDT        ;IS THIS A SINGLE PORT RP06 ?
2932  022422  001457                      BEQ     6$                  ;TYPE THE NO IF SO
2933
2934
2935                                      ;*NO...IT'S NOT AN RP04/RP05/RP06 DEVICE
2936                                      ;*SO TYPE OUT THE DEVICE TYPE
2937
2938  022424  104401  022432              TYPE    ,65$                ;;TYPE ASCIZ STRING
2939  022430  000410                      BR      64$                 ;;GET OVER THE ASCIZ
2940  022452  010546                      MOV     R5,-(SP)            ;GET READY TO TYPE UNIT NUMBER
2941  022454  104405                      TYPDS
2942  022456  104401  022464              TYPE    ,67$                ;;TYPE ASCIZ STRING
2943  022462  000406                      BR      66$                 ;;GET OVER THE ASCIZ
2944  022500  017746  172276              MOV     @RHDT,-(SP)         ;GET READY TO TYPE RHDT
2945  022504  104402                      TYPOC
2946  022506  104401  022514              TYPE    ,69$                ;;TYPE ASCIZ STRING
2947  022512  000422                      BR      68$                 ;;GET OVER THE ASCIZ
2948  022560  000407                      BR      5$                  ;NO RP04/5/6 FOUND SO BRANCH
2949  022562  010523              6$:     MOV     R5,(R3)+
2950  022564  104401  001223              TYPE    ,$CRLF
2951  022570  010546                      MOV     R5,-(SP)            ;PUT DRIVE NO. ON STACK
2952  022572  104405                      TYPDS                       ;TYPE DRIVE NO.
2953  022574  005237  015114              INC     @#NOUNIT            ;INCR TOTAL NO. OF UNITS
2954
2955  022600  005205              5$:     INC     R5                  ;'RHCS2' UNIT ADDRESS
2956  022602  005300                      DEC     R0                  ;DRIVE COUNTER DOWN ONE
2957  022604  001252                      BNE     4$                  ;TEST AND DO NEXT UNIT IF 8 NOT DONE
2958
2959  022606  013737  015072  015112      MOV     @#UNITS,@#UNIT      ;SET UNIT NO. TO FIRST ONE FOUND/OR 0
2960  022614  013737  015114  015116      MOV     @#NOUNIT,@#NUNIT    ;SAVE NO. OF UNITS
2961  022622  005337  015116              DEC     @#NUNIT             ;IF NUNIT = 0 THEN ONLY ONE UNIT
2962                                                                  ;IF NUNIT > 0 THEN MORE THAN ONE UNIT
2963
2964  022626  005737  015120        SELTST: TST     @#SELECT            ;STARTING ADDRESS 200 ?
2965  022632  001403                      BEQ     TST5                ;BRANCH IF STARTING FROM 200
2966  022634  013737  015122  015112      MOV     @#UNITSL,@#UNIT     ;CHANGE UNIT NUMBER TO SELECTED ONE
2967
```

```
2968
2969  022642  000004                    TST5:   SCOPE
2970  022644  012737  000001  001212            MOV     #1,$TIMES       ;;DO 1 ITERATION
2971  022652  012737  023322  001106            MOV     #1$,$LPADR      ;;SET SCOPE LOOP ADDRESS
2972
2973  022660  004737  045770                    JSR     PC,@#CLDISK     ;FILL UNIT NO.
2974  022664  005037  015134                    CLR     @#ATTENT        ;CLEAR
2975
2976                                            ;*TEST FOR UNIT #0
2977
2978  022670  005737  015112                    TST     @#UNIT          ;IS UNIT #0 NEXT IN THE UNITS TABLE ?
2979  022674  001022                            BNE     10$             ;IF NOT, TEST THIS UNIT
2980  022676  012700  000041                    MOV     #41,R0          ;IF SO, CHECK THE LOAD MEDIA LOCATION
2981  022702  122710  000011                    CMPB    #11,(R0)        ;WAS IS AN RP04/5/6 ?
2982  022706  001015                            BNE     10$             ;NO... GO AHEAD AND TEST UNIT #0
2983  022710  005737  015120                    TST     @#SELECT        ;WAS UNIT #0 SELECTED ?
2984                                                                    ;(IE. WAS IT A 210 START ?)
2985  022714  001012                            BNE     10$             ;IF SO...TEST UNIT #0
2986
2987                                            ;*INCREMENT THE UNITS TABLE TO NEXT DRIVE (IF ANY)
2988                                            ;* & DECREMENT THE 'NOUNITS' PRESENT (TO BE TESTED)
2989
2990  022716  012700  015072                    MOV     #UNITS,R0       ;LOAD UNITS TABLE POINTER
2991  022722  005720                            TST     (R0)+           ;SELECT THE NEXT UNIT IN THE TABLE
2992                                                                    ;(DOUBLE INCREMENT THE POINTER)
2993  022724  022710  177777                    CMP     #-1,(R0)        ;IS THERE ANOTHER TABLE ENTRY PRESENT ?
2994  022730  001404                            BEQ     10$             ;IF NOT (LOC = -1) ...MUST USE UNIT #0
2995  022732  011037  015112                    MOV     (R0),@#UNIT     ;SET UP TO BE THE UNIT UNDER TEST
2996  022736  005337  015114                    DEC     @#NOUNITS       ;DECREMENT BECAUSE UNIT #0 WON'T BE TESTED
2997  022742  013700  015112            10$:    MOV     @#UNIT,R0       ;R0 CONTAINS THE UNIT UNDER TEST
2998
2999
3000
3001                                    ;       CLR     @#RP06          ;CLEAR RP06 DEVICE TYPE FLAG
3002                                    ;       CMP     #24022,@RHDT    ;DUAL PORT RP06 ?
3003                                    ;       BEQ     2$              ;YES...SET THE FLAG
3004                                    ;       CMP     #20022,@RHDT    ;SINGLE PORT RP06 ?
3005                                    ;       BEQ     2$              ;YES...SET FLAG
3006                                    ;       BR      3$              ;DON'T SET THE RP06 FLAG
3007                                    ;2$:    MOV     #-1,@#RP06      ;SET THE FLAG
3008
3009                                    ;3$:                            ;ASSUME THE NEXT UNIT IS AN RP04
3010
3011
3012  022746  116037  017336  015134            MOVB    ATABLE(R0),@#ATTENT  ;SET APPROPRIATE ATTENTION BIT
3013  022754  104401  022762                    TYPE    ,65$            ;;TYPE ASCIZ STRING
3014  022760  000414                            BR      64$             ;;GET OVER THE ASCIZ
3015  023012  013746  015112                    MOV     @#UNIT,-(SP)    ;UNIT NO. TO STACK
3016  023016  104405                            TYPDS                   ;TYPE  DRIVE NO.
3017  023020  104401  023026                    TYPE    ,67$            ;;TYPE ASCIZ STRING
3018  023024  000410                            BR      66$             ;;GET OVER THE ASCIZ
3019  023046  017746  171732                    MOV     @RHSN,-(SP)     ;;SAVE @RHSN FOR TYPEOUT
3020  023052  104402                            TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3021  023054  104401  023062                    TYPE    ,69$            ;;TYPE ASCIZ STRING
3022  023060  000410                            BR      68$             ;;GET OVER THE ASCIZ
3023  023102  017746  171674                    MOV     @RHDT,-(SP)     ;;SAVE @RHDT FOR TYPEOUT
```

```
3024  023106  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3025
3026
3027
3028
3029
3030  023110  022777  024020  171664          CMP     #24020,@RHDT    ;DUAL PORT RP04 ?
3031  023116  001425                          BEQ     4$              ;TYPE ASCII MSG OUT
3032  023120  022777  020020  171654          CMP     #20020,@RHDT    ;SINGLE PORT RP04 ?
3033  023126  001421                          BEQ     4$              ;TYPE THE MESSAGE
3034
3035  023130  022777  024021  171644          CMP     #24021,@RHDT    ;DUAL PORT RP05 ?
3036  023136  001434                          BEQ     5$              ;TYPE THE MESSAGE
3037  023140  022777  020021  171634          CMP     #20021,@RHDT    ;SINGLE PORT RP05 ?
3038  023146  001430                          BEQ     5$              ;TYPE THE MESSAGE
3039
3040  023150  022777  024022  171624          CMP     #24022,@RHDT    ;DUAL PORT RP06 ?
3041  023156  001443                          BEQ     6$              ;TYPE THE MESSAGE
3042  023160  022777  020022  171614          CMP     #20022,@RHDT    ;SINGLE PORT RP06 ?
3043  023166  001437                          BEQ     6$              ;TYPE IT OUT
3044  023170  000454                          BR      1$              ;DRIVE IS NOT RP04/5/6 - SO
3045                                                                  ;DO NOT TYPE ANY MESSAGE OUT
3046
3047                                                                  ;-SHOULD NEVER HAPPEN AT THIS POINT
3048                                                                  ;UNLESS DRIVE GOT SICK WHILE TESTING
3049                                                                  ;WAS IN PROGRESS
3050
3051  023172                          4$:
3052  023172  104401  023200                  TYPE    ,71$            ;;TYPE ASCIZ STRING
3053  023176  000413                          BR      70$             ;;GET OVER THE ASCIZ
3054  023226  000435                          BR      1$              ;SKIP NEXT ONES
3055  023230                          5$:
3056  023230  104401  023236                  TYPE    ,73$            ;;TYPE ASCIZ STRING
3057  023234  000413                          BR      72$             ;;GET OVER THE ASCIZ
3058  023264  000416                          BR      1$              ;SKIP NEXT
3059  023266                          6$:
3060  023266  104401  023274                  TYPE    ,75$            ;;TYPE ASCIZ STRING
3061  023272  000413                          BR      74$             ;;GET OVER THE ASCIZ
3062
3063  023322  005777  171456          1$:     TST     @RHSN           ;READ SERIAL NO. AND DRIVE TYPE
3064  023326  005777  171450                  TST     @RHDT           ;THESE TWO ARE TO HELP SCOPE LOOPS
3065  023332  017737  171446  015130          MOV     @RHSN,@#SAVSN   ;SAVE TO CHECK IF CLR RHCS2 BIT 5 CLEARS ANY BITS
3066  023340  017737  171436  015126          MOV     @RHDT,@#SAVDT   ;SAVE TO CHECK IF CLR RHCS2 BIT 5 CLEARS ANY BITS
```

```
3067
3068
3069  023346  000004                    TST6:   SCOPE
3070  023350  012737  000006  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3071  023356  004737  045770                    JSR     PC,@#CLDISK     ;GIVE INITILIZE
3072  023362  032713  010000                    BIT     #MOL,@R3        ;CHECK MOL IN RHDS1
3073  023366  001550                            BEQ     TST7            ;BRANCH IF MOL LOW
3074
3075  023370  104401  023376                    TYPE    ,65$            ;;TYPE ASCIZ STRING
3076  023374  000421                            BR      64$             ;;GET OVER THE ASCIZ
3077  023440  104401  023446                    TYPE    ,67$            ;;TYPE ASCIZ STRING
3078  023444  000424                            BR      66$             ;;GET OVER THE ASCIZ
3079  023516  104401  023524                    TYPE    ,69$            ;;TYPE ASCIZ STRING
3080  023522  000430                            BR      68$             ;;GET OVER THE ASCIZ
3081
3082  023604  032713  010000            1$:     BIT     #MOL,@R3        ;CHECK MOL IN RHDS1
3083  023610  001375                            BNE     1$              ;BRANCH IF MOL IS HIGH
3084  023612  104401  023620                    TYPE    ,71$            ;;TYPE ASCIZ STRING
3085  023616  000434                            BR      70$             ;;GET OVER THE ASCIZ
3086
3087
3088
```

```
3089
3090  023710  000004                    TST7:   SCOPE
3091  023712  012706  001000                    MOV     #STACK,SP       ;RESET STACK
3092  023716  012737  000007  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3093
3094  023724  004737  045770                    JSR     PC,@#CLDISK     ;INIT AND SET UP GENERAL REG.
3095                                                                    ;AND UNIT NUMBER
3096  023730  012777  000001  171040            MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
3097
3098  023736  013777  015206  171012            MOV     @#PKACK,@RHCS1  ;LOAD PACK ACKNOWLEDGE COMMAND INTO RHCS1
3099
3100                                            ;SAVE REGISTERS FOR COMPARISON AFTER GO
3101  023744  004037  046462                    JSR     R0,@#SAVER      ;SAVE
3102  023750  014750                            RHWC                    ;FROM
3103  023752  016264                            REINTO                  ;TO
3104  023754  000023                            19.                     ;NUMBER OF REGISTERS SAVED
3105
3106                                            ;GIVE GO TO PACK ACKNOWLEDGE COMMAND
3107  023756  052777  000001  170772            BIS     #GO,@RHCS1      ;GO TO PACK ACKNOWLEDGE COMMAND
3108
3109                                            ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
3110  023764  052737  000100  016314            BIS     #VV,@#REINTO+30 ;SAVED RHDS1
3111
3112                                            ;AFTER GO HAS BEEN GIVEN TO PACK ACKNOWLEDGE COMMAND
3113                                            ;SAVE REGISTERS AGAIN SO THAT COMPARISONS CAN
3114                                            ;BE DONE
3115  023772  004037  046462                    JSR     R0,@#SAVER      ;SAVE
3116  023776  014750                            RHWC                    ;FROM
3117  024000  015220                            WRFROM
3118  024002  000023                            19.                     ;NUMBER OF REGISTERS SAVED
3119
3120                                            ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
3121                                            ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
3122                                            ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
3123  024004  113737  016311  015245            MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
3124
3125
3126                                            ;COMPARE REGISTERS BEFORE PACK ACKNOWLEDGE COMMAND
3127                                            ;WITH AFTER GO
3128  024012  004037  046664                    JSR     R0,@#COMPAR     ;COMPARE
3129  024016  016264                            REINTO                  ;GOOD BUFFER
3130  024020  015220                            WRFROM                  ;TEST BUFFER
3131  024022  000023                            19.                     ;NUMBER
3132  024024  024032                            1$                      ;RETURN FOR ERROR
3133  024026  024032                            1$                      ;SAME
3134  024030  024052                            2$                      ;RETURN FOR GOOD COMPARISON
3135  024032  013705  052774            1$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
3136  024036  060505                            ADD     R5,R5           ;DOUBLE ERROR WORD
3137  024040  016537  014746  045534            MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
3138
3139  024046  104001                            ERROR   1               ;IMPROPER REGISTER CHANGE
3140                                                                    ;AFTER PACK ACKNOWLEDGE COMMAND
3141                                                                    ;WITH GO IS GIVEN
3142  024050  000207                            RTS     PC              ;RETURN TO COMPARISION
3143
3144  024052                            2$:
```

```
3145
3146  024052  000004                    TST10:  SCOPE
3147  024054  012706  001000                    MOV     #STACK,SP       ;RESET STACK
3148  024060  012737  000010  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3149  024066  004737  045770                    JSR     PC,@#CLDISK     ;INIT DRIVE
3150  024072  012777  000001  170676            MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
3151  024100  004037  050342                    JSR     R0,@#MAKECYL    ;SUBROUTINE TO GIVE A SEEK
3152  024104  000000                            0                       ;THIS SHUOLD CHANGE RHCC TO 0
3153
3154                                            .SBTTL
3155                                            .SBTTL  ***DIAGNOSTIC CODE***
3156                                            .SBTTL
```

```
3157            000004                  TIMOT=4
3158                                    .REM %
3159                                    THE FOLLOWING TESTS WILL ATTEMPT TO CATCH THOSE BUGS WHICH FAULT
3160                                    INSERTION HAS SHOWN US WE MISSED IN THE FIRST PART OF THIS TEST.
3161
3162                                     THIS TEST WILL ASCERTAIN THAT THE ALL RH11 REGISTERS WIIL
3163                                    RESPOND TO I.E. NOT CAUSE A NON-EXISTANT MEMORY ERROR WHEN ACCESSED
3164                                    BY A 'TST'' INSRUCTION.
3165                                    %
3166  024106  000004                    TST11:  SCOPE
3167  024110  012737  000001  001212            MOV     #1,$TIMES       ;;DO 1 ITERATION
3168  024116  012737  177777  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3169  024124  012777  000040  170622            MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3170
3171  024132  012705  014756                    MOV     #RHCS1,R5       ;R5=LIST POINTER.
3172
3173  024136  013737  000004  001176            MOV     @#TIMOT,$TMP0   ;SAVE TIMEOUT VECTOR.
3174  024144  012737  024202  000004            MOV     #E00,@#TIMOT    ;SET VECTOR TO E00
3175
3176  024152  012706  001000          L00:      MOV     #STACK,SP       ;SET STACK POINTER.
3177
3178  024156  011502                  I00:      MOV     (R5),R2         ;R2=RH11 ADDRESS.
3179  024160  010237  045534                    MOV     R2,REGADR
3180  024164  005712                            TST     (R2)            ;DOES THE RH11 REGISTER RESPOND?
3181
3182  024166  062705  000002                    ADD     #2,R5           ;UPDATE ADDRESS
3183  024172  020527  015010                    CMP     R5,#RHEC2       ;AT THE END OF LEGAL RH11 REGISTERS?
3184  024176  101767                            BLOS    I00             ;NOPE
3185  024200  000401                            BR      000             ;YES! GOTO 000.
3186
3187  024202  104042                  E00:      ERROR   42
3188
3189  024204  013737  001176  000004  000:      MOV     $TMP0,@#TIMOT   ;RESTORE TIMEOUT VECTOR.
```

```
3190
```

```
3191                                    .REM %
3192                                    THE FOLLOWING 6 TESTS WILL TEST THE BYTE LOGIC FOR THE RH11 REGISTERS
3193                                    NO ATTEMPT IS MADE TO DETERMINE IF ALL POSSIBLE DATA PATTERNS CAN BE
3194                                    LOADED, ONLY THAT THE RH11 HARDWARE WILL ALLOW THE PROGRAM TO SELECTIVLY
3195                                    MANIPULATE BYTES USING THE FOLLOWING COMMANDS:
3196
3197                                            1). MOVE BYTE BOTH TO AND FROM THE WORD COUNT REGISTER
3198
3199                                            2). BIT SET INTO THE WORD COUNT REGISTER.
3200
3201                                            3). BIT CLEAR INTO THE WORD COUNT REGISTER.
3202
3203                                    %
3204
3205  024212  000004                  TST12:  SCOPE
3206  024214  012737  000001  001212          MOV     #1,$TIMES       ;;DO 1 ITERATION
3207  024222  012737  000012  017330          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3208  024230  012777  000040  170516          MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3209
3210  024236  012706  001000                  MOV     #STACK,SP       ;SET STACK POINTER.
3211
3212  024242  013702  014750                  MOV     RHWC,R2         ;R2=RH11 WC ADDRESS.
3213  024246  010237  045534                  MOV     R2,REGADR       ;REGISTER ADDRESS TO REGADR FOR TYPING.
3214  024252  012737  000377  001124          MOV     #377,$GDDAT     ;$GDDAT=S/B.
3215
3216  024260  005012                  L02:    CLR     (R2)            ;CLEAR RH11 WC.
3217
3218  024262  112712  000377          I02:    MOVB    #377,(R2)       ;SET WC TO LO BYTE.
3219  024266  011237  001126                  MOV     (R2),$BDDAT     ;$BDDAT=ACTUAL WAS
3220
3221  024272  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS
3222  024300  001401                          BEQ     TST13           ;NEXT TEST
3223  024302  104044                          ERROR   44
3224
```

```
3225
3226  024304  000004                    TST13:  SCOPE
3227  024306  012737  000001  001212            MOV     #1,$TIMES       ;;DO 1 ITERATION
3228  024314  012737  000013  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3229  024322  012777  000040  170424            MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3230
3231  024330  012706  001000                    MOV     #STACK,SP       ;SET STACK POINTER.
3232
3233  024334  013702  014750                    MOV     RHWC,R2         ;R2=RH11 WC HI BYTE ADDRESS.
3234  024340  010237  045534                    MOV     R2,REGADR
3235  024344  012737  177400  001124            MOV     #177400,$GDDAT  ;$GDDAT=S/B.
3236
3237  024352  005012                    L03:    CLR     (R2)            ;CLEAR RH11 WC.
3238
3239  024354  005202                            INC     R2              ;R2=HI BYTE ADDRESS.
3240
3241  024356  112712  000377          I03:    MOVB    #377,(R2)       ;SET WC HI BYTE.
3242
3243  024362  005302                            DEC     R2              ;R2=FULL WORD ADDRESS.
3244
3245  024364  011237  001126                    MOV     (R2),$BDDAT     ;$BDDAT=ACTUAL.
3246
3247  024370  023737  001126  001124            CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS.
3248  024376  001401                            BEQ     TST14           ;NEXT TEST
3249  024400  104044                            ERROR   44
3250
3251
```

```
3252
3253  024402  000004                      TST14:  SCOPE
3254  024404  012737  000001  001212              MOV     #1,$TIMES       ;;DO 1 ITERATION
3255  024412  012737  000014  017330              MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3256  024420  012777  000040  170326              MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3257
3258  024426  012706  001000                      MOV     #STACK,SP       ;SET STACK POINTER.
3259
3260  024432  013702  014750                      MOV     RHWC,R2         ;R2=RH11 WC ADDRESS
3261  024436  010237  045534                      MOV     R2,REGADR
3262  024442  012737  000377  001124              MOV     #377,$GDDAT     ;$GDDAT=S/B.
3263
3264  024450  005012                      L04:    CLR     (R2)            ;CLEAR RH11 WC.
3265
3266  024452  012712  000252                      MOV     #252,(R2)       ;SET UP WC
3267  024456  152712  000125              I04:    BISB    #125,(R2)       ;DO A BISB
3268
3269  024462  011237  001126                      MOV     (R2),$BDDAT     ;$BDDAT=WAS.
3270
3271  024466  023737  001126  001124              CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS
3272  024474  001401                              BEQ     TST15           ;NEXT TEST
3273  024476  104045                              ERROR   45
3274
3275
```

```
3276
3277  024500  000004                    TST15:  SCOPE
3278  024502  012737  000001  001212            MOV     #1,$TIMES       ;;DO 1 ITERATION
3279  024510  012737  000015  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3280  024516  012777  000040  170230            MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3281
3282  024524  012706  001000                    MOV     #STACK,SP       ;SET STACK POINTER.
3283
3284  024530  013702  014750                    MOV     RHWC,R2         ;R2=RH11 WC ADDRESS.
3285  024534  010237  045534                    MOV     R2,REGADR
3286  024540  012737  177400  001124            MOV     #177400,$GDDAT  ;$GDDAT=S/B.
3287
3288  024546  005012                    L05:    CLR     (R2)            ;CLEAR RH11 WC.
3289
3290  024550  005202                            INC     R2              ;R2 =HI BYTE.
3291
3292  024552  112712  000125                    MOVB    #125,(R2)       ;SET UP RH11 WC.
3293  024556  152712  000252            I05:    BISB    #252,(R2)       ;DO A BISB.
3294
3295  024562  005302                            DEC     R2              ;R2=FULL WORD ADDRESS.
3296
3297  024564  011237  001126                    MOV     (R2),$BDDAT     ;$BDDAT=WAS.
3298
3299  024570  023737  001126  001124            CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS.
3300  024576  001401                            BEQ     TST16           ;NEXT TEST
3301  024600  104045                            ERROR   45
3302
3303
```

```
3304
3305  024602  000004                  TST16:  SCOPE
3306  024604  012737  000001  001212          MOV     #1,$TIMES       ;;DO 1 ITERATION
3307  024612  012737  000016  017330          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3308  024620  012777  000040  170126          MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3309
3310  024626  012706  001000                  MOV     #STACK,SP       ;SET STACK POINTER.
3311
3312  024632  013702  014750                  MOV     RHWC,R2         ;R2=RH11 WC ADDRESS.
3313  024636  010237  045534                  MOV     R2,REGADR
3314  024642  012737  000252  001124          MOV     #252,$GDDAT     ;$GDDAT=S/B.
3315
3316  024650  005012                  L06:    CLR     (R2)            ;CLEAR RH11 WC.
3317
3318  024652  012712  000377                  MOV     #377,(R2)       ;SET WC=0000377
3319
3320  024656  142712  000125          I06:    BICB    #125,(R2)       ;DO A BICB
3321
3322  024662  011237  001126                  MOV     (R2),$BDDAT     ;$BDDAT=WAS.
3323
3324  024666  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS.
3325  024674  001401                          BEQ     TST17           ;NEXT TEST
3326  024676  104046                          ERROR   46
3327
3328
```

```
3329
3330  024700  000004                    TST17:  SCOPE
3331  024702  012737  000001  001212            MOV     #1,$TIMES       ;;DO 1 ITERATION
3332  024710  012737  000017  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3333  024716  012777  000040  170030            MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3334
3335  024724  012706  001000                    MOV     #STACK,SP       ;SET STACK POINTER.
3336
3337  024730  013702  014750                    MOV     RHWC,R2         ;R2=RH11 WC ADDRESS.
3338  024734  010237  045534                    MOV     R2,REGADR
3339  024740  012737  125000  001124            MOV     #125000,$GDDAT  ;$GDDAT=S/B.
3340
3341  024746  005012                    L07:    CLR     (R2)            ;CLEAR RH11 WC.
3342
3343  024750  005202                            INC     R2              ;R2=HI BYTE.
3344
3345  024752  112712  000377                    MOVB    #377,(R2)       ;SET WC=177400
3346  024756  142712  000125            I07:    BICB    #125,(R2)       ;DO A BICB
3347
3348  024762  005302                            DEC     R2              ;R2=FULL WORD.
3349
3350  024764  011237  001126                    MOV     (R2),$BDDAT     ;$BDDAT=WAS.
3351
3352  024770  023737  001126  001124            CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS.
3353  024776  001401                            BEQ     TST20           ;NEXT TEST
3354  025000  104046                            ERROR   46
3355
3356
```

```
3357
3358                                    .REM    %
3359                                    THIS TEST ATTEMPT TO ACCESS AN ILLEGAL REGISTER WITHIN THE RANGE
3360                                    OF ADDRESSES SELECTED BY THE XOR'S. THE RH11 ADDRESS DECODE LOGIC WILL ALLOW UP TO 32
3361                                    CONTIGUOUS REGISTERS TO EXIST, IN OUR CONFIGURATION ONLY 16 WILL REALLY
3362                                    EXIST, THIS TEST ATTEMPTS TO ACCESS THE 20(8) REGISTER - IF IT RESPONDS
3363                                    THE TEST WILL TYPE OUT AN ERROR.
3364
3365                                    FOR THE CASE OF THE RH70, THE CONFIGURATION ALLOWS 18 OR 32 REGISTERS, SO
3366                                    WE WILL ATTEMPT TO ADDRESS REGISTER 23(8), 33(8) OR AS BEFORE THE LAST
3367                                    REGISTER + 2.
3368                                            %
3369
3370  025002  000004                    TST20:  SCOPE
3371  025004  012737  000001  001212            MOV     #1,$TIMES       ;;DO 1 ITERATION
3372  025012  012737  000020  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3373  025020  012777  000040  167726            MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3374
3375  025026  005737  017334                    TST     @#RH70          ;CHECK TO SEE IF RUNNING WITH RH70
3376  025032  001403                            BEQ     1$              ;IF NOT...SKIP NEXT & DO FOLLOWING
3377  025034  013702  015016                    MOV     RHCS3,R2        ;R2 = LAST LEGAL RH70 REG ADDRESS
3378  025040  000402                            BR      2$              ;SKIP NEXT
3379
3380  025042  013702  015010            1$:     MOV     RHEC2,R2        ;R2 = LAST LEGAL RH11 REG ADDRESS.
3381  025046  062702  000002            2$:     ADD     #2,R2           ;R2 = FIRST ILLEGAL ADDRESS.
3382  025052  010237  045534                    MOV     R2,REGADR
3383  025056  005037  001126                    CLR     $BDDAT          ;$BDDAT=WAS.
3384  025062  005037  001124                    CLR     $GDDAT          ;$GDDAT=S/B.
3385
3386  025066  013737  000004  001176            MOV     @#TIMOT,$TMP0   ;SAVE TIMEOUT VECTOR.
3387  025074  012737  025120  000004            MOV     #010,@#TIMOT    ;SET TIMEOUT VECTOR TO 010.
3388
3389  025102  012706  001000            L10:    MOV     #STACK,SP       ;SET THE STACK POINTER.
3390
3391  025106  005712                            TST     (R2)            ;TEST ADDRESS.
3392
3393  025110  062702  000024                    ADD     #24,R2          ;THIS MIGHT BE THE CASE OF 32 REGISTERS
3394  025114  005712                            TST     (R2)            ;TEST IT AGAIN
3395  025116  104047                            ERROR   47
3396
3397  025120  013737  001176  000004    010:    MOV     $TMP0,@#TIMOT   ;RESTORE TIMEOUT VECTOR.
3398
```

```
3399                                    ;
3400                                    ;NOW WE ATTACK BTCB
3401                                    ;
3402                                    .REM    %
3403                                    THE FOLLOWING TWO TESTS DETERMINE IF ALL NON DRIVES SET THE
3404                                    NED BIT AND THAT ALL EXISTING DRIVES CLEAR THE NED BIT.
3405                                    THE TESTS LOOK AT TOTALAT TO DETERMINE WHICH DRIVES EXIST AND TEST THAT
3406                                    THESE DRIVES WILL CLEAR THE NED BIT. AND ALSO THAT NON EXISTANT DRIVES
3407                                    WILL SET THE NED BIT.
3408                                    ON FIRST PASS ONLY, IF ALL DRIVES ARE ON LINE A MESSAGE WILL SO ADVISE THE
3409                                    OPERATOR. THE TEST WILL CONTINUE REGARLESS OF THE OPERATORS ACTION.
3410                                    %
3411  025126  000004                    TST21:  SCOPE
3412  025130  012737  000001  001212            MOV     #1,$TIMES       ;;DO 1 ITERATION
3413  025136  012737  000021  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3414  025144  012777  000040  167602            MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3415
3416  025152  123727  015136  000377            CMPB    @#TOTALAT,#377          ;ARE ALL DRIVES ON LINE.
3417  025160  001150                            BNE     U11             ;NO GO RUN THE TEST.
3418
3419  025162  023727  001100  000000            CMP     $PASS,#0        ;IF PASS #0 THEN TYPE MESSAGE
3420                                                                    ;ADVISING OPERATOR
3421                                                                    ;THAT THIS TEST WILL NOT BE RUN
3422  025170  001002                            BNE     Y11
3423  025172  000137  025604                    JMP     X11
3424
3425  025176                            Y11:
3426  025176  104401  025204                    TYPE    ,65$            ;;TYPE ASCIZ STRING
3427  025202  000426                            BR      64$             ;;GET OVER THE ASCIZ
3428  025260  104401  025266                    TYPE    ,67$            ;;TYPE ASCIZ STRING
3429  025264  000426                            BR      66$             ;;GET OVER THE ASCIZ
3430  025342  104401  025350                    TYPE    ,69$            ;;TYPE ASCIZ STRING
3431  025346  000425                            BR      68$             ;;GET OVER THE ASCIZ
3432  025422  104401  025430                    TYPE    ,71$            ;;TYPE ASCIZ STRING
3433  025426  000425                            BR      70$             ;;GET OVER THE ASCIZ
3434
3435  025502  012706  001000            U11:    MOV     #STACK,SP       ;SET STACK POINTER.
3436
3437  025506  013702  014754                    MOV     RHCS2,R2        ;R2=RH11 RHCS2 ADDRESS.
3438  025512  010237  045534                    MOV     R2,REGADR
3439
3440  025516  013700  015136                    MOV     @#TOTALAT,R0
3441  025522  005001                            CLR     R1
3442  025524  006000            S11:    ROR     R0
3443  025526  103423                            BCS     N11
3444
3445  025530  010137  001124            R11:    MOV     R1,$GDDAT       ;$GDDAT=S/B.
3446  025534  052737  010000  001124            BIS     #NED,$GDDAT
3447
3448  025542  013705  014756                    MOV     RHCS1,R5        ;ADDRESS OF A DEVICE REGISTER.
3449
3450  025546  010112            L11:    MOV     R1,(R2)         ;LOAD A NON-EXISTANT DRIVE.
3451
3452  025550  011537  001176                    MOV     (R5),$TMP0      ;ATTEMPT TO READ FROM DEVICE REGISTER.
3453
3454  025554  011237  001126                    MOV     (R2),$BDDAT     ;$BDDAT=WAS.
```

```
 3455  025560  042737  167770  001126          BIC     #^C<NED!US4!US2!US1>,$BDDAT;$BDDAT=SAVED DATA.
 3456
 3457  025566  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS.
 3458  025574  001005                          BNE     E11
 3459
 3460  025576  005201                  N11:    INC     R1
 3461  025600  020127  000010                  CMP     R1,#8.          ;TESTED ALL DRIVES YET.
 3462  025604                          X11:
 3463  025604  001402                          BEQ     TST22           ;NEXT TEST
 3464  025606  000746                          BR      S11
 3465
 3466  025610  104050                  E11:    ERROR   50
 3467
 3468
```

```
3469
3470  025612  000004                    TST22:  SCOPE
3471  025614  012737  000001  001212            MOV     #1,$TIMES       ;;DO 1 ITERATION
3472  025622  012737  000022  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3473  025630  012777  000040  167116            MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3474
3475  025636  012706  001000                    MOV     #STACK,SP       ;SET STACK POINTER.
3476
3477  025642  013702  014754                    MOV     RHCS2,R2        ;R2=RH11 RHCS2 ADDRESS.
3478  025646  010237  045534                    MOV     R2,REGADR
3479  025652  013700  015136                    MOV     @#TOTALAT,R0
3480  025656  005001                            CLR     R1
3481  025660  006000                    S12:    ROR     R0
3482  025662  103020                            BCC     N12
3483
3484  025664  010137  001124                    MOV     R1,$GDDAT
3485
3486  025670  013705  014756                    MOV     RHCS1,R5        ;ADDRESS OF A DEVICE REGISTER.
3487
3488  025674  010112                    L12:    MOV     R1,(R2)         ;SELECT UNIT.
3489
3490  025676  011537  001176                    MOV     (R5),$TMP0      ;ATTEMPT TO READ FROM DEVICE.
3491
3492  025702  011237  001126                    MOV     (R2),$BDDAT     ;$BDDAT=WAS.
3493  025706  042737  167770  001126            BIC     #^C<NED!US4!US2!US1>,$BDDAT     ;$BDDAT=SAVED DATA.
3494
3495  025714  023737  001126  001124            CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS.
3496  025722  001005                            BNE     E12
3497
3498  025724  005201                    N12:    INC     R1
3499  025726  020127  000010                    CMP     R1,#8.          ;TESTED ALL DRIVES.
3500  025732  001402                            BEQ     TST23           ;NEXT TEST
3501
3502  025734  000751                            BR      S12
3503  025736  104050                    E12:    ERROR   50
3504
3505
```

```
3506
3507                                    .REM    %
3508                                    TEST TO SEE IF WE CAN READ FROM THE ''AS'' REGISTER AND NOT CAUSE
3509                                    A NED NON-EXISTANT DRIVE ERROR
3510                                    %
3511  025740  000004                    TST23:  SCOPE
3512  025742  012737  000001  001212            MOV     #1,$TIMES       ;;DO 1 ITERATION
3513  025750  012737  000023  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3514  025756  012777  000040  166770            MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3515
3516  025764  012706  001000                    MOV     #STACK,SP       ;SET STACK POINTER.
3517
3518  025770  013702  014754                    MOV     RHCS2,R2        ;R2=RH11 CS2 ADDRESS.
3519  025774  012737  000007  001124            MOV     #US4!US2!US1,$GDDAT;$GDDAT=S/B.
3520
3521  026002  013705  014756                    MOV     RHCS1,R5        ;ADDRESS OF A DEVICE REGISTER.
3522
3523  026006  012712  000007            L13:    MOV     #US4!US2!US1,(R2);LOAD A NON-EXISTANT DEVICE.
3524
3525  026012  013702  014774                    MOV     RHAS,R2         ;R2= ''AS''.
3526
3527  026016  011237  001176                    MOV     (R2),$TMP0      ;ATTEMPT TO READ FROM DEVICE AS.
3528
3529  026022  005012                            CLR     (R2)            ;CLEAR AS REGISTER.
3530  026024  013702  014754                    MOV     RHCS2,R2        ;R2=RH11 CS2 ADDRESS.
3531  026030  010237  045534                    MOV     R2,REGADR
3532
3533  026034  011237  001126                    MOV     (R2),$BDDAT     ;$BDDAT=WAS.
3534  026040  042737  167770  001126            BIC     #^C<NED!US4!US2!US1>,$BDDAT;SAVE NED AND DEVICE SELECTED.
3535
3536  026046  023737  001126  001124            CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS.
3537  026054  001401                            BEQ     TST24           ;NEXT TEST
3538  026056  104051                            ERROR   51
3539
3540
```

```
3541                                    .REM    %
3542                                    THE NEXT THREE TESTS WILL TEST THE BUS ADDRESS REGISTER IN BOTH WORD
3543                                    AND BYTE MODE. THE PATTERNS ARE CHOSEN TO PICK UP ANY DROPPED OR STUCK
3544                                    BITS.
3545                                            %
3546
3547  026060  000004                    TST24:  SCOPE
3548  026062  012737  000001  001212            MOV     #1,$TIMES       ;;DO 1 ITERATION
3549  026070  012737  000024  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3550  026076  012777  000040  166650            MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3551
3552  026104  012706  001000                    MOV     #STACK,SP       ;SET STACK POINTER.
3553
3554  026110  013702  014752                    MOV     RHBA,R2         ;R2=RH11 BA ADDRESS.
3555  026114  010237  045534                    MOV     R2,REGADR
3556  026120  012705  026160                    MOV     #LST14A,R5      ;R5=TEST LIST ADDRESS.
3557
3558  026124  012537  001124            L14:    MOV     (R5)+,$GDDAT    ;$GDDAT=S/B.
3559
3560  026130  013712  001124            I14:    MOV     $GDDAT,(R2)     ;SET BUS ADDRESS REGISTER.
3561  026134  011237  001126                    MOV     (R2),$BDDAT     ;READ BUS ADDRESS REGISTER.
3562
3563  026140  023737  001126  001124            CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS.
3564  026146  001401                            BEQ     N14             ;GET NEXT TEST DATA.
3565  026150  104052                            ERROR   52
3566
3567  026152  005715                    N14:    TST     (R5)            ;AT END OF LIST
3568  026154  001363                            BNE     L14             ;NO!
3569  026156  000440                            BR      TST25           ;NEXT TEST
3570  026160  000002                    LST14A: 2
3571  026162  000004                            4
3572  026164  000010                            10
3573  026166  000020                            20
3574  026170  000040                            40
3575  026172  000100                            100
3576  026174  000200                            200
3577  026176  000400                            400
3578  026200  001000                            1000
3579  026202  002000                            2000
3580  026204  004000                            4000
3581  026206  010000                            10000
3582  026210  020000                            20000
3583  026212  040000                            40000
3584  026214  100000                            100000
3585  026216  177776                            177776
3586  026220  177774                            177774
3587  026222  177772                            177772
3588  026224  177766                            177766
3589  026226  177756                            177756
3590  026230  177736                            177736
3591  026232  177676                            177676
3592  026234  177576                            177576
3593  026236  177376                            177376
3594  026240  176776                            176776
3595  026242  175776                            175776
3596  026244  173776                            173776
```

```
3597  026246  167776                    167776
3598  026250  157776                    157776
3599  026252  137776                    137776
3600  026254  077776                    077776
3601  026256  000000                    0
3602
```

```
3603
3604  026260  000004                      TST25:  SCOPE
3605  026262  012737  000001  001212              MOV     #1,$TIMES       ;;DO 1 ITERATION
3606  026270  012737  000025  017330              MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3607  026276  012777  000040  166450              MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3608
3609  026304  012706  001000                      MOV     #STACK,SP       ;RESET STACK.
3610
3611  026310  013702  014752                      MOV     RHBA,R2         ;R2=RH11 BA ADDRESS.
3612  026314  010237  045534                      MOV     R2,REGADR
3613  026320  012705  026362                      MOV     #LST15A,R5      ;R5=TEST LIST ADDRESS.
3614
3615  026324  005012                      L15:    CLR     (R2)            ;CLEAR RH11 BA REGISTER.
3616
3617  026326  012537  001124                      MOV     (R5)+,$GDDAT    ;$GDDAT=S/B.
3618
3619  026332  113712  001124              I15:    MOVB    $GDDAT,(R2)     ;SET BUS ADDRESS REGISTER.
3620  026336  011237  001126                      MOV     (R2),$BDDAT     ;READ BUS ADDRESS REGISTER.
3621
3622  026342  023737  001126  001124              CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS.
3623  026350  001401                              BEQ     R15
3624
3625  026352  104053                              ERROR   53
3626
3627  026354  005715                      R15:    TST     (R5)            ;AT END OF TEST LIST.
3628  026356  001362                              BNE     L15             ;NO!
3629  026360  000417                              BR      TST26           ;NEXT TEST
3630                                  ;THIS LIST WILL BE USED TO LOAD THE LOWER BYTE OF THE BA REGISTER.
3631                                  ;
3632  026362  000002                  LST15A: 2
3633  026364  000004                          4
3634  026366  000010                          10
3635  026370  000020                          20
3636  026372  000040                          40
3637  026374  000100                          100
3638  026376  000200                          200
3639  026400  000376                          376
3640  026402  000372                          372
3641  026404  000366                          366
3642  026406  000356                          356
3643  026410  000336                          336
3644  026412  000276                          276
3645  026414  000176                          176
3646  026416  000000                          0
```

```
3647
3648  026420  000004                      TST26:  SCOPE
3649  026422  012737  000001  001212              MOV     #1,$TIMES       ;;DO 1 ITERATION
3650  026430  012737  000026  017330              MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3651  026436  012777  000040  166310              MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3652
3653  026444  012706  001000                      MOV     #STACK,SP       ;RESET STACK.
3654
3655  026450  013702  014752                      MOV     RHBA,R2         ;R2=RH11 BA ADDRESS.
3656  026454  010237  045534                      MOV     R2,REGADR
3657  026460  012705  026532                      MOV     #LST16A,R5      ;R5=TEST LIST ADDRESS.
3658
3659  026464  005012                      L16:    CLR     (R2)            ;CLEAR BA REGISTER.
3660
3661  026466  012537  001124                      MOV     (R5)+,$GDDAT    ;$GDDAT=S/B.
3662
3663  026472  005202                              INC     R2              ;R2=HI BYTE ADDRESS.
3664
3665  026474  113712  001124              I16:    MOVB    $GDDAT,(R2)     ;SET BUS ADDRESS REGISTER HI-BYTE.
3666
3667  026500  005302                              DEC     R2              ;R2=FULL WORD ADDRESS.
3668
3669  026502  011237  001126                      MOV     (R2),$BDDAT     ;READ BUS ADDRESS.
3670  026506  000337  001124                      SWAB    $GDDAT          ;ADJUST DATA FOR HI BYTE.
3671
3672  026512  023737  001126  001124              CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS.
3673  026520  001401                              BEQ     R16             ;OKAY
3674
3675  026522  104053                              ERROR   53
3676
3677  026524  005715                      R16:    TST     (R5)            ;AT END OF TEST LIST.
3678  026526  001356                              BNE     L16             ;NOPE
3679  026530  000420                              BR      TST27           ;NEXT TEST
3680                                  ;THIS LIST WILL BE USED TO LOAD THE UPPER BYTE OF THE BA REGISTER.
3681                                  ;
3682  026532  000002                  LST16A: 2
3683  026534  000004                          4
3684  026536  000010                          10
3685  026540  000020                          20
3686  026542  000040                          40
3687  026544  000100                          100
3688  026546  000200                          200
3689  026550  000376                          376
3690  026552  000374                          374
3691  026554  000376                          376
3692  026556  000366                          366
3693  026560  000356                          356
3694  026562  000336                          336
3695  026564  000276                          276
3696  026566  000176                          176
3697  026570  000000                          0
3698
```

```
3699                                    .REM    %
3700                                    THIS TEST CAUSE AN RH11 INTERRUPT VIA THE SPECIAL COMMAND SEQUENCE
3701                                    BIS     #IE,@RHCS2. THIS TEST PROVES ONLY THAT THE HARDWARE CAN HANDLE
3702                                    INTERRUPTS PROPERLY
3703                                            %
3704  026572  000004                    TST27:  SCOPE
3705  026574  012737  000001  001212            MOV     #1,$TIMES       ;;DO 1 ITERATION
3706  026602  012737  000027  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3707  026610  012777  000040  166136            MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3708
3709  026616  012706  001000                    MOV     #STACK,SP       ;SET STACK POINTER.
3710
3711  026622  013702  014756                    MOV     RHCS1,R2        ;R2=RH11 ADDRESS.
3712  026626  010237  045534                    MOV     R2,REGADR
3713  026632  005037  001126                    CLR     $BDDAT          ;$BDDAT=WAS.
3714  026636  005037  001124                    CLR     $GDDAT          ;$GDDAT=S/B.
3715
3716  026642  017737  166076  001176            MOV     @RPVEC,$TMP0    ;SAVE RH11 INTERRUPT VECTOR.
3717  026650  012777  026674  166066            MOV     #021,@RPVEC     ;SET RH11 INTERRUPT VECTOR TO 021.
3718
3719  026656  052712  000100          L21:      BIS     #IE,(R2)        ;CAUSE INTERRUPT.
3720  026662  000240                            NOP                     ;WAIT FOR INTERRUPT.
3721  026664  000240                            NOP
3722  026666  011237  001124                    MOV     (R2),$GDDAT     ;SAVE CONTENTS OF REGISTER.
3723
3724  026672  104054                            ERROR   54
3725
3726  026674  013777  001176  166042  021:      MOV     $TMP0,@RPVEC    ;RESTORE RH11 INTERRUPT VECTOR.
3727
3728
```

```
3729
3730                                    ;TEST THAT RESET CAN GENERATE THE SIGNAL CLR L.
3731                                    .REM    %
3732                                            THE PROGRAM SETS THE PORT SELECT BIT .THEN DOES A RESET
3733                                            IF THE SIGNAL CLR L WAS GENERATED THE PORT SELECT BIT WILL CLEAR.
3734                                            %
3735  026702  000004                    TST30:  SCOPE
3736  026704  012737  000001  001212            MOV     #1,$TIMES       ;;DO 1 ITERATION
3737  026712  012737  000030  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3738  026720  012777  000040  166026            MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3739
3740  026726  013702  014756                    MOV     RHCS1,R2        ;R2=RH11 CS1 ADDRESS
3741  026732  010237  045534                    MOV     R2,REGADR
3742  026736  005037  001124                    CLR     $GDDAT          ;$GDDAT=S/B
3743
3744  026742  012706  001000            L22:    MOV     #STACK,SP       ;SET STACK POINTER.
3745  026746  012712  002000                    MOV     #PSEL,(R2)      ;SET PORT SELECT FLOP.
3746  026752  000005                            RESET                   ;DO A BUSA INIT.
3747
3748  026754  011237  001126                    MOV     (R2),$BDDAT     ;$BDDAT=WAS.
3749  026760  042737  177677  001126            BIC     #^C<IE>,$BDDAT  ;SAVE ONLY I.E. BIT.
3750
3751  026766  023737  001126  001124            CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS.
3752  026774  001401                            BEQ     TST31           ;NEXT TEST
3753  026776  104055                            ERROR   55
3754
3755
```

```
3756                                    .REM    %
3757                                    SET THE MXF FLOP AND READ IT BACK.
3758                                    %
3759  027000  000004                   TST31:  SCOPE
3760  027002  012737  000001  001212           MOV     #1,$TIMES       ;;DO 1 ITERATION
3761  027010  012737  000031  017330           MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3762  027016  005737  017334                   TST     @#RH70          ;TEST FOR RH70 CONTROLLER
3763  027022  001402                           BEQ     30$             ;IF FLAG = 1, SKIP THIS TEST
3764  027024  000137  027110                   JMP     TST32           ;JUMP TO NEXT TEST ----------)
3765  027030  012777  000040  165716           MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3766
3767  027036  012706  001000                   MOV     #STACK,SP       ;SET STACK POINTER.
3768  027042  013702  014754                   MOV     RHCS2,R2        ;R2=RH11 CS2 ADDRESS.
3769  027046  010237  045534                   MOV     R2,REGADR
3770  027052  012737  001000  001124           MOV     #MXF,$GDDAT     ;$GDDAT=S/B.
3771
3772  027060  012712  001000           I24:    MOV     #MXF,(R2)       ;LOAD MXF
3773
3774  027064  011237  001126                   MOV     (R2),$BDDAT     ;$BDDAT=WAS.
3775
3776  027070  042737  176777  001126           BIC     #^C<MXF>,$BDDAT ;SAVE ONLY MXF
3777
3778  027076  023737  001126  001124           CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS
3779  027104  001401                           BEQ     TST32           ;NEXT TEST
3780  027106  104056                           ERROR   56
3781
3782
```

```
3783                                        .REM    %
3784                                        SET THE UNIBUS PARITY ERROR FLOP DIRECTLY VIA A MOV #UPE TO RHCS2
3785                                        ATTEMPT TO READ IT BACK.
3786                                        %
3787  027110  000004                TST32:  SCOPE
3788  027112  012737  000001  001212        MOV     #1,$TIMES       ;;DO 1 ITERATION
3789  027120  012737  000032  017330        MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3790  027126  005737  017334                TST     @#RH70          ;TEST FOR RH70 CONTROLLER
3791  027132  001402                        BEQ     30$             ;IF FLAG = 1, SKIP THIS TEST
3792  027134  000137  027224                JMP     TST33           ;JUMP TO NEXT TEST ----------)
3793
3794  027140  012777  000040  165606        MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3795
3796  027146  012706  001000                MOV     #STACK,SP       ;SET STACK POINTER.
3797
3798  027152  013702  014754                MOV     RHCS2,R2        ;R2=RHCS2 ADDRESS.
3799  027156  010237  045534                MOV     R2,REGADR
3800  027162  012737  020000  001124        MOV     #UPE,$GDDAT     ;$GDDAT=S/B.
3801
3802  027170  013712  001124        L30:    MOV     $GDDAT,(R2)     ;ATTEMPT TO SET UPE.
3803  027174  000240                        NOP
3804  027176  000240                        NOP
3805
3806  027200  011237  001126                MOV     (R2),$BDDAT     ;$BDDAT=ACTUAL DATA.
3807  027204  042737  157777  001126        BIC     #^C<UPE>,$BDDAT ;SAVE ONLY UPE.
3808
3809  027212  023737  001126  001124        CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS.
3810  027220  001401                        BEQ     TST33           ;NEXT TEST
3811  027222  104057                        ERROR   57
3812
3813
```

```
3814
3815                                    .REM    %
3816                                    SET THE UNIBUS PARITY ERROR FLOP AND ATTEMPT TO CLEAR IT WITH A
3817                                    CONTROLLER CLEAR.
3818                                    %
3819  027224  000004                    TST33:  SCOPE
3820  027226  012737  000001  001212            MOV     #1,$TIMES       ;;DO 1 ITERATION
3821  027234  012737  000033  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3822  027242  012777  000040  165504            MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3823
3824  027250  012706  001000                    MOV     #STACK,SP       ;SET STACK POINTER.
3825
3826  027254  013702  014754                    MOV     RHCS2,R2        ;R2=RHCS2 ADDRESS.
3827  027260  010237  045534                    MOV     R2,REGADR
3828  027264  005037  001124                    CLR     $GDDAT          ;$GDDAT=S/B.
3829
3830  027270  012712  020000            L31:    MOV     #UPE,(R2)       ;SET UNIBUS PARITY ERROR SET.
3831  027274  000240                            NOP
3832  027276  000240                            NOP
3833
3834  027300  012712  000040                    MOV     #CLR,(R2)       ;CONTROLLER CLEAR.
3835
3836  027304  011237  001126                    MOV     (R2),$BDDAT     ;READ STATUS
3837  027310  042737  157777  001126            BIC     #^C<UPE>,$BDDAT ;SAVE ERROR.
3838
3839  027316  023737  001126  001124            CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS.
3840  027324  001401                            BEQ     TST34           ;NEXT TEST
3841  027326  104057                            ERROR   57
3842
3843
```

```
3844
3845                                    .REM    %
3846                                    SET THE UNIT SELECT REGISTER TO DRIVE #7 ALL BITS SET. GENERATE A
3847                                    CONTROLLER CLEAR AND VERIFY THAT THE UNIT SELECT REGISTER IS CLEARED.
3848                                    %
3849  027330  000004                    TST34:  SCOPE
3850  027332  012737  000001  001212            MOV     #1,$TIMES       ;;DO 1 ITERATION
3851  027340  012737  000034  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3852  027346  012777  000040  165400            MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3853
3854  027354  012706  001000                    MOV     #STACK,SP       ;SET STACK POINTER.
3855
3856  027360  013702  014754                    MOV     RHCS2,R2        ;R2=CS2 ADDRESS.
3857  027364  010237  045534                    MOV     R2,REGADR
3858  027370  005037  001124                    CLR     $GDDAT          ;$GDDAT=S/B.
3859
3860  027374  012712  000007            L34:    MOV     #US4!US2!US1,(R2)  ;LOAD RHCS2 DRIVE SELECT
3861  027400  000240                            NOP
3862  027402  000240                            NOP
3863
3864  027404  012712  000040                    MOV     #CLR,(R2)       ;GENERATE CLR.
3865
3866  027410  011237  001126                    MOV     (R2),$BDDAT     ;READ RHCS2
3867  027414  042737  177770  001126            BIC     #^C<US4!US2!US1>,$BDDAT;SAVE DRIVE SELECT BITS.
3868
3869  027422  023737  001126  001124            CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS.
3870  027430  001401                            BEQ     TST35           ;NEXT TEST
3871  027432  104060                            ERROR   60
3872
3873
```

```
3874
3875                                    .REM    %
3876                                    SET THE UPE FLOP UNIBUS PARITY ERROR. VERIFY THAT THE SETTING OF (UPE)
3877                                    ALSO SETS THE TRANSFER ERROR (TRE) FLOP
3878                                    %
3879  027434  000004                    TST35:  SCOPE
3880  027436  012737  000001  001212            MOV     #1,$TIMES       ;;DO 1 ITERATION
3881  027444  012737  000035  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3882  027452  005737  017334                    TST     @#RH70          ;TEST FOR RH70 CONTROLLER
3883  027456  001402                            BEQ     30$             ;IF FLAG = 1, SKIP THIS TEST
3884  027460  000137  027550                    JMP     TST36           ;JUMP TO NEXT TEST ----------)
3885
3886  027464  012777  000040  165262            MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3887
3888  027472  012706  001000                    MOV     #STACK,SP       ;SET STACK POINTER.
3889
3890  027476  013702  014754                    MOV     RHCS2,R2        ;R2=RHCS2 ADDRESS.
3891  027502  012737  040000  001124            MOV     #TRE,$GDDAT     ;$GDDAT=S/B.
3892
3893  027510  012712  020000            L35:    MOV     #UPE,(R2)       ;SET UNIBUS PARITY ERROR.
3894
3895  027514  013702  014756                    MOV     RHCS1,R2        ;R2=RHCS1 ADDRESS.
3896  027520  010237  045534                    MOV     R2,REGADR
3897
3898  027524  011237  001126                    MOV     (R2),$BDDAT     ;READ REGISTER
3899  027530  042737  137777  001126            BIC     #^C<TRE>,$BDDAT ;SAVE TRANSFER ERROR.
3900
3901  027536  023737  001126  001124            CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS
3902  027544  001401                            BEQ     TST36           ;NEXT TEST
3903  027546  104061                            ERROR   61
3904
3905
```

```
3906
3907                                    .REM    %
3908                                    SET THE NED BIT NON EXISTANT DRIVE VERIFY THAT THIS SETS THE (TRE) BIT.
3909                                    %
3910  027550  000004                    TST36:  SCOPE
3911  027552  012737  000001  001212            MOV     #1,$TIMES       ;;DO 1 ITERATION
3912  027560  012737  000036  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3913  027566  012777  000040  165160            MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3914
3915  027574  012706  001000                    MOV     #STACK,SP       ;SET STACK POINTER.
3916
3917  027600  013702  014754                    MOV     RHCS2,R2        ;R2=CS2 ADDRESS.
3918  027604  012737  040000  001124            MOV     #TRE,$GDDAT     ;$GDDAT=S/B.
3919
3920  027612  013700  015136                    MOV     @#TOTALAT,R0    ;GET ALL AVAILABLE DRIVES DATA
3921  027616  005001                            CLR     R1
3922  027620  006000                    S36:    ROR     R0
3923  027622  103420                            BCS     N36
3924  027624  010112                    L36:    MOV     R1,(R2)           ;SET NED.
3925
3926  027626  013702  014756                    MOV     RHCS1,R2        ;R2=RHCS1 ADDRESS.
3927  027632  010237  045534                    MOV     R2,REGADR
3928
3929  027636  011237  001126                    MOV     (R2),$BDDAT     ;READ REGISTER.
3930  027642  042737  137777  001126            BIC     #^C<TRE>,$BDDAT ;SAVE TRE.
3931
3932  027650  023737  001126  001124            CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS.
3933  027656  001406                            BEQ     TST37           ;NEXT TEST
3934  027660  104062                            ERROR   62
3935  027662  000404                            BR      TST37           ;NEXT TEST
3936
3937  027664  005201                    N36:    INC     R1
3938  027666  020127  000010                    CMP     R1,#8.
3939  027672  001352                            BNE     S36
3940
3941
```

```
3942
3943                                    .REM    %
3944                                    SET THE PORT SELECT FLOP VERIFY THAT WE CAN READ IT BACK.
3945                                    %
3946  027674  000004                    TST37:  SCOPE
3947  027676  012737  000001  001212            MOV     #1,$TIMES       ;;DO 1 ITERATION
3948  027704  012737  000037  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3949  027712  012777  000040  165034            MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3950
3951  027720  012706  001000                    MOV     #STACK,SP       ;SET STACK POINTER.
3952
3953  027724  013702  014756                    MOV     RHCS1,R2        ;R2=RHCS1 ADDRESS.
3954  027730  010237  045534                    MOV     R2,REGADR
3955  027734  005037  001124                    CLR     $GDDAT          ;$GDDAT=S/B.
3956
3957  027740  012712  002000            L40:    MOV     #PSEL,(R2)      ;SET P SELECT.
3958
3959  027744  012777  000040  165002            MOV     #CLR,@RHCS2     ;DO A CONTROLLER CLEAR.
3960
3961  027752  011237  001126                    MOV     (R2),$BDDAT     ;READ BACK P SELECT BIT.
3962  027756  042737  175777  001126            BIC     #^C<PSEL>,$BDDAT;SAVE ONLY PORT SELECT.
3963
3964  027764  023737  001126  001124            CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS.
3965  027772  001401                            BEQ     TST40           ;NEXT TEST
3966  027774  104064                            ERROR   64
3967
```

```
3968                                        .REM    %
3969                                        VERIFY THAT CONTROLLER CLEAR WILL CLEAR THE COMMAND REGISTER.
3970                                                %
3971
3972  027776  000004                  TST40:  SCOPE
3973  030000  012737  000001  001212          MOV     #1,$TIMES           ;;DO 1 ITERATION
3974  030006  012737  000040  017330          MOV     #TTNO,@#TSTNM       ;THIS SAVES TEST NUMBER
3975  030014  012777  000040  164732          MOV     #CLR,@RHCS2         ;CLEAR RH11 CONTROLLER
3976
3977  030022  012706  001000                  MOV     #STACK,SP           ;SET STACK POINTER.
3978
3979  030026  013702  014756                  MOV     RHCS1,R2            ;R2=RHCS1 ADDRESS.
3980  030032  010237  045534                  MOV     R2,REGADR
3981  030036  005037  001124                  CLR     $GDDAT              ;$GDDAT=S/B.
3982
3983  030042  012712  000011          L41:    MOV     #11,(R2)            ;ISSUE A DRIVE CLR AND GO.
3984
3985  030046  012777  000040  164700          MOV     #CLR,@RHCS2         ;CONTROLLER CLEAR.
3986
3987  030054  011237  001126                  MOV     (R2),$BDDAT         ;READ COMMAND REGISTER.
3988  030060  042737  177770  001126          BIC     #^C<7>,$BDDAT       ;SAVE ONLY THE COMMAND BITS.
3989
3990  030066  023737  001126  001124          CMP     $BDDAT,$GDDAT       ;COMPARE RESULTS.
3991  030074  001401                          BEQ     TST41               ;NEXT TEST
3992  030076  104065                          ERROR   65
3993
```

```
3994
3995                                    .REM    %
3996                                    THE COMMAND REGISTER IS SUPPOSED TO BE SELF CLEARING THAT IS WHEN THE
3997                                    RH11 IS SELECTED WHEATHER OR NOT WE ADDRESS IT DIRECTLY THE LOGIC
3998                                    SHOULD CLEAR THE COMMAND REGISTER.
3999                                    THIS IS TESTED BY LOADING A COMMAND INTO IT READING ANOTHER REGISTER
4000                                    THAN RETURNING TO CHECK THE COMMAND REGISTER TO DETERMINE IF IT CLEARED.
4001                                    %
4002  030100  000004                    TST41:  SCOPE
4003  030102  012737  000001  001212            MOV     #1,$TIMES       ;;DO 1 ITERATION
4004  030110  012737  000041  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
4005  030116  012777  000040  164630            MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
4006
4007  030124  012706  001000                    MOV     #STACK,SP       ;SET STACK POINTER.
4008
4009  030130  013702  014756                    MOV     RHCS1,R2        ;R2=RHCS1 ADDRESS.
4010  030134  010237  045534                    MOV     R2,REGADR
4011  030140  005037  001124                    CLR     $GDDAT          ;$GDDAT=S/B.
4012
4013  030144  012737  000340  177776            MOV     #340,PS         ;SET PRIORITY TO #7.
4014
4015  030152  012712  000011            L42:    MOV     #11,(R2)        ;ISSUE A DRIVE CLR AND GO
4016
4017  030156  011237  001176                    MOV     (R2),$TMP0      ;DO A RESELECT OF THE REGISTER.
4018
4019  030162  011237  001126                    MOV     (R2),$BDDAT     ;READ THE COMMAND REGISTER.
4020  030166  042737  177770  001126            BIC     #^C<7>,$BDDAT   ;SAVE ONLY THE COMMAND BITS.
4021
4022  030174  023737  001126  001124            CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS.
4023  030202  001401                            BEQ     TST42           ;NEXT TEST
4024  030204  104066                            ERROR   66
4025
```

```
4026
4027                                        .REM    %
4028                                        HERE WE TEST TO SEE IF SETTING (IE) AND (RDY) WILL CAUSE AN INTERRUPT.
4029                                        %
4030  030206  000004                TST42:  SCOPE
4031  030210  012737  000001  001212        MOV     #1,$TIMES       ;;DO 1 ITERATION
4032  030216  012737  000042  017330        MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
4033  030224  012777  000040  164522        MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
4034
4035  030232  013702  014756                MOV     RHCS1,R2        ;R2=RPCS1 ADDRESS.
4036  030236  010237  045534                MOV     R2,REGADR
4037  030242  005037  001126                CLR     $BDDAT          ;$BDDAT=WAS.
4038  030246  005037  001124                CLR     $GDDAT          ;$GDDAT=S/B.
4039
4040  030252  017737  164466  001176        MOV     @RPVEC,$TMP0    ;SAVE RH11 INTERRUPT VECTOR.
4041  030260  012777  030314  164456        MOV     #043,@RPVEC     ;SET RH11 INTERRUPT VECTOR 043
4042
4043  030266  012706  001000          L43:  MOV     #STACK,SP       ;SET STACK POINTER.
4044  030272  005037  177776                CLR     PS
4045
4046  030276  052712  000300                BIS     #IE!RDY,(R2)    ;THIS WILL CAUSE AN INTERRUPT.
4047  030302  000240                        NOP
4048  030304  000240                        NOP
4049
4050  030306  011237  001124                MOV     (R2),$GDDAT
4051  030312  104067                        ERROR   67
4052
4053  030314  013777  001176  164422  043:  MOV     $TMP0,@RPVEC    ;RESTORE RH11 INTERRUPT VECTOR.
```

```
4054
4055                                        .REM    %
4056                                        THE (IE) BIT IS SET CAUSING AN INTERRUPT. AT THE COMPLETION OF THE INTERRUPT
4057                                        WE CHECK TO SEE IF THE (IE) BIT HAS CLEARED.
4058                                        %
4059  030322  000004                TST43:  SCOPE
4060  030324  012737  000001  001212        MOV     #1,$TIMES       ;;DO 1 ITERATION
4061  030332  012737  000043  017330        MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
4062  030340  012777  000040  164406        MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
4063
4064  030346  013702  014756                MOV     RHCS1,R2        ;R2=RPCS1 ADDRESS.
4065  030352  010237  045534                MOV     R2,REGADR
4066  030356  005037  001124                CLR     $GDDAT          ;$GDDAT=S/B.
4067
4068  030362  017737  164356  001176        MOV     @RPVEC,$TMP0    ;SAVE RH11 INTERRUPT VECTOR.
4069  030370  012777  030420  164346        MOV     #044,@RPVEC     ;SET VECTOR TO 044.
4070
4071  030376  012706  001000          L44:  MOV     #STACK,SP       ;SET STACK POINTER.
4072  030402  005037  177776                CLR     PS
4073
4074  030406  052712  000100                BIS     #IE,(R2)        ;THIS WILL CAUSE AN INTERRUPT
4075  030412  000240                        NOP
4076  030414  000240                        NOP
4077
4078  030416  000411                        BR      E44             ;NO INTERRUPT.
4079
4080  030420  011237  001126          044:  MOV     (R2),$BDDAT     ;READ RHCS1.
4081  030424  042737  177677  001126        BIC     #^C<IE>,$BDDAT  ;SAVE ONLY THE ''IE'' BIT.
4082
4083  030432  023737  001126  001124        CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS.
4084  030440  001401                        BEQ     R44             ;OK!
4085
4086  030442  104070                E44:    ERROR   70
4087
4088  030444  013777  001176  164272  R44:  MOV     $TMP0,@RPVEC    ;RESTORE RH11 INTERRUPT VECTOR.
4089
```

```
4090
4091                                    .REM    %
4092                                    HERE WE VERIFY THAT WRITING INTO THE "AS" REGISTER OWILL NOT CAUSE
4093                                    AN (NED) ERROR.
4094                                    %
4095  030452  000004                    TST44:  SCOPE
4096  030454  012737  000001  001212            MOV     #1,$TIMES       ;;DO 1 ITERATION
4097  030462  012737  000044  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
4098  030470  012777  000040  164256            MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
4099
4100  030476  012706  001000                    MOV     #STACK,SP       ;SET STACK POINTER.
4101
4102  030502  013702  014754                    MOV     RHCS2,R2        ;R2=RH11 CS2 ADDRESS.
4103  030506  012737  000007  001124            MOV     #US4!US2!US1,$GDDAT;$GDDAT=S/B.
4104
4105  030514  013705  014756                    MOV     RHCS1,R5        ;ADDRESS OF A DEVICE REGISTER.
4106
4107  030520  012712  000007            L45:    MOV     #US4!US2!US1,(R2);LOAD A NON EXISTANT DEVICE.
4108
4109  030524  013702  014774                    MOV     RHAS,R2         ;R2="AS" ADDRESS.
4110
4111  030530  012712  000377                    MOV     #377,(R2)       ;ATTEMPT TO LOAD "AS".
4112
4113  030534  005012                            CLR     (R2)            ;CLEAR  "AS".
4114
4115  030536  013702  014754                    MOV     RHCS2,R2        ;R2=RH11 CS2 ADDRESS.
4116  030542  010237  045534                    MOV     R2,REGADR
4117
4118  030546  011237  001126                    MOV     (R2),$BDDAT     ;$BDDAT=WAS.
4119  030552  042737  167770  001126            BIC     #^C<NED!US4!US2!US1>,$BDDAT;SAVE NED AND DEVICE SELECTED.
4120
4121  030560  023737  001126  001124            CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS.
4122  030566  001401                            BEQ     TST45           ;NEXT TEST
4123  030570  104071                            ERROR   71
4124
```

```
4125                                    .SBTTL  EXTENDED MEMORY, DRIVE TIMING & SECTOR SELECT TESTS
4126
4127
4128  030572  000004                    TST45:  SCOPE
4129  030574  012706  001000                    MOV     #STACK,SP       ;RESET STACK
4130  030600  012737  000045  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
4131  030606  004737  045770                    JSR     PC,@#CLDISK     ;INIT DRIVE
4132  030612  012777  000001  164156            MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
4133  030620  004037  050342                    JSR     R0,@#MAKECYL    ;SUBROUTINE TO GIVE A SEEK
4134  030624  000000                            0                       ;THIS SHUOLD CHANGE RHCC TO 0
4135
4136
4137
4138
4139
4140
4141  030626  000004                    TST46:  SCOPE
4142
4143  030630  012706  001000                    MOV     #STACK,SP       ;RESET STACK
4144  030634  012737  000046  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
4145  030642  005737  017334                    TST     @#RH70          ;TEST FOR RH70 CONTROLLER
4146  030646  001402                            BEQ     30$             ;IF FLAG = 1, SKIP THIS TEST
4147  030650  000137  031160                    JMP     TST47           ;JUMP TO NEXT TEST ----------------------)
4148
4149  030654  004037  045706                    JSR     R0,@#CLAREA     ;CLEAR SIMULATED DISK
4150  030660  054572                            .WORD   DISK            ;FROM
4151  030662  055616                            .WORD   TOLGAP+16       ;TO
4152  030664  000000                            .WORD   0               ;DATA
4153
4154                                    ;THESE ARE SETUP FOR DISKLESS USE ONLY
4155
4156  030666  012737  010000  052654            MOV     #FMT22,@#CYL;CYLINDER 0
4157                                                                    ;16 BITS PER WORD
4158  030674  005037  052656                    CLR     @#SECOTR        ;SECTOR 0 TRACK 0
4159  030700  005037  052660                    CLR     @#KEY1          ;KEY1 0
4160  030704  005037  052662                    CLR     @#KEY2          ;KEY2 0
4161  030710  012737  000400  052722            MOV     #256.,@#NOWORD  ;NO OF DATA WORDS
4162  030716  012737  000001  052664            MOV     #1,@#X          ;WRITE DATA
4163  030724  004537  047176                    JSR     R5,@#CRC        ;GO TO CALCULATE CRC
4164  030730  052654                            CYL
4165  030732  054554                            WCRC
4166
4167                                    ;THESE ARE REGULAR SETUPS
4168
4169
4170  030734  004737  045770                    JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
4171  030740  012777  177400  164002            MOV     #-256.,@RHWC    ;256 DATA WORDS
4172  030746  013777  001144  163776            MOV     @#$TKS,@RHBA    ;STARTING ADDRESS OF WRITE BUFFER
4173  030754  017737  150164  015140            MOV     @$TKS,@#TMPILL  ;TEMPORARY STORAGE OF DATA
4174  030762  005077  163774                    CLR     @RHDST          ;SECTOR 0 TRACK 0
4175  030766  012777  010000  163772            MOV     #FMT22,@RHOF    ;16 BITS PER WORD FORMAT
4176  030774  005077  163770                    CLR     @RHCA           ;CYLINDER 0
4177  031000  004737  046024                    JSR     PC,@#CHECKT     ;CHECK DVA,RDY,DPR,DRY = 1 AND OTHERS DON'T
4178  031004  104401  005123                    TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF THE
4179  031010  000000                            HALT                    ;STOP THE TEST
4180  031012  013746  015170                    MOV     @#WRIDAT,-(SP)  ;WRITE DATA=60
```

```
 4181  031016  052716  001400                BIS     #A16!A17,(SP)   ;SET HIGH ORDER UNIBUS BITS
 4182  031022  012611                        MOV     (SP)+,@R1       ;FILL RHCS1
 4183  031024  052777  000010  163722        BIS     #BAI,@RHCS2     ;SET BUS ADDRESS INHIBIT
 4184  031032  005037  015124                CLR     @#ERFLG$        ;CLEAR ERROR FLAG
 4185  031036  004737  052514                JSR     PC,@#COMHD      ;WRITE DATA
```

```
4186
4187                                    ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
4188                                    ;FROM THE ''COMHD'' ROUTINE IT MEANS SECTOR GAP, SYNC BYTE
4189                                    ;HEADER, HEADER CRC, HEADER GAP AND SYNC BYTE HAVE GONE BY
4190                                    ;AND SYNCS WERE CORRECTLY DETECTED.
4191
4192                                    ;DATA IS TO BE CHECKED.
4193
4194  031042  004737  045470                      JSR     PC,@#PUTREG     ;SAVE REGISTERS
4195  031046  005737  015124                      TST     @#ERFLG$        ;HAVE ANY ERRORS OCCURED?
4196  031052  001042                              BNE     TST47           ;       BRANCH OUT IF YES
4197  031054  013700  015140                      MOV     @#TMPILL,R0     ;GOOD DATA
4198  031060  012701  054572                      MOV     #DISK,R1        ;DATA WRITTEN INTO ''DISK''
4199  031064  012702  000400                      MOV     #256.,R2        ;COUNTER
4200  031070  012737  000401  052774  1$:         MOV     #257.,@#ERWORD  ;FOR ERROR WORD
4201  031076  020021                              CMP     R0,(R1)+        ;COMPARE GOOD DATA WITH DATA ON DISK
4202  031100  001425                              BEQ     3$              ;BRANCH IF GOOD
4203  031102  013737  015140  001124              MOV     @#TMPILL,@#$GDDAT;GOOD DATA
4204  031110  014137  001126                      MOV     -(R1),@#$BDDAT  ;BAD DATA
4205  031114  160237  052774                      SUB     R2,@#ERWORD     ;ERROR WORD NO
4206  031120  005737  015124                      TST     @#ERFLG$        ;ANY ERRORS ALREADY THERE?
4207  031124  001002                              BNE     2$              ;BRANCH IF YES
4208  031126  104037                              ERROR   37              ;ERROR ON WRITE DATA COMMAND
4209                                                                      ;SEE NEXT ERROR COMMENTS
4210  031130  000401                              BR      64$             ;BRANCH TO AVOID PRINTING NEXT ERROR
4211  031132  104040                      2$:     ERROR   40              ;WORD NO GIVES WORD IN ERROR
4212                                                                      ;ERROR OCCURED WHILE WRITING
4213                                                                      ;WITH A16 A17 OF RHCS1 SET
4214  031134  005721                      64$:    TST     (R1)+           ;UNDO -(R1) FOR BAD DATA
4215  031136  017746  147776                      MOV     @SWR,-(SP)      ;GET SWITCH SETTING
4216  031142  042716  177177                      BIC     #177177,(SP)    ;KEEP ONLY SWITCH 7 AND 8
4217  031146  022726  000200                      CMP     #SW07,(SP)+     ;IS 7 SET AND 8 RESET
4218  031152  001402                              BEQ     TST47           ;       BRANCH OUT IF YES
4219  031154  005302                      3$:     DEC     R2              ;IF NOT COUNT 256 WORDS
4220  031156  001344                              BNE     1$              ;BRANCH IF 256 NOT DONE
```

```
4221
4222  031160  000004                    TST47:  SCOPE
4223  031162  012706  001000                    MOV     #STACK,SP       ;RESET STACK
4224  031166  012737  000047  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
4225
4226                                            ;*THESE ARE TO SETUP FOR DISKLESS USE
4227
4228  031174  012737  010000  052654            MOV     #FMT22,@#CYL    ;16 BITS PER WORD
4229                                                                    ;CYLINDER 0
4230  031202  005037  052656                    CLR     @#SECOTR        ;SECTOR 0
4231                                                                    ;TRACK 0
4232  031206  005037  052660                    CLR     @#KEY1          ;KEY1 = 0
4233  031212  005037  052662                    CLR     @#KEY2          ;KEY2 = 0
4234  031216  005037  052664                    CLR     @#X             ;THIS IS A READ COMMAND
4235  031222  004537  047176                    JSR     R5,@#CRC        ;GO TO CALCULATE CRC
4236  031226  052654                            CYL
4237  031230  054554                            WCRC
4238
4239                                            ;* THESE        ARE REGULAR SETUPS
4240
4241  031232  004737  045770                    JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
4242  031236  012777  177374  163504            MOV     #-260.,@RHWC    ;256 DATA WORDS, 4 HEADER WORDS
4243  031244  012777  016264  163500            MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF BUFFER
4244  031252  005077  163504                    CLR     @RHDST          ;TRACK = 0
4245                                                                    ;SECTOR = 0
4246  031256  012777  014000  163502            MOV     #FMT22!ECI,@RHOF ;16 BITS PER WORD
4247                                                                    ;ECC CORRECTION INHIBITED
4248  031264  005077  163500                    CLR     @RHCA           ;CYLINDER = 0
4249  031270  004737  046024                    JSR     PC,@#CHECKT     ;CHECK DVA,RDY,DPR,DRY = 1 AND OTHERS DON'T
4250  031274  104401  005123                    TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF THE
4251  031300  000000                            HALT                    ;STOP THE TEST
4252  031302  013711  015176                    MOV     @#REFOR,@R1     ;READ HEADER AND DATA = 72
4253
4254                                            ;*READ & SAVE REGISTERS FOR COMPARISON AFTER SIMULATED 'DTE'
4255
4256  031306  004037  046462                    JSR     R0,@#SAVER      ;READ IN SEQUENCE
4257  031312  014750                            RHWC                    ;FROM HARDWARE REGISTER
4258  031314  015024                            WC                      ;INTO CORE AT LOCATION
4259  031316  000023                            19.                     ;NUMBER OF REGISTERS TO READ
4260
4261
4262                                            ;*NOW 'GO' WILL BE GIVEN, EVERYTHING WILL BE TREATED
4263                                            ;*NORMALLY FOR THE HEADER. BUT WHEN IT IS TIME TO READ
4264                                            ;*DATA, ONLY SECTOR CLOCKS WILL BE GIVEN. NO DIAGNOSTIC DATA
4265                                            ;*CLOCKS WILL BE GIVEN. THIS SHOULD BRING SECTOR PULSE HIGH
4266                                            ;*WITHOUT PUTTING 'READ'' DOWN HENCE 'DTE' WILL COME UP.
4267
4268  031320  012737  177777  015144            MOV     #-1,@#TESDTE    ;SET DTE TEST
4269  031326  012737  177777  015124            MOV     #-1,@#ERFLG$    ;THIS WILL BRING THE READ HEADER
4270                                                                    ;AND DATA PROCESS OUT AFTER THE
4271                                                                    ;HEADER HAS BEEN CORRECTLY READ
4272
4273  031334  004737  052514                    JSR     PC,@#COMHD      ;ISSUE 'GO', SEARCH FOR THE SECTOR
4274                                                                    ;AND READ THE HEADER
4275
4276  031340  017737  163412  001176  SETCK1: MOV     @RHCS1,@#$TMP0  ;READ CS1 TO CHECK FOR ANY READ ERRORS
```

```
4277  031346  032737  100000  001176          BIT     #SC,@#$TMP0     ;TEST FOR ''SPECIAL CONDITION'' - 'SC'
4278  031354  001405                          BEQ     8$              ;CONTINUE WITH TEST IF 'SC' = 0 (NO ERRORS)
4279  031356  004737  045470                  JSR     PC,@#PUTREG     ;READ & SAVE ALL REGISTERS AGAIN IF AN
4280                                                                  ;UNDEFINED DATA TRANSFER ERROR OCCURRED
4281  031362  104040                          ERROR   40              ;READ/WRITE HEADER & DATA ERROR DURING
4282  031364  000137  031652                  JMP     TST50           ;       'DTE' TEST - ABORT THE TEST
4283
4284  031370                          8$:     ;NOW THE HEADER HAS BEEN READ OK
4285                                          ;*NOW 560 SECTOR CLOCKS WILL BE GIVEN
4286                                          ;*GAP 11 BYTES, SYNC 1 BYTE, DATA 512, ECC 4 BYTES
4287                                          ;*GAP 2 BYTES, TOLERANCE 28 BYTES, EXTRA 2
4288
4289                                          ;*THESE 560 SECTOR CLOCKS ARE DIVIDED INTO TWO GROUPS
4290                                          ;*24 SECTOR CLOCKS WITH NORMAL DIAGNOSTIC DATA CLOCKS,
4291                                          ;*AND 536 SECTOR CLOCKS WITHOUT ANY DIAGNOSTIC DATA CLOCKS.
4292
4293
4294                                          ;*THIS GIVES 24 SECTOR CLOCKS WITH DIAGNOSTIC DATA CLOCKS
4295                                          ;*WHICH EQUALS 3 BYTES OF DATA
4296
4297  031370  012701  000030                  MOV     #24.,R1         ;LOAD COUNTER
4298  031374  013700  014776                  MOV     @#RHMR,R0       ;GET RHMR ADDRESS
4299  031400  012710  000001                  MOV     #DMD,@R0        ;SET DIAGNOSTIC MODE
4300  031404  052710  000012          1$:     BIS     #MSTCK!MCLK,@R0 ;SET SECTOR CLOCK AND DATA CLOCK
4301  031410  042710  000012                  BIC     #MSTCK!MCLK,@R0 ;CLEAR SECTOR CLOCK AND DATA CLOCK
4302  031414  012702  000007                  MOV     #7,R2           ;LOAD COUNTER FOR DIAGNOSTIC CLOCKS
4303  031420  052710  000002          4$:     BIS     #MCLK,@R0       ;SET CLOCK
4304  031424  042710  000002                  BIC     #MCLK,@R0       ;CLEAR CLOCK
4305  031430  005302                          DEC     R2              ;COUNT TO 7
4306  031432  001372                          BNE     4$              ;BRANCH IF 7 NOT DONE
4307  031434  005301                          DEC     R1              ;COUNT TO 24
4308  031436  001362                          BNE     1$              ;BRANCH IF 24 NOT DONE
4309
4310                                          ;*THIS GIVES 536 SECTOR CLOCKS WITHOUT DIAGNOSTIC DATA CLOCKS
4311
4312  031440  012701  001030                  MOV     #536.,R1        ;LOAD SECTOR CLOCK COUNTER
4313  031444  052710  000010          5$:     BIS     #MSTCK,@R0      ;SET SECTOR CLOCK
4314  031450  042710  000010                  BIC     #MSTCK,@R0      ;CLEAR SECTOR CLOCK
4315  031454  005301                          DEC     R1              ;COUNT
4316  031456  001372                          BNE     5$              ;BRANCH IF 536 NOT DONE
4317
4318
4319                                          ;*NOW 'DTE' SHOULD BE SET
4320                                          ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
4321
4322  031460  012737  177400  015024          MOV     #-256.,@#WC     ;SAVED RHWC
4323  031466  012737  016274  015026          MOV     #REINTO+<4.*2>,@#BA ;SAVED RHBA
4324  031474  052737  140000  015032          BIS     #SC!TRE,@#CS1   ;SAVED RHCS1
4325  031502  052737  010000  015034          BIS     #DTE,@#ER1      ;SAVED RHER1
4326  031510  012737  000401  015052          MOV     #401,@#MR       ;SAVED RHMR
4327  031516  052737  140000  015054          BIS     #ATA!ERR,@#DS1 ;SAVED RHDS1
4328  031524  012737  000100  015066          MOV     #100,@#LA       ;SAVED RHLA
4329  031532  012737  000001  015036          MOV     #1,@#DST        ;SAVED RHDST
4330  031540  013737  015134  015050          MOV     @#ATTENT,@#AS   ;SAVED RHAS
4331
4332
```

```
4333                                        ;*NOW READ & SAVE REGISTERS AGAIN SO THAT COMPARISONS
4334                                        ;*CAN BE DONE (USE THE 'WRFROM' SAVE BUFFER THIS TIME)
4335
4336  031546  004037  046462                JSR     R0,@#SAVER      ;READ IN SEQUENCE
4337  031552  014750                        RHWC                    ;FROM HARDWARE REGISTER
4338  031554  015220                        WRFROM                  ;INTO CORE BUFFER
4339  031556  000023                        19.                     ;NUMBER OF REGISTERS TO READ
4340
4341                                        ;*FOR RHAS UPPER BYTE
4342  031560  113737  015051  015245        MOVB    @#AS+1,@#WRFROM+25 ;UPPER RHAS
4343
4344                                        ;*COMPARE THE HEADER READ
4345
4346  031566  004037  046664                JSR     R0,@#COMPAR     ;COMPARE
4347  031572  052654                        CYL                     ;GOOD BUFFER
4348  031574  016264                        REINTO                  ;TEST BUFFER
4349  031576  000004                        4.                      ;NUMBER
4350  031600  031606                        6$                      ;RETURN FOR ERROR
4351  031602  031606                        6$                      ;SAME
4352  031604  031612                        7$                      ;RETURN FOR GOOD COMPARISON
4353  031606  104010              6$:       ERROR   10              ;HEADER READ IN DURING THIS TEST IS
4354                                                                ;IN ERROR
4355
4356  031610  000207                        RTS     PC              ;RETURN
4357
4358  031612                    7$:                                 ;GOOD COMPARISON, CONTINUE
4359
4360
4361                                        ;*COMPARE REGISTERS BEFORE COMMAND WITH REGISTERS AFTER COMMAND
4362
4363  031612  004037  046664                JSR     R0,@#COMPAR     ;COMPARE
4364  031616  015024                        WC                      ;INITIAL SNAPSHOT BUFFER (CHANGED)
4365  031620  015220                        WRFROM                  ;TEST SNAPSHOT BUFFER
4366  031622  000022                        18.                     ;NUMBER OF REGISTERS
4367  031624  031632                        2$                      ;RETURN FOR ERROR
4368  031626  031632                        2$                      ;SAME
4369  031630  031652                        3$                      ;RETURN FOR GOOD COMPARISON
4370
4371  031632  013705  052774          2$:   MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
4372  031636  060505                        ADD     R5,R5           ;DOUBLE ERROR WORD
4373  031640  016537  014746  045534        MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER
4374  031646  104001                        ERROR   1               ;IMPROPER REGISTER
4375                                                                ;CHANGE AFTER FORCING
4376                                                                ;'DTE' ERROR
4377  031650  000207                        RTS     PC              ;RETURN
4378
4379  031652                    3$:                                 ;GOOD - REGISTERS OK, GO ON TO NEXT TEST
4380
4381
```

```
4382
4383  031652  000004                      TST50:  SCOPE
4384  031654  012706  001000                      MOV     #STACK,SP       ;RESET STACK
4385  031660  012737  000050  017330              MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
4386                                              ;*THESE ARE TO SETUP FOR DISKLESS USE
4387
4388  031666  012737  010000  052654              MOV     #FMT22,@#CYL    ;16 BITS PER WORD
4389                                                                      ;CYLINDER 0
4390  031674  005037  052656                      CLR     @#SECOTR        ;SECTOR 0
4391                                                                      ;TRACK 0
4392  031700  005037  052660                      CLR     @#KEY1          ;KEY1 = 0
4393  031704  005037  052662                      CLR     @#KEY2          ;KEY2 = 0
4394  031710  012737  177777  052664              MOV     #-1,@#X         ;THIS IS FOR WRITE DATA COMMAND
4395  031716  004537  047176                      JSR     R5,@#CRC        ;GO TO CALCULATE CRC
4396  031722  052654                              CYL
4397  031724  054554                              WCRC
4398
4399                                              ;* THESE        ARE REGULAR SETUPS & CHECKS
4400
4401  031726  004737  045770                      JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
4402  031732  012777  177400  163010              MOV     #-256.,@RHWC    ;256 DATA WORDS
4403  031740  012777  015220  163004              MOV     #WRFROM,@RHBA   ;STARTING ADDRESS OF BUFFER
4404  031746  005077  163010                      CLR     @RHDST          ;TRACK = 0
4405                                                                      ;SECTOR = 0
4406  031752  012777  010000  163006              MOV     #FMT22,@RHOF    ;16 BITS PER WORD
4407                                                                      ;ECC CORRECTION INHIBITED
4408  031760  005077  163004                      CLR     @RHCA           ;CYLINDER = 0
4409  031764  004737  046024                      JSR     PC,@#CHECKT     ;CHECK DVA,RDY,DPR,DRY = 1 AND OTHERS DON'T
4410  031770  104401  005123                      TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF THE
4411  031774  000000                              HALT                    ;STOP THE TEST
4412  031776  013711  015170                      MOV     @#WRIDAT,@R1    ;WRITE DATA = 60
4413
4414                                              ;*READ & SAVE REGISTERS FOR COMPARISON AFTER SIMULATED 'DTE'
4415
4416  032002  004037  046462                      JSR     R0,@#SAVER      ;READ REGISTERS IN SEQUENCE
4417  032006  014750                              RHWC                    ;FROM HARDWARE REGISTER
4418  032010  015024                              WC                      ;INTO CORE BUFFER LOCATION
4419  032012  000023                              19.                     ;NUMBER OF REGISTERS TO READ
4420
4421                                              ;*NOW 'GO' WILL BE GIVEN, EVERYTHING WILL BE TREATED
4422                                              ;*NORMALLY FOR THE HEADER.  WHEN IT IS TIME TO WRITE
4423                                              ;*DATA, ONLY SECTOR CLOCKS WILL BE GIVEN, NO DIAGNOSTIC DATA
4424                                              ;*CLOCKS WILL BE GIVEN. THIS SHOULD BRING SECTOR PULSE HIGH
4425                                              ;*WITHOUT PUTTING READ DOWN, HENCE 'DTE' WILL COME UP.
4426
4427  032014  012737  177777  015144              MOV     #-1,@#TESDTE    ;SET DTE TEST
4428  032022  012737  177777  015124              MOV     #-1,@#ERFLG$    ;THIS WILL BRING THE READ HEADER
4429                                                                      ;AND DATA PROCESS OUT AFTER THE
4430                                                                      ;HEADER HAS BEEN CORRECTLY READ
4431
4432  032030  004737  052514                      JSR     PC,@#COMHD      ;ISSUE 'GO', SEARCH FOR SECTOR,
4433                                                                      ;READ HEADER AND DATA.
4434
4435  032034  017737  162716  001176  SETCK2:     MOV     @RHCS1,@#$TMP0  ;READ CS1 TO CHECK FOR READ ERRORS
4436  032042  032737  100000  001176              BIT     #SC,@#$TMP0     ;TEST FOR "SPECIAL CONDITION" - 'SC'
4437  032050  001405                              BEQ     6$              ;CONTINUE TESTING IF NO ERROR
```

```
4438  032052  004737  045470              JSR     PC,@#PUTREG     ;READ & SAVE REGISTERS AGAIN IF UNDE-
4439                                                              ;FINED ERROR HAS OCCURRED
4440  032056  104040                      ERROR   40              ;THERE WAS A READ/WRITE HEADER ERROR
4441                                                              ;DURING 'DTE' TEST SETUP
4442  032060  000137  032376              JMP     TST51           ;       ABORT THE TEST
4443
4444  032064                      6$:     ;NOW THE HEADER HAS BEEN READ,
4445                                      ;*560 SECTOR CLOCKS WILL BE GIVEN -
4446                                      ;*GAP 11 BYTES, SYNC 1 BYTE, DATA 512, ECC 4 BYTES
4447                                      ;*GAP 2 BYTES, TOLERANCE 28 BYTES, EXTRA 2
4448
4449                                      ;*THESE 560 SECTOR CLOCKS ARE DIVIDED INTO TWO GROUPS
4450                                      ;*24 SECTOR CLOCKS WITH NORMAL DIAGNOSTIC DATA CLOCKS
4451                                      ;*AND 536 SECTOR CLOCKS WITHOUT ANY DIAGNOSTIC DATA CLOCKS
4452
4453
4454                                      ;*THIS GIVES 24 SECTOR CLOCKS WITH DIAGNOSTIC DATA CLOCKS
4455
4456  032064  012701  000030              MOV     #24.,R1         ;LOAD SECTOR CLOCK COUNTER
4457  032070  013700  014776              MOV     @#RHMR,R0       ;GET RHMR ADDRESS
4458  032074  012710  000001              MOV     #DMD,@R0        ;SET DIAGNOSTIC MODE
4459  032100  052710  000012      1$:     BIS     #MSTCK!MCLK,@R0 ;SET SECTOR CLOCK AND DATA CLOCK
4460  032104  042710  000012              BIC     #MSTCK!MCLK,@R0 ;CLEAR SECTOR CLOCK AND DATA CLOCK
4461  032110  012702  000007              MOV     #7,R2           ;LOAD COUNTER FOR DIAGNOSTIC DATA CLOCKS
4462  032114  052710  000002      4$:     BIS     #MCLK,@R0       ;SET CLOCK (DATA)
4463  032120  042710  000002              BIC     #MCLK,@R0       ;CLEAR CLOCK (DATA)
4464  032124  005302                      DEC     R2              ;COUNT
4465  032126  001372                      BNE     4$              ;BRANCH IF 7 NOT DONE
4466  032130  005301                      DEC     R1              ;COUNT
4467  032132  001362                      BNE     1$              ;BRANCH IF 24 NOT DONE
4468
4469                                      ;*THIS GIVES 536 SECTOR CLOCKS WITHOUT DIAGNOSTIC DATA CLOCKS
4470
4471  032134  012701  001030              MOV     #536.,R1        ;LOAD SECTOR CLOCK COUNTER
4472  032140  052710  000010      5$:     BIS     #MSTCK,@R0      ;SET SECTOR CLOCK
4473  032144  042710  000010              BIC     #MSTCK,@R0      ;CLEAR SECTOR CLOCK
4474  032150  005301                      DEC     R1              ;COUNT
4475  032152  001372                      BNE     5$              ;BRANCH IF 536 NOT DONE
4476
4477
4478
4479                                      ;*ECC PATTERN REGISTER IS NOT CHECKED
4480  032154  017737  162630  015064      MOV     @RHEC2,@#EC2    ;RHEC2 IS NOT CHECKED
4481
4482
4483                                      ;*NOW 'DTE' SHOULD BE SET, CHANGE SAVED REGISTERS TO EXPECTED VALUES
4484
4485  032162  005737  017334              TST     @#RH70          ;CHECK FOR RH70 CONTROLLER
4486  032166  001412                      BEQ     7$              ;SKIP RH70 CODE AND DO RH11 IF NOT
4487
4488  032170  012737  177416  015024      MOV     #-242.,@#WC     ;SAVED RHWC
4489  032176  012737  015254  015026      MOV     #WRFROM+<14.*2>,@#BA ;SAVED RHBA
4490  032204  052737  000300  015030      BIS     #IR!OR,@#CS2    ;SAVED RHCS2
4491  032212  000414                      BR      8$              ;SKIP NEXT RH11 CODE
4492
4493  032214  012737  177511  015024  7$: MOV     #-183.,@#WC     ;SAVED RHWC
```

```
4494  032222  012737  015442  015026          MOV     #WRFROM+<73.*2>,@#BA ;SAVED RHBA
4495  032230  042737  000100  015030          BIC     #IR,@#CS2           ;SAVED RHCS2
4496  032236  052737  000200  015030          BIS     #OR,@#CS2           ;SAVED RHCS2
4497
4498  032244  052737  140000  015032  8$:     BIS     #SC!TRE,@#CS1       ;SAVED RHCS1
4499  032252  052737  010000  015034          BIS     #DTE,@#ER1          ;SAVED RHER1
4500  032260  012737  000201  015052          MOV     #DENVL!DMD,@#MR     ;SAVED RHMR
4501  032266  052737  140000  015054          BIS     #ATA!ERR,@#DS1 ;SAVED RHDS1
4502  032274  012737  000100  015066          MOV     #100,@#LA           ;SAVED RHLA
4503  032302  012737  000001  015036          MOV     #1,@#DST            ;SAVED RHDST
4504  032310  013737  015134  015050          MOV     @#ATTENT,@#AS       ;SAVED RHAS
4505
4506                                          ;*NOW READ & SAVE REGISTERS AGAIN SO THAT COMPARISONS
4507                                          ;CAN BE DONE (USING 'WRFROM' BUFFER THIS TIME)
4508
4509  032316  004037  046462                  JSR     R0,@#SAVER          ;READ IN SEQUENCE
4510  032322  014750                          RHWC                        ;FROM HARDWARE REGISTER
4511  032324  016264                          REINTO                      ;INTO CORE BUFFER LOCATION
4512  032326  000023                          19.                         ;NUMBER OF REGISTERS TO READ
4513
4514                                          ;*FOR RHAS UPPER BYTE
4515  032330  113737  015051  016311          MOVB    @#AS+1,@#REINTO+25 ;UPPER RHAS
4516
4517
4518                                          ;*COMPARE CHANGED REGISTER SNAPSHOT BEFORE COMMAND WITH
4519                                          ;*SNAPSHOT AFTER COMMAND
4520
4521  032336  004037  046664                  JSR     R0,@#COMPAR         ;COMPARE
4522  032342  015024                          WC                          ;CHANGED INITIAL SNAPSHOT BUFFER
4523  032344  016264                          REINTO                      ;SNAPSHORT BUFFER AFTER COMMAND
4524  032346  000022                          18.                         ;NUMBER OF REGISTERS TO COMPARE
4525  032350  032356                          2$                          ;RETURN FOR ERROR
4526  032352  032356                          2$                          ;SAME
4527  032354  032376                          3$                          ;RETURN FOR GOOD COMPARISON
4528
4529  032356  013705  052774          2$:     MOV     @#ERWORD,R5         ;GETTING READY TO INDEX
4530  032362  060505                          ADD     R5,R5               ;DOUBLE ERROR WORD
4531  032364  016537  014746  045534          MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER
4532  032372  104001                          ERROR   1                   ;IMPROPER REGISTER
4533                                                                      ;CHANGE AFTER FORCING
4534                                                                      ;'DTE' ERROR
4535  032374  000207                          RTS     PC                  ;RETURN
4536
4537  032376                          3$:                                 ;GOOD, REGISTERS OK - GO ON TO NEXT TEST
4538
4539
```

```
4540
4541  032376  000004                    TST51:  SCOPE
4542  032400  012706  001000                    MOV     #STACK,SP       ;RESET STACK
4543  032404  012737  000051  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
4544
4545                                            ;*THESE ARE TO SET UP FOR DISKLESS USE ONLY
4546
4547  032412  012737  010000  056012            MOV     #FMT22,@#WCYL   ;FORMAT 22=16 BITWORDS AND
4548                                                                    ;CYLINDER 0
4549  032420  005037  056014                    CLR     @#WSECTR        ;TRACK=0, SECTOR=0
4550  032424  005037  056016                    CLR     @#WKEY1         ;KEY1=0
4551  032430  005037  056020                    CLR     @#WKEY2         ;KEY2=0
4552  032434  012737  000400  056052            MOV     #256.,@#FNWORD  ;256 DATAWORDS
4553  032442  004537  047176                    JSR     R5,@#CRC        ;GO TO CALCULATE CRC
4554  032446  056012                            WCYL
4555  032450  056022                            GCRC
4556
4557                                            ;* THESE        ARE REGULAR SETUPS & CHECKS
4558
4559  032452  004737  045770                    JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
4560  032456  012777  177374  162264            MOV     #-260.,@RHWC    ;256 DATA WORDS & 4 HEADER WORDS
4561  032464  012777  015220  162260            MOV     #WRFROM,@RHBA   ;STARTING ADDRESS OF BUFFER
4562  032472  005077  162264                    CLR     @RHDST          ;TRACK = 0
4563                                                                    ;SECTOR = 0
4564  032476  012777  014000  162262            MOV     #FMT22!ECI,@RHOF ;16 BITS PER WORD
4565                                                                    ;ECC CORRECTION INHIBITED
4566  032504  005077  162260                    CLR     @RHCA           ;CYLINDER = 0
4567  032510  004737  046024                    JSR     PC,@#CHECKT     ;CHECK DVA,RDY,DPR,DRY = 1 AND OTHERS DON'T
4568  032514  104401  005123                    TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF THE
4569  032520  000000                            HALT                    ;STOP THE TEST
4570  032522  013711  015172                    MOV     @#WRIFOR,@R1    ;WRITE HEADER AND DATA = 62
4571
4572                                            ;*READ & SAVE REGISTERS FOR COMPARISON AFTER SIMULATED 'DTE'
4573
4574  032526  004037  046462                    JSR     R0,@#SAVER      ;READ IN SEQUENCE
4575  032532  014750                            RHWC                    ;FROM HARDWARE REGISTER
4576  032534  015024                            WC                      ;INTO CORE BUFFER LOCATION
4577  032536  000023                            19.                     ;NUMBER OF REGISTERS TO READ
```

```
4578
4579                                    ;*NOW 'GO' WILL BE GIVEN. EVERYTHING WILL BE TREATED
4580                                    ;*NORMALLY TILL HEADER IS TO BE GIVEN, THEN ONLY
4581                                    ;*SECTOR CLOCKS WILL BE GIVEN. NO DIAGNOSTIC DATA
4582                                    ;*CLOCKS WILL BE GIVEN. THIS SHOULD BRING SECTOR PULSE HIGH
4583                                    ;*WITHOUT PUTTING ''READ'' DOWN, HENCE 'DTE' WILL COME UP.
4584
4585  032540  012737  177777  015144        MOV     #-1,@#TESDTE    ;SET DTE TEST
4586  032546  012737  177777  015124        MOV     #-1,@#ERFLG$    ;THIS WILL BRING THE READ HEADER
4587                                                                ;AND DATA PROCESS OUT AFTER THE
4588                                                                ;HEADER HAS BEEN CORRECTLY READ
4589
4590  032554  004737  055636               JSR     PC,@#COMWHD     ;ISSUE 'GO', SEARCH FOR SECTOR,
4591                                                                ;WRITE HEADER AND DATA.
4592
4593  032560  017737  162172  001176  SETCK3: MOV    @RHCS1,@#$TMP0  ;READ CS1 TO CHECK FOR ERRORS DURING WRITE
4594  032566  032737  100000  001176        BIT     #SC,@#$TMP0     ;TEST FOR ''SPECIAL CONDITION'' - 'SC'
4595  032574  001405                       BEQ     4$              ;CONTINUE TEST IF NO ERROR ('SC' = 0)
4596  032576  004737  045470               JSR     PC,@#PUTREG     ;READ & SAVE REGISTERS AGAIN IF ERROR
4597  032602  104040                       ERROR   40              ;THERE WAS A READ/WRITE HEADER ERROR
4598                                                                ;DURING 'DTE' TEST SETUP
4599  032604  000137  033036               JMP     TST52           ;       ABORT THE TEST AT THAT POINT
4600
4601  032610                          4$:     ;NOW SECTOR HAS BEEN FOUND OK
4602
4603                                    ;*609 SECTOR CLOCKS WILL BE GIVEN,
4604                                    ;*39 BYTES FOR SECTOR GAP,
4605                                    ;*1 BYTE FOR HEADER SYNC,
4606                                    ;*8 BYTES FOR HEADER,
4607                                    ;*GAP 11 BYTES, SYNC 1 BYTE, DATA 512, ECC 4 BYTES
4608                                    ;*GAP 2 BYTES, TOLERANCE 28 BYTES, EXTRA 3
4609
4610
4611                                    ;*THIS GIVES 609 SECTOR CLOCKS WITHOUT DIAGNOSTIC DATA CLOCKS
4612
4613  032610  012701  001141           MOV     #609.,R1        ;LOAD SECTOR CLOCK COUNTER
4614  032614  052710  000010      5$:  BIS     #MSTCK,@R0      ;SET SECTOR CLOCK
4615  032620  042710  000010           BIC     #MSTCK,@R0      ;CLEAR SECTOR CLOCK
4616  032624  005301                   DEC     R1              ;COUNT
4617  032626  001372                   BNE     5$              ;BRANCH IF 536 NOT DONE
4618
4619                                    ;*NOW 'DTE' SHOULD BE SET, CHANGE SAVED REGISTERS
4620                                    ;TO EXPECTED VALUES
4621
4622  032630  005737  017334           TST     @#RH70          ;CHECK FOR RH70 CONTROLLER
4623  032634  001407                   BEQ     6$              ;SKIP RH70 CODE AND DO RH11 IF NOT
4624
4625  032636  012737  177404  015024   MOV     #-252.,@#WC     ;SAVED RHWC
4626  032644  012737  015240  015026   MOV     #WRFROM+<8.*2>,@#BA ;SAVED RHBA
4627  032652  000406                   BR      7$              ;SKIP NEXT RH11 CODE
4628
4629  032654  012737  177477  015024  6$:  MOV     #-193.,@#WC     ;SAVED RHWC
4630  032662  012737  015426  015026       MOV     #WRFROM+<67.*2>,@#BA ;SAVED RHBA
4631
4632  032670  052737  140000  015032  7$:  BIS     #SC!TRE,@#CS1   ;SAVED RHCS1
4633  032676  042737  000100  015030       BIC     #IR,@#CS2       ;SAVED RHCS2
```

```
4634  032704  052737  000200  015030          BIS     #OR,@#CS2       ;SAVED RHCS2
4635  032712  052737  010000  015034          BIS     #DTE,@#ER1      ;SAVED RHER1
4636  032720  012737  000401  015052          MOV     #401,@#MR       ;SAVED RHMR
4637  032726  052737  140000  015054          BIS     #ATA!ERR,@#DS1  ;SAVED RHDS1
4638  032734  012737  000100  015066          MOV     #100,@#LA       ;SAVED RHLA
4639  032742  012737  000001  015036          MOV     #1,@#DST        ;SAVED RHDST
4640  032750  013737  015134  015050          MOV     @#ATTENT,@#AS   ;SAVED RHAS
4641
4642                                          ;*NOW READ & SAVE REGISTERS AGAIN SO THAT COMPARISONS CAN BE DONE
4643
4644  032756  004037  046462                  JSR     R0,@#SAVER      ;READ IN SEQUENCE
4645  032762  014750                          RHWC                    ;FROM HARDWARE REGISTER
4646  032764  016264                          REINTO                  ;INTO CORE BUFFER LOCATION
4647  032766  000023                          19.                     ;NUMBER OF REGISTERS TO READ
4648
4649                                          ;*FOR RHAS UPPER BYTE
4650  032770  113737  015051  016311          MOVB    @#AS+1,@#REINTO+25 ;UPPER RHAS
4651
4652                                          ;*COMPARE CHANGED REGISTER SNAPSHOT BEFORE COMMAND
4653                                          ;*WITH REGISTER SNAPSHOT AFTER COMMAND
4654
4655  032776  004037  046664                  JSR     R0,@#COMPAR     ;COMPARE
4656  033002  015024                          WC                      ;CHANGED REGISTER SNAPSHOT
4657  033004  016264                          REINTO                  ;SNAPSHOT AFTER COMMAND
4658  033006  000022                          18.                     ;NUMBER OF REGISTERS TO COMPARE
4659  033010  033016                          2$                      ;RETURN FOR ERROR
4660  033012  033016                          2$                      ;SAME
4661  033014  033036                          3$                      ;RETURN FOR GOOD COMPARISON
4662
4663  033016  013705  052774          2$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
4664  033022  060505                  ADD     R5,R5                   ;DOUBLE ERROR WORD
4665  033024  016537  014746  045534          MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER
4666  033032  104001                  ERROR   1                       ;IMPROPER REGISTER
4667                                                                  ;CHANGE AFTER FORCING
4668                                                                  ;'DTE' ERROR
4669  033034  000207                  RTS     PC                      ;RETURN
4670
4671  033036                  3$:                                     ;GOOD, REGISTERS COMPARE OK
4672                                                                  ;GO ON TO THE NEXT TEST
4673
```

```
 4674  033036  000004                    TST52:  SCOPE
 4675  033040  012706  001000                    MOV     #STACK,SP       ;RESET STACK
 4676  033044  012737  000052  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
 4677  033052  004737  045770                    JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS & CLEAR
 4678                                                                    ;THE DRIVE
 4679  033056  012737  000026  015146            MOV     #22.,@#TAGDTE   ;22 SECTORS
 4680                                                                    ;THIS TEST REPEATS
 4681                                                                    ;ITSELF 22 TIMES
 4682
 4683                                            ;*THE FOLLOWING INITIALIZES FOR SECTOR 0
 4684
 4685  033064  005037  033170                    CLR     @#SS3+2         ;HEADER (SECTOR)
 4686  033070  012737  000025  033174            MOV     #21.,@#SS4+2    ;HEADER (KEY1)
 4687  033076  012737  000025  033200            MOV     #21.,@#SS5+2    ;HEADER (KEY2)
 4688  033104  005037  033226                    CLR     @#SS7+2         ;DATA (SECTOR)
 4689  033110  005037  033310                    CLR     @#SS10+2        ;DATA
 4690  033114  005037  033340                    CLR     @#SS12+2        ;SECTOR (SIMULATED DISK)
 4691  033120  012737  000025  033346            MOV     #21.,@#SS13+2   ;KEY1 (SIMULATED DISK)
 4692  033126  012737  000025  033354            MOV     #21.,@#SS14+2   ;KEY2 (SIMULATED DISK)
 4693  033134  005037  033414                    CLR     @#SS15+2        ;SECTOR (RHDST)
 4694
 4695                                            ;*CLEAR SIMULATED DISK AREA
 4696  033140                            SS1:
 4697  033140  012700  054474           1$:     MOV     #SECGAP,R0      ;POINTER
 4698  033144  012701  000460                    MOV     #304.,R1        ;COUNTER
 4699  033150  005020                    2$:     CLR     (R0)+           ;CLEAR SIMULATED DISK AREA
 4700  033152  005301                            DEC     R1              ;COUNT
 4701  033154  001375                            BNE     2$
 4702
 4703
 4704                                            ;*SETUP WRITE FROM BUFFER
 4705  033156  012700  015220                    MOV     #WRFROM,R0
 4706
 4707                                            ;*HEADER
 4708  033162  012720  010000                    MOV     #FMT22,(R0)+    ;FORMAT 16 BITS PER WORD
 4709                                                                    ;CYLINDER 0
 4710  033166  012720  000000            SS3:    MOV     #0,(R0)+        ;SECTOR TO VARY
 4711  033172  012720  000025            SS4:    MOV     #21.,(R0)+      ;KEY1 TO VARY
 4712  033176  012720  000025            SS5:    MOV     #21.,(R0)+      ;KEY2 TO VARY
 4713
 4714                                            ;*DATA IN WRITE FROM BUFFER ALTHOUGH THIS IS DATA AND NOT
 4715                                            ;*HEADER, THE SECTOR WITH SYNC BYTES WILL BE GIVEN AS DATA.
 4716
 4717                                            ;*DATA IS - 19 WORDS OF ZEROS - SYNC WORDS, 4 HEADER WORDS
 4718                                            ;*1 CRC WORD, 5 WORDS OF ZEROS, 1 SYNC WORD, 100 ZEROS
 4719                                            ;*(DATA), 1 SYNC WORD, 70 SECTOR NUMBER TO VARY
 4720
 4721  033202  012705  000023                    MOV     #19.,R5         ;COUNTER
 4722  033206  005020                    6$:     CLR     (R0)+           ;19 ZEROS
 4723  033210  005305                            DEC     R5              ;COUNT
 4724  033212  001375                            BNE     6$              ;19 DONE?
 4725  033214  013720  052756                    MOV     @#RSYNC,(R0)+   ;SYNC = 14400
 4726  033220  012720  010000                    MOV     #FMT22,(R0)+    ;CYLINDER 0
 4727  033224  012720  000000            SS7:    MOV     #0,(R0)+        ;SECTOR TO VARY
 4728  033230  005020                            CLR     (R0)+
 4729  033232  005020                            CLR     (R0)+
```

```
4730  033234  004537  047176                JSR     R5,@#CRC        ;CALCULATE CRC FOR ABOVE 4 WORDS
4731  033240  015270                        WRFROM+50               ;4 WORDS START FROM HERE
4732  033242  015300                        WRFROM+60               ;PUT CRC HERE
4733
4734  033244  005720                        TST     (R0)+           ;INCREMENT R0
4735
4736  033246  012705  000005                MOV     #5.,R5
4737  033252  005020              8$:       CLR     (R0)+           ;5 WORDS OF ZEROS
4738  033254  005305                        DEC     R5              ;COUNT
4739  033256  001375                        BNE     8$              ;BRANCH IF 5 NOT DONE
4740
4741  033260  013720  052756                MOV     @#RSYNC,(R0)+   ;SYNC = 14400
4742
4743  033264  012705  000144                MOV     #100.,R5
4744  033270  005020              9$:       CLR     (R0)+           ;100 WORDS OF ZEROS
4745  033272  005305                        DEC     R5
4746  033274  001375                        BNE     9$
4747
4748  033276  013720  052756                MOV     @#RSYNC,(R0)+   ;SYNC = 14400
4749  033302  012705  000106                MOV     #70.,R5
4750  033306  012720  000000              SS10:     MOV     #0,(R0)+        ;SECTOR TO VARY
4751  033312  005305                        DEC     R5
4752  033314  001374                        BNE     SS10
4753
4754                                        ;*CLEAR REST OF 256 WORDS THAT IS 54 WORDS OF ZEROS
4755
4756  033316  012705  000066                MOV     #54.,R5
4757  033322  005020              11$:      CLR     (R0)+
4758  033324  005305                        DEC     R5
4759  033326  001375                        BNE     11$
4760
4761                                        ;*THESE ARE TO BE SET UP FOR DISKLESS USE ONLY
4762
4763  033330  012737  010000  056012        MOV     #FMT22,@#WCYL   ;FORMAT = 16 BIT WORDS
4764                                                                ;CYLINDER = 0
4765  033336  012737  000000  056014  SS12:     MOV     #0,@#WSECTR     ;SECTOR TO VARY
4766  033344  012737  000025  056016  SS13:     MOV     #21.,@#WKEY1    ;KEY1 TO VARY
4767  033352  012737  000025  056020  SS14:     MOV     #21.,@#WKEY2    ;KEY2 TO VARY
4768  033360  012737  000312  056052        MOV     #202.,@#FNWORD  ;202 DATA WORDS
4769  033366  004537  047176                JSR     R5,@#CRC        ;CALCULATE CRC
4770  033372  056012                        WCYL                    ;FIRST WORD
4771  033374  056022                        GCRC                    ;PUT HERE
4772
4773                                        ;*THESE ARE REGULAR SETUPS
4774
4775  033376  012777  177400  161344        MOV     #-256.,@RHWC    ;202 DATA, 4 HEADER
4776  033404  012777  015220  161340        MOV     #WRFROM,@RHBA   ;FILL BUS ADDRESS
4777  033412  012777  000000  161342  SS15:     MOV     #0,@RHDST       ;SECTOR TO VARY
4778  033420  013777  015172  161330        MOV     @#WRIFOR,@RHCS1 ;GET READY TO DO
4779                                                                ;WRITE HEADER AND DATA
4780                                                                ;WITH 62 FUNCTION CODE IN RHCS1
4781  033426  012777  010000  161332        MOV     #FMT22,@RHOF    ;16 BITS PER WORD FORMAT
4782  033434  005077  161330                CLR     @RHCA           ;CYLINDER = 0
4783
4784  033440  005037  015124                CLR     @#ERFLG$        ;CLEAR ERROR FLAG
4785
```

```
 4786  033444  004737  046024                 JSR     PC,@#CHECKT     ;CHECK DVA,RDY,DPR,DRY = 1 AND OTHERS DON'T
 4787  033450  104401  005123                 TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF THE
 4788  033454  000000                         HALT                    ;STOP THE TEST
 4789
 4790  033456  004737  055636                 JSR     PC,@#COMWHD     ;ISSUE 'GO', COUNT SECTOR CLOCKS,
 4791                                                                 ;WRITE HEADER AND DATA
 4792  033462  005737  015124                 TST     @#ERFLG$        ;HAVE ANY ERRORS OCCURRED ?
 4793  033466  001051                         BNE     TST53           ;       EXIT IF YES----------------------)
 4794
 4795  033470  004737  046214                 JSR     PC,@#CHECKE     ;CHECK THAT DVA,RDY,DPR,DRY = 1
 4796  033474  104401  005123                 TYPE    ,CPHALT         ;CANNOT CONTINUE IF THEY DON'T
 4797  033500  000000                         HALT                    ;STOP THE TEST AND RESTART PROGRAM
 4798
 4799                                         ;*NOW COMPARE 'DISK' BUFFER WITH 'REINTO' BUFFER
 4800  033502  004037  046664                 JSR     R0,@#COMPAR     ;CHECK
 4801  033506  015230                         WRFROM+8.               ;GOOD BUFFER
 4802  033510  054572                         DISK                    ;TEST BUFFER
 4803  033512  000400                         256.                    ;NUMBER OF WORDS
 4804  033514  033522                         16$                     ;RETURN POINT FOR ERROR  HEADER
 4805  033516  033526                         17$                     ;RETURN POINT FOR ERROR DATA
 4806  033520  033532                         18$                     ;RETURN FOR GOOD COMPARISON
 4807  033522  104007             16$:        ERROR   7
 4808  033524  000207                         RTS     PC
 4809  033526  104010             17$:        ERROR   10
 4810  033530  000207                         RTS     PC
 4811
 4812                                         ;*THE FOLLOWING INCREMENTS ARE TO CHANGE THE ABOVE SET UP
 4813                                         ;*TO WRITE ON THE NEXT SECTOR
 4814
 4815  033532  005237  033170     18$:        INC     @#SS3+2         ;HEADER (SECTOR)
 4816  033536  005337  033174                 DEC     @#SS4+2         ;HEADER (KEY1)
 4817  033542  005337  033200                 DEC     @#SS5+2         ;HEADER (KEY2)
 4818  033546  005237  033226                 INC     @#SS7+2         ;DATA (SECTOR)
 4819  033552  005237  033310                 INC     @#SS10+2        ;DATA
 4820  033556  005237  033340                 INC     @#SS12+2        ;SECTOR (SIMULATED DISK)
 4821  033562  005337  033346                 DEC     @#SS13+2        ;KEY1 (SIMULATED DISK)
 4822  033566  005337  033354                 DEC     @#SS14+2        ;KEY2 (SIMULATED DISK)
 4823  033572  005237  033414                 INC     @#SS15+2        ;SECTOR (RHDST)
 4824
 4825  033576  005337  015146     SS2:        DEC     @#TAGDTE        ;COUNT DOWN FOR 22 SECTORS
 4826  033602  001001                         BNE     1$              ;BRANCH IF 22 SECTORS NOT DONE
 4827  033604  000402                         BR      TST53           ;ALL DONE - GO TO NEXT TEST
 4828  033606  000137  033140     1$:         JMP     @#SS1           ;GO BACK TO NEXT SECTOR
 4829
 4830
 4831
```

```
                                        .SBTTL  DATA TRANSFER TESTS USING ECC


033612  000004                  TST53:  SCOPE
033614  012706  001000                MOV     #STACK,SP       ;RESET STACK

033620  012737  000053  017330        MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
033626  012700  054474                MOV     #SECGAP,R0      ;POINTER
033632  012701  000402                MOV     #258.,R1        ;COUNTER
033636  012720  177777          1$:   MOV     #-1,(R0)+       ;FILL SIMULATER DISK WITH ONES
033642  005301                        DEC     R1
033644  001374                        BNE     1$
033646  004737  045770                JSR     PC,@#CLDISK     ;THIS IS USED TO SET GENERAL REGISTERS

                                        ;*THESE ARE FOR ECC TEST ONLY

033652  012737  177777  015142        MOV     #-1,@#TSECC     ;THIS IS AN ECC TEST
033660  005037  050476                CLR     @#POSITI        ;CLEAR ERROR POSITION COUNTER
033664  013737  050472  050474        MOV     @#NCODE,@#NCOUNT  ;TEMPORARY N-CODE COUNTER
033672  013737  050500  050506        MOV     @#HARDER,@#HADTMP  ;TEMPORARY HARD ERROR COUNTER
033700  005037  050464                CLR     @#GECC1         ;ECC LOW ORDER TO BE GENERATED
033704  005037  050466                CLR     @#GECC2         ;ECC HIGH ORDER TO BE GENERATED
033710  005037  050502                CLR     @#DATENV        ;CLEAR DATA ENVELOPE CLOCK COUNT
033714  005037  050504                CLR     @#ZCODE         ;CLEAR LEADING ZEROS CLOCK COUNT


                                        ;*THESE ARE TO BE SETUP FOR DISKLESS USE ONLY

033720  012737  010000  056012        MOV     #FMT22,@#WCYL   ;FORMAT22=16BIT WORDS AND
                                                                ;CYLINDER 0
033726  012737  000001  056014        MOV     #1,@#WSECTR     ;TRACK=0, SECTOR=1
033734  005037  056016                CLR     @#WKEY1         ;KEY1=0
033740  005037  056020                CLR     @#WKEY2         ;KEY2=0
033744  012737  000400  056052        MOV     #256.,@#FNWORD  ;256 DATA WORDS
033752  004537  047176                JSR     R5,@#CRC        ;GO TO CALCULATE CRC
033756  056012                        WCYL
033760  056022                        GCRC

                                        ;*THESE ARE REGULAR SETUPS

033762  012777  177374  160760        MOV     #-260.,@RHWC    ;256 DATA WORDS 4 HEADER WORDS
033770  012700  015220                MOV     #WRFROM,R0      ;THESE TWO INSTRUCTIONS GETS
033774  010077  160752                MOV     R0,@RHBA        ;ADDR. OF WRFROM INTO R0 AND
                                                                ;BUS ADDRESS REGISTER
034000  012720  010000                MOV     #FMT22,(R0)+    ;FORMAT=16 BIT WORDS
                                                                ;CYLINDER=0
034004  012720  000001          2$:   MOV     #1,(R0)+        ;TRACK=0, SECTOR=1, KEYS=0
034010  005020                        CLR     (R0)+           ;KEY1=0
034012  005020                        CLR     (R0)+           ;KEY2=0
034014  012705  000400                MOV     #256.,R5        ;COUNTER
034020  012720  000000          3$:   MOV     #0,(R0)+        ;MOVE ALL ZEROS FOR DATA
034024  005305                        DEC     R5
034026  001374                        BNE     3$              ;BRANCH IF DATA NOT COMPLETE
```

```
4888  034030  012777  000001  160724          MOV     #1,@RHDST       ;TRACK=0 SECTOR=1
4889
4890  034036  004737  046024                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,DPR,DRY = 1 AND OTHERS DON'T
4891  034042  104401  005123                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF THE
4892  034046  000000                          HALT                    ;STOP THE TEST
4893
4894  034050  013711  015172                  MOV     @#WRIFOR,@R1    ;GET READY FOR WRITE HEADER AND
4895                                                                    ;DATA WITH 62 IN RHCS1
4896  034054  005037  015124                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
4897  034060  012777  010000  160700          MOV     #FMT22,@RHOF    ;FORMAT BIT=1 (16 BIT WORDS)
4898  034066  005077  160676                  CLR     @RHCA           ;CYLINDER =0
4899  034072  004737  055636                  JSR     PC,@#COMWHD     ;WRITE HEADER AND DATA
4900
4901                                          ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
4902                                          ;*FROM THE ''COMWHD'' ROUTINE THAT MEANS ALL HEADER ON DISK
4903                                          ;*IS GOOD IE. ONLY DATA IS TO BE CHECKED TO SEE IF THEY ARE
4904                                          ;*ALL ZEROS AND WRITE DATA GAP AND TOLERANCE GAP TO SEE IF
4905                                          ;*THEY ARE ALL ZEROS
4906
4907  034076  005737  015124                  TST     @#ERFLG$        ;HAS ANY ERRORS OCCURED?
4908                                                                    ;IF WRITE ERROR OCCURS ECC IS NOT CHECKED
4909  034102  001056                          BNE     TST54           ;;BRANCH IF YES
4910
4911
4912                                          ;*COMPARE SOFTWARE GENERATED ECC WITH THAT GENERATED BY HARDWARE
4913
4914  034104  023737  050464  055572          CMP     @#GECC1,@#WECC1;COMPARE SOFTWARE ECC WITH HARDWARE ECC
4915  034112  001402                          BEQ     6$              ;BRANCH IF GOOD
4916  034114  104031                          ERROR   31              ;LOW ORDER ECC IN ERROR
4917  034116  000405                          BR      7$              ;BRANCH TO CONTINUE
4918  034120  023737  050466  055574  6$:     CMP     @#GECC2,@#WECC2;COMPARE SOFTWARE ECC WITH HARDWARE ECC
4919  034126  001401                          BEQ     7$              ;BRANCH IF GOOD
4920  034130  104031                          ERROR   31              ;HIGH ORDER ECC IN ERROR
4921
4922
4923  034132                          7$:
4924  034132  004737  046214                  JSR     PC,@#CHECKE     ;CHECK THAT DVA,RDY,DPR,DRY = 1
4925  034136  104401  005123                  TYPE    ,CPHALT         ;CANNOT CONTINUE IF THEY DON'T
4926  034142  000000                          HALT                    ;STOP THE TEST AND RESTART PROGRAM
4927
4928
4929
4930                                          ;*FILL ''REINTO'' BUFFER WITH EXPECTED DATA
4931
4932  034144  004037  045706                  JSR     R0,@#CLAREA     ;FILL REINTO BUFFER
4933  034150  016264                          REINTO                  ;FROM
4934  034152  017262                          REINTO+<255.*2>         ;TO
4935  034154  000000                  .WORD   0                       ;DATA
4936
4937  034156  013737  050464  017264          MOV     @#GECC1,@#REINTO+<256.*2>;FILL ECC1
4938  034164  013737  050466  017266          MOV     @#GECC2,@#REINTO+<257.*2>;FILL ECC2
4939  034172  004037  045706                  JSR     R0,@#CLAREA     ;FILL REST
4940  034176  017270                  REINTO+<258.*2>                 ;FROM
4941  034200  017324                  REINTO+<272.*2>                 ;TO
4942  034202  000000                  0                               ;DATA
4943
```

```
4944
4945  034204  005037  015124                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
4946
4947
4948                                          ;*NOW COMPARE 'DISK' BUFFER WITH 'REINTO'
4949
4950  034210  004037  046664                  JSR     R0,@#COMPAR     ;CHECK
4951  034214  016264                          REINTO                  ;GOOD BUFFER
4952  034216  054572                          DISK                    ;TEST BUFFER
4953  034220  000402                          258.                    ;NUMBER OF WORDS CHECKED
4954  034222  034230                          4$                      ;RETURN POINT FOR ERROR HEADER
4955  034224  034234                          5$                      ;RETURN POINT FOR ERROR DATA
4956  034226  034240                          TST54                   ;RETURN FOR GOOD COMPARISON
4957  034230  104007              4$:         ERROR   7               ;READ ERROR 10 NEXT
4958  034232  000207                          RTS     PC              ;RETURN TO COMPARE
4959  034234  104010              5$:         ERROR   10              ;WORD NOS 1 TO 256 ARE
4960                                                                  ;DATA WORDS
4961                                                                  ;WORD NOS 257 AND 258
4962                                                                  ;ARE ECC WHICH ARE CHECKED
4963                                                                  ;WORD NOS 259
4964                                                                  ;IS DATA GAP
4965                                                                  ;WORD NOS 260 TO 273
4966                                                                  ;ARE TOLERANCE GAP
4967  034236  000207                          RTS     PC              ;RETURN TO COMPARE
4968
4969
4970
4971
4972
```

```
4973
4974  034240  000004                    TST54:  SCOPE
4975  034242  012706  001000                    MOV     #STACK,SP       ;RESET STACK
4976
4977  034246  012737  000054  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
4978
4979
4980                                      ;       SETUP FOR WHAT IS TO BE READ
4981                                      ;       HEADER CRC IS RESTORED FROM A SUBROUTINE
4982
4983  034254  012746  000000                    MOV     #0,-(SP)        ;DATA TO BE READ
4984  034260  012705  000400                    MOV     #256.,R5        ;COUNTER
4985  034264  012700  054572                    MOV     #DISK,R0        ;START OF SIMULATED DISK DATA
4986  034270  011620                    1$:     MOV     (SP),(R0)+      ;MOVE IN DATA ON TO SIMULATED DISK
4987  034272  005305                            DEC     R5              ;COUNT
4988  034274  001375                            BNE     1$              ;BRANCH IF 256 NOT COMPLETE
4989  034276  005726                            TST     (SP)+           ;UNDO -(SP)
4990  034300  022020                            CMP     (R0)+,(R0)+     ;JUMP OVER THE TWO ECC WORDS
4991  034302  012705  000017                    MOV     #15.,R5         ;1 DATA GAP
4992                                                                    ;14 TOLERANCE GAP
4993  034306  005020                    2$:     CLR     (R0)+           ;CLEAR DATA GAP, AND
4994  034310  005305                            DEC     R5              ;TOLERANCE GAP
4995  034312  001375                            BNE     2$              ;BRANCH IF NOT COMPLETE
4996
4997
4998  034314  004737  051260                    JSR     PC,@#FILLEC     ;INSERT THE TWO ECC WORDS ON THE DISK
4999                                                                    ;IN THE CORRECT PLACE
5000
5001                                      ;*THESE ARE FOR ECC TEST ONLY
5002
5003  034320  012737  177777  015142            MOV     #-1,@#TSECC     ;THIS IS AN ECC TEST
5004  034326  005037  050476                    CLR     @#POSITI        ;CLEAR ERROR POSITION COUNTER
5005  034332  013737  050472  050474            MOV     @#NCODE,@#NCOUNT  ;TEMPORARY N-CODE COUNTER
5006  034340  013737  050500  050506            MOV     @#HARDER,@#HADTMP  ;TEMPORARY HARD ERROR COUNTER
5007  034346  005037  050464                    CLR     @#GECC1         ;ECC LOW ORDER TO BE GENERATED
5008  034352  005037  050466                    CLR     @#GECC2         ;ECC HIGH ORDER TO BE GENERATED
5009  034356  005037  050502                    CLR     @#DATENV        ;CLEAR DATA ENVELOPE CLOCK COUNT
5010  034362  005037  050504                    CLR     @#ZCODE         ;CLEAR LEADING ZEROS CLOCK COUNT
5011
5012
5013                                      ;*THESE ARE TO SETUP FOR DISKLESS USE ONLY
5014
5015  034366  012737  010000  052654            MOV     #FMT22,@#CYL    ;16 BITS PER WORD
5016                                                                    ;CYLINDER 0, FORMAT 16 BITS
5017  034374  112737  000000  052657            MOVB    #0,@#SECOTR+1   ;TRACK 0
5018  034402  112737  000000  052656            MOVB    #0,@#SECOTR     ;SECTOR 0
5019  034410  012737  000000  052660            MOV     #0,@#KEY1       ;KEY1=0
5020  034416  012737  000000  052662            MOV     #0,@#KEY2       ;KEY2=0
5021  034424  012737  000400  052734            MOV     #256.,  @#DAWORD ;NO. OF DATA WORDS
5022  034432  005037  052664                    CLR     @#X             ;THIS IS A READ COMMAND
5023  034436  004537  047176                    JSR     R5,@#CRC        ;GO TO CALCULATE CRC
5024  034442  052654                            CYL
5025  034444  054554                            WCRC
5026
5027
5028
```

```
5029
5030                                        ;*THESE ARE REGULAR SETUPS
5031
5032  034446  004737  045770                JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
5033  034452  012777  177374  160270        MOV     #-256.-4.,@RHWC ;256. DATA 4 HEADER WORDS
5034  034460  012777  016264  160264        MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
5035  034466  112746  000000                MOVB    #0,-(SP)        ;IN LOWER BYTE GET SECTOR
5036  034472  112766  000000  000001        MOVB    #0,1(SP)        ;GET TRACK IN HIGHER BYTE
5037  034500  012677  160256                MOV     (SP)+,  @RHDST  ;TRACK/SECTOR IN RHDST
5038  034504  012777  010000  160254        MOV     #FMT22,@RHOF    ;16 BITS PER WORD
5039                                                                ;ECC CORRECTION NOT INHIBIT
5040                                                                ;BECAUSE ECC IS NOT GOING
5041                                                                ;TO BE CHECKED
5042  034512  005077  160252                CLR     @RHCA           ;CYLINDER 0
5043
5044  034516  004737  046024                JSR     PC,@#CHECKT     ;CHECK DVA,RDY,DPR,DRY = 1 AND OTHERS DON'T
5045  034522  104401  005123                TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF THE
5046  034526  000000                        HALT                    ;STOP THE TEST
5047
5048  034530  013711  015176                MOV     @#REFOR,@R1     ;READ HEADER AND DATA=72
5049  034534  005037  015124                CLR     @#ERFLG$        ;CLEAR ERROR FLAG
5050  034540  004737  052514                JSR     PC,@#COMHD      ;READ HEADER AND DATA
5051                                                                ;IF THERE ARE READ ERRORS THEN
5052                                                                ;ECC WILL NOT BE CHECKED
5053
5054
5055                                        ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
5056                                        ;*FROM THE ''COMHD'' ROUTINE THAT MEANS SECTOR GAP,
5057                                        ;*FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
5058                                        ;*SYNC BYTE HAVE GONE BY AND SYNCS WERE CORRECTLY
5059                                        ;*DETECTED
5060                                        ;*HEADER AND DATA ARE TO BE CHECKED.
5061                                        ;*IN CHECKING READ DATA THE WRITE FROM BUFFER
5062                                        ;*'WRFROM'' IS FILLED WITH EXPECTED DATA AND
5063                                        ;*COMPARISONS ARE MADE
5064
5065  034544  005737  015124                TST     @#ERFLG$        ;ANY ERRORS ALREADY THERE
5066  034550  001102                        BNE     TST55           ;BRANCH IF YES
5067  034552  004737  045470                JSR     PC,@#PUTREG     ;SAVE REGISTERS
5068  034556  005737  015034                TST     @#ER1           ;NO ERRORS SHOULD BE SET
5069  034562  001401                        BEQ     6$              ;BRANCH IF NO ERRORS SET
5070  034564  104032                        ERROR   32              ;32 BIT ECC REGISTER SHOULD BE ZERO
5071                                                                ;ONLY 11 OF THE 32 BITS CAN BE SEEN
5072                                                                ;IN THE PATERN REGISTER
5073                                                                ;DCK SHOULD BE SET IN RHER1
5074  034566  013746  050464        6$:     MOV     @#GECC1,-(SP)   ;GET PATTERN REGISTER
5075  034572  042716  174000                BIC     #174000,(SP)    ;KEEP ONLY 11 BITS
5076  034576  022637  015064                CMP     (SP)+,@#EC2     ;COMPARE PATTERN REGISTER
5077  034602  001401                        BEQ     7$              ;BRANCH IF GOOD
5078  034604  104032                        ERROR   32              ;11 BITS OF THE 32 BIT ECC REGISTER INCORRECT
5079
5080
5081
5082
5083                                        ;*ADD 16 MAINTENANCE CLOCKS TO
5084                                        ;*BRING EBL DOWN
```

```
5085
5086  034606  012700  000020          7$:   MOV   #16.,R0          ;COUNTER
5087  034612  052777  000002  160156  8$:   BIS   #MCLK,@RHMR      ;SET CLOCK
5088  034620  042777  000002  160150        BIC   #MCLK,@RHMR      ;CLEAR CLOCK
5089  034626  005300                        DEC   R0               ;COUNT
5090  034630  001370                        BNE   8$               ;BRANCH IF 16 CLOCKS NOT DONE
5091  034632  004737  046214                JSR   PC,@#CHECKE      ;CHECK THAT DVA,RDY,DPR,DRY = 1
5092  034636  104401  005123                TYPE  ,CPHALT          ;CANNOT CONTINUE IF THEY DON'T
5093  034642  000000                        HALT                   ;STOP THE TEST AND RESTART PROGRAM
5094  034644  012700  015220                MOV   #WRFROM,R0       ;GETTING READY TO FILL EXPECTED DATA
5095  034650  012720  010000                MOV   #0!FMT22,(R0)+   ;CYLINDER 0
5096  034654  112746  000000                MOVB  #0,-(SP)         ;IN LOWER BYTE GET SECTOR
5097  034660  112766  000000  000001        MOVB  #0,1(SP)         ;GET TRACK IN HIGHER BYTE
5098  034666  012620                        MOV   (SP)+,(R0)+      ;GET TRACK/SECTOR IN BUFFER
5099  034670  012720  000000                MOV   #0,(R0)+         ;KEY1 IN BUFFER
5100  034674  012720  000000                MOV   #0,(R0)+         ;KEY2 IN BUFFER
5101  034700  012701  000400                MOV   #256.,R1         ;DATA WORD COUNTER
5102  034704  012702  000000                MOV   #0,R2            ;DATA
5103  034710  010220                3$:     MOV   R2,(R0)+         ;DATA INTO BUFFER
5104  034712  005301                        DEC   R1               ;COUNT
5105  034714  001375                        BNE   3$               ;BRANCH IF 256 NOT DONE
5106
5107
5108  034716  005037  015124                CLR   @#ERFLG$         ;CLEAR ERROR FLAG
5109  034722  004737  045470                JSR   PC,@#PUTREG      ;SAVE REGISTERS
5110
5111
5112
5113                                        ;*NOW READ DATA BUFFER WILL BE CHECKED
5114
5115  034726  004037  046664                JSR   R0,@#COMPAR      ;CHECK
5116  034732  015220                        WRFROM                 ;GOOD BUFFER
5117  034734  016264                        REINTO                 ;TEST BUFFER
5118  034736  000404                        4+256.                 ;NUMBER OF WORDS CHECKED
5119  034740  034746                        4$                     ;RETURN POINT FOR ERROR HEADER
5120  034742  034752                        5$                     ;RETURN POINT FOR ERROR DATA
5121  034744  034756                        TST55                  ;RETURN FOR GOOD COMPARISON
5122  034746  104004                4$:     ERROR 4                ;READ NEXT ERROR
5123  034750  000207                        RTS   PC               ;RETURN TO ''COMPAR''
5124  034752  104005                5$:     ERROR 5                ;WORD NOS 1 TO 4 ARE
5125                                                               ;HEADER WORDS
5126                                                               ;5 TO 260 ARE DATA WORDS
5127  034754  000207                        RTS   PC               ;RETURN TO ''COMPAR''
5128
5129
5130
5131
```

```
5132
5133  034756  000004                    TST55:  SCOPE
5134  034760  012706  001000                    MOV     #STACK,SP       ;RESET STACK
5135
5136  034764  012737  000055  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
5137
5138
5139                                            ;*SETUP FOR WHAT IS TO BE READ
5140                                            ;*HEADER CRC IS RESTORED FROM A SUBROUTINE
5141
5142  034772  012746  000000                    MOV     #0,-(SP)        ;DATA TO BE READ
5143  034776  012705  000400                    MOV     #256.,R5        ;COUNTER
5144  035002  012700  054572                    MOV     #DISK,R0        ;START OF SIMULATED DISK DATA
5145  035006  011620                    1$:     MOV     (SP),(R0)+      ;MOVE IN DATA ON TO SIMULATED DISK
5146  035010  005305                            DEC     R5              ;COUNT
5147  035012  001375                            BNE     1$              ;BRANCH IF 256 NOT COMPLETE
5148  035014  005726                            TST     (SP)+           ;UNDO -(SP)
5149  035016  022020                            CMP     (R0)+,(R0)+     ;JUMP OVER THE TWO ECC WORDS
5150  035020  012705  000017                    MOV     #15.,   R5      ;1 DATA GAP
5151                                                                    ;14 TOLERANCE GAP
5152  035024  005020                    2$:     CLR     (R0)+           ;CLEAR DATA GAP, AND
5153  035026  005305                            DEC     R5              ;TOLERANCE GAP
5154  035030  001375                            BNE     2$              ;BRANCH IF NOT COMPLETE
5155
5156
5157  035032  004737  051260                    JSR     PC,@#FILLEC     ;INSERT ECC IN PROPER PLACE ON DISK
5158
5159
5160
5161                                            ;*THESE ARE FOR ECC TEST ONLY
5162
5163  035036  012737  177777  015142            MOV     #-1,@#TSECC     ;THIS IS AN ECC TEST
5164  035044  005037  050476                    CLR     @#POSITI        ;CLEAR ERROR POSITION COUNTER
5165  035050  013737  050472  050474            MOV     @#NCODE,@#NCOUNT  ;TEMPORARY N-CODE COUNTER
5166  035056  013737  050500  050506            MOV     @#HARDER,@#HADTMP  ;TEMPORARY HARD ERROR COUNTER
5167  035064  005037  050464                    CLR     @#GECC1         ;ECC LOW ORDER TO BE GENERATED
5168  035070  005037  050466                    CLR     @#GECC2         ;ECC HIGH ORDER TO BE GENERATED
5169  035074  005037  050502                    CLR     @#DATENV        ;CLEAR DATA ENVELOPE CLOCK COUNT
5170  035100  005037  050504                    CLR     @#ZCODE         ;CLEAR LEADING ZEROS CLOCK COUNT
5171
5172
5173                                            ;*THESE ARE TO SETUP FOR DISKLESS USE ONLY
5174
5175  035104  012737  010000  052654            MOV     #FMT22,@#CYL    ;16 BITS PER WORD
5176                                                                    ;CYLINDER 0, FORMAT 16 BITS
5177  035112  112737  000000  052657            MOVB    #0,@#SECOTR+1   ;TRACK 0
5178  035120  112737  000000  052656            MOVB    #0,@#SECOTR     ;SECTOR 0
5179  035126  012737  000000  052660            MOV     #0,@#KEY1       ;KEY1=0
5180  035134  012737  000000  052662            MOV     #0,@#KEY2       ;KEY2=0
5181  035142  012737  000400  052734            MOV     #256.,  @#DAWORD ;NO. OF DATA WORDS
5182  035150  005037  052664                    CLR     @#X             ;THIS IS A READ COMMAND
5183  035154  004537  047176                    JSR     R5,@#CRC        ;GO TO CALCULATE CRC
5184  035160  052654                            CYL
5185  035162  054554                            WCRC
5186
5187
```

```
5188                                        ;*THIS IS TO INSERT ERROR
5189                                        ;*THE DISK DATA IS IN LOCATION STARTING FROM 'DISK'
5190                                        ;*THE POSITION OF THE ERROR CAN BE CHANGED BY CHANGING
5191                                        ;*THIS MOVE
5192
5193  035164  012737  100000  054574        MOV     #100000,@#DISK+2 ;FORCE ERROR ON BIT NUMBER 32
5194                                                                 ;SO ERROR POSITION REGISTER WILL SHOW
5195                                                                 ;22
5196  035172  012737  000026  035346        MOV     #22.,@#8$        ;INSERT POSITION REG.
5197
5198
5199                                        ;*THESE ARE REGULAR SETUPS
5200
5201  035200  004737  045770                JSR     PC,@#CLDISK      ;SETUP GENERAL REGISTERS
5202  035204  012777  177374  157536        MOV     #-256.-4.,@RHWC  ;256. DATA 4 HEADER WORDS
5203  035212  012777  016264  157532        MOV     #REINTO,@RHBA    ;STARTING ADDRESS OF READ BUFFER
5204  035220  112746  000000                MOVB    #0,-(SP)         ;IN LOWER BYTE GET SECTOR
5205  035224  112766  000000  000001        MOVB    #0,1(SP)         ;GET TRACK IN HIGHER BYTE
5206  035232  012677  157524                MOV     (SP)+,  @RHDST   ;TRACK/SECTOR IN RHDST
5207  035236  012777  010000  157522        MOV     #FMT22,@RHOF     ;16 BITS PER WORD
5208                                                                 ;ECC CORRECTION NOT INHIBIT
5209                                                                 ;BECAUSE ECC IS NOT GOING
5210                                                                 ;TO BE CHECKED
5211  035244  005077  157520                CLR     @RHCA            ;CYLINDER 0
5212
5213  035250  004737  046024                JSR     PC,@#CHECKT      ;CHECK DVA,RDY,DPR,DRY = 1 AND OTHERS DON'T
5214  035254  104401  005123                TYPE    ,CPHALT          ;CANNOT CONTINUE TESTING IF ANY OF THE
5215  035260  000000                        HALT                     ;STOP THE TEST
5216
5217  035262  013711  015176                MOV     @#REFOR,@R1      ;READ HEADER AND DATA=72
5218  035266  005037  015124                CLR     @#ERFLG$         ;CLEAR ERROR FLAG
5219  035272  004737  052514                JSR     PC,@#COMHD       ;READ HEADER AND DATA
5220                                                                 ;IF THERE ARE READ ERRORS THEN
5221                                                                 ;ECC WILL NOT BE CHECKED
5222
5223
5224                                        ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
5225                                        ;*FROM THE ''COMHD'' ROUTINE THAT MEANS SECTOR GAP,
5226                                        ;*FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
5227                                        ;*SYNC BYTE HAVE GONE BY AND SYNCS WERE CORRECTLY
5228                                        ;*DETECTED
5229                                        ;*HEADER AND DATA ARE TO BE CHECKED.
5230                                        ;*IN CHECKING READ DATA THE WRITE FROM BUFFER
5231                                        ;*''WRFROM'' IS FILLED WITH EXPECTED DATA AND
5232                                        ;*COMPARISONS ARE MADE
5233
5234  035276  005737  015124                TST     @#ERFLG$         ;ANY ERRORS ALREADY THERE
5235  035302  001077                        BNE     TST56            ;BRANCH IF YES
5236  035304  004737  045470                JSR     PC,@#PUTREG      ;SAVE REGISTERS
5237  035310  022737  100000  015034        CMP     #DCK,@#ER1       ;ONLY DATA CHECK ERROR SHOULD BE SET
5238  035316  001401                        BEQ     6$               ;BRANCH IF YES
5239  035320  104032                        ERROR   32               ;32 BIT ECC REGISTER SHOULD BE NON
5240                                                                 ;ZERO
5241                                                                 ;ONLY 11 OF THE 32 BITS CAN BE SEEN
5242                                                                 ;IN THE PATERN REGISTER
5243                                                                 ;DCK SHOULD BE SET IN RHER1
```

```
5244  035322  013746  050464          6$:   MOV   @#GECC1,-(SP)   ;GET PATTERN REGISTER
5245  035326  042716  174000                BIC   #174000,(SP)    ;KEEP ONLY 11 BITS
5246  035332  022637  015064                CMP   (SP)+,@#EC2     ;COMPARE PATTERN REGISTER
5247  035336  001401                        BEQ   7$              ;BRANCH IF GOOD
5248  035340  104032                        ERROR 32              ;11 BITS OF THE 32 BIT ECC REGISTER INCORRECT
5249
5250  035342  004037  051106          7$:   JSR   R0,@#ECORR      ;GO TO ECC CORRECTION PROCESS
5251  035346  000026                  8$:   22.                   ;EXPECTED POSITION REG. WHEN CORRECTION
5252                                                              ;IS COMPLETE
5253
5254
5255
5256  035350  004737  046214                JSR   PC,@#CHECKE     ;CHECK THAT DVA,RDY,DPR,DRY = 1
5257  035354  104401  005123                TYPE  ,CPHALT         ;CANNOT CONTINUE IF THEY DON'T
5258  035360  000000                        HALT                  ;STOP THE TEST AND RESTART PROGRAM
5259  035362  012700  015220                MOV   #WRFROM,R0      ;GETTING READY TO FILL EXPECTED DATA
5260  035366  012720  010000                MOV   #0!FMT22,(R0)+  ;CYLINDER 0
5261  035372  112746  000000                MOVB  #0,-(SP)        ;IN LOWER BYTE GET SECTOR
5262  035376  112766  000000  000001        MOVB  #0,1(SP)        ;GET TRACK IN HIGHER BYTE
5263  035404  012620                        MOV   (SP)+,  (R0)+   ;GET TRACK/SECTOR IN BUFFER
5264  035406  012720  000000                MOV   #0,(R0)+        ;KEY1 IN BUFFER
5265  035412  012720  000000                MOV   #0,(R0)+        ;KEY2 IN BUFFER
5266  035416  012701  000400                MOV   #256.,  R1      ;DATA WORD COUNTER
5267  035422  012702  000000                MOV   #0,R2           ;DATA
5268  035426  010220                  3$:   MOV   R2,(R0)+        ;DATA INTO BUFFER
5269  035430  005301                        DEC   R1              ;COUNT
5270  035432  001375                        BNE   3$              ;BRANCH IF 256 NOT DONE
5271
5272                                        ;*ONLY GOOD DATA HAS BEEN PUT IN 'WRFROM'
5273                                        ;*NOW THE INSERTED ERROR WILL BE PUT IN
5274  035434  012737  100000  015232        MOV   #100000,@#WRFROM+<5*2>  ;INSERTED ERROR
5275
5276
5277
5278  035442  005037  015124                CLR   @#ERFLG$        ;CLEAR ERROR FLAG
5279  035446  004737  045470                JSR   PC,@#PUTREG     ;SAVE REGISTERS
5280
5281
5282                                        ;*NOW READ DATA BUFFER WILL BE CHECKED
5283
5284  035452  004037  046664                JSR   R0,@#COMPAR     ;CHECK
5285  035456  015220                        WRFROM                ;GOOD BUFFER
5286  035460  016264                        REINTO                ;TEST BUFFER
5287  035462  000404                        4+256.                ;NUMBER OF WORDS CHECKED
5288  035464  035472                        4$                    ;RETURN POINT FOR ERROR HEADER
5289  035466  035476                        5$                    ;RETURN POINT FOR ERROR DATA
5290  035470  035502                        TST56                 ;RETURN FOR GOOD COMPARISON
5291  035472  104004            4$:         ERROR 4               ;READ NEXT ERROR
5292  035474  000207                        RTS   PC              ;RETURN TO ''COMPAR''
5293  035476  104005            5$:         ERROR 5               ;WORD NOS 1 TO 4 ARE
5294                                                              ;HEADER WORDS
5295                                                              ;5 TO 260 ARE DATA WORDS
5296  035500  000207                        RTS   PC              ;RETURN TO ''COMPAR''
5297
```

```
5298
5299  035502  000004                    TST56:  SCOPE
5300  035504  012706  001000                    MOV     #STACK,SP       ;RESET STACK
5301
5302  035510  012737  000056  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
5303
5304
5305                                      ;       SETUP FOR WHAT IS TO BE READ
5306                                      ;       HEADER CRC IS RESTORED FROM A SUBROUTINE
5307
5308  035516  012746  000000                    MOV     #0,-(SP)        ;DATA TO BE READ
5309  035522  012705  000400                    MOV     #256.,R5        ;COUNTER
5310  035526  012700  054572                    MOV     #DISK,R0        ;START OF SIMULATED DISK DATA
5311  035532  011620                    1$:     MOV     (SP),(R0)+      ;MOVE IN DATA ON TO SIMULATED DISK
5312  035534  005305                            DEC     R5              ;COUNT
5313  035536  001375                            BNE     1$              ;BRANCH IF 256 NOT COMPLETE
5314  035540  005726                            TST     (SP)+           ;UNDO -(SP)
5315  035542  022020                            CMP     (R0)+,(R0)+     ;JUMP OVER THE TWO ECC WORDS
5316  035544  012705  000017                    MOV     #15.,R5         ;1 DATA GAP
5317                                                                    ;14 TOLERANCE GAP
5318  035550  005020                    2$:     CLR     (R0)+           ;CLEAR DATA GAP, AND
5319  035552  005305                            DEC     R5              ;TOLERANCE GAP
5320  035554  001375                            BNE     2$              ;BRANCH IF NOT COMPLETE
5321
5322
5323  035556  004737  051260                    JSR     PC,@#FILLEC     ;INSERT THE TWO ECC WORDS ON THE DISK
5324                                                                    ;IN THE CORRECT PLACE
5325
5326                                          ;*THESE ARE FOR ECC TEST ONLY
5327
5328  035562  012737  177777  015142            MOV     #-1,@#TSECC     ;THIS IS AN ECC TEST
5329  035570  005037  050476                    CLR     @#POSITI        ;CLEAR ERROR POSITION COUNTER
5330  035574  013737  050472  050474            MOV     @#NCODE,@#NCOUNT  ;TEMPORARY N-CODE COUNTER
5331  035602  013737  050500  050506            MOV     @#HARDER,@#HADTMP  ;TEMPORARY HARD ERROR COUNTER
5332  035610  005037  050464                    CLR     @#GECC1         ;ECC LOW ORDER TO BE GENERATED
5333  035614  005037  050466                    CLR     @#GECC2         ;ECC HIGH ORDER TO BE GENERATED
5334  035620  005037  050502                    CLR     @#DATENV        ;CLEAR DATA ENVELOPE CLOCK COUNT
5335  035624  005037  050504                    CLR     @#ZCODE         ;CLEAR LEADING ZEROS CLOCK COUNT
5336
5337
5338                                          ;*THESE ARE TO SETUP FOR DISKLESS USE ONLY
5339
5340  035630  012737  010000  052654            MOV     #FMT22,@#CYL    ;16 BITS PER WORD
5341                                                                    ;CYLINDER 0, FORMAT 16 BITS
5342  035636  112737  000000  052657            MOVB    #0,@#SECOTR+1   ;TRACK 0
5343  035644  112737  000000  052656            MOVB    #0,@#SECOTR     ;SECTOR 0
5344  035652  012737  000000  052660            MOV     #0,@#KEY1       ;KEY1=0
5345  035660  012737  000000  052662            MOV     #0,@#KEY2       ;KEY2=0
5346  035666  012737  000400  052734            MOV     #256.,@#DAWORD  ;NO. OF DATA WORDS
5347  035674  005037  052664                    CLR     @#X             ;THIS IS A READ COMMAND
5348  035700  004537  047176                    JSR     R5,@#CRC        ;GO TO CALCULATE CRC
5349  035704  052654                            CYL
5350  035706  054554                            WCRC
5351
5352
5353                                          ;*THIS IS TO INSERT ERROR
```

```
5354                                        ;*THE DISK DATA IS IN LOCATION STARTING FROM 'DISK'
5355                                        ;*THE POSITION OF THE ERROR CAN BE CHANGED BY CHANGING
5356                                        ;*THIS MOVE
5357  035710  012737  177760  054574        MOV     #177760,@#DISK+2  ;FORCE ERROR ON BIT NUMBER 21 THRU 32
5358                                                                  ;SO ERROR POSITION REGISTER WILL SHOW
5359                                                                  ;22
5360  035716  012737  010040  036072        MOV     #4128.,@#8$       ;INSERT POSITION REG.
5361
5362
5363                                        ;*THESE ARE REGULAR SETUPS
5364
5365  035724  004737  045770                JSR     PC,@#CLDISK       ;SETUP GENERAL REGISTERS
5366  035730  012777  177374  157012        MOV     #-256.-4.,@RHWC   ;256. DATA 4 HEADER WORDS
5367  035736  012777  016264  157006        MOV     #REINTO,@RHBA     ;STARTING ADDRESS OF READ BUFFER
5368  035744  112746  000000                MOVB    #0,-(SP)          ;IN LOWER BYTE GET SECTOR
5369  035750  112766  000000  000001        MOVB    #0,1(SP)          ;GET TRACK IN HIGHER BYTE
5370  035756  012677  157000                MOV     (SP)+,  @RHDST    ;TRACK/SECTOR IN RHDST
5371  035762  012777  010000  156776        MOV     #FMT22,@RHOF      ;16 BITS PER WORD
5372                                                                  ;ECC CORRECTION NOT INHIBIT
5373                                                                  ;BECAUSE ECC IS NOT GOING
5374                                                                  ;TO BE CHECKED
5375  035770  005077  156774                CLR     @RHCA             ;CYLINDER 0
5376  035774  004737  046024                JSR     PC,@#CHECKT       ;CHECK DVA,RDY,DPR,DRY = 1 AND OTHERS DON'T
5377  036000  104401  005123                TYPE    ,CPHALT           ;CANNOT CONTINUE TESTING IF ANY OF THE
5378  036004  000000                        HALT                      ;STOP THE TEST
5379  036006  013711  015176                MOV     @#REFOR,@R1       ;READ HEADER AND DATA=72
5380  036012  005037  015124                CLR     @#ERFLG$          ;CLEAR ERROR FLAG
5381  036016  004737  052514                JSR     PC,     @#COMHD   ;READ HEADER AND DATA
5382                                                                  ;IF THERE ARE READ ERRORS THEN
5383                                                                  ;ECC WILL NOT BE CHECKED
5384
```

```
5385
5386                                        ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
5387                                        ;*FROM THE ''COMHD'' ROUTINE THAT MEANS SECTOR GAP,
5388                                        ;*FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
5389                                        ;*SYNC BYTE HAVE GONE BY AND SYNCS WERE CORRECTLY
5390                                        ;*DETECTED
5391                                        ;*HEADER AND DATA ARE TO BE CHECKED.
5392                                        ;*IN CHECKING READ DATA THE WRITE FROM BUFFER
5393                                        ;*'WRFROM'' IS FILLED WITH EXPECTED DATA AND
5394                                        ;*COMPARISONS ARE MADE
5395
5396  036022  005737  015124                  TST     @#ERFLG$        ;ANY ERRORS ALREADY THERE
5397  036026  001106                          BNE     TST57           ;BRANCH IF YES
5398  036030  004737  045470                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
5399  036034  022737  100000  015034          CMP     #DCK,@#ER1      ;ONLY DATA CHECK ERROR SHOULD BE SET
5400  036042  001401                          BEQ     6$              ;BRANCH IF YES
5401  036044  104032                          ERROR   32              ;32 BIT ECC REGISTER SHOULD BE NON
5402                                                                  ;ZERO
5403                                                                  ;ONLY 11 OF THE 32 BITS CAN BE SEEN
5404                                                                  ;IN THE PATERN REGISTER
5405                                                                  ;DCK SHOULD BE SET IN RHER1
5406  036046  013746  050464          6$:     MOV     @#GECC1,-(SP)   ;GET PATTERN REGISTER
5407  036052  042716  174000                  BIC     #174000,(SP)    ;KEEP ONLY 11 BITS
5408  036056  022637  015064                  CMP     (SP)+,@#EC2     ;COMPARE PATTERN REGISTER
5409  036062  001401                          BEQ     7$              ;BRANCH IF GOOD
5410  036064  104032                          ERROR   32              ;11 BITS OF THE 32 BIT ECC REGISTER INCORRECT
5411
5412  036066  004037  051106          7$:     JSR     R0,@#ECORR      ;GO TO ECC CORRECTION PROCESS
5413  036072  000000                  8$:     .WORD                   ;EXPECTED POSITION REG. WHEN CORRECTION
5414                                                                  ;IS COMPLETE
5415
5416
5417
5418  036074  004737  045470                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
5419  036100  022737  100100  015034          CMP     #DCK!ECH,@#ER1  ;WITH ERRORS INSERTED IN BIT POSITION 21
5420                                                                  ;THRU 32 HARD ERROR BIT SHOULD SET
5421  036106  001401                          BEQ     9$              ;BRANCH IF GOOD
5422  036110  104036                          ERROR   36              ;WITH ERROR INSERTED IN BIT POSITION 21 THRU
5423                                                                  ;32 ECH SHOULD SET
5424
5425
5426
5427  036112                          9$:
5428  036112  004737  046214                  JSR     PC,@#CHECKE     ;CHECK THAT DVA,RDY,DPR,DRY = 1
5429  036116  104401  005123                  TYPE    ,CPHALT         ;CANNOT CONTINUE IF THEY DON'T
5430  036122  000000                          HALT                    ;STOP THE TEST AND RESTART PROGRAM
5431  036124  012700  015220                  MOV     #WRFROM,R0      ;GETTING READY TO FILL EXPECTED DATA
5432  036130  012720  010000                  MOV     #0!FMT22,(R0)+  ;CYLINDER 0
5433  036134  112746  000000                  MOVB    #0,-(SP)        ;IN LOWER BYTE GET SECTOR
5434  036140  112766  000000  000001          MOVB    #0,1(SP)        ;GET TRACK IN HIGHER BYTE
5435  036146  012620                          MOV     (SP)+,(R0)+     ;GET TRACK/SECTOR IN BUFFER
5436  036150  012720  000000                  MOV     #0,(R0)+        ;KEY1 IN BUFFER
5437  036154  012720  000000                  MOV     #0,(R0)+        ;KEY2 IN BUFFER
5438  036160  012701  000400                  MOV     #256.,R1        ;DATA WORD COUNTER
5439  036164  012702  000000                  MOV     #0,R2           ;DATA
5440  036170  010220                  3$:     MOV     R2,(R0)+        ;DATA INTO BUFFER
```

```
 5441  036172  005301                    DEC     R1              ;COUNT
 5442  036174  001375                    BNE     3$              ;BRANCH IF 256 NOT DONE
 5443
```

```
5444                                        ;ONLY GOOD DATA HAS BEEN PUT IN 'WRFROM'
5445                                        ;*NOW THE INSERTED ERROR WILL BE PUT IN
5446
5447  036176  012737  177760  015232        MOV     #177760,@#WRFROM+<5*2>  ;INSERTED ERROR
5448
5449
5450
5451  036204  005037  015124                CLR     @#ERFLG$                ;CLEAR ERROR FLAG
5452  036210  004737  045470                JSR     PC,@#PUTREG     ;SAVE REGISTERS
5453
5454
5455                                        ;*NOW READ DATA BUFFER WILL BE CHECKED
5456
5457  036214  004037  046664                JSR     R0,@#COMPAR     ;CHECK
5458  036220  015220                        WRFROM                  ;GOOD BUFFER
5459  036222  016264                        REINTO                  ;TEST BUFFER
5460  036224  000404                        4+256.                  ;NUMBER OF WORDS CHECKED
5461  036226  036234                        4$                      ;RETURN POINT FOR ERROR HEADER
5462  036230  036240                        5$                      ;RETURN POINT FOR ERROR DATA
5463  036232  036244                        TST57                   ;RETURN FOR GOOD COMPARISON
5464  036234  104004              4$:       ERROR   4               ;READ NEXT ERROR
5465  036236  000207                        RTS     PC              ;RETURN TO ''COMPAR''
5466  036240  104005              5$:       ERROR   5               ;WORD NOS 1 TO 4 ARE
5467                                                                ;HEADER WORDS
5468                                                                ;5 TO 260 ARE DATA WORDS
5469  036242  000207                        RTS     PC              ;RETURN TO ''COMPAR''
5470
5471
5472
5473
5474
5475
```

```
5476
5477  036244  000004                    TST57:  SCOPE
5478  036246  012706  001000                    MOV     #STACK,SP       ;RESET STACK
5479
5480  036252  012737  000057  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
5481  036260  012700  054474                    MOV     #SECGAP,R0      ;POINTER
5482  036264  012701  000460                    MOV     #304.,R1        ;COUNTER
5483  036270  005020                    1$:     CLR     (R0)+           ;CLEAR SIMULATED DISK AREA
5484  036272  005301                            DEC     R1
5485  036274  001375                            BNE     1$
5486  036276  004737  045770                    JSR     PC,CLDISK       ;THIS IS USED TO SET GENERAL REGISTERS
5487
5488                                            ;*THESE ARE FOR ECC TEST ONLY
5489
5490  036302  012737  177777  015142            MOV     #-1,@#TSECC     ;THIS IS AN ECC TEST
5491  036310  005037  050476                    CLR     @#POSITI        ;CLEAR ERROR POSITION COUNTER
5492  036314  013737  050472  050474            MOV     @#NCODE,@#NCOUNT  ;TEMPORARY N-CODE COUNTER
5493  036322  013737  050500  050506            MOV     @#HARDER,@#HADTMP  ;TEMPORARY HARD ERROR COUNTER
5494  036330  005037  050464                    CLR     @#GECC1         ;ECC LOW ORDER TO BE GENERATED
5495  036334  005037  050466                    CLR     @#GECC2         ;ECC HIGH ORDER TO BE GENERATED
5496  036340  005037  050502                    CLR     @#DATENV        ;CLEAR DATA ENVELOPE CLOCK COUNT
5497  036344  005037  050504                    CLR     @#ZCODE         ;CLEAR LEADING ZEROS CLOCK COUNT
5498
5499
5500
5501
5502                                            ;*THESE ARE TO BE SETUP FOR DISKLESS USE ONLY
5503
5504  036350  012737  010000  056012            MOV     #FMT22,@#WCYL   ;FORMAT22=16BIT WORDS AND
5505                                                                    ;CYLINDER 0
5506  036356  012737  000001  056014            MOV     #1,@#WSECTR     ;TRACK=0, SECTOR=1
5507  036364  005037  056016                    CLR     @#WKEY1         ;KEY1=0
5508  036370  005037  056020                    CLR     @#WKEY2         ;KEY2=0
5509  036374  012737  000400  056052            MOV     #256.,@#FNWORD  ;256 DATA WORDS
5510  036402  004537  047176                    JSR     R5,@#CRC        ;GO TO CALCULATE CRC
5511  036406  056012                            WCYL
5512  036410  056022                            GCRC
5513
5514                                            ;*THESE ARE REGULAR SETUPS
5515
5516  036412  012777  177374  156330            MOV     #-260.,@RHWC    ;256 DATA WORDS 4 HEADER WORDS
5517  036420  012700  015220                    MOV     #WRFROM,R0      ;THESE TWO INSTRUCTIONS GETS
5518  036424  010077  156322                    MOV     R0,@RHBA        ;ADDR. OF WRFROM INTO R0 AND
5519                                                                    ;BUS ADDRESS REGISTER
5520  036430  012720  010000                    MOV     #FMT22,(R0)+    ;FORMAT=16 BIT WORDS
5521                                                                    ;CYLINDER=0
5522  036434  012720  000001            2$:     MOV     #1,(R0)+        ;TRACK=0, SECTOR=1, KEYS=0
5523  036440  005020                            CLR     (R0)+           ;KEY1=0
5524  036442  005020                            CLR     (R0)+           ;KEY2=0
5525  036444  012705  000400                    MOV     #256.,R5        ;COUNTER
5526  036450  012720  177777            3$:     MOV     #-1,(R0)+       ;MOVE ALL ONES FOR DATA
5527  036454  005305                            DEC     R5
5528  036456  001374                            BNE     3$              ;BRANCH IF DATA NOT COMPLETE
5529  036460  012777  000001  156274            MOV     #1,@RHDST       ;TRACK=0 SECTOR=1
5530
5531  036466  004737  046024                    JSR     PC,@#CHECKT     ;CHECK DVA,RDY,DPR,DRY = 1 AND OTHERS DON'T
```

```
5532  036472  104401  005123                  TYPE    ,CPHALT          ;CANNOT CONTINUE TESTING IF ANY OF THE
5533  036476  000000                          HALT                     ;STOP THE TEST
5534
5535  036500  013711  015172                  MOV     @#WRIFOR,@R1     ;GET READY FOR WRITE HEADER AND
5536                                                                     ;DATA WITH 62 IN RHCS1
5537  036504  005037  015124                  CLR     @#ERFLG$         ;CLEAR ERROR FLAG
5538  036510  012777  010000  156250          MOV     #FMT22,@RHOF     ;FORMAT BIT=1 (16 BIT WORDS)
5539  036516  005077  156246                  CLR     @RHCA            ;CYLINDER =0
5540  036522  004737  055636                  JSR     PC,@#COMWHD      ;WRITE HEADER AND DATA
5541
5542                                          ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
5543                                          ;*FROM THE ''COMWHD'' ROUTINE THAT MEANS ALL HEADER ON DISK
5544                                          ;*IS GOOD IE. ONLY DATA IS TO BE CHECKED TO SEE IF THEY ARE
5545                                          ;*ALL ONES AND WRITE DATA GAP AND TOLERANCE GAP TO SEE IF
5546                                          ;*THEY ARE ALL ZEROS
5547
5548  036526  005737  015124                  TST     @#ERFLG$         ;HAS ANY ERRORS OCCURED?
5549                                                                     ;IF WRITE ERROR OCCURS ECC IS NOT CHECKED
5550  036532  001056                          BNE     TST60            ;;BRANCH IF YES
5551
5552
5553                                          ;*COMPARE SOFTWARE GENERATED ECC WITH THAT GENERATED BY HARDWARE
5554
5555  036534  023737  050464  055572          CMP     @#GECC1,@#WECC1;COMPARE SOFTWARE ECC WITH HARDWARE ECC
5556  036542  001402                          BEQ     6$               ;BRANCH IF GOOD
5557  036544  104031                          ERROR   31               ;LOW ORDER ECC IN ERROR
5558  036546  000405                          BR      7$               ;BRANCH TO CONTINUE
5559  036550  023737  050466  055574  6$:     CMP     @#GECC2,@#WECC2;COMPARE SOFTWARE ECC WITH HARDWARE ECC
5560  036556  001401                          BEQ     7$               ;BRANCH IF GOOD
5561  036560  104031                          ERROR   31               ;HIGH ORDER ECC IN ERROR
5562
5563
5564  036562                          7$:
5565  036562  004737  046214                  JSR     PC,@#CHECKE      ;CHECK THAT DVA,RDY,DPR,DRY = 1
5566  036566  104401  005123                  TYPE    ,CPHALT          ;CANNOT CONTINUE IF THEY DON'T
5567  036572  000000                          HALT                     ;STOP THE TEST AND RESTART PROGRAM
5568
5569
5570
5571
5572                                          ;*FILL ''REINTO'' BUFFER WITH EXPECTED DATA
5573
5574  036574  004037  045706                  JSR     R0,@#CLAREA      ;FILL REINTO BUFFER
5575  036600  016264                                  REINTO           ;FROM
5576  036602  017262                                  REINTO+<255.*2>  ;TO
5577  036604  177777                          .WORD   -1               ;DATA
5578
5579  036606  013737  050464  017264          MOV     @#GECC1,@#REINTO+<256.*2>;FILL ECC1
5580  036614  013737  050466  017266          MOV     @#GECC2,@#REINTO+<257.*2>;FILL ECC2
5581  036622  004037  045706                  JSR     R0,@#CLAREA      ;FILL REST
5582  036626  017270                  REINTO+<258.*2>                  ;FROM
5583  036630  017324                  REINTO+<272.*2>                  ;TO
5584  036632  000000                  0                                ;DATA
5585
5586
5587  036634  005037  015124                  CLR     @#ERFLG$         ;CLEAR ERROR FLAG
```

```
5588
5589
5590                                        ;*NOW COMPARE 'DISK' BUFFER WITH 'REINTO'
5591
5592  036640  004037  046664                JSR     R0,@#COMPAR     ;CHECK
5593  036644  016264                        REINTO                  ;GOOD BUFFER
5594  036646  054572                        DISK                    ;TEST BUFFER
5595  036650  000402                        258.                    ;NUMBER OF WORDS CHECKED
5596  036652  036660                        4$                      ;RETURN POINT FOR ERROR HEADER
5597  036654  036664                        5$                      ;RETURN POINT FOR ERROR DATA
5598  036656  036670                        TST60                   ;RETURN FOR GOOD COMPARISON
5599  036660  104007              4$:       ERROR   7               ;READ ERROR 10 NEXT
5600  036662  000207                        RTS     PC              ;RETURN TO COMPARE
5601  036664  104010              5$:       ERROR   10              ;WORD NOS 1 TO 256 ARE
5602                                                                ;DATA WORDS
5603                                                                ;WORD NOS 257 AND 258
5604                                                                ;ARE ECC WHICH ARE CHECKED
5605                                                                ;WORD NOS 259
5606                                                                ;IS DATA GAP
5607                                                                ;WORD NOS 260 TO 273
5608                                                                ;ARE TOLERANCE GAP
5609  036666  000207                        RTS     PC              ;RETURN TO COMPARE
5610
5611
5612
5613
5614
```

```
5615
5616  036670  000004                    TST60:  SCOPE
5617  036672  012706  001000                    MOV     #STACK,SP       ;RESET STACK
5618
5619  036676  012737  000060  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
5620
5621
5622                                      ;       SETUP FOR WHAT IS TO BE READ
5623                                      ;       HEADER CRC IS RESTORED FROM A SUBROUTINE
5624
5625  036704  012746  177777                    MOV     #-1,-(SP)       ;DATA TO BE READ
5626  036710  012705  000400                    MOV     #256.,R5        ;COUNTER
5627  036714  012700  054572                    MOV     #DISK,R0        ;START OF SIMULATED DISK DATA
5628  036720  011620                    1$:     MOV     (SP),(R0)+      ;MOVE IN DATA ON TO SIMULATED DISK
5629  036722  005305                            DEC     R5              ;COUNT
5630  036724  001375                            BNE     1$              ;BRANCH IF 256 NOT COMPLETE
5631  036726  005726                            TST     (SP)+           ;UNDO -(SP)
5632  036730  022020                            CMP     (R0)+,(R0)+     ;JUMP OVER THE TWO ECC WORDS
5633  036732  012705  000017                    MOV     #15.,R5         ;1 DATA GAP
5634                                                                    ;14 TOLERANCE GAP
5635  036736  005020                    2$:     CLR     (R0)+           ;CLEAR DATA GAP, AND
5636  036740  005305                            DEC     R5              ;TOLERANCE GAP
5637  036742  001375                            BNE     2$              ;BRANCH IF NOT COMPLETE
5638
5639
5640  036744  004737  051260                    JSR     PC,@#FILLEC     ;INSERT THE TWO ECC WORDS ON THE DISK
5641                                                                    ;IN THE CORRECT PLACE
5642
5643                                              ;*THESE ARE FOR ECC TEST ONLY
5644
5645  036750  012737  177777  015142            MOV     #-1,@#TSECC     ;THIS IS AN ECC TEST
5646  036756  005037  050476                    CLR     @#POSITI        ;CLEAR ERROR POSITION COUNTER
5647  036762  013737  050472  050474            MOV     @#NCODE,@#NCOUNT  ;TEMPORARY N-CODE COUNTER
5648  036770  013737  050500  050506            MOV     @#HARDER,@#HADTMP  ;TEMPORARY HARD ERROR COUNTER
5649  036776  005037  050464                    CLR     @#GECC1         ;ECC LOW ORDER TO BE GENERATED
5650  037002  005037  050466                    CLR     @#GECC2         ;ECC HIGH ORDER TO BE GENERATED
5651  037006  005037  050502                    CLR     @#DATENV        ;CLEAR DATA ENVELOPE CLOCK COUNT
5652  037012  005037  050504                    CLR     @#ZCODE         ;CLEAR LEADING ZEROS CLOCK COUNT
5653
5654
5655                                              ;*THESE ARE TO SETUP FOR DISKLESS USE ONLY
5656
5657  037016  012737  010000  052654            MOV     #FMT22,@#CYL    ;16 BITS PER WORD
5658                                                                    ;CYLINDER 0, FORMAT 16 BITS
5659  037024  112737  000000  052657            MOVB    #0,@#SECOTR+1   ;TRACK 0
5660  037032  112737  000000  052656            MOVB    #0,@#SECOTR     ;SECTOR 0
5661  037040  012737  000000  052660            MOV     #0,@#KEY1       ;KEY1=0
5662  037046  012737  000000  052662            MOV     #0,@#KEY2       ;KEY2=0
5663  037054  012737  000400  052734            MOV     #256.,@#DAWORD  ;NO. OF DATA WORDS
5664  037062  005037  052664                    CLR     @#X             ;THIS IS A READ COMMAND
5665  037066  004537  047176                    JSR     R5,@#CRC        ;GO TO CALCULATE CRC
5666  037072  052654                            CYL
5667  037074  054554                            WCRC
5668
5669
5670
```

```
5671
5672                                        ;*THESE ARE REGULAR SETUPS
5673
5674  037076  004737  045770                JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
5675  037102  012777  177374  155640        MOV     #-256.-4.,@RHWC ;256. DATA 4 HEADER WORDS
5676  037110  012777  016264  155634        MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
5677  037116  112746  000000                MOVB    #0,-(SP)        ;IN LOWER BYTE GET SECTOR
5678  037122  112766  000000  000001        MOVB    #0,1(SP)        ;GET TRACK IN HIGHER BYTE
5679  037130  012677  155626                MOV     (SP)+,@RHDST    ;TRACK/SECTOR IN RHDST
5680  037134  012777  010000  155624        MOV     #FMT22,@RHOF    ;16 BITS PER WORD
5681                                                                ;ECC CORRECTION NOT INHIBIT
5682                                                                ;BECAUSE ECC IS NOT GOING
5683                                                                ;TO BE CHECKED
5684  037142  005077  155622                CLR     @RHCA           ;CYLINDER 0
5685
5686  037146  004737  046024                JSR     PC,@#CHECKT     ;CHECK DVA,RDY,DPR,DRY = 1 AND OTHERS DON'T
5687  037152  104401  005123                TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF THE
5688  037156  000000                        HALT                    ;STOP THE TEST
5689
5690  037160  013711  015176                MOV     @#REFOR,@R1     ;READ HEADER AND DATA=72
5691  037164  005037  015124                CLR     @#ERFLG$        ;CLEAR ERROR FLAG
5692  037170  004737  052514                JSR     PC,@#COMHD      ;READ HEADER AND DATA
5693                                                                ;IF THERE ARE READ ERRORS THEN
5694                                                                ;ECC WILL NOT BE CHECKED
5695
5696
5697                                        ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
5698                                        ;*FROM THE ''COMHD'' ROUTINE THAT MEANS SECTOR GAP,
5699                                        ;*FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
5700                                        ;*SYNC BYTE HAVE GONE BY AND SYNCS WERE CORRECTLY
5701                                        ;*DETECTED
5702                                        ;*HEADER AND DATA ARE TO BE CHECKED.
5703                                        ;*IN CHECKING READ DATA THE WRITE FROM BUFFER
5704                                        ;*'WRFROM'' IS FILLED WITH EXPECTED DATA AND
5705                                        ;*COMPARISONS ARE MADE
5706
5707  037174  005737  015124                TST     @#ERFLG$        ;ANY ERRORS ALREADY THERE
5708  037200  001102                        BNE     TST61           ;BRANCH IF YES
5709  037202  004737  045470                JSR     PC,@#PUTREG     ;SAVE REGISTERS
5710  037206  005737  015034                TST     @#ER1           ;NO ERRORS SHOULD BE SET
5711  037212  001401                        BEQ     6$              ;BRANCH IF NO ERRORS SET
5712  037214  104032                        ERROR   32              ;32 BIT ECC REGISTER SHOULD BE ZERO
5713                                                                ;ONLY 11 OF THE 32 BITS CAN BE SEEN
5714                                                                ;IN THE PATERN REGISTER
5715                                                                ;DCK SHOULD BE SET IN RHER1
5716  037216  013746  050464        6$:     MOV     @#GECC1,-(SP)   ;GET PATTERN REGISTER
5717  037222  042716  174000                BIC     #174000,(SP)    ;KEEP ONLY 11 BITS
5718  037226  022637  015064                CMP     (SP)+,@#EC2     ;COMPARE PATTERN REGISTER
5719  037232  001401                        BEQ     7$              ;BRANCH IF GOOD
5720  037234  104032                        ERROR   32              ;11 BITS OF THE 32 BIT ECC REGISTER INCORRECT
5721
5722
5723
5724
5725                                        ;*ADD 16 MAINTENANCE CLOCKS TO
5726                                        ;*BRING EBL DOWN
```

```
5727
5728  037236  012700  000020          7$:   MOV   #16.,R0          ;COUNTER
5729  037242  052777  000002  155526  8$:   BIS   #MCLK,@RHMR      ;SET CLOCK
5730  037250  042777  000002  155520        BIC   #MCLK,@RHMR      ;CLEAR CLOCK
5731  037256  005300                        DEC   R0               ;COUNT
5732  037260  001370                        BNE   8$               ;BRANCH IF 16 CLOCKS NOT DONE
5733  037262  004737  046214                JSR   PC,@#CHECKE      ;CHECK THAT DVA,RDY,DPR,DRY = 1
5734  037266  104401  005123                TYPE  ,CPHALT          ;CANNOT CONTINUE IF THEY DON'T
5735  037272  000000                        HALT                   ;STOP THE TEST AND RESTART PROGRAM
5736  037274  012700  015220                MOV   #WRFROM,R0       ;GETTING READY TO FILL EXPECTED DATA
5737  037300  012720  010000                MOV   #0!FMT22,(R0)+   ;CYLINDER 0
5738  037304  112746  000000                MOVB  #0,     -(SP)    ;IN LOWER BYTE GET SECTOR
5739  037310  112766  000000  000001        MOVB  #0,     1(SP)    ;GET TRACK IN HIGHER BYTE
5740  037316  012620                        MOV   (SP)+,  (R0)+    ;GET TRACK/SECTOR IN BUFFER
5741  037320  012720  000000                MOV   #0,     (R0)+    ;KEY1 IN BUFFER
5742  037324  012720  000000                MOV   #0,     (R0)+    ;KEY2 IN BUFFER
5743  037330  012701  000400                MOV   #256.,  R1       ;DATA WORD COUNTER
5744  037334  012702  177777                MOV   #-1,    R2       ;DATA
5745
5746  037340  010220                  3$:   MOV   R2,     (R0)+    ;DATA INTO BUFFER
5747  037342  005301                        DEC   R1               ;COUNT
5748  037344  001375                        BNE   3$               ;BRANCH IF 256 NOT DONE
5749  037346  005037  015124                CLR   @#ERFLG$         ;CLEAR ERROR FLAG
5750  037352  004737  045470                JSR   PC,@#PUTREG      ;SAVE REGISTERS
5751
5752                                        ;NOW READ DATA BUFFER WILL BE CHECKED
5753
5754  037356  004037  046664                JSR   R0,@#COMPAR      ;CHECK
5755  037362  015220                        WRFROM                 ;GOOD BUFFER
5756  037364  016264                        REINTO                 ;TEST BUFFER
5757  037366  000404                        4+256.                 ;NUMBER OF WORDS CHECKED
5758  037370  037376                        4$                     ;RETURN POINT FOR ERROR HEADER
5759  037372  037402                        5$                     ;RETURN POINT FOR ERROR DATA
5760  037374  037406                        TST61                  ;RETURN FOR GOOD COMPARISON
5761  037376  104004            4$:   ERROR 4                      ;READ NEXT ERROR
5762  037400  000207                  RTS   PC                     ;RETURN TO ''COMPAR''
5763  037402  104005            5$:   ERROR 5                      ;WORD NOS 1 TO 4 ARE
5764                                                               ;HEADER WORDS
5765                                                               ;5 TO 260 ARE DATA WORDS
5766  037404  000207                  RTS   PC                     ;RETURN TO ''COMPAR''
5767
5768
5769
5770
```

```
5771
5772  037406  000004                    TST61:  SCOPE
5773  037410  012706  001000                    MOV     #STACK,SP       ;RESET STACK
5774
5775  037414  012737  000061  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
5776
5777
5778                                            ;*SETUP FOR WHAT IS TO BE READ
5779                                            ;*HEADER CRC IS RESTORED FROM A SUBROUTINE
5780
5781  037422  012746  177777                    MOV     #-1,-(SP)       ;DATA TO BE READ
5782  037426  012705  000400                    MOV     #256.,R5        ;COUNTER
5783  037432  012700  054572                    MOV     #DISK,R0        ;START OF SIMULATED DISK DATA
5784  037436  011620                    1$:     MOV     (SP),(R0)+      ;MOVE IN DATA ON TO SIMULATED DISK
5785  037440  005305                            DEC     R5              ;COUNT
5786  037442  001375                            BNE     1$              ;BRANCH IF 256 NOT COMPLETE
5787  037444  005726                            TST     (SP)+           ;UNDO -(SP)
5788  037446  022020                            CMP     (R0)+,(R0)+     ;JUMP OVER THE TWO ECC WORDS
5789  037450  012705  000017                    MOV     #15.,   R5      ;1 DATA GAP
5790                                                                    ;14 TOLERANCE GAP
5791  037454  005020                    2$:     CLR     (R0)+           ;CLEAR DATA GAP, AND
5792  037456  005305                            DEC     R5              ;TOLERANCE GAP
5793  037460  001375                            BNE     2$              ;BRANCH IF NOT COMPLETE
5794
5795
5796  037462  004737  051260                    JSR     PC,@#FILLEC     ;INSERT ECC IN PROPER PLACE ON DISK
5797
5798
5799
5800                                            ;*THESE ARE FOR ECC TEST ONLY
5801
5802  037466  012737  177777  015142            MOV     #-1,@#TSECC     ;THIS IS AN ECC TEST
5803  037474  005037  050476                    CLR     @#POSITI        ;CLEAR ERROR POSITION COUNTER
5804  037500  013737  050472  050474            MOV     @#NCODE,@#NCOUNT  ;TEMPORARY N-CODE COUNTER
5805  037506  013737  050500  050506            MOV     @#HARDER,@#HADTMP  ;TEMPORARY HARD ERROR COUNTER
5806  037514  005037  050464                    CLR     @#GECC1         ;ECC LOW ORDER TO BE GENERATED
5807  037520  005037  050466                    CLR     @#GECC2         ;ECC HIGH ORDER TO BE GENERATED
5808  037524  005037  050502                    CLR     @#DATENV        ;CLEAR DATA ENVELOPE CLOCK COUNT
5809  037530  005037  050504                    CLR     @#ZCODE         ;CLEAR LEADING ZEROS CLOCK COUNT
5810
5811
5812                                            ;*THESE ARE TO SETUP FOR DISKLESS USE ONLY
5813
5814  037534  012737  010000  052654            MOV     #FMT22,@#CYL    ;16 BITS PER WORD
5815                                                                    ;CYLINDER 0, FORMAT 16 BITS
5816  037542  112737  000000  052657            MOVB    #0,@#SECOTR+1   ;TRACK 0
5817  037550  112737  000000  052656            MOVB    #0,@#SECOTR     ;SECTOR 0
5818  037556  012737  000000  052660            MOV     #0,@#KEY1       ;KEY1=0
5819  037564  012737  000000  052662            MOV     #0,@#KEY2       ;KEY2=0
5820  037572  012737  000400  052734            MOV     #256.,@#DAWORD  ;NO. OF DATA WORDS
5821  037600  005037  052664                    CLR     @#X             ;THIS IS A READ COMMAND
5822  037604  004537  047176                    JSR     R5,@#CRC        ;GO TO CALCULATE CRC
5823  037610  052654                            CYL
5824  037612  054554                            WCRC
5825
5826
```

```
                                        ;*THIS IS TO INSERT ERROR
                                        ;*THE DISK DATA IS IN LOCATION STARTING FROM 'DISK'
                                        ;*THE POSITION OF THE ERROR CAN BE CHANGED BY CHANGING
                                        ;*THIS MOVE
                                        
037614  012737  077777  054574        MOV     #77777,@#DISK+2 ;FORCE ERROR ON BIT NUMBER 32
                                                                ;SO ERROR POSITION REGISTER WILL SHOW
                                                                ;22
037622  012737  000026  037776        MOV     #22.,@#8$       ;INSERT POSITION REG.
                                        
                                        ;*THESE ARE REGULAR SETUPS
                                        
037630  004737  045770                JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
037634  012777  177374  155106        MOV     #-256.-4.,@RHWC ;256. DATA 4 HEADER WORDS
037642  012777  016264  155102        MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
037650  112746  000000                MOVB    #0,-(SP)        ;IN LOWER BYTE GET SECTOR
037654  112766  000000  000001        MOVB    #0,1(SP)        ;GET TRACK IN HIGHER BYTE
037662  012677  155074                MOV     (SP)+,@RHDST    ;TRACK/SECTOR IN RHDST
037666  012777  010000  155072        MOV     #FMT22,@RHOF    ;16 BITS PER WORD
                                                                ;ECC CORRECTION NOT INHIBIT
                                                                ;BECAUSE ECC IS NOT GOING
                                                                ;TO BE CHECKED
037674  005077  155070                CLR     @RHCA           ;CYLINDER 0
037700  004737  046024                JSR     PC,@#CHECKT     ;CHECK DVA,RDY,DPR,DRY = 1 AND OTHERS DON'T
037704  104401  005123                TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF THE
037710  000000                        HALT                    ;STOP THE TEST
037712  013711  015176                MOV     @#REFOR,@R1     ;READ HEADER AND DATA=72
037716  005037  015124                CLR     @#ERFLG$        ;CLEAR ERROR FLAG
037722  004737  052514                JSR     PC,@#COMHD      ;READ HEADER AND DATA
                                                                ;IF THERE ARE READ ERRORS THEN
                                                                ;ECC WILL NOT BE CHECKED
                                        
                                        ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
                                        ;*FROM THE ''COMHD'' ROUTINE THAT MEANS SECTOR GAP,
                                        ;*FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
                                        ;*SYNC BYTE HAVE GONE BY AND SYNCS WERE CORRECTLY
                                        ;*DETECTED
                                        ;*HEADER AND DATA ARE TO BE CHECKED.
                                        ;*IN CHECKING READ DATA THE WRITE FROM BUFFER
                                        ;*'WRFROM'' IS FILLED WITH EXPECTED DATA AND
                                        ;*COMPARISONS ARE MADE
                                        
037726  005737  015124                TST     @#ERFLG$        ;ANY ERRORS ALREADY THERE
037732  001077                        BNE     TST62           ;BRANCH IF YES
037734  004737  045470                JSR     PC,@#PUTREG     ;SAVE REGISTERS
037740  022737  100000  015034        CMP     #DCK,@#ER1      ;ONLY DATA CHECK ERROR SHOULD BE SET
037746  001401                        BEQ     6$              ;BRANCH IF YES
037750  104032                        ERROR   32              ;32 BIT ECC REGISTER SHOULD BE NON
                                                                ;ZERO
                                                                ;ONLY 11 OF THE 32 BITS CAN BE SEEN
                                                                ;IN THE PATERN REGISTER
                                                                ;DCK SHOULD BE SET IN RHER1
037752  013746  050464        6$:     MOV     @#GECC1,-(SP)   ;GET PATTERN REGISTER
037756  042716  174000                BIC     #174000,(SP)    ;KEEP ONLY 11 BITS
```

```
5883  037762  022637  015064              CMP     (SP)+,@#EC2     ;COMPARE PATTERN REGISTER
5884  037766  001401                      BEQ     7$              ;BRANCH IF GOOD
5885  037770  104032                      ERROR   32              ;11 BITS OF THE 32 BIT ECC REGISTER INCORRECT
5886
5887  037772  004037  051106        7$:   JSR     R0,@#ECORR      ;GO TO ECC CORRECTION PROCESS
5888  037776  000026              8$:   22.                     ;EXPECTED POSITION REG. WHEN CORRECTION
5889                                                            ;IS COMPLETE
5890
5891
5892
5893  040000  004737  046214              JSR     PC,@#CHECKE     ;CHECK THAT DVA,RDY,DPR,DRY = 1
5894  040004  104401  005123              TYPE    ,CPHALT         ;CANNOT CONTINUE IF THEY DON'T
5895  040010  000000                      HALT                    ;STOP THE TEST AND RESTART PROGRAM
5896  040012  012700  015220              MOV     #WRFROM,R0      ;GETTING READY TO FILL EXPECTED DATA
5897  040016  012720  010000              MOV     #0!FMT22,(R0)+  ;CYLINDER 0
5898  040022  112746  000000              MOVB    #0,-(SP)        ;IN LOWER BYTE GET SECTOR
5899  040026  112766  000000  000001      MOVB    #0,1(SP)        ;GET TRACK IN HIGHER BYTE
5900  040034  012620                      MOV     (SP)+,(R0)+     ;GET TRACK/SECTOR IN BUFFER
5901  040036  012720  000000              MOV     #0,(R0)+        ;KEY1 IN BUFFER
5902  040042  012720  000000              MOV     #0,(R0)+        ;KEY2 IN BUFFER
5903  040046  012701  000400              MOV     #256.,R1        ;DATA WORD COUNTER
5904  040052  012702  177777              MOV     #-1,R2          ;DATA
5905  040056  010220              3$:     MOV     R2,(R0)+        ;DATA INTO BUFFER
5906  040060  005301                      DEC     R1              ;COUNT
5907  040062  001375                      BNE     3$              ;BRANCH IF 256 NOT DONE
5908
5909                                      ;*ONLY GOOD DATA HAS BEEN PUT IN 'WRFROM'
5910                                      ;*NOW THE INSERTED ERROR WILL BE PUT IN
5911
5912  040064  012737  077777  015232      MOV     #77777,@#WRFROM+<5*2>   ;INSERTED ERROR
5913  040072  004737  045470              JSR     PC,@#PUTREG     ;SAVE REGISTERS
5914  040076  005037  015124              CLR     @#ERFLG$        ;CLEAR ERROR FLAG
5915
5916
5917                                      ;*NOW READ DATA BUFFER WILL BE CHECKED
5918
5919  040102  004037  046664              JSR     R0,@#COMPAR     ;CHECK
5920  040106  015220                      WRFROM                  ;GOOD BUFFER
5921  040110  016264                      REINTO                  ;TEST BUFFER
5922  040112  000404                      4+256.                  ;NUMBER OF WORDS CHECKED
5923  040114  040122                      4$                      ;RETURN POINT FOR ERROR HEADER
5924  040116  040126                      5$                      ;RETURN POINT FOR ERROR DATA
5925  040120  040132                      TST62                   ;RETURN FOR GOOD COMPARISON
5926  040122  104004              4$:     ERROR   4               ;READ NEXT ERROR
5927  040124  000207                      RTS     PC              ;RETURN TO ''COMPAR''
5928  040126  104005              5$:     ERROR   5               ;WORD NOS 1 TO 4 ARE
5929                                                            ;HEADER WORDS
5930                                                            ;5 TO 260 ARE DATA WORDS
5931  040130  000207                      RTS     PC              ;RETURN TO ''COMPAR''
5932
```

```
5933
5934  040132  000004                    TST62:  SCOPE
5935  040134  012706  001000                    MOV     #STACK,SP       ;RESET STACK
5936
5937  040140  012737  000062  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
5938
5939
5940                                      ;       SETUP FOR WHAT IS TO BE READ
5941                                      ;       HEADER CRC IS RESTORED FROM A SUBROUTINE
5942
5943  040146  012746  177777                    MOV     #-1,-(SP)       ;DATA TO BE READ
5944  040152  012705  000400                    MOV     #256.,R5        ;COUNTER
5945  040156  012700  054572                    MOV     #DISK,R0        ;START OF SIMULATED DISK DATA
5946  040162  011620                    1$:     MOV     (SP),(R0)+      ;MOVE IN DATA ON TO SIMULATED DISK
5947  040164  005305                            DEC     R5              ;COUNT
5948  040166  001375                            BNE     1$              ;BRANCH IF 256 NOT COMPLETE
5949  040170  005726                            TST     (SP)+           ;UNDO -(SP)
5950  040172  022020                            CMP     (R0)+,(R0)+     ;JUMP OVER THE TWO ECC WORDS
5951  040174  012705  000017                    MOV     #15.,R5         ;1 DATA GAP
5952                                                                    ;14 TOLERANCE GAP
5953  040200  005020                    2$:     CLR     (R0)+           ;CLEAR DATA GAP, AND
5954  040202  005305                            DEC     R5              ;TOLERANCE GAP
5955  040204  001375                            BNE     2$              ;BRANCH IF NOT COMPLETE
5956
5957
5958  040206  004737  051260                    JSR     PC,@#FILLEC     ;INSERT THE TWO ECC WORDS ON THE DISK
5959                                                                    ;IN THE CORRECT PLACE
5960
5961                                            ;*THESE ARE FOR ECC TEST ONLY
5962
5963  040212  012737  177777  015142            MOV     #-1,@#TSECC     ;THIS IS AN ECC TEST
5964  040220  005037  050476                    CLR     @#POSITI        ;CLEAR ERROR POSITION COUNTER
5965  040224  013737  050472  050474            MOV     @#NCODE,@#NCOUNT  ;TEMPORARY N-CODE COUNTER
5966  040232  013737  050500  050506            MOV     @#HARDER,@#HADTMP  ;TEMPORARY HARD ERROR COUNTER
5967  040240  005037  050464                    CLR     @#GECC1         ;ECC LOW ORDER TO BE GENERATED
5968  040244  005037  050466                    CLR     @#GECC2         ;ECC HIGH ORDER TO BE GENERATED
5969  040250  005037  050502                    CLR     @#DATENV        ;CLEAR DATA ENVELOPE CLOCK COUNT
5970  040254  005037  050504                    CLR     @#ZCODE         ;CLEAR LEADING ZEROS CLOCK COUNT
5971
5972
5973                                            ;*THESE ARE TO SETUP FOR DISKLESS USE ONLY
5974
5975  040260  012737  010000  052654            MOV     #FMT22,@#CYL    ;16 BITS PER WORD
5976                                                                    ;CYLINDER 0, FORMAT 16 BITS
5977  040266  112737  000000  052657            MOVB    #0,@#SECOTR+1   ;TRACK 0
5978  040274  112737  000000  052656            MOVB    #0,@#SECOTR     ;SECTOR 0
5979  040302  012737  000000  052660            MOV     #0,@#KEY1       ;KEY1=0
5980  040310  012737  000000  052662            MOV     #0,@#KEY2       ;KEY2=0
5981  040316  012737  000400  052734            MOV     #256.,@#DAWORD  ;NO. OF DATA WORDS
5982  040324  005037  052664                    CLR     @#X             ;THIS IS A READ COMMAND
5983  040330  004537  047176                    JSR     R5,@#CRC        ;GO TO CALCULATE CRC
5984  040334  052654                            CYL
5985  040336  054554                            WCRC
5986
5987
5988                                            ;*THIS IS TO INSERT ERROR
```

```
 5989                                   ;*THE DISK DATA IS IN LOCATION STARTING FROM 'DISK'
 5990                                   ;*THE POSITION OF THE ERROR CAN BE CHANGED BY CHANGING
 5991                                   ;*THIS MOVE
 5992
 5993  040340  012737  077757  054574   MOV     #77757,@#DISK+2 ;FORCE ERROR ON BIT NUMBER 32 AND 21
 5994                                                           ;SO ERROR POSITION REGISTER WILL SHOW
 5995                                                           ;22
 5996  040346  012737  010040  040522   MOV     #4128.,@#8$     ;INSERT POSITION REG.
 5997
 5998
 5999                                   ;*THESE ARE REGULAR SETUPS
 6000
 6001  040354  004737  045770           JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
 6002  040360  012777  177374  154362   MOV     #-256.-4.,@RHWC ;256. DATA 4 HEADER WORDS
 6003  040366  012777  016264  154356   MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
 6004  040374  112746  000000           MOVB    #0,-(SP)        ;IN LOWER BYTE GET SECTOR
 6005  040400  112766  000000  000001   MOVB    #0,1(SP)        ;GET TRACK IN HIGHER BYTE
 6006  040406  012677  154350           MOV     (SP)+,  @RHDST  ;TRACK/SECTOR IN RHDST
 6007  040412  012777  010000  154346   MOV     #FMT22,@RHOF    ;16 BITS PER WORD
 6008                                                           ;ECC CORRECTION NOT INHIBIT
 6009                                                           ;BECAUSE ECC IS NOT GOING
 6010                                                           ;TO BE CHECKED
 6011  040420  005077  154344           CLR     @RHCA           ;CYLINDER 0
 6012
 6013  040424  004737  046024           JSR     PC,@#CHECKT     ;CHECK DVA,RDY,DPR,DRY = 1 AND OTHERS DON'T
 6014  040430  104401  005123           TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF THE
 6015  040434  000000                   HALT                    ;STOP THE TEST
 6016
 6017  040436  013711  015176           MOV     @#REFOR,@R1     ;READ HEADER AND DATA=72
 6018  040442  005037  015124           CLR     @#ERFLG$        ;CLEAR ERROR FLAG
 6019  040446  004737  052514           JSR     PC,@#COMHD      ;READ HEADER AND DATA
 6020                                                           ;IF THERE ARE READ ERRORS THEN
 6021                                                           ;ECC WILL NOT BE CHECKED
 6022
 6023
 6024                                   ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
 6025                                   ;*FROM THE ''COMHD'' ROUTINE THAT MEANS SECTOR GAP,
 6026                                   ;*FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
 6027                                   ;*SYNC BYTE HAVE GONE BY AND SYNCS WERE CORRECTLY
 6028                                   ;*DETECTED
 6029                                   ;*HEADER AND DATA ARE TO BE CHECKED.
 6030                                   ;*IN CHECKING READ DATA THE WRITE FROM BUFFER
 6031                                   ;*'WRFROM'' IS FILLED WITH EXPECTED DATA AND
 6032                                   ;*COMPARISONS ARE MADE
 6033
 6034  040452  005737  015124           TST     @#ERFLG$        ;ANY ERRORS ALREADY THERE
 6035  040456  001106                   BNE     TST63           ;BRANCH IF YES
 6036  040460  004737  045470           JSR     PC,@#PUTREG     ;SAVE REGISTERS
 6037  040464  022737  100000  015034   CMP     #DCK,@#ER1      ;ONLY DATA CHECK ERROR SHOULD BE SET
 6038  040472  001401                   BEQ     6$              ;BRANCH IF YES
 6039  040474  104032                   ERROR   32              ;32 BIT ECC REGISTER SHOULD BE NON
 6040                                                           ;ZERO
 6041                                                           ;ONLY 11 OF THE 32 BITS CAN BE SEEN
 6042                                                           ;IN THE PATERN REGISTER
 6043                                                           ;DCK SHOULD BE SET IN RHER1
 6044  040476  013746  050464   6$:     MOV     @#GECC1,-(SP)   ;GET PATTERN REGISTER
```

```
6045  040502  042716  174000                  BIC     #174000,(SP)      ;KEEP ONLY 11 BITS
6046  040506  022637  015064                  CMP     (SP)+,@#EC2       ;COMPARE PATTERN REGISTER
6047  040512  001401                          BEQ     7$                ;BRANCH IF GOOD
6048  040514  104032                          ERROR   32                ;11 BITS OF THE 32 BIT ECC REGISTER INCORRECT
6049
6050  040516  004037  051106            7$:   JSR     R0,@#ECORR        ;GO TO ECC CORRECTION PROCESS
6051  040522  000000            8$:   .WORD                             ;EXPECTED POSITION REG. WHEN CORRECTION
6052                                                                    ;IS COMPLETE
6053
6054  040524  004737  045470                  JSR     PC,@#PUTREG       ;SAVE REGISTERS
6055
6056
6057  040530  022737  100100  015034          CMP     #DCK!ECH,@#ER1    ;WITH ERRORS INSERTED IN BIT POSITION 21
6058                                                                    ;AND 32 HARD ERROR BIT SHOULD SET
6059  040536  001401                          BEQ     9$                ;BRANCH IF GOOD
6060  040540  104036                          ERROR   36                ;WITH ERROR INSERTED IN BIT POSITION 21 THRU
6061                                                                    ;32 HCE SHOULD SET
6062
6063
6064
6065
6066  040542                          9$:
6067  040542  004737  046214                  JSR     PC,@#CHECKE       ;CHECK THAT DVA,RDY,DPR,DRY = 1
6068  040546  104401  005123                  TYPE    ,CPHALT           ;CANNOT CONTINUE IF THEY DON'T
6069  040552  000000                          HALT                      ;STOP THE TEST AND RESTART PROGRAM
6070  040554  012700  015220                  MOV     #WRFROM,R0        ;GETTING READY TO FILL EXPECTED DATA
6071  040560  012720  010000                  MOV     #0!FMT22,(R0)+    ;CYLINDER 0
6072  040564  112746  000000                  MOVB    #0,-(SP)          ;IN LOWER BYTE GET SECTOR
6073  040570  112766  000000  000001          MOVB    #0,1(SP)          ;GET TRACK IN HIGHER BYTE
6074  040576  012620                          MOV     (SP)+,(R0)+       ;GET TRACK/SECTOR IN BUFFER
6075  040600  012720  000000                  MOV     #0,(R0)+          ;KEY1 IN BUFFER
6076  040604  012720  000000                  MOV     #0,(R0)+          ;KEY2 IN BUFFER
6077  040610  012701  000400                  MOV     #256.,R1          ;DATA WORD COUNTER
6078  040614  012702  177777                  MOV     #-1,R2            ;DATA
6079  040620  010220                  3$:     MOV     R2,(R0)+          ;DATA INTO BUFFER
6080  040622  005301                          DEC     R1                ;COUNT
6081  040624  001375                          BNE     3$                ;BRANCH IF 256 NOT DONE
6082
6083                                          ;*ONLY GOOD DATA HAS BEEN PUT IN 'WRFROM'
6084                                          ;*NOW THE INSERTED ERROR WILL BE PUT IN
6085
6086  040626  012737  077757  015232          MOV     #77757,@#WRFROM+<5*2>   ;INSERTED ERROR
6087  040634  004737  045470                  JSR     PC,@#PUTREG       ;SAVE REGISTERS
6088  040640  005037  015124                  CLR     @#ERFLG$          ;CLEAR ERROR FLAG
6089
6090
6091                                          ;*NOW READ DATA BUFFER WILL BE CHECKED
6092
6093  040644  004037  046664                  JSR     R0,@#COMPAR       ;CHECK
6094  040650  015220                          WRFROM                    ;GOOD BUFFER
6095  040652  016264                          REINTO                    ;TEST BUFFER
6096  040654  000404                          4+256.                    ;NUMBER OF WORDS CHECKED
6097  040656  040664                          4$                        ;RETURN POINT FOR ERROR HEADER
6098  040660  040670                          5$                        ;RETURN POINT FOR ERROR DATA
6099  040662  040674                          TST63                     ;RETURN FOR GOOD COMPARISON
6100  040664  104004                  4$:     ERROR   4                 ;READ NEXT ERROR
```

```
 6101  040666  000207                    RTS     PC              ;RETURN TO ''COMPAR''
 6102  040670  104005              5$:   ERROR   5               ;WORD NOS 1 TO 4 ARE
 6103                                                            ;HEADER WORDS
 6104                                                            ;5 TO 260 ARE DATA WORDS
 6105  040672  000207                    RTS     PC              ;RETURN TO ''COMPAR''
```

```
 6106
 6107  040674  000004                    TST63:  SCOPE
 6108  040676  012706  001000                    MOV     #STACK,SP       ;RESET STACK
 6109
 6110  040702  012737  000063  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
 6111  040710  012700  054474                    MOV     #SECGAP,R0      ;POINTER
 6112  040714  012701  000460                    MOV     #304.,R1        ;COUNTER
 6113  040720  005020                    1$:     CLR     (R0)+           ;CLEAR SIMULATED DISK AREA
 6114  040722  005301                            DEC     R1
 6115  040724  001375                            BNE     1$
 6116  040726  004737  045770                    JSR     PC,CLDISK       ;THIS IS USED TO SET GENERAL REGISTERS
 6117
 6118                                            ;*THESE ARE FOR ECC TEST ONLY
 6119
 6120  040732  012737  177777  015142            MOV     #-1,@#TSECC     ;THIS IS AN ECC TEST
 6121  040740  005037  050476                    CLR     @#POSITI        ;CLEAR ERROR POSITION COUNTER
 6122  040744  013737  050472  050474            MOV     @#NCODE,@#NCOUNT  ;TEMPORARY N-CODE COUNTER
 6123  040752  013737  050500  050506            MOV     @#HARDER,@#HADTMP  ;TEMPORARY HARD ERROR COUNTER
 6124  040760  005037  050464                    CLR     @#GECC1         ;ECC LOW ORDER TO BE GENERATED
 6125  040764  005037  050466                    CLR     @#GECC2         ;ECC HIGH ORDER TO BE GENERATED
 6126  040770  005037  050502                    CLR     @#DATENV        ;CLEAR DATA ENVELOPE CLOCK COUNT
 6127  040774  005037  050504                    CLR     @#ZCODE         ;CLEAR LEADING ZEROS CLOCK COUNT
 6128
 6129
 6130
 6131
 6132                                            ;*THESE ARE TO BE SETUP FOR DISKLESS USE ONLY
 6133
 6134  041000  012737  010000  056012            MOV     #FMT22,@#WCYL   ;FORMAT22=16BIT WORDS AND
 6135                                            ;CYLINDER 0
 6136  041006  012737  000001  056014            MOV     #1,@#WSECTR     ;TRACK=0, SECTOR=1
 6137  041014  005037  056016                    CLR     @#WKEY1         ;KEY1=0
 6138  041020  005037  056020                    CLR     @#WKEY2         ;KEY2=0
 6139  041024  012737  000400  056052            MOV     #256.,@#FNWORD  ;256 DATA WORDS
 6140  041032  004537  047176                    JSR     R5,@#CRC        ;GO TO CALCULATE CRC
 6141  041036  056012                            WCYL
 6142  041040  056022                            GCRC
 6143
 6144                                            ;*THESE ARE REGULAR SETUPS
 6145
 6146  041042  012777  177374  153700            MOV     #-260.,@RHWC    ;256 DATA WORDS 4 HEADER WORDS
 6147  041050  012700  015220                    MOV     #WRFROM,R0      ;THESE TWO INSTRUCTIONS GETS
 6148  041054  010077  153672                    MOV     R0,@RHBA        ;ADDR. OF WRFROM INTO R0 AND
 6149                                            ;BUS ADDRESS REGISTER
 6150  041060  012720  010000                    MOV     #FMT22,(R0)+    ;FORMAT=16 BIT WORDS
 6151                                            ;CYLINDER=0
 6152  041064  012720  000001            2$:     MOV     #1,(R0)+        ;TRACK=0, SECTOR=1, KEYS=0
 6153  041070  005020                            CLR     (R0)+           ;KEY1=0
 6154  041072  005020                            CLR     (R0)+           ;KEY2=0
 6155  041074  012705  000400                    MOV     #256.,R5        ;COUNTER
 6156  041100  012720  052525            3$:     MOV     #52525,(R0)+    ;MOVE ALL 52525 FOR DATA
 6157  041104  005305                            DEC     R5
 6158  041106  001374                            BNE     3$              ;BRANCH IF DATA NOT COMPLETE
 6159  041110  012777  000001  153644            MOV     #1,@RHDST       ;TRACK=0 SECTOR=1
 6160
 6161  041116  004737  046024                    JSR     PC,@#CHECKT     ;CHECK DVA,RDY,DPR,DRY = 1 AND OTHERS DON'T
```

```
6162  041122  104401  005123                TYPE    ,CPHALT          ;CANNOT CONTINUE TESTING IF ANY OF THE
6163  041126  000000                        HALT                     ;STOP THE TEST
6164
6165  041130  013711  015172                MOV     @#WRIFOR,@R1     ;GET READY FOR WRITE HEADER AND
6166                                                                   ;DATA WITH 62 IN RHCS1
6167  041134  005037  015124                CLR     @#ERFLG$         ;CLEAR ERROR FLAG
6168  041140  012777  010000  153620        MOV     #FMT22,@RHOF     ;FORMAT BIT=1 (16 BIT WORDS)
6169  041146  005077  153616                CLR     @RHCA            ;CYLINDER =0
6170  041152  004737  055636                JSR     PC,@#COMWHD      ;WRITE HEADER AND DATA
6171
6172                                        ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
6173                                        ;*FROM THE ''COMWHD'' ROUTINE THAT MEANS ALL HEADER ON DISK
6174                                        ;*IS GOOD IE. ONLY DATA IS TO BE CHECKED TO SEE IF THEY ARE
6175                                        ;*ALL 52525 AND WRITE DATA GAP AND TOLERANCE GAP TO SEE IF
6176                                        ;*THEY ARE ALL ZEROS
6177
6178  041156  005737  015124                TST     @#ERFLG$         ;HAS ANY ERRORS OCCURED?
6179                                                                   ;IF WRITE ERROR OCCURS ECC IS NOT CHECKED
6180  041162  001056                        BNE     TST64            ;;BRANCH IF YES
6181
6182
6183                              ;COMPARE SOFTWARE GENERATED ECC WITH THAT GENERATED BY HARDWARE
6184  041164  023737  050464  055572        CMP     @#GECC1,@#WECC1;COMPARE SOFTWARE ECC WITH HARDWARE ECC
6185  041172  001402                        BEQ     6$               ;BRANCH IF GOOD
6186  041174  104031                        ERROR   31               ;LOW ORDER ECC IN ERROR
6187  041176  000405                        BR      7$               ;BRANCH TO CONTINUE
6188  041200  023737  050466  055574  6$:   CMP     @#GECC2,@#WECC2 ;COMPARE SOFTWARE ECC WITH HARDWARE ECC
6189  041206  001401                        BEQ     7$               ;BRANCH IF GOOD
6190  041210  104031                        ERROR   31               ;HIGH ORDER ECC IN ERROR
6191
6192
6193  041212                          7$:
6194  041212  004737  046214                JSR     PC,@#CHECKE      ;CHECK THAT DVA,RDY,DPR,DRY = 1
6195  041216  104401  005123                TYPE    ,CPHALT          ;CANNOT CONTINUE IF THEY DON'T
6196  041222  000000                        HALT                     ;STOP THE TEST AND RESTART PROGRAM
6197
6198
6199
6200
6201                                        ;*FILL ''REINTO'' BUFFER WITH EXPECTED DATA
6202
6203  041224  004037  045706                JSR     R0,@#CLAREA      ;FILL REINTO BUFFER
6204  041230  016264                                REINTO           ;FROM
6205  041232  017262                                REINTO+<255.*2>  ;TO
6206  041234  052525                        .WORD   52525            ;DATA
6207
6208  041236  013737  050464  017264        MOV     @#GECC1,@#REINTO+<256.*2>;FILL ECC1
6209  041244  013737  050466  017266        MOV     @#GECC2,@#REINTO+<257.*2>;FILL ECC2
6210  041252  004037  045706                JSR     R0,@#CLAREA      ;FILL REST
6211  041256  017270                        REINTO+<258.*2>          ;FROM
6212  041260  017324                        REINTO+<272.*2>          ;TO
6213  041262  000000                        0                        ;DATA
6214
6215
6216  041264  005037  015124                CLR     @#ERFLG$         ;CLEAR ERROR FLAG
6217
```

```
6218
6219                                    ;*NOW COMPARE 'DISK' BUFFER WITH 'REINTO'
6220
6221  041270  004037  046664            JSR     R0,@#COMPAR     ;CHECK
6222  041274  016264                    REINTO                  ;GOOD BUFFER
6223  041276  054572                    DISK                    ;TEST BUFFER
6224  041300  000402                    258.                    ;NUMBER OF WORDS CHECKED
6225  041302  041310                    4$                      ;RETURN POINT FOR ERROR HEADER
6226  041304  041314                    5$                      ;RETURN POINT FOR ERROR DATA
6227  041306  041320                    TST64                   ;RETURN FOR GOOD COMPARISON
6228  041310  104007            4$:     ERROR   7               ;READ ERROR 10 NEXT
6229  041312  000207                    RTS     PC              ;RETURN TO COMPARE
6230  041314  104010            5$:     ERROR   10              ;WORD NOS 1 TO 256 ARE
6231                                                            ;DATA WORDS
6232                                                            ;WORD NOS 257 AND 258
6233                                                            ;ARE ECC WHICH ARE CHECKED
6234                                                            ;WORD NOS 259
6235                                                            ;IS DATA GAP
6236                                                            ;WORD NOS 260 TO 273
6237                                                            ;ARE TOLERANCE GAP
6238  041316  000207                    RTS     PC              ;RETURN TO COMPARE
```

```
6239
6240  041320  000004                          TST64:  SCOPE
6241  041322  012706  001000                          MOV     #STACK,SP       ;RESET STACK
6242  041326  012737  000064  017330                  MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
6243
6244
6245                                                  ;*SETUP FOR WHAT IS TO BE READ
6246                                                  ;*HEADER CRC IS RESTORED FROM A SUBROUTINE
6247
6248  041334  012746  052525                          MOV     #52525,-(SP)    ;DATA TO BE READ
6249  041340  012705  000400                          MOV     #256.,R5        ;COUNTER
6250  041344  012700  054572                          MOV     #DISK,R0        ;START OF SIMULATED DISK DATA
6251  041350  011620                          1$:     MOV     (SP),(R0)+      ;MOVE IN DATA ON TO SIMULATED DISK
6252  041352  005305                                  DEC     R5              ;COUNT
6253  041354  001375                                  BNE     1$              ;BRANCH IF 256 NOT COMPLETE
6254  041356  005726                                  TST     (SP)+           ;UNDO -(SP)
6255  041360  022020                                  CMP     (R0)+,(R0)+     ;JUMP OVER THE TWO ECC WORDS
6256  041362  012705  000017                          MOV     #15.,R5         ;1 DATA GAP
6257                                                                          ;14 TOLERANCE GAP
6258  041366  005020                          2$:     CLR     (R0)+           ;CLEAR DATA GAP, AND
6259  041370  005305                                  DEC     R5              ;TOLERANCE GAP
6260  041372  001375                                  BNE     2$              ;BRANCH IF NOT COMPLETE
6261
6262  041374  004737  051260                          JSR     PC,@#FILLEC     ;INSERT THE TWO ECC WORDS ON THE DISK
6263                                                                          ;IN THE CORRECT PLACE
6264
6265                                                  ;*THESE ARE FOR ECC TEST ONLY
6266
6267  041400  012737  177777  015142                  MOV     #-1,@#TSECC     ;THIS IS AN ECC TEST
6268  041406  005037  050476                          CLR     @#POSITI        ;CLEAR ERROR POSITION COUNTER
6269  041412  013737  050472  050474                  MOV     @#NCODE,@#NCOUNT  ;TEMPORARY N-CODE COUNTER
6270  041420  013737  050500  050506                  MOV     @#HARDER,@#HADTMP  ;TEMPORARY HARD ERROR COUNTER
6271  041426  005037  050464                          CLR     @#GECC1         ;ECC LOW ORDER TO BE GENERATED
6272  041432  005037  050466                          CLR     @#GECC2         ;ECC HIGH ORDER TO BE GENERATED
6273  041436  005037  050502                          CLR     @#DATENV        ;CLEAR DATA ENVELOPE CLOCK COUNT
6274  041442  005037  050504                          CLR     @#ZCODE         ;CLEAR LEADING ZEROS CLOCK COUNT
6275
6276
6277                                                  ;*THESE ARE TO SETUP FOR DISKLESS USE ONLY
6278
6279  041446  012737  010000  052654                  MOV     #FMT22,@#CYL    ;16 BITS PER WORD
6280                                                                          ;CYLINDER 0, FORMAT 16 BITS
6281  041454  112737  000000  052657                  MOVB    #0,@#SECOTR+1   ;TRACK 0
6282  041462  112737  000000  052656                  MOVB    #0,@#SECOTR     ;SECTOR 0
6283  041470  012737  000000  052660                  MOV     #0,@#KEY1       ;KEY1=0
6284  041476  012737  000000  052662                  MOV     #0,@#KEY2       ;KEY2=0
6285  041504  012737  000400  052734                  MOV     #256.,@#DAWORD  ;NO. OF DATA WORDS
6286  041512  005037  052664                          CLR     @#X             ;THIS IS A READ COMMAND
6287  041516  004537  047176                          JSR     R5,@#CRC        ;GO TO CALCULATE CRC
6288  041522  052654                                  CYL
6289  041524  054554                                  WCRC
6290
6291
6292
6293
6294                                                  ;*THESE ARE REGULAR SETUPS
```

```
6295
6296  041526  004737  045770                  JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
6297  041532  012777  177374  153210          MOV     #-256.-4.,@RHWC ;256. DATA 4 HEADER WORDS
6298  041540  012777  016264  153204          MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
6299  041546  112746  000000                  MOVB    #0,-(SP)        ;IN LOWER BYTE GET SECTOR
6300  041552  112766  000000  000001          MOVB    #0,1(SP)        ;GET TRACK IN HIGHER BYTE
6301  041560  012677  153176                  MOV     (SP)+,  @RHDST  ;TRACK/SECTOR IN RHDST
6302  041564  012777  010000  153174          MOV     #FMT22,@RHOF ;16 BITS PER WORD
6303                                                                  ;ECC CORRECTION NOT INHIBIT
6304                                                                  ;BECAUSE ECC IS NOT GOING
6305                                                                  ;TO BE CHECKED
6306  041572  005077  153172                  CLR     @RHCA           ;CYLINDER 0
6307
6308  041576  004737  046024                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,DPR,DRY = 1 AND OTHERS DON'T
6309  041602  104401  005123                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF THE
6310  041606  000000                          HALT                    ;STOP THE TEST
6311
6312  041610  013711  015176                  MOV     @#REFOR,@R1     ;READ HEADER AND DATA=72
6313  041614  005037  015124                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
6314  041620  004737  052514                  JSR     PC,@#COMHD      ;READ HEADER AND DATA
6315                                                                  ;IF THERE ARE READ ERRORS THEN
6316                                                                  ;ECC WILL NOT BE CHECKED
6317
6318
6319                                          ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
6320                                          ;*FROM THE ''COMHD'' ROUTINE THAT MEANS SECTOR GAP,
6321                                          ;*FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
6322                                          ;*SYNC BYTE HAVE GONE BY AND SYNCS WERE CORRECTLY
6323                                          ;*DETECTED
6324                                          ;*HEADER AND DATA ARE TO BE CHECKED.
6325                                          ;*IN CHECKING READ DATA THE WRITE FROM BUFFER
6326                                          ;*''WRFROM'' IS FILLED WITH EXPECTED DATA AND
6327                                          ;*COMPARISONS ARE MADE
6328
6329  041624  005737  015124                  TST     @#ERFLG$        ;ANY ERRORS ALREADY THERE
6330  041630  001102                          BNE     TST65           ;BRANCH IF YES
6331  041632  004737  045470                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
6332  041636  005737  015034                  TST     @#ER1           ;NO ERRORS SHOULD BE SET
6333  041642  001401                          BEQ     6$              ;BRANCH IF NO ERRORS SET
6334  041644  104032                          ERROR   32              ;32 BIT ECC REGISTER SHOULD BE ZERO
6335                                                                  ;ONLY 11 OF THE 32 BITS CAN BE SEEN
6336                                                                  ;IN THE PATERN REGISTER
6337                                                                  ;DCK SHOULD BE SET IN RHER1
6338  041646  013746  050464          6$:     MOV     @#GECC1,-(SP)   ;GET PATTERN REGISTER
6339  041652  042716  174000                  BIC     #174000,(SP)    ;KEEP ONLY 11 BITS
6340  041656  022637  015064                  CMP     (SP)+,@#EC2     ;COMPARE PATTERN REGISTER
6341  041662  001401                          BEQ     7$              ;BRANCH IF GOOD
6342  041664  104032                          ERROR   32              ;11 BITS OF THE 32 BIT ECC REGISTER INCORRECT
6343
6344
6345
6346
6347                                          ;*ADD 16 MAINTENANCE CLOCKS TO
6348                                          ;*BRING EBL DOWN
6349
6350  041666  012700  000020          7$:     MOV     #16.,R0         ;COUNTER
```

```
6351  041672  052777  000002  153076  8$:     BIS     #MCLK,@RHMR     ;SET CLOCK
6352  041700  042777  000002  153070          BIC     #MCLK,@RHMR     ;CLEAR CLOCK
6353  041706  005300                          DEC     R0              ;COUNT
6354  041710  001370                          BNE     8$              ;BRANCH IF 16 CLOCKS NOT DONE
6355  041712  004737  046214                  JSR     PC,@#CHECKE     ;CHECK THAT DVA,RDY,DPR,DRY = 1
6356  041716  104401  005123                  TYPE    ,CPHALT         ;CANNOT CONTINUE IF THEY DON'T
6357  041722  000000                          HALT                    ;STOP THE TEST AND RESTART PROGRAM
6358  041724  012700  015220                  MOV     #WRFROM,R0      ;GETTING READY TO FILL EXPECTED DATA
6359  041730  012720  010000                  MOV     #0!FMT22,(R0)+  ;CYLINDER 0
6360  041734  112746  000000                  MOVB    #0,-(SP)        ;IN LOWER BYTE GET SECTOR
6361  041740  112766  000000  000001          MOVB    #0,1(SP)        ;GET TRACK IN HIGHER BYTE
6362  041746  012620                          MOV     (SP)+,(R0)+     ;GET TRACK/SECTOR IN BUFFER
6363  041750  012720  000000                  MOV     #0,(R0)+        ;KEY1 IN BUFFER
6364  041754  012720  000000                  MOV     #0,(R0)+        ;KEY2 IN BUFFER
6365  041760  012701  000400                  MOV     #256.,R1        ;DATA WORD COUNTER
6366  041764  012702  052525                  MOV     #52525,R2       ;DATA
6367  041770  010220                  3$:     MOV     R2,(R0)+        ;DATA INTO BUFFER
6368  041772  005301                          DEC     R1              ;COUNT
6369  041774  001375                          BNE     3$              ;BRANCH IF 256 NOT DONE
6370  041776  005037  015124                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
6371  042002  004737  045470                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
6372
6373                                          ;*NOW READ DATA BUFFER WILL BE CHECKED
6374
6375  042006  004037  046664                  JSR     R0,@#COMPAR     ;CHECK
6376  042012  015220                          WRFROM                  ;GOOD BUFFER
6377  042014  016264                          REINTO                  ;TEST BUFFER
6378  042016  000404                          4+256.                  ;NUMBER OF WORDS CHECKED
6379  042020  042026                          4$                      ;RETURN POINT FOR ERROR HEADER
6380  042022  042032                          5$                      ;RETURN POINT FOR ERROR DATA
6381  042024  042036                          TST65                   ;RETURN FOR GOOD COMPARISON
6382  042026  104004          4$:     ERROR   4                       ;READ NEXT ERROR
6383  042030  000207                  RTS     PC                      ;RETURN TO ''COMPAR''
6384  042032  104005          5$:     ERROR   5                       ;WORD NOS 1 TO 4 ARE
6385                                                                  ;HEADER WORDS
6386                                                                  ;5 TO 260 ARE DATA WORDS
6387  042034  000207                  RTS     PC                      ;RETURN TO ''COMPAR''
```

```
6388
6389  042036  000004                    TST65:  SCOPE
6390  042040  012706  001000                    MOV     #STACK,SP       ;RESET STACK
6391  042044  012737  000065  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
6392
6393                                            ;*SETUP FOR WHAT IS TO BE READ
6394                                            ;*HEADER CRC IS RESTORED FROM A SUBROUTINE
6395
6396  042052  012746  052525                    MOV     #52525, -(SP)   ;DATA TO BE READ
6397  042056  012705  000400                    MOV     #256.,R5        ;COUNTER
6398  042062  012700  054572                    MOV     #DISK,R0        ;START OF SIMULATED DISK DATA
6399  042066  011620                    1$:     MOV     (SP),(R0)+      ;MOVE IN DATA ON TO SIMULATED DISK
6400  042070  005305                            DEC     R5              ;COUNT
6401  042072  001375                            BNE     1$              ;BRANCH IF 256 NOT COMPLETE
6402  042074  005726                            TST     (SP)+           ;UNDO -(SP)
6403  042076  022020                            CMP     (R0)+,(R0)+     ;JUMP OVER THE TWO ECC WORDS
6404  042100  012705  000017                    MOV     #15.,   R5      ;1 DATA GAP
6405                                                                    ;14 TOLERANCE GAP
6406  042104  005020                    2$:     CLR     (R0)+           ;CLEAR DATA GAP, AND
6407  042106  005305                            DEC     R5              ;TOLERANCE GAP
6408  042110  001375                            BNE     2$              ;BRANCH IF NOT COMPLETE
6409
6410
6411  042112  004737  051260                    JSR     PC,@#FILLEC     ;INSERT ECC IN PROPER PLACE ON DISK
6412
6413
6414
6415                                            ;*THESE ARE FOR ECC TEST ONLY
6416
6417  042116  012737  177777  015142            MOV     #-1,@#TSECC     ;THIS IS AN ECC TEST
6418  042124  005037  050476                    CLR     @#POSITI        ;CLEAR ERROR POSITION COUNTER
6419  042130  013737  050472  050474            MOV     @#NCODE,@#NCOUNT  ;TEMPORARY N-CODE COUNTER
6420  042136  013737  050500  050506            MOV     @#HARDER,@#HADTMP  ;TEMPORARY HARD ERROR COUNTER
6421  042144  005037  050464                    CLR     @#GECC1         ;ECC LOW ORDER TO BE GENERATED
6422  042150  005037  050466                    CLR     @#GECC2         ;ECC HIGH ORDER TO BE GENERATED
6423  042154  005037  050502                    CLR     @#DATENV        ;CLEAR DATA ENVELOPE CLOCK COUNT
6424  042160  005037  050504                    CLR     @#ZCODE         ;CLEAR LEADING ZEROS CLOCK COUNT
6425
6426
6427                                            ;*THESE ARE TO SETUP FOR DISKLESS USE ONLY
6428
6429  042164  012737  010000  052654            MOV     #FMT22,@#CYL    ;16 BITS PER WORD
6430                                                                    ;CYLINDER 0, FORMAT 16 BITS
6431  042172  112737  000000  052657            MOVB    #0,@#SECOTR+1   ;TRACK 0
6432  042200  112737  000000  052656            MOVB    #0,@#SECOTR     ;SECTOR 0
6433  042206  012737  000000  052660            MOV     #0,@#KEY1       ;KEY1=0
6434  042214  012737  000000  052662            MOV     #0,@#KEY2       ;KEY2=0
6435  042222  012737  000400  052734            MOV     #256.,@#DAWORD  ;NO. OF DATA WORDS
6436  042230  005037  052664                    CLR     @#X             ;THIS IS A READ COMMAND
6437  042234  004537  047176                    JSR     R5,@#CRC        ;GO TO CALCULATE CRC
6438  042240  052654                            CYL
6439  042242  054554                            WCRC
6440
6441
6442                                            ;*THIS IS TO INSERT ERROR
6443                                            ;*THE DISK DATA IS IN LOCATION STARTING FROM 'DISK'
```

```
6444                                        ;*THE POSITION OF THE ERROR CAN BE CHANGED BY CHANGING
6445                                        ;*THIS MOVE
6446                                        ;*THIS CHANGES THE LAST BIT OF THE ECC
6447
6448  042244  013746  055574              MOV     @#WECC2,-(SP)    ;GET LAST ECC
6449  042250  005116                      COM     (SP)             ;INVERT ALL BITS OF WECC2
6450  042252  042716  077777              BIC     #^C100000,(SP)   ;KEEP BIT 16
6451  042256  042737  100000  055574      BIC     #100000,@#WECC2  ;CLEAR BIT 16 IN ECC
6452  042264  052637  055574              BIS     (SP)+,@#WECC2    ;THIS WILL SET BIT 16 IF IT WAS 0
6453                                                               ;OR WILL SET NOTHING IF IT WAS A 1
6454
6455
6456  042270  012737  010026  042444      MOV     #4118.,@#8$      ;INSERT POSITION REG.
6457
6458
6459                                        ;*THESE ARE REGULAR SETUPS
6460
6461  042276  004737  045770              JSR     PC,@#CLDISK      ;SETUP GENERAL REGISTERS
6462  042302  012777  177374  152440      MOV     #-256.-4.,@RHWC  ;256. DATA 4 HEADER WORDS
6463  042310  012777  016264  152434      MOV     #REINTO,@RHBA    ;STARTING ADDRESS OF READ BUFFER
6464  042316  112746  000000              MOVB    #0,     -(SP)    ;IN LOWER BYTE GET SECTOR
6465  042322  112766  000000  000001      MOVB    #0,     1(SP)    ;GET TRACK IN HIGHER BYTE
6466  042330  012677  152426              MOV     (SP)+,  @RHDST   ;TRACK/SECTOR IN RHDST
6467  042334  012777  010000  152424      MOV     #FMT22,@RHOF ;16 BITS PER WORD
6468                                                               ;ECC CORRECTION NOT INHIBIT
6469                                                               ;BECAUSE ECC IS NOT GOING
6470                                                               ;TO BE CHECKED
6471  042342  005077  152422              CLR     @RHCA            ;CYLINDER 0
6472
6473  042346  004737  046024              JSR     PC,@#CHECKT      ;CHECK DVA,RDY,DPR,DRY = 1 AND OTHERS DON'T
6474  042352  104401  005123              TYPE    ,CPHALT          ;CANNOT CONTINUE TESTING IF ANY OF THE
6475  042356  000000                      HALT                     ;STOP THE TEST
6476
6477  042360  013711  015176              MOV     @#REFOR,@R1      ;READ HEADER AND DATA=72
6478  042364  005037  015124              CLR     @#ERFLG$         ;CLEAR ERROR FLAG
6479  042370  004737  052514              JSR     PC,@#COMHD       ;READ HEADER AND DATA
6480                                                               ;IF THERE ARE READ ERRORS THEN
6481                                                               ;ECC WILL NOT BE CHECKED
6482
6483
6484                                        ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
6485                                        ;*FROM THE ''COMHD'' ROUTINE THAT MEANS SECTOR GAP,
6486                                        ;*FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
6487                                        ;*SYNC BYTE HAVE GONE BY AND SYNCS WERE CORRECTLY
6488                                        ;*DETECTED
6489                                        ;*HEADER AND DATA ARE TO BE CHECKED.
6490                                        ;*IN CHECKING READ DATA THE WRITE FROM BUFFER
6491                                        ;*'WRFROM'' IS FILLED WITH EXPECTED DATA AND
6492                                        ;*COMPARISONS ARE MADE
6493
6494  042374  005737  015124              TST     @#ERFLG$         ;ANY ERRORS ALREADY THERE
6495  042400  001074                      BNE     TST66            ;BRANCH IF YES
6496  042402  004737  045470              JSR     PC,@#PUTREG      ;SAVE REGISTERS
6497  042406  022737  100000  015034      CMP     #DCK,@#ER1       ;ONLY DATA CHECK ERROR SHOULD BE SET
6498  042414  001401                      BEQ     6$               ;BRANCH IF YES
6499  042416  104032                      ERROR   32               ;32 BIT ECC REGISTER SHOULD BE NON
```

```
6500                                                    ;ZERO
6501                                                    ;ONLY 11 OF THE 32 BITS CAN BE SEEN
6502                                                    ;IN THE PATERN REGISTER
6503                                                    ;DCK SHOULD BE SET IN RHER1
6504  042420  013746  050464          6$:   MOV   @#GECC1,-(SP)    ;GET PATTERN REGISTER
6505  042424  042716  174000                BIC   #174000,(SP)    ;KEEP ONLY 11 BITS
6506  042430  022637  015064                CMP   (SP)+,@#EC2     ;COMPARE PATTERN REGISTER
6507  042434  001401                        BEQ   7$              ;BRANCH IF GOOD
6508  042436  104032                        ERROR 32              ;11 BITS OF THE 32 BIT ECC REGISTER INCORRECT
6509
6510  042440  004037  051106          7$:   JSR   R0,@#ECORR      ;GO TO ECC CORRECTION PROCESS
6511  042444  010026                8$:   4118.                   ;EXPECTED POSITION REG. WHEN CORRECTION
6512                                                            ;IS COMPLETE
6513
6514
6515
6516  042446  004737  046214                JSR   PC,@#CHECKE     ;CHECK THAT DVA,RDY,DPR,DRY = 1
6517  042452  104401  005123                TYPE  ,CPHALT         ;CANNOT CONTINUE IF THEY DON'T
6518  042456  000000                        HALT                  ;STOP THE TEST AND RESTART PROGRAM
6519  042460  012700  015220                MOV   #WRFROM,R0      ;GETTING READY TO FILL EXPECTED DATA
6520  042464  012720  010000                MOV   #0!FMT22,(R0)+  ;CYLINDER 0
6521  042470  112746  000000                MOVB  #0,-(SP)        ;IN LOWER BYTE GET SECTOR
6522  042474  112766  000000  000001        MOVB  #0,1(SP)        ;GET TRACK IN HIGHER BYTE
6523  042502  012620                        MOV   (SP)+,(R0)+     ;GET TRACK/SECTOR IN BUFFER
6524  042504  012720  000000                MOV   #0,(R0)+        ;KEY1 IN BUFFER
6525  042510  012720  000000                MOV   #0,(R0)+        ;KEY2 IN BUFFER
6526  042514  012701  000400                MOV   #256.,R1        ;DATA WORD COUNTER
6527  042520  012702  052525                MOV   #52525,R2       ;DATA
6528  042524  010220                3$:   MOV   R2,(R0)+        ;DATA INTO BUFFER
6529  042526  005301                        DEC   R1              ;COUNT
6530  042530  001375                        BNE   3$              ;BRANCH IF 256 NOT DONE
6531
6532                                                    ;*ONLY GOOD DATA HAS BEEN PUT IN 'WRFROM'
6533                                                    ;*NOW THE INSERTED ERROR WILL BE PUT IN
6534                                                    ;*BUT INSERTED ERROR IS IN ECC SO DATA IS NOT WRONG
6535
6536
6537  042532  004737  045470                JSR   PC,@#PUTREG     ;SAVE REGISTERS
6538  042536  005037  015124                CLR   @#ERFLG$        ;CLEAR ERROR FLAG
6539
6540
6541                                                    ;*NOW READ DATA BUFFER WILL BE CHECKED
6542
6543  042542  004037  046664                JSR   R0,@#COMPAR     ;CHECK
6544  042546  015220                        WRFROM                ;GOOD BUFFER
6545  042550  016264                        REINTO                ;TEST BUFFER
6546  042552  000404                        4+256.                ;NUMBER OF WORDS CHECKED
6547  042554  042562                        4$                    ;RETURN POINT FOR ERROR HEADER
6548  042556  042566                        5$                    ;RETURN POINT FOR ERROR DATA
6549  042560  042572                        TST66                 ;RETURN FOR GOOD COMPARISON
6550  042562  104004          4$:   ERROR 4                       ;READ NEXT ERROR
6551  042564  000207                        RTS   PC              ;RETURN TO ''COMPAR''
6552  042566  104005          5$:   ERROR 5                       ;WORD NOS 1 TO 4 ARE
6553                                                            ;HEADER WORDS
6554                                                            ;5 TO 260 ARE DATA WORDS
6555  042570  000207                        RTS   PC              ;RETURN TO ''COMPAR''
```

```
6556
```

```
6557
6558  042572  000004                    TST66:  SCOPE
6559  042574  012706  001000                    MOV     #STACK,SP       ;RESET STACK
6560  042600  012737  000066  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
6561
6562
6563                                      ;*SETUP FOR WHAT IS TO BE READ
6564                                      ;*HEADER CRC IS RESTORED FROM A SUBROUTINE
6565
6566  042606  012746  052525                    MOV     #52525,-(SP)    ;DATA TO BE READ
6567  042612  012705  000400                    MOV     #256.,R5        ;COUNTER
6568  042616  012700  054572                    MOV     #DISK,R0        ;START OF SIMULATED DISK DATA
6569  042622  011620                    1$:     MOV     (SP),(R0)+      ;MOVE IN DATA ON TO SIMULATED DISK
6570  042624  005305                            DEC     R5              ;COUNT
6571  042626  001375                            BNE     1$              ;BRANCH IF 256 NOT COMPLETE
6572  042630  005726                            TST     (SP)+           ;UNDO -(SP)
6573  042632  022020                            CMP     (R0)+,(R0)+     ;JUMP OVER THE TWO ECC WORDS
6574  042634  012705  000017                    MOV     #15.,   R5      ;1 DATA GAP
6575                                                                    ;14 TOLERANCE GAP
6576  042640  005020                    2$:     CLR     (R0)+           ;CLEAR DATA GAP, AND
6577  042642  005305                            DEC     R5              ;TOLERANCE GAP
6578  042644  001375                            BNE     2$              ;BRANCH IF NOT COMPLETE
6579
6580
6581  042646  004737  051260                    JSR     PC,@#FILLEC     ;INSERT THE TWO ECC WORDS ON THE DISK
6582                                                                    ;IN THE CORRECT PLACE
6583
6584                                      ;*THESE ARE FOR ECC TEST ONLY
6585
6586  042652  012737  177777  015142            MOV     #-1,@#TSECC     ;THIS IS AN ECC TEST
6587  042660  005037  050476                    CLR     @#POSITI        ;CLEAR ERROR POSITION COUNTER
6588  042664  013737  050472  050474            MOV     @#NCODE,@#NCOUNT  ;TEMPORARY N-CODE COUNTER
6589  042672  013737  050500  050506            MOV     @#HARDER,@#HADTMP  ;TEMPORARY HARD ERROR COUNTER
6590  042700  005037  050464                    CLR     @#GECC1         ;ECC LOW ORDER TO BE GENERATED
6591  042704  005037  050466                    CLR     @#GECC2         ;ECC HIGH ORDER TO BE GENERATED
6592  042710  005037  050502                    CLR     @#DATENV        ;CLEAR DATA ENVELOPE CLOCK COUNT
6593  042714  005037  050504                    CLR     @#ZCODE         ;CLEAR LEADING ZEROS CLOCK COUNT
6594
6595
6596                                      ;*THESE ARE TO SETUP FOR DISKLESS USE ONLY
6597
6598  042720  012737  010000  052654            MOV     #FMT22,@#CYL    ;16 BITS PER WORD
6599                                                                    ;CYLINDER 0, FORMAT 16 BITS
6600  042726  112737  000000  052657            MOVB    #0,@#SECOTR+1   ;TRACK 0
6601  042734  112737  000000  052656            MOVB    #0,@#SECOTR     ;SECTOR 0
6602  042742  012737  000000  052660            MOV     #0,@#KEY1       ;KEY1=0
6603  042750  012737  000000  052662            MOV     #0,@#KEY2       ;KEY2=0
6604  042756  012737  000400  052734            MOV     #256.,@#DAWORD  ;NO. OF DATA WORDS
6605  042764  005037  052664                    CLR     @#X             ;THIS IS A READ COMMAND
6606  042770  004537  047176                    JSR     R5,@#CRC        ;GO TO CALCULATE CRC
6607  042774  052654                            CYL
6608  042776  054554                            WERC
6609
6610
6611                                      ;*THIS IS TO INSERT ERROR
6612                                      ;*THE DISK DATA IS IN LOCATION STARTING FROM 'DISK'
```

```
6613                                        ;*THE POSITION OF THE ERROR CAN BE CHANGED BY CHANGING
6614                                        ;*THIS MOVE
6615
6616  043000  012737  152652  054636        MOV     #152652,@#DISK+44;INSERT ERROR IN POSITION 296 THRU 304
6617                                                                  ;IN WORD NUMBER 19
6618  043006  012737  052532  054640        MOV     #52532,@#DISK+46;INSERT ERROR IN POSITION 305 THRU 308
6619                                                                  ;IN WORD NUMBER 20
6620  043014  012737  010040  043170        MOV     #4128.,@#8$     ;INSERT POSITION REG.
6621
6622
6623                                        ;*THESE ARE REGULAR SETUPS
6624
6625  043022  004737  045770                JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
6626  043026  012777  177374  151714        MOV     #-256.-4.,@RHWC ;256. DATA 4 HEADER WORDS
6627  043034  012777  016264  151710        MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
6628  043042  112746  000000                MOVB    #0,-(SP)        ;IN LOWER BYTE GET SECTOR
6629  043046  112766  000000  000001        MOVB    #0,1(SP)        ;GET TRACK IN HIGHER BYTE
6630  043054  012677  151702                MOV     (SP)+,  @RHDST  ;TRACK/SECTOR IN RHDST
6631  043060  012777  010000  151700        MOV     #FMT22,@RHOF    ;16 BITS PER WORD
6632                                                                  ;ECC CORRECTION NOT INHIBIT
6633                                                                  ;BECAUSE ECC IS NOT GOING
6634                                                                  ;TO BE CHECKED
6635  043066  005077  151676                CLR     @RHCA           ;CYLINDER 0
6636  043072  004737  046024                JSR     PC,@#CHECKT     ;CHECK DVA,RDY,DPR,DRY = 1 AND OTHERS DON'T
6637  043076  104401  005123                TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF THE
6638  043102  000000                        HALT                    ;STOP THE TEST
6639  043104  013711  015176                MOV     @#REFOR,@R1     ;READ HEADER AND DATA=72
6640  043110  005037  015124                CLR     @#ERFLG$        ;CLEAR ERROR FLAG
6641  043114  004737  052514                JSR     PC,@#COMHD      ;READ HEADER AND DATA
6642                                                                  ;IF THERE ARE READ ERRORS THEN
6643                                                                  ;ECC WILL NOT BE CHECKED
6644
6645
6646                                        ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
6647                                        ;*FROM THE ''COMHD'' ROUTINE THAT MEANS SECTOR GAP,
6648                                        ;*FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
6649                                        ;*SYNC BYTE HAVE GONE BY AND SYNCS WERE CORRECTLY
6650                                        ;*DETECTED
6651                                        ;*HEADER AND DATA ARE TO BE CHECKED.
6652                                        ;*IN CHECKING READ DATA THE WRITE FROM BUFFER
6653                                        ;*''WRFROM'' IS FILLED WITH EXPECTED DATA AND
6654                                        ;*COMPARISONS ARE MADE
6655
6656  043120  005737  015124                TST     @#ERFLG$        ;ANY ERRORS ALREADY THERE
6657  043124  001111                        BNE     TST67           ;BRANCH IF YES
6658  043126  004737  045470                JSR     PC,@#PUTREG     ;SAVE REGISTERS
6659  043132  022737  100000  015034        CMP     #DCK,@#ER1      ;ONLY DATA CHECK ERROR SHOULD BE SET
6660  043140  001401                        BEQ     6$              ;BRANCH IF YES
6661  043142  104032                        ERROR   32              ;32 BIT ECC REGISTER SHOULD BE NON
6662                                                                  ;ZERO
6663                                                                  ;ONLY 11 OF THE 32 BITS CAN BE SEEN
6664                                                                  ;IN THE PATERN REGISTER
6665                                                                  ;DCK SHOULD BE SET IN RHER1
6666  043144  013746  050464        6$:     MOV     @#GECC1,-(SP)   ;GET PATTERN REGISTER
6667  043150  042716  174000                BIC     #174000,(SP)    ;KEEP ONLY 11 BITS
6668  043154  022637  015064                CMP     (SP)+,@#EC2     ;COMPARE PATTERN REGISTER
```

```
6669  043160  001401                    BEQ     7$              ;BRANCH IF GOOD
6670  043162  104032                    ERROR   32              ;11 BITS OF THE 32 BIT ECC REGISTER INCORRECT
6671
6672  043164  004037  051106      7$:   JSR     R0,@#ECORR      ;GO TO ECC CORRECTION PROCESS
6673  043170  000000                8$:   .WORD                  ;EXPECTED POSITION REG. WHEN CORRECTION
6674                                                            ;IS COMPLETE
6675
6676
6677
6678  043172  004737  045470              JSR     PC,@#PUTREG     ;SAVE REGISTERS
6679  043176  022737  100100  015034      CMP     #DCK!ECH,@#ER1  ;WITH ERRORS INSERTED IN BIT POSITION 21
6680                                                            ;THRU 32 HARD ERROR BIT SHOULD SET
6681  043204  001401                    BEQ     9$              ;BRANCH IF GOOD
6682  043206  104036                    ERROR   36              ;WITH ERROR INSERTED IN BIT POSITION 21 THRU
6683                                                            ;32 HCE SHOULD SET
6684
6685
6686
6687  043210                      9$:
6688  043210  004737  046214              JSR     PC,@#CHECKE     ;CHECK THAT DVA,RDY,DPR,DRY = 1
6689  043214  104401  005123              TYPE    ,CPHALT         ;CANNOT CONTINUE IF THEY DON'T
6690  043220  000000                      HALT                    ;STOP THE TEST AND RESTART PROGRAM
6691  043222  012700  015220              MOV     #WRFROM,R0      ;GETTING READY TO FILL EXPECTED DATA
6692  043226  012720  010000              MOV     #0!FMT22,(R0)+  ;CYLINDER 0
6693  043232  112746  000000              MOVB    #0,-(SP)        ;IN LOWER BYTE GET SECTOR
6694  043236  112766  000000  000001      MOVB    #0,1(SP)        ;GET TRACK IN HIGHER BYTE
6695  043244  012620                      MOV     (SP)+,(R0)+     ;GET TRACK/SECTOR IN BUFFER
6696  043246  012720  000000              MOV     #0,(R0)+        ;KEY1 IN BUFFER
6697  043252  012720  000000              MOV     #0,(R0)+        ;KEY2 IN BUFFER
6698  043256  012701  000400              MOV     #256.,R1        ;DATA WORD COUNTER
6699  043262  012702  052525              MOV     #52525,R2       ;DATA
6700  043266  010220              3$:     MOV     R2,(R0)+        ;DATA INTO BUFFER
6701  043270  005301                      DEC     R1              ;COUNT
6702  043272  001375                      BNE     3$              ;BRANCH IF 256 NOT DONE
6703
6704                                      ;*ONLY GOOD DATA HAS BEEN PUT IN 'WRFROM'
6705                                      ;*NOW THE INSERTED ERROR WILL BE PUT IN
6706
6707  043274  012737  152652  015274      MOV     #152652,@#WRFROM+54;INSERT ERROR IN POSITION 296 THRU 304
6708                                                            ;IN WORD NUMBER 19 IN DATA
6709  043302  012737  052532  015276      MOV     #52532,@#WRFROM+56;INSERT ERROR IN POSITION 305 THRU 308
6710                                                            ;IN WORD NUMBER 20 IN DATA
6711
6712  043310  004737  045470              JSR     PC,@#PUTREG     ;SAVE REGISTERS
6713  043314  005037  015124              CLR     @#ERFLG$        ;CLEAR ERROR FLAG
6714
6715
6716                                      ;*NOW READ DATA BUFFER WILL BE CHECKED
6717
6718  043320  004037  046664              JSR     R0,@#COMPAR     ;CHECK
6719  043324  015220                      WRFROM                  ;GOOD BUFFER
6720  043326  016264                      REINTO                  ;TEST BUFFER
6721  043330  000404                      4+256.                  ;NUMBER OF WORDS CHECKED
6722  043332  043340                      4$                      ;RETURN POINT FOR ERROR HEADER
6723  043334  043344                      5$                      ;RETURN POINT FOR ERROR DATA
6724  043336  043350                      TST67                   ;RETURN FOR GOOD COMPARISON
```

```
 6725  043340  104004                  4$:     ERROR   4               ;READ NEXT ERROR
 6726  043342  000207                          RTS     PC              ;RETURN TO ''COMPAR''
 6727  043344  104005                  5$:     ERROR   5               ;WORD NOS 1 TO 4 ARE
 6728                                                                  ;HEADER WORDS
 6729                                                                  ;5 TO 260 ARE DATA WORDS
 6730  043346  000207                          RTS     PC              ;RETURN TO ''COMPAR''
 6731
```

```
6732
6733  043350  000004                    TST67:  SCOPE
6734  043352  012706  001000                    MOV     #STACK,SP       ;RESET STACK
6735
6736  043356  012737  000067  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
6737
6738
6739                                            ;*SETUP FOR WHAT IS TO BE READ
6740                                            ;*HEADER CRC IS RESTORED FROM A SUBROUTINE
6741
6742  043364  012746  052525                    MOV     #52525,-(SP)    ;DATA TO BE READ
6743  043370  012705  000400                    MOV     #256.,R5        ;COUNTER
6744  043374  012700  054572                    MOV     #DISK,R0        ;START OF SIMULATED DISK DATA
6745  043400  011620                    1$:     MOV     (SP),(R0)+      ;MOVE IN DATA ON TO SIMULATED DISK
6746  043402  005305                            DEC     R5              ;COUNT
6747  043404  001375                            BNE     1$              ;BRANCH IF 256 NOT COMPLETE
6748  043406  005726                            TST     (SP)+           ;UNDO -(SP)
6749  043410  022020                            CMP     (R0)+,(R0)+     ;JUMP OVER THE TWO ECC WORDS
6750  043412  012705  000017                    MOV     #15.,R5         ;1 DATA GAP
6751                                                                    ;14 TOLERANCE GAP
6752  043416  005020                    2$:     CLR     (R0)+           ;CLEAR DATA GAP, AND
6753  043420  005305                            DEC     R5              ;TOLERANCE GAP
6754  043422  001375                            BNE     2$              ;BRANCH IF NOT COMPLETE
6755
6756
6757  043424  004737  051260                    JSR     PC,@#FILLEC     ;INSERT THE TWO ECC WORDS ON THE DISK
6758                                                                    ;IN THE CORRECT PLACE
6759
6760                                            ;*THESE ARE FOR ECC TEST ONLY
6761
6762  043430  012737  177777  015142            MOV     #-1,@#TSECC     ;THIS IS AN ECC TEST
6763  043436  005037  050476                    CLR     @#POSITI        ;CLEAR ERROR POSITION COUNTER
6764  043442  013737  050472  050474            MOV     @#NCODE,@#NCOUNT  ;TEMPORARY N-CODE COUNTER
6765  043450  013737  050500  050506            MOV     @#HARDER,@#HADTMP  ;TEMPORARY HARD ERROR COUNTER
6766  043456  005037  050464                    CLR     @#GECC1         ;ECC LOW ORDER TO BE GENERATED
6767  043462  005037  050466                    CLR     @#GECC2         ;ECC HIGH ORDER TO BE GENERATED
6768  043466  005037  050502                    CLR     @#DATENV        ;CLEAR DATA ENVELOPE CLOCK COUNT
6769  043472  005037  050504                    CLR     @#ZCODE         ;CLEAR LEADING ZEROS CLOCK COUNT
6770
6771
6772                                            ;*THESE ARE TO SETUP FOR DISKLESS USE ONLY
6773
6774  043476  012737  010000  052654            MOV     #FMT22,@#CYL    ;16 BITS PER WORD
6775                                                                    ;CYLINDER 0, FORMAT 16 BITS
6776  043504  112737  000000  052657            MOVB    #0,@#SECOTR+1   ;TRACK 0
6777  043512  112737  000000  052656            MOVB    #0,@#SECOTR     ;SECTOR 0
6778  043520  012737  000000  052660            MOV     #0,@#KEY1       ;KEY1=0
6779  043526  012737  000000  052662            MOV     #0,@#KEY2       ;KEY2=0
6780  043534  012737  000400  052734            MOV     #256.,@#DAWORD  ;NO. OF DATA WORDS
6781  043542  005037  052664                    CLR     @#X             ;THIS IS A READ COMMAND
6782  043546  004537  047176                    JSR     R5,@#CRC        ;GO TO CALCULATE CRC
6783  043552  052654                            CYL
6784  043554  054554                            WCRC
6785
6786
6787                                            ;*THIS IS TO INSERT ERROR
```

```
6788                                    ;*THE DISK DATA IS IN LOCATION STARTING FROM 'DISK'
6789                                    ;*THE POSITION OF THE ERROR CAN BE CHANGED BY CHANGING
6790                                    ;*THIS MOVE
6791
6792  043556  012737  152525  054574    MOV     #152525,@#DISK+2  ;FORCE ERROR ON BIT NUMBER 32
6793  043564  012737  152525  055570    MOV     #152525,@#DISK+<255.*2>;FORCE ERROR IN BIT 4096
6794  043572  012737  010040  043746    MOV     #4128.,@#8$       ;INSERT POSITION REG.
6795
6796
6797                                    ;*THESE ARE REGULAR SETUPS
6798
6799  043600  004737  045770            JSR     PC,@#CLDISK       ;SETUP GENERAL REGISTERS
6800  043604  012777  177374  151136    MOV     #-256.-4.,@RHWC   ;256. DATA 4 HEADER WORDS
6801  043612  012777  016264  151132    MOV     #REINTO,@RHBA     ;STARTING ADDRESS OF READ BUFFER
6802  043620  112746  000000            MOVB    #0,-(SP)          ;IN LOWER BYTE GET SECTOR
6803  043624  112766  000000  000001    MOVB    #0,1(SP)          ;GET TRACK IN HIGHER BYTE
6804  043632  012677  151124            MOV     (SP)+,  @RHDST    ;TRACK/SECTOR IN RHDST
6805  043636  012777  010000  151122    MOV     #FMT22,@RHOF      ;16 BITS PER WORD
6806                                                              ;ECC CORRECTION NOT INHIBIT
6807                                                              ;BECAUSE ECC IS NOT GOING
6808                                                              ;TO BE CHECKED
6809  043644  005077  151120            CLR     @RHCA             ;CYLINDER 0
6810
6811  043650  004737  046024            JSR     PC,@#CHECKT       ;CHECK DVA,RDY,DPR,DRY = 1 AND OTHERS DON'T
6812  043654  104401  005123            TYPE    ,CPHALT           ;CANNOT CONTINUE TESTING IF ANY OF THE
6813  043660  000000                    HALT                      ;STOP THE TEST
6814
6815  043662  013711  015176            MOV     @#REFOR,@R1       ;READ HEADER AND DATA=72
6816  043666  005037  015124            CLR     @#ERFLG$          ;CLEAR ERROR FLAG
6817  043672  004737  052514            JSR     PC,@#COMHD        ;READ HEADER AND DATA
6818                                                              ;IF THERE ARE READ ERRORS THEN
6819                                                              ;ECC WILL NOT BE CHECKED
6820
6821
6822                                    ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
6823                                    ;*FROM THE ''COMHD'' ROUTINE THAT MEANS SECTOR GAP,
6824                                    ;*FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
6825                                    ;*SYNC BYTE HAVE GONE BY AND SYNCS WERE CORRECTLY
6826                                    ;*DETECTED
6827                                    ;*HEADER AND DATA ARE TO BE CHECKED.
6828                                    ;*IN CHECKING READ DATA THE WRITE FROM BUFFER
6829                                    ;*''WRFROM'' IS FILLED WITH EXPECTED DATA AND
6830                                    ;*COMPARISONS ARE MADE
6831
6832  043676  005737  015124            TST     @#ERFLG$          ;ANY ERRORS ALREADY THERE
6833  043702  001111                    BNE     TST70             ;       BRANCH IF YES
6834  043704  004737  045470            JSR     PC,@#PUTREG       ;SAVE REGISTERS
6835  043710  022737  100000  015034    CMP     #DCK,@#ER1        ;ONLY DATA CHECK ERROR SHOULD BE SET
6836  043716  001401                    BEQ     6$                ;BRANCH IF YES
6837  043720  104032                    ERROR   32                ;32 BIT ECC REGISTER SHOULD BE NON
6838                                                              ;ZERO
6839                                                              ;ONLY 11 OF THE 32 BITS CAN BE SEEN
6840                                                              ;IN THE PATERN REGISTER
6841                                                              ;DCK SHOULD BE SET IN RHER1
6842  043722  013746  050464    6$:     MOV     @#GECC1,-(SP)     ;GET PATTERN REGISTER
6843  043726  042716  174000            BIC     #174000,(SP)      ;KEEP ONLY 11 BITS
```

```
6844  043732  022637  015064          CMP     (SP)+,@#EC2     ;COMPARE PATTERN REGISTER
6845  043736  001401                  BEQ     7$              ;BRANCH IF GOOD
6846  043740  104032                  ERROR   32              ;11 BITS OF THE 32 BIT ECC REGISTER INCORRECT
6847
6848  043742  004037  051106  7$:     JSR     R0,@#ECORR      ;GO TO ECC CORRECTION PROCESS
6849  043746  000000          8$:     .WORD                   ;EXPECTED POSITION REG. WHEN CORRECTION
6850                                                          ;IS COMPLETE
6851
6852
6853
6854  043750  004737  045470          JSR     PC,@#PUTREG     ;SAVE REGISTERS
6855  043754  022737  100100  015034  CMP     #DCK!ECH,@#ER1  ;WITH ERRORS INSERTED IN BIT POSITION 32
6856                                                          ;AND 4096 HARD ERROR BIT SHOULD SET
6857  043762  001401                  BEQ     9$              ;BRANCH IF GOOD
6858  043764  104036                  ERROR   36              ;WITH ERROR INSERTED IN BIT POSITION 21 THRU
6859                                                          ;32 HCE SHOULD SET
6860
6861
6862
6863
6864  043766          9$:
6865  043766  004737  046214          JSR     PC,@#CHECKE     ;CHECK THAT DVA,RDY,DPR,DRY = 1
6866  043772  104401  005123          TYPE    ,CPHALT         ;CANNOT CONTINUE IF THEY DON'T
6867  043776  000000                  HALT                    ;STOP THE TEST AND RESTART PROGRAM
6868  044000  012700  015220          MOV     #WRFROM,R0      ;GETTING READY TO FILL EXPECTED DATA
6869  044004  012720  010000          MOV     #0!FMT22,(R0)+  ;CYLINDER 0
6870  044010  112746  000000          MOVB    #0,-(SP)        ;IN LOWER BYTE GET SECTOR
6871  044014  112766  000000  000001  MOVB    #0,1(SP)        ;GET TRACK IN HIGHER BYTE
6872  044022  012620                  MOV     (SP)+,(R0)+     ;GET TRACK/SECTOR IN BUFFER
6873  044024  012720  000000          MOV     #0,(R0)+        ;KEY1 IN BUFFER
6874  044030  012720  000000          MOV     #0,(R0)+        ;KEY2 IN BUFFER
6875  044034  012701  000400          MOV     #256.,R1        ;DATA WORD COUNTER
6876  044040  012702  052525          MOV     #52525,R2       ;DATA
6877  044044  010220          3$:     MOV     R2,(R0)+        ;DATA INTO BUFFER
6878  044046  005301                  DEC     R1              ;COUNT
6879  044050  001375                  BNE     3$              ;BRANCH IF 256 NOT DONE
6880
6881                                  ;*ONLY GOOD DATA HAS BEEN PUT IN 'WRFROM'
6882                                  ;*NOW THE INSERTED ERROR WILL BE PUT IN
6883
6884  044052  012737  152525  015232  MOV     #152525,@#WRFROM+<5*2>  ;INSERTED ERROR IN BIT 32
6885  044060  012737  152525  016226  MOV     #152525,@#WRFROM+<259.*2>  ;INSERT ERROR IN BIT 4096
6886
6887  044066  005037  015124          CLR     @#ERFLG$        ;CLEAR ERROR FLAG.
6888  044072  004737  045470          JSR     PC,@#PUTREG     ;SAVE REGISTERS
6889
6890                                  ;*NOW READ DATA BUFFER WILL BE CHECKED
6891
6892  044076  004037  046664          JSR     R0,@#COMPAR     ;CHECK
6893  044102  015220                  WRFROM                  ;GOOD BUFFER (CHANGED)
6894  044104  016264                  REINTO                  ;TEST BUFFER
6895  044106  000404                  4+256.                  ;NUMBER OF WORDS CHECKED
6896  044110  044116                  4$                      ;RETURN POINT FOR ERROR HEADER
6897  044112  044122                  5$                      ;RETURN POINT FOR ERROR DATA
6898  044114  044126                  TST70                   ;RETURN FOR GOOD COMPARISON
6899  044116  104004          4$:     ERROR   4               ;READ NEXT ERROR
```

```
 6900  044120  000207                    RTS     PC              ;RETURN TO ''COMPAR''
 6901  044122  104005              5$:   ERROR   5               ;WORD NOS 1 TO 4 ARE
 6902                                                            ;HEADER WORDS
 6903                                                            ;5 TO 260 ARE DATA WORDS
 6904  044124  000207                    RTS     PC              ;RETURN TO ''COMPAR''
```

```
6905                                            .SBTTL  CURSORY INTERRUPT LOGIC TESTS
6906
6907
6908
6909  044126  000004                    TST70:  SCOPE
6910
6911  044130  012737  000070  017330            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
6912  044136  012706  001000                    MOV     #STACK,SP       ;RESET STACK
6913
6914  044142  004737  045770                    JSR     PC,@#CLDISK     ;CLEAR DISK
6915  044146  013700  014744                    MOV     @#RPVEC,R0      ;GET VECTOR ADDRESS
6916  044152  012720  044220                    MOV     #RPTRP1,(R0)+   ;SET INTERRUPT VECTOR
6917  044156  012710  000340                    MOV     #340,(R0)       ;SET SERVICE ROUTINE PRIORITY
6918  044162  012737  000200  177776            MOV     #200,PS         ;SET PROCESSOR PRIORITY
6919  044170  012711  000300                    MOV     #RDY!IE,@R1     ;RDY, IE IN RHSC1 SHOULD CAUSE INTERRUPT
6920  044174  013737  046340  001200            MOV     @#TIMCNT,@#$TMP1;COUNTER
6921  044202  005337  001200            1$:     DEC     @#$TMP1         ;WAIT FOR INTERRUPT
6922  044206  001375                    BNE     1$                      ;BRANCH IF NOT ZERO
6923                                                                    ;BEFORE THIS IS ZERO INTERRUPT SHOULD
6924                                                                    ;OCCUR
6925  044210  004737  045470                    JSR     PC,@#PUTREG     ;SAVE REGISTERS
6926  044214  104021                    ERROR   21                      ;INTERRUPT DID NOT OCCUR
6927
6928  044216  000410                    BR      TST71           ;       BRANCH TO NEXT TEST
6929
6930  044220  022626                    RPTRP1: CMP     (SP)+,(SP)+     ;RESTORE STACK
6931  044222  004737  045470                    JSR     PC,@#PUTREG     ;SAVE REGISTERS
6932  044226  022737  004200  015032            CMP     #DVA!RDY,@#CS1  ;'IE' SHOULD BE LOW
6933  044234  001401                    BEQ     TST71           ;       BRANCH IF GOOD
6934  044236  104021                    ERROR   21                      ;INTERRUPT OCCURRED BUT
6935                                                                    ;'IE' FAILED TO RESET
```

```
6936
6937
6938  044240  000004                  TST71:  SCOPE
6939
6940  044242  012737  000071  017330          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
6941  044250  012706  001000                  MOV     #STACK,SP       ;RESET STACK
6942
6943  044254  004737  045770                  JSR     PC,@#CLDISK     ;CLEAR DISK
6944  044260  013700  014744                  MOV     @#RPVEC,R0      ;GET VECTOR ADDRESS
6945  044264  012720  044324                  MOV     #RPTRP2,(R0)+   ;SET INTERRUPT VECTOR
6946  044270  012710  000340                  MOV     #340,(R0)       ;SET SERVICE ROUTINE PRIORITY
6947  044274  012737  000240  177776          MOV     #240,PS         ;SET PROCESSOR PRIORITY
6948  044302  012711  000300                  MOV     #RDY!IE,@R1     ;RDY, IE IN RHSC1 SHOULD CAUSE INTERRUPT
6949  044306  013737  046340  001200          MOV     @#TIMCNT,@#$TMP1;COUNTER
6950  044314  005337  001200          1$:     DEC     @#$TMP1         ;WAIT FOR INTERRUPT
6951  044320  001375                  BNE     1$                      ;BRANCH IF NOT ZERO
6952                                                                  ;BEFORE THIS IS ZERO INTERRUPT SHOULD
6953                                                                  ;OCCUR
6954  044322  000404                  BR      TST72                   ;       NO INTERRUPT SO BRANCH
6955
6956  044324  022626          RPTRP2: CMP     (SP)+,(SP)+             ;RESTORE STACK
6957  044326  004737  045470          JSR     PC,@#PUTREG             ;SAVE REGISTERS
6958  044332  104021                  ERROR   21                      ;INTERRUPT OCCURRED WITH
6959                                                                  ;PROCESSOR STATUS SAME
6960                                                                  ;AS DISK
6961
6962
6963
6964
```

```
6965  044334  000004                    TST72:  SCOPE
6966  044336  012737  000001  001212            MOV     #1,$TIMES       ;;DO 1 ITERATION
6967  044344  004737  045770                    JSR     PC,@#CLDISK
6968
6969  044350  012737  000000  177776            MOV     #0,PS           ;REINSTATE PS TO 0
6970  044356  104401  044364                    TYPE    ,65$            ;;TYPE ASCIZ STRING
6971  044362  000425                            BR      64$             ;;GET OVER THE ASCIZ
6972  044436  013746  015112                    MOV     @#UNIT,-(SP)    ;GET READY TO TYPE UNIT NUMBER
6973  044442  104405                            TYPDS
6974  044444  104401  044452                    TYPE    ,67$            ;;TYPE ASCIZ STRING
6975  044450  000402                            BR      66$             ;;GET OVER THE ASCIZ
6976  044456  013746  001112                    MOV     @#$ERTTL,-(SP)  ;GET READY TO TYPE NUMBER OF ERRORS
6977  044462  104405                            TYPDS
6978  044464  005037  001112                    CLR     @#$ERTTL        ;CLEAR TOTAL NUMBER OF ERRORS
6979  044470  005037  001102                    CLR     @#$TSTNM        ;CLEAR TEST NUMBER
6980  044474  005737  015120                    TST     @#SELECT        ;STARTING FROM 200 ?
6981  044500  001413                            BEQ     3$              ;TEST NEXT DRIVE IF SO
6982
6983  044502  005237  001100                    INC     @#$PASS         ;INCREASE PASS COUNT
6984  044506  104401  044671                    TYPE    ,$ENDMG         ;TYPE END PASS #
6985  044512  013746  001100                    MOV     @#$PASS,-(SP)
6986  044516  104405                            TYPDS
6987  044520  104401  044666                    TYPE    ,$ENULL
6988  044524  000137  022642                    JMP     @#TST5          ;CONTINUE TESTING THIS DRIVE ---------->
6989
6990  044530  005337  015114            3$:     DEC     @#NOUNITS       ;NO. OF UNITS PRESENT DECREMENTED
6991  044534  001413                            BEQ     $EOP            ;BRANCH IF ALL DRIVES COMPLETE
6992  044536  013700  015112                    MOV     @#UNIT,R0       ;UNIT UNDER TEST
6993  044542  012701  015072                    MOV     #UNITS,R1       ;TABLE
6994  044546  022100                    1$:     CMP     (R1)+,R0        ;IS THIS UNIT JUST TESTED
6995  044550  001401                            BEQ     2$              ;BRANCH IF YES
6996  044552  000775                            BR      1$              ;BRANCH IF NO
6997  044554  011137  015112            2$:     MOV     (R1),@#UNIT     ;THIS IS NEXT UNIT
6998  044560  000137  022642                    JMP     @#TST5          ;TEST NEXT DRIVE ---------->
```

```
6999
7000                                            .SBTTL
7001                                            .SBTTL  ***SUBROUTINES***
7002                                            .SBTTL
7003
7004
7005
7006
7007  044564  000004                            SCOPE
7008  044566  005037  001102                    CLR     $TSTNM          ;;ZERO THE TEST NUMBER
7009  044572  005037  001212                    CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
7010  044576  005237  001100                    INC     $PASS           ;;INCREMENT THE PASS NUMBER
7011  044602  042737  100000  001100            BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
7012  044610  005327                            DEC     (PC)+           ;;LOOP?
7013  044612  000001                  $EOPCT:   .WORD   1
7014  044614  003022                            BGT     $DOAGN          ;;YES
7015  044616  012737                            MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
7016  044620  000001                  $ENDCT:   .WORD   1
7017  044622  044612                            $EOPCT
7018  044624  104401  044671                    TYPE    ,$ENDMG         ;;TYPE 'END PASS #''
7019  044630  013746  001100                    MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
7020  044634  104405                            TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
7021  044636  104401  044666                    TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
7022  044642  013700  000042          $GET42:   MOV     @#42,R0         ;;GET MONITOR ADDRESS
7023  044646  001405                            BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
7024  044650  000005                            RESET                   ;;CLEAR THE WORLD
7025  044652  004710                  $ENDAD:   JSR     PC,(R0)         ;;GO TO MONITOR
7026  044654  000240                            NOP                     ;;SAVE ROOM
7027  044656  000240                            NOP                     ;;FOR
7028  044660  000240                            NOP                     ;;ACT11
7029  044662  000137                            JMP     @(PC)+          ;;RETURN
7030  044664  021034                  $RTNAD:   .WORD   TST1
7031  044666     377        377        000     $ENULL:   .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
7032  044671     015  042412  042116            $ENDMG:   .ASCIZ  <15><12>/END PASS #/
7033  044676  050040  051501  020123
7034  044704  000043
7035
7036
7037
7038
7039
7040                                    ;*HERE IS A DETAILED EXPLAINATION OF HOW THE LOOP ON ERROR WORKS.
7041                                    ;*ON HITTING AN ERROR IF THE LOOP ON ERROR SWITCH IS SET, THE
7042                                    ;*PROGRAM GOES BACK - USUALLY BACK TO THE BEGINNING OF THE TEST.
7043
7044                                    ;*WHEN THIS OPERATOR SELECTABLE SCOPE LOOP IS USED THEN THE POINT
7045                                    ;*THE PROGRAM GOES BACK TO CAN BE CHANGED.
7046                                    ;*THE RESTRICTIONS TO THE POINT WHERE THE PROGRAM CAN GO ARE: -
7047                                    ;*1.  IT MUST BE WITHIN THE TEST UNDER CONSIDERATION
7048                                    ;*2.  LOOP ON ERROR SWITCH MUST BE SET
7049                                    ;*3.  THE ERROR MUST OCCUR WITHIN THE TEST UNDER CONSIDERATION
7050                                    ;*IF THE ERROR DOES NOT OCCUR WITHIN THE TEST UNDER CONSIDERATION
7051                                    ;*THE PROGRAM WILL REVERT TO NORMAL OPERATION.  HOWEVER, IF LOOP ON
7052                                    ;*TEST SWITCH IS SET AND THIS OPERATOR SELECTABLE SCOPE LOOP IS USED
7053                                    ;*THEN THE PROGRAM WILL LOOP BACK TO THE SELECTED POINT WHEN IT
7054                                    ;*COMES TO THE END OF THE TEST UNDER CONSIDERATION.
```

```
7055                                    ;*
7056                                    ;*AFTER LOOPING FOR SOME TIME IF THE LOOP SWITCH IS PUT DOWN THEN
7057                                    ;*NORMAL OPERATION WILL CONTINUE.
7058
7059
7060  044706  000000                    TESTAD: 0                       ;FIRST ADDRESS OF TEST
7061
7062  044710                            OPERSEL:
7063  044710  005037  177776                    CLR     PS              ;MAKE PROCESSOR STATUS ZERO
7064  044714  104401  044722                    TYPE    ,65$            ;;TYPE ASCIZ STRING
7065  044720  000421                            BR      64$             ;;GET OVER THE ASCIZ
7066  044764  013746  017330                    MOV     @#TSTNM,-(SP)   ;GET READY TO TYPE TEST
7067  044770  104402                            TYPOC                   ;NUMBER
7068  044772  104401  045000                    TYPE    ,67$            ;;TYPE ASCIZ STRING
7069  044776  000414                            BR      66$             ;;GET OVER THE ASCIZ
7070  045030  013746  001110                    MOV     @#$LPERR,-(SP)  ;GET READY TO TYPE LOOP BACK PC
7071  045034  104402                            TYPOC
7072  045036  104401  001223                    TYPE    ,$CRLF
7073  045042  104401  045050                    TYPE    ,69$            ;;TYPE ASCIZ STRING
7074  045046  000426                            BR      68$             ;;GET OVER THE ASCIZ
7075  045124  104401  045132                    TYPE    ,71$            ;;TYPE ASCIZ STRING
7076  045130  000420                            BR      70$             ;;GET OVER THE ASCIZ
7077  045172  104401  045200                    TYPE    ,73$            ;;TYPE ASCIZ STRING
7078  045176  000423                            BR      72$             ;;GET OVER THE ASCIZ
7079  045246  104412                            RDOCT
7080  045250  062716  000002                    ADD     #2,(SP)         ;GET LPADR
7081  045254  012637  001106                    MOV     (SP)+,@#$LPADR
7082  045260  104401  045266                    TYPE    ,75$            ;;TYPE ASCIZ STRING
7083  045264  000417                            BR      74$             ;;GET OVER THE ASCIZ
7084  045324  104401  045332                    TYPE    ,77$            ;;TYPE ASCIZ STRING
7085  045330  000441                            BR      76$             ;;GET OVER THE ASCIZ
7086  045434  104412                            RDOCT
7087  045436  012637  001110                    MOV     (SP)+,@#$LPERR  ;GET LPERR
7088  045442  013746  001106                    MOV     @#$LPADR,-(SP)
7089                                    ;THIS CLEARS UP GARBAGE
7090  045446  005037  052770                    CLR     @#NOSYNC                ;CLEAR FLAG FOR HEADER ERROR COMMANDS
7091  045452  005037  015142                    CLR     @#TSECC                 ;CLEAR FLAG FOR ECC TEST
7092                                                                            ;WHEN =177777 IT IS AN ECC TEST
7093                                                                            ;WHEN =0 IT IS NOT AN ECC TEST
7094
7095  045456  005037  050470                    CLR     @#TSECCG                ;EVEN IN AN ECC TEST EVERY CLOCK
7096                                                                            ;IS NOT TO GENERATE ECC
7097                                                                            ;IF =177777 GENERATE ECC
7098                                                                            ;IF =0 DO NOT GENERATE ECC
7099  045462  005037  015144                    CLR     @#TESDTE                ;DRIVE TIMING ERROR TEST
7100  045466  000002                            RTI
```

```
7101
7102                                    .SBTTL  SAVE REGISTERS ROUTINE
7103
7104                                    ;THIS SAVES THE CONTENTS OF ALL HARDWARE REGISTERS
7105                                    ;IN MEMORY LOCATIONS TAGED FROM "WC" TO "EC2"
7106                                    ;
7107                                    ;
7108                                    ;THIS IS DONE SO THAT COMPARES ARE DONE WITH SAVED LOCATIONS
7109                                    ;AND NOT THE REGISTERS THEMSELVES.   THIS WILL MAKE
7110                                    ;ERROR PRINTOUTS FOR GOOD AND BAD DATA ALWAYS DIFFRENT
7111
7112  045470                    PUTREG:
7113  045470  010046                   MOV     R0,-(SP)        ;;PUSH R0 ON STACK
7114  045472  010146                   MOV     R1,-(SP)        ;;PUSH R1 ON STACK
7115  045474  010246                   MOV     R2,-(SP)        ;;PUSH R2 ON STACK
7116  045476  012700  014750           MOV     #RHWC,R0        ;STARTING ADDRESS OF REG
7117  045502  012701  015024           MOV     #WC,R1          ;STARTING ADDRESS OF SAVE LOCATIONS
7118  045506  012702  000023           MOV     #RHCC-RHWC+2/2,R2 ;NUMBER OF REG. INTO R2
7119  045512  013021            10$:   MOV     @(R0)+,(R1)+    ;SAVE HARDWARE REG.
7120  045514  005302                   DEC     R2
7121  045516  001375                   BNE     10$
7122  045520  012602                   MOV     (SP)+,R2        ;;POP STACK INTO R2
7123  045522  012601                   MOV     (SP)+,R1        ;;POP STACK INTO R1
7124  045524  012600                   MOV     (SP)+,R0        ;;POP STACK INTO R0
7125  045526  000207                   RTS     PC
```

```
7126
7127                                    .SBTTL  FLOAT 1 AND 0
7128
7129                                    ;*FLOAT A ONE AND A ZERO THRU A DESIGNATED REGISTER
7130                                    ;*ABSOLUTE ADDRESS OF REG. UNDER TEST IS IN R4
7131
7132  045530  000000                    MASK:   0                       ;BITS UNDER TEST
7133  045532  000000                    LERR:   0                       ;ERROR HLT ADDRESS
7134  045534  000000                    REGADR: 0
7135
7136  045536  012537  045530            BITST:  MOV     (R5)+,  MASK    ;FETCH DATA MASK
7137  045542  012504                            MOV     (R5)+,  R4      ;GET ADDRESS OF REG. UNDER TEST
7138  045544  010437  045534                    MOV     R4,     REGADR
7139  045550  010537  045532                    MOV     R5,     LERR    ;GET ERROR RETURN ADDR.
7140  045554  062705  000004                    ADD     #4,     R5      ;MODIFY RETURN ADDR. TO JUMP OVER RTS
7141  045560  012703  000001                    MOV     #1,     R3      ;INITIALIZE DATA PATTERN
7142  045564  004737  045606            BLT1:   JSR     PC,     BLT2    ;OUTPUT FLOATING ZERO
7143  045570  004737  045606                    JSR     PC,     BLT2    ;OUTPUT FLOATING ONE
7144  045574  000241                            CLC
7145  045576  006103                            ROL     R3              ;SHIFT PATTERN
7146  045600  005703                            TST     R3
7147  045602  001370                            BNE     BLT1            ;BRANCH IF NOT COMPLETE
7148  045604  000205                            RTS     R5              ;RETURN TO TEST
7149  045606  005103                    BLT2:   COM     R3              ;COMPLEMENT PATTERN
7150  045610  012737  045616  045750            MOV     #BLT3,  @#LAD   ;SET SCOPE LOOP
7151  045616  010337  001124            BLT3:   MOV     R3,@#$GDDAT     ;STORE GOOD DATA
7152  045622  005137  045530                    COM     @#MASK          ;AND MASK WITH PATTERN
7153  045626  043737  045530  001124            BIC     @#MASK, @#$GDDAT;CLEAR THE REST
7154  045634  005137  045530                    COM     @#MASK          ;RESTORE MASK
7155  045640  013714  001124                    MOV     @#$GDDAT,(R4)   ;OUTPUT TO REGISTER
7156  045644  011437  001126                    MOV     (R4),@#$BDDAT   ;INPUT FROM REGISTER
7157  045650  005137  045530                    COM     @#MASK
7158  045654  043737  045530  001126            BIC     @#MASK,@#$BDDAT ;AND MASK OUT RECEIVED DATA
7159  045662  005137  045530                    COM     @#MASK          ;RESTORE MASK
7160  045666  023737  001124  001126            CMP     @#$GDDAT,@#$BDDAT;IS DATA CORRECT
7161  045674  001403                            BEQ     1$              ;BRANCH IF GOOD
7162  045676  004777  177630                    JSR     PC,     @LERR   ;GO TO REPORT ERROR
7163  045702  104413                            SCOP1                   ;LOCAL SCOPE LOOP
7164  045704  000207                    1$:     RTS     PC
```

```
7165
7166                                    .SBTTL  CLEAR MEMORY ROUTINE
7167
7168
7169                                ;*      THIS CLEARS ANY BLOCK OF MEMORY
7170                                ;*      FILLING IT WITH ANY DATA
7171                                ;*
7172                                ;*      CALL
7173                                ;*      JSR     R0,CLAREA
7174                                ;*      X                       ;STARTING ADDRESS OF BLOCK
7175                                ;*      Y
7176                                ;*      Z                       ;DATA TO BE FILLED
7177                                ;*R1 WILL HAVE STARTING ADDRESS OF BLOCK TO BE FILLED
7178                                ;*R2 AFTER SUBTRACTION WILL HAVE TWICE NUMBER OF LOCATIONS
7179                                ;*R3 WILL HAVE DATA TO BE FILLED
7180                                ;*TO AVOID DIVIDE ROUTINE TWO DECREMENT R2 WILL BE USED
7181
7182  045706                        CLAREA:
7183  045706  010146                       MOV     R1,-(SP)        ;;PUSH R1 ON STACK
7184  045710  010246                       MOV     R2,-(SP)        ;;PUSH R2 ON STACK
7185  045712  010346                       MOV     R3,-(SP)        ;;PUSH R3 ON STACK
7186  045714  012001                       MOV     (R0)+,R1        ;FROM
7187  045716  012002                       MOV     (R0)+,R2        ;TO
7188  045720  012003                       MOV     (R0)+,R3        ;DATA
7189  045722  160102                       SUB     R1,R2           ;NO. OF LOCATIONS MINUS TWO
7190  045724  062702  000002               ADD     #2,R2           ;GET TWICE NO OF LOCATIONS
7191  045730  010321                1$:    MOV     R3,(R1)+        ;MOVE IN DATA
7192  045732  005302                       DEC     R2
7193  045734  005302                       DEC     R2
7194  045736  001374                       BNE     1$              ;BRANCH IF NOT COMPLETE
7195  045740  012603                       MOV     (SP)+,R3        ;;POP STACK INTO R3
7196  045742  012602                       MOV     (SP)+,R2        ;;POP STACK INTO R2
7197  045744  012601                       MOV     (SP)+,R1        ;;POP STACK INTO R1
7198  045746  000200                       RTS     R0              ;RETURN
```

```
7199                                            .SBTTL  LOCAL TRAPS
7200  045750  000000                    LAD:    0
7201
7202  045752  032777  001000  133160    T.SCOP: BIT     #SW09,  @SWR
7203  045760  001402                            BEQ     1$
7204  045762  013716  045750                    MOV     @#LAD,  (SP)
7205  045766  000002                    1$:     RTI
7206
7207                                    ;*EXAMPLE OF THE USE OF THE ABOVE
7208                                    ;*THIS WILL LOOP BETWEEN X: AND SCOP1 PROVIDED THERE IS NO 'NEWTST''
7209                                    ;*MOV   #X,     @#LAD
7210                                    ;*X:    ---     ---
7211                                    ;*      ---     ---
7212                                    ;*      ---     ---
7213                                    ;*      SCOP1
7214
7215
7216                                            .SBTTL  CLEAR DISK ROUTINE
7217
7218  045770  013701  014756            CLDISK: MOV     @#RHCS1,R1      ;R1 WILL BE CONTROL AND STATUS1
7219  045774  013702  014754                    MOV     @#RHCS2,R2      ;R2 WILL BE CONTROL AND STATUS2
7220  046000  013703  015000                    MOV     @#RHDS1,R3      ;R3 WILL BE DISK STATUS REGISTER1
7221  046004  013704  014760                    MOV     @#RHER1,R4      ;R4 WILL BE ERROR REGISTER #1
7222
7223  046010  012712  000040                    MOV     #CLR,@R2        ;CLEAR ALL REG.
7224  046014  013712  015112                    MOV     @#UNIT,@R2      ;REINSTATE UNIT NO.
7225  046020  005011                            CLR     @R1             ;CLEAR FUNCTION BITS
7226  046022  000207                            RTS     PC
7227
```

```
7228
7229                                    .SBTTL  CHECK DISK STATUS ROUTINES
7230
7231                                    ;*THIS CHECKS THAT DEVICE AVAILABLE (DVA) AND READY (RDY) IN RHCS1 = 1
7232                                    ;*AND CHECKS THAT DEVICE PRESENT (DPR), DEVICE READY (DRY) IN RHDS1 = 1
7233                                    ;*IT ALSO CHECKS THAT ALL OTHER BITS IN THESE REGISTERS = 0
7234
7235
7236  046024  011637  015132          CHECKT: MOV     (SP),@#PCJSR    ;SAVE PC OF JSR+4
7237  046030  162737  000004  015132          SUB     #4,@#PCJSR      ;GET PC OF JSR
7238  046036  004737  045470                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
7239  046042  022737  004200  015032          CMP     #DVA!RDY,@#CS1  ;RHCS1 SHOULD HAVE DEVICE AVAILABLE
7240                                                                  ;AND BE READY
7241  046050  001423                          BEQ     3$              ;BRANCH IF GOOD TO RHDS1 CHECK
7242
7243  046052  032737  004000  015032          BIT     #DVA,@#CS1      ;BAD SO TEST DEVICE AVAILABLE
7244  046060  001004                          BNE     1$              ;TEST READY IF DVA THERE
7245  046062  010137  001122                  MOV     R1,@#$BDADR     ;ADDRESS OF BAD REGISTER (RHCS1)
7246  046066  104026                          ERROR   26              ;RHCS1 DID NOT HAVE DEVICE
7247                                                                  ;AVAILABLE AT START OF TEST
7248  046070  000413                          BR      3$              ;BRANCH TO RHDS1 CHECK
7249
7250  046072  032737  000200  015032  1$:     BIT     #RDY,@#CS1      ;TEST READY
7251  046100  001003                          BNE     2$              ;IF RDY THERE BRANCH
7252  046102  010137  001122                  MOV     R1,@#$BDADR     ;ADDRESS OF BAD REGISTER (RHCS1)
7253  046106  104026                          ERROR   26              ;RHCS1 DID NOT HAVE READY
7254                                                                  ;AT THE START OF TEST
7255  046110  000403                  2$:     BR      3$              ;BRANCH TO NEXT COMPARE
7256  046112  010137  001122                  MOV     R1,@#$BDADR     ;ADDRESS OF BAD REGISTER (RHCS1)
7257  046116  104026                          ERROR   26              ;RHCS1 HAD SOME BITS OTHER
7258                                                                  ;THAN DVA AND RDY SET
7259                                                                  ;ALL OTHER BITS SHOULD BE 0
7260                                                                  ;AT START OF TEST
7261
7262  046120  013746  015054          3$:     MOV     @#DS1,-(SP)     ;GET RHDS1
7263  046124  042716  001100                  BIC     #VV!PROG,(SP)   ;CLEAR VV AND PROGRAMABLE BIT
7264  046130  022726  000600                  CMP     #DPR!DRY,(SP)+  ;RHDS1 SHOULD HAVE THESE SET
7265  046134  001424                          BEQ     8$              ;RETURN TO TEST IF GOOD
7266
7267  046136  032737  000400  015054  4$:     BIT     #DPR,@#DS1      ;BAD SO TEST DRIVE PRESENT
7268  046144  001004                          BNE     5$              ;CHECK DRY IF GOOD
7269  046146  010337  001122                  MOV     R3,@#$BDADR     ;ADDRESS OF BAD REGISTER (RHDS1)
7270  046152  104026                          ERROR   26              ;RHDS1 DOES NOT HAVE DPR
7271  046154  000413                          BR      7$              ;BRANCH OUT
7272  046156  032737  000200  015054  5$:     BIT     #DRY,@#DS1      ;TEST DRIVE READY
7273  046164  001004                          BNE     6$              ;IF DPR WAS THERE SO BRANCH
7274  046166  010337  001122                  MOV     R3,@#$BDADR     ;ADDRESS OF BAD REGISTER (RHDS1)
7275  046172  104026                          ERROR   26              ;RHDS1 DOES NOT HAVE DRY
7276  046174  000403                          BR      7$              ;BRANCH OUT
7277  046176  010337  001122          6$:     MOV     R3,@#$BDADR     ;ADDRESS OF BAD REGISTER (RHDS1)
7278  046202  104026                          ERROR   26              ;RHDS1 HAS SOME BITS OTHER
7279                                                                  ;THAN MOL, DRY, DPR, SET
7280                                                                  ;ALL OTHER BITS SHOULD BE 0
7281  046204  000207                  7$:     RTS     PC              ;RETURN TO TEST AND HALT - FATAL ERROR
7282
7283  046206  062716  000006          8$:     ADD     #6,(SP)         ;ADJUST STACK PTR TO GET OVER HALT IN TEST
```

```
7284  046212  000207                          RTS     PC              ;RETURN TO TEST AND CONTINUE TESTING
7285
7286
7287                                  ;*THIS CHECKS THAT DEVICE AVAILABLE (DVA) AND READY (RDY) IN RHCS1 = 1
7288                                  ;*AND CHECKS THAT DEVICE PRESENT (DPR), DEVICE READY (DRY) IN RHDS1 = 1
7289                                  ;*IT DOES NOT CHECK THAT OTHER BITS IN THESE REGISTERS = 0
7290
7291  046214  011637  015132          CHECKE: MOV     (SP),@#PCJSR    ;SAVE PC OF JSR+4
7292  046220  162737  000004  015132          SUB     #4,@#PCJSR      ;GET PC OF JSR
7293  046226  004737  045470                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
7294  046232  032737  000200  015032          BIT     #RDY,@#CS1      ;RHCS1 SHOULD HAVE DEVICE AVAILABLE
7295                                                                  ;AND BE READY
7296  046240  001004                          BNE     1$              ;BRANCH IF GOOD
7297  046242  010137  001122                  MOV     R1,@#$BDADR     ;FAILING REGISTER
7298  046246  104026                          ERROR   26              ;RHCS1 IS IN ERROR
7299                                                                  ;DOES NOT HAVE DVA, RDY
7300  046250  000427                          BR      4$              ;BRANCH
7301
7302  046252  032737  004000  015032  1$:     BIT     #DVA,@#CS1      ;RHCS1 SHOULD HAVE DEVICE AVAILABLE
7303                                                                  ;AND BE READY
7304  046260  001004                          BNE     2$              ;BRANCH IF GOOD
7305  046262  010137  001122                  MOV     R1,@#$BDADR     ;FAILING REGISTER
7306  046266  104026                          ERROR   26              ;RHCS1 IS IN ERROR
7307                                                                  ;DOES NOT HAVE DVA, RDY
7308  046270  000417                          BR      4$              ;BRANCH OUT
7309  046272  032737  000200  015054  2$:     BIT     #DRY,@#DS1      ;RHDS1 SHOULD HAVE DPR,DRY
7310  046300  001004                          BNE     3$              ;BRANCH IF THERE
7311  046302  010337  001122                  MOV     R3,@#$BDADR     ;FAILING REGISTER RHDS1
7312  046306  104026                          ERROR   26              ;RHDS1 DOES NOT HAVE DPR,DRY
7313  046310  000407                          BR      4$              ;BRANCH OUT
7314  046312  032737  000400  015054  3$:     BIT     #DPR,@#DS1      ;RHDS1 SHOULD HAVE DPR,DRY
7315  046320  001004                          BNE     5$              ;BRANCH IF THERE
7316  046322  010337  001122                  MOV     R3,@#$BDADR     ;FAILING REGISTER RHDS1
7317  046326  104026                          ERROR   26              ;RHDS1 DOES NOT HAVE DPR,DRY
7318  046330  000207                  4$:     RTS     PC              ;RETURN TO TEST AND HALT - FATAL ERROR
7319
7320  046332  062716  000006          5$:     ADD     #6,(SP)         ;ADJUST STACK TO GET OVER HALT IN TEST
7321  046336  000207                          RTS     PC              ;RETURN TO TEST AND CONTINUE TESTING
7322
7323                                          .SBTTL  WAIT LOOP
7324                                  ;*      WAIT LOOP
7325                                  ;*      ONE  LOOP OR ONE COUNT = 5.15 MICROSEC WITH BIPOLAR MEMORY (MIN)
7326                                  ;*      ONE LOOP OR ONE COUNT = 11.86 MICROSEC WITH CORE (MIN)
7327                                  ;*      WITH CORE ERROR IS INDICATED AFTER ABOUT 650 MILLISEC (MIN)
7328
7329  046340  177777                  TIMCNT: 177777                  ;WAITING COUNT
7330
7331  046342  010046                  WAIT.T: MOV     R0,-(SP)        ;SAVE R0
7332  046344  016600  000002                  MOV     2(SP),R0        ;GET ADRESS OF REG. ADDRESS
7333  046350  010037  001204                  MOV     R0,@#$TMP3      ;WAT PC+2 IN $TMP3
7334  046354  162737  000002  001204          SUB     #2,@#$TMP3      ;WAT PC FOR TYPEOUT
7335  046362  012037  001176                  MOV     (R0)+,@#$TMP0   ;WAIT REGISTER ADDRESS
7336  046366  012037  001200                  MOV     (R0)+,@#$TMP1   ;WAIT ON BIT
7337  046372  010066  000002                  MOV     R0,2(SP)        ;RESTORE RETURN ON STACK
7338  046376  012600                          MOV     (SP)+,R0        ;RESTORE R0
7339  046400  013737  046340  001202          MOV     @#TIMCNT,@#$TMP2;TEMPORARY COUNT
```

```
7340
7341  046406  033777  001200  132562  1$:     BIT     @#$TMP1,@$TMP0  ;IS REQUIRED BIT THERE?
7342  046414  001021                          BNE     2$              ;BRANCH IF YES
7343  046416  005337  001202                  DEC     @#$TMP2         ;COUNT
7344  046422  001371                          BNE     1$              ;BRANCH IF NOT TIME UP
7345  046424  013737  046340  001202          MOV     @#TIMCNT,@#$TMP2;TEMPORARY COUNT
7346  046432  033777  001200  132536  3$:     BIT     @#$TMP1,@$TMP0  ;IS REQUIRED BIT THERE?
7347  046440  001007                          BNE     2$              ;BRANCH IF YES
7348  046442  005337  001202                  DEC     @#$TMP2         ;COUNT
7349  046446  001371                          BNE     3$              ;BRANCH IF NOT TIME UP
7350  046450  017737  132522  001126          MOV     @$TMP0,@#$BDDAT ;REGISTER CONTENTS
7351  046456  104016                          ERROR   16              ;WAITED ON BIT FAILED TO SET
7352  046460  000002                  2$:     RTI
7353
7354
7355                                  ;*      CALL FOR THE ABOVE WAITLOOP IS
7356                                  ;*
7357                                  ;*      MOV     @A,@#X$         ;A CONTAINS REGISTER ADDRESS
7358                                  ;*      -       -       -       ;HENCE X$ WILL HAVE ABSOLUTE REG. ADR.
7359                                  ;*      -       -       -
7360                                  ;*      -       -       -
7361                                  ;*      WAT
7362                                  ;*X$:   0                       ;ABSOLUTE REG. ADDRESS UNDER WAIT
7363                                  ;*      .WORD   0               ;BIT WAITED FOR
7364                                  ;*                              ;CONTINUE
```

```
7365
7366                                    .SBTTL  SAVE ROUTINE
7367
7368                                    ;THIS IS A SUBROUTINE TO READ & SAVE REGISTERS
7369                                    ;IN THE REGISTER TABLE TO ANY LOCATION
7370                                    ;THE CALL IS
7371                                    ;JSR    R0,@#SAVER
7372                                    ;FROM
7373                                    ;TO
7374                                    ;NUMBER OF WORDS SAVED
7375
7376  046462                    SAVER:
7377  046462  010146                   MOV     R1,-(SP)        ;;PUSH R1 ON STACK
7378  046464  010246                   MOV     R2,-(SP)        ;;PUSH R2 ON STACK
7379  046466  010346                   MOV     R3,-(SP)        ;;PUSH R3 ON STACK
7380  046470  012001                   MOV     (R0)+,R1        ;FROM
7381  046472  012002                   MOV     (R0)+,R2        ;TO
7382  046474  012003                   MOV     (R0)+,R3        ;NUMBER
7383  046476  013122            1$:    MOV     @(R1)+,(R2)+    ;SAVE REGISTER CONTENTS
7384  046500  005303                   DEC     R3              ;COUNT
7385  046502  001375                   BNE     1$              ;BRANCH IF NOT DONE
7386  046504  012603                   MOV     (SP)+,R3        ;;POP STACK INTO R3
7387  046506  012602                   MOV     (SP)+,R2        ;;POP STACK INTO R2
7388  046510  012601                   MOV     (SP)+,R1        ;;POP STACK INTO R1
7389  046512  000200                   RTS     R0
7390
7391
7392
7393
7394                                    .SBTTL  WRITE CHECK ROUTINE
7395
7396                                    ;THIS IS A SUBROUTINE TO DO WRITE CHECK HEADER AND DATA
7397                                    ;CYLINDER 0, TRACK 1, SECTOR 1, KEYS 0
7398
7399                                    ;THESE ARE TO SET UP FOR DISKLESS USE ONLY
7400
7401  046514  012737  010000  052654  WRCHHD: MOV     #FMT22,@#CYL    ;CYLINDER 0 FORMAT 16 BIT WORDS
7402  046522  112737  000001  052657          MOVB    #1,@#SECOTR+1   ;TRACK=1
7403  046530  112737  000001  052656          MOVB    #1,@#SECOTR     ;SECTOR=1
7404  046536  005037  052660                  CLR     @#KEY1          ;KEY1=0
7405  046542  005037  052662                  CLR     @#KEY2          ;KEY2=0
7406  046546  012737  000044  052734          MOV     #36.,DAWORD     ;NO OF DATA WORDS
7407  046554  005037  052664                  CLR     @#X             ;THIS IS A READ OPERATION
7408  046560  004537  047176                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
7409  046564  052654                          CYL
7410  046566  054554                          WCRC
7411
7412                                    ;THESE ARE REGULAR SETUPS
7413
7414  046570  004737  045770                  JSR     PC,@#CLDISK     ;SET UP GENERAL REGISTERS
7415                                                                  ;AND CLEAR DISK REGISTERS
7416  046574  012777  177730  146146          MOV     #-40.,@RHWC     ;36 DATA WORDS 4 HEADER WORDS
7417  046602  012777  016264  146142          MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
7418  046610  112746  000001                  MOVB    #1,-(SP)        ;SECTOR=1
7419  046614  112766  000001  000001          MOVB    #1,1(SP)        ;TRACK=1 IN UPPER BYTE
7420  046622  012677  146134                  MOV     (SP)+,@RHDST    ;TRACK=1, SECTOR=1 IN RHDST
```

```
 7421  046626  012777  014000  146132          MOV     #FMT22!ECI,@RHOF ;16 BIT WORDS
 7422                                                                    ;ECC CORRECTION INHIBIT BECAUSE
 7423                                                                    ;ECC LOGIC IS NOT CHECKED YET
 7424  046634  005077  146130                  CLR     @RHCA            ;CYLINDER=0
 7425  046640  004737  046024                  JSR     PC,@#CHECKT      ;CHECK DVA,RDY,DPR,DRY = 1 AND OTHERS DON'T
 7426  046644  104401  005123                  TYPE    ,CPHALT          ;CANNOT CONTINUE TESTING IF ANY OF THE
 7427  046650  000000                          HALT                     ;STOP THE TEST
 7428  046652  013711  015166                  MOV     @#WRCHDT,@R1     ;WRITE CHECK HEADER AND DATA=52
 7429                                                                    ;INTO RHCS1
 7430  046656  004737  052514                  JSR     PC,@#COMHD       ;WRITE CHECK HEADER AND DATA
 7431                                                                    ;SAME AS READ HEADER AND DATA
 7432
 7433  046662  000207                          RTS     PC               ;RETURN TO WRITE CHECK TEST
```

```
                                        .SBTTL  COMPARE ROUTINE

                                ;*THIS IS A SUBROUTINE TO COMPARE TWO BLOCKS IN MEMORY
                                ;*R1 HAS GOOD DATA BUFFER ADDRESS
                                ;*R2 HAS TEST DATA BUFFER ADDRESS
                                ;*$TMP0 HAS ADDRESS OF RETURN ON ERROR TO PRINT HEADER
                                ;*$TMP1 HAS ADDRESS OF RETURN ON ERROR TO PRINT DATA
                                ;*R3 HAS NUMBER OF WORDS TO BE COMPARED
                                ;*R4 HAS ONE MORE THAN NUMBER OF WORDS TO BE COMPARED

046664                          COMPAR:
046664  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
046666  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
046670  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
046672  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
046674  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
046676  012001                          MOV     (R0)+,R1        ;ADDRESS OF GOOD DATA BUFFER
046700  012002                          MOV     (R0)+,R2        ;ADDRESS OF TEST DATA BUFFER
046702  012003                          MOV     (R0)+,R3        ;NO OF WORDS TO BE COMPARED
046704  012037  001176                  MOV     (R0)+,$TMP0     ;RETURN ON ERROR TO PRINT HEADER
046710  012037  001200                  MOV     (R0)+,$TMP1     ;RETURN ON ERROR TO PRINT DATA
046714  011000                          MOV     (R0),R0         ;RETURN ON NO ERROR
046716  010304                          MOV     R3,R4           ;NO OF WORDS TO BE COMPARED
046720  005204                          INC     R4
046722  010437  052774          1$:     MOV     R4,@#ERWORD     ;FOR ERROR WORD NO
046726  022122                          CMP     (R1)+,(R2)+     ;COMPARE GOOD WITH TEST DATA
046730  001426                          BEQ     3$              ;BRANCH IF GOOD

046732  014137  001124                  MOV     -(R1),@#$GDDAT  ;GOOD DATA
046736  014237  001126                  MOV     -(R2),@#$BDDAT  ;BAD DATA
046742  160337  052774                  SUB     R3,@#ERWORD     ;ERROR WORD NO.
046746  005737  015124                  TST     @#ERFLG$        ;ANY ERRORS ALREAY THERE
046752  001003                          BNE     2$              ;BRANCH IF YES
046754  004777  132216                  JSR     PC,@$TMP0       ;RETURN TO PRINT HEADER
046760  000402                          BR      5$              ;BRANCH TO AVOID PRINTING NEXT ERROR
046762  004777  132212          2$:     JSR     PC,@$TMP1       ;RETURN TO PRINT DATA
046766  022122                  5$:     CMP     (R1)+,(R2)+     ;UNDO -(R1) AND -(R2) FOR ERRORS
046770  017746  132144                  MOV     @SWR,-(SP)      ;GET SWITCH SETTING
046774  042716  177177                  BIC     #^C600,(SP)     ;KEEP ONLY SWITCH 7 AND 8
047000  022726  000200                  CMP     #SW07,(SP)+     ;IS 7 SET AND 8 RESET
047004  001402                          BEQ     4$              ;BRANCH OUT IF YES
047006  005303                  3$:     DEC     R3              ;COUNT
047010  001344                          BNE     1$              ;BRANCH IF ALL NOT DEVICE
047012                          4$:
047012  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
047014  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
047016  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
047020  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
047022  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
047024  000200                          RTS     R0              ;RETURN TO MAIN PROGRAM
```

```
                                        .SBTTL  WRITE CHECK DATA

                                        ;THIS IS A SUBROUTINE TO DO WRITE CHECK DATA
                                        ;CYLINDER 0, TRACK 1, SECTOR 1, KEYS 0


                                        ;THESE ARE TO SET UP FOR DISKLESS USE ONLY

047026  012737  010000  052654  WRCHDA: MOV     #FMT22,@#CYL    ;CYLINDER 0 FORMAT 16 BIT WORDS
047034  112737  000001  052657          MOVB    #1,@#SECOTR+1   ;TRACK=1
047042  112737  000001  052656          MOVB    #1,@#SECOTR     ;SECTOR=1
047050  005037  052660                  CLR     @#KEY1          ;KEY1=0
047054  005037  052662                  CLR     @#KEY2          ;KEY2=0
047060  012737  000040  052734          MOV     #32.,@#DAWORD   ;NO OF DATA WORDS
047066  005037  052664                  CLR     @#X             ;THIS IS A READ OPERATION

047072  004537  047176                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
047076  052654                          CYL
047100  054554                          WCRC

                                        ;THESE ARE REGULAR SETUPS

047102  004737  045770                  JSR     PC,@#CLDISK     ;SET UP GENERAL REGISTERS
                                                                ;AND CLEAR DISK REGISTERS

047106  012777  177740  145634          MOV     #-32.,@RHWC     ;36 DATA WORDS 4 HEADER WORDS
047114  012777  016264  145630          MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
047122  112746  000001                  MOVB    #1,-(SP)        ;SECTOR=1
047126  112766  000001  000001          MOVB    #1,1(SP)        ;TRACK=1 IN UPPER BYTE
047134  012677  145622                  MOV     (SP)+,@RHDST    ;TRACK=1, SECTOR=1 IN RHDST
047140  012777  014000  145620          MOV     #FMT22!ECI,@RHOF        ;16 BIT WORDS
                                                                ;ECC CORRECTION INHIBIT BECAUSE
                                                                ;ECC LOGIC IS NOT CHECKED YET
047146  005077  145616                  CLR     @RHCA           ;CYLINDER=0
047152  004737  046024                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,DPR,DRY = 1 AND OTHERS DON'T
047156  104401  005123                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF THE
047162  000000                          HALT                    ;STOP THE TEST
047164  013711  015164                  MOV     @#WRCHEK,@R1    ;WRITE CHECK DATA=50 INTO RHCS1
047170  004737  052514                  JSR     PC,@#COMHD      ;WRITE CHECK HEADER AND DATA
                                                                ;SAME AS READ HEADER AND DATA

047174  000207                          RTS     PC              ;RETURN TO WRITE CHECK TEST
```

```
                                        .SBTTL  CRC GENERATION ROUTINE

                                        ;THIS IS A SUBROUTINE TO CALCULATE CRC FOR THE FOUR
                                        ;HEADER WORDS AND STORE THEM IN ''WCRC'' AND ''GCRC''
                                        ;R1 - REGISTER FOR CRC, INCREMENTED CRC VALUE IS HERE
                                        ;R2 - THIS HAS BIT POSITION 2 VALUE C
                                        ;R3 - THIS HAS BIT POSITION 16 I.E. OUTPUT BIT VALUE B
                                        ;R4 - THIS HAS BIT POSITION 15 VALUE E
                                        ;$TMP0 - NUMBER OF WORDS
                                        ;$TMP2 - NUMBER OF BITS PER WORD = 16
                                        ;$TMP3 - TEMPORARY REG.
                                        ;$TMP4 - TEMPORARY REG TO TRANSFER CARRY
                                        ;$TMP5 - THIS HAS DATA BIT VALUE D

                                        ;FETCH DATA BIT D
                                        ;B = D XOR 16
                                        ;C = B XOR 2
                                        ;E = B XOR 15
                                        ;ROTATE RIGHT ONE POSITION
                                        ;B GOES TO POSITION 1
                                        ;C GOES TO POSITION 3
                                        ;E GOES TO POSITION 16
                                        ;REPEAT 64 TIMES

                                        ;CALL   JSR     R5,@#CRC
                                        ;X      ;FIRST LOCATION AT
                                        ;Y      ;PUT CRC IN WCRC FOR READ GCRC FOR WRITE

047176                          CRC:
047176  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
047200  012500                          MOV     (R5)+,R0        ;GET POINTER TO CYL NO.
047202  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
047204  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
047206  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
047210  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
047212  005001                          CLR     R1              ;CLEAR WORKING LOCATION
047214  005037  001210                  CLR     @#$TMP5
047220  012737  000004  001176          MOV     #4,@#$TMP0      ;WORD COUNT
047226  012037  001204          16$:    MOV     (R0)+,@#$TMP3   ;TEMPORARY WORD STORAGE
047232  012737  000020  001202          MOV     #16.,$TMP2      ;BIT COUNT
047240  013737  001204  001206          MOV     @#$TMP3,@#$TMP4 ;TEMPORARY WORD STORAGE
047246  006037  001204          15$:    ROR     @#$TMP3         ;GET LSB INTO ''C''
047252  006037  001210                  ROR     @#$TMP5         ;GET ABOVE ''C'' INTO $TMP5
047256  032701  000001                  BIT     #BIT0,R1        ;IS POSITION 15 HIGH
047262  001403                          BEQ     1$              ;BRANCH IF POSITION 16 LOW
047264  012703  100000                  MOV     #BIT15,R3       ;GET POSITION 16
047270  000401                          BR      2$
047272  005003                  1$:     CLR     R3              ;GET POSITION 16
047274  063703  001210          2$:     ADD     @#$TMP5,R3      ;XOR POSITION 16 WITH D
                                                                ;TO GIVE B
047300  032701  040000                  BIT     #BIT14,R1       ;IS POSITION 2 HIGH
047304  001403                          BEQ     3$              ;BRANCH IF POSITION 2 LOW
047306  012702  100000                  MOV     #BIT15,R2       ;GET POSITION 2
047312  000401                          BR      4$
047314  005002                  3$:     CLR     R2              ;GET POSITION 2
047316  060302                  4$:     ADD     R3,R2           ;XOR B WITH POSITION 2
```

```
7584
7585  047320  032701  000002          BIT     #BIT1,R1        ;TO GIVE C
7586  047324  001403                  BEQ     5$              ;IS POSITION 15 HIGH
7587  047326  012704  100000          MOV     #BIT15,R4       ;BRANCH IF POSITION 15 LOW
7588  047332  000401                  BR      6$              ;GET POSITION 15
7589  047334  005004          5$:     CLR     R4              ;XOR POSITION 15 WITH B
7590  047336  060304          6$:     ADD     R3,R4           ;TO GIVE E
7591
7592  047340  006037  001206          ROR     @#$TMP4         ;GET LSB INTO "C"
7593  047344  006001                  ROR     R1              ;GET ABOVE C INTO R1
7594  047346  005703                  TST     R3              ;TEST B
7595  047350  100403                  BMI     7$              ;BRANCH IF B=1
7596  047352  042701  100000          BIC     #BIT15,R1       ;SET B IN POSITION 1
7597  047356  000402                  BR      10$
7598  047360  052701  100000  7$:     BIS     #BIT15,R1       ;SET B IN POSITION 1
7599  047364  005702          10$:    TST     R2              ;TEST C
7600  047366  100403                  BMI     11$             ;BRANCH IF C=1
7601  047370  042701  020000          BIC     #BIT13,R1       ;GET C IN POSITION 3
7602  047374  000402                  BR      12$
7603  047376  052701  020000  11$:    BIS     #BIT13,R1       ;GET C IN POSITION 3
7604  047402  005704          12$:    TST     R4              ;TEST E
7605  047404  100403                  BMI     13$             ;BRANCH IF E=1
7606  047406  042701  000001          BIC     #BIT0,R1        ;GET E IN POSITION 16
7607  047412  000402                  BR      14$
7608  047414  052701  000001  13$:    BIS     #BIT0,R1        ;GET E IN POSITION 16
7609  047420  005337  001202  14$:    DEC     @#$TMP2         ;BIT COUNTER
7610  047424  001310                  BNE     15$             ;BRANCH IF 16 NOT DONE
7611  047426  005337  001176          DEC     @#$TMP0         ;WORD COUNTER
7612  047432  001275                  BNE     16$             ;BRANCH IF 4 NOT DONE
7613  047434  010135                  MOV     R1,@(R5)+       ;PUT CRC WHERE DESIRED
7614  047436  012604                  MOV     (SP)+,R4        ;;POP STACK INTO R4
7615  047440  012603                  MOV     (SP)+,R3        ;;POP STACK INTO R3
7616  047442  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
7617  047444  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
7618  047446  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
7619  047450  000205                  RTS     R5
```

```
                                        .SBTTL  SIMULATED DISK SETUP

                                        ;THIS IS A SUBROUTINE TO SET UP THE SIMULATOR DISK FOR
                                        ;CYLINDER 0 (16 BITS PER WORD)
                                        ;TRACK 1, SECTOR 1
                                        ;KEY1 1
                                        ;KEY2 1
                                        ;CRC THROUGH THE JSR R5,@#CRC
                                        ;256 WORDS OF 177400

                                        ;CALL   JSR     PC,@#SETDSK

047452                          SETDSK:
047452  010046                         MOV     R0,-(SP)        ;;PUSH R0 ON STACK
047454  010146                         MOV     R1,-(SP)        ;;PUSH R1 ON STACK
047456  010246                         MOV     R2,-(SP)        ;;PUSH R2 ON STACK
047460  012700  177400                 MOV     #177400,R0      ;DATA IN THE DISK
047464  012701  000400                 MOV     #256.,R1        ;COUNTER
047470  012702  054572                 MOV     #DISK,R2        ;START OF SIMULATOR DISK
047474  010022                  1$:    MOV     R0,(R2)+        ;MOVE IN DATA
047476  005301                         DEC     R1              ;COUNT FOR 256
047500  001375                         BNE     1$              ;BRANCH IF 256 NOT COMPLETE
047502  012701  000021                 MOV     #17.,R1         ;2 ECC WORDS, 1 DATA GAP
                                                                ;14 TOLERANCE GAP
047506  005022                  2$:    CLR     (R2)+           ;CLEAR ECC,DATA GAP AND
                                                                ;TOLERANCE GAP
047510  005301                         DEC     R1              ;COUNT
047512  001375                         BNE     2$              ;BRANCH IF NOT COMPLETE

                                        ;NOW SET UP FOR DISKLESS USE

047514  012737  010000  052654         MOV     #FMT22,@#CYL    ;CYLINDER 0 (16 BIT WORDS)
047522  112737  000001  052657         MOVB    #1,@#SECOTR+1   ;TRACK=1
047530  112737  000001  052656         MOVB    #1,@#SECOTR     ;SECTOR=1
047536  012737  000001  052660         MOV     #1,@#KEY1       ;KEY1=1
047544  012737  000001  052662         MOV     #1,@#KEY2       ;KEY2=1
047552  013737  000400  052734         MOV     256.,@#DAWORD   ;NO. OF DATA WORDS
047560  004537  047176                 JSR     R5,@#CRC        ;GO TO CALCULATE CRC
047564  052654                         CYL                     ;FIRST CRC WORD
047566  054554                         WCRC                    ;PUT CALCULATED CRC
047570  012602                         MOV     (SP)+,R2        ;;POP STACK INTO R2
047572  012601                         MOV     (SP)+,R1        ;;POP STACK INTO R1
047574  012600                         MOV     (SP)+,R0        ;;POP STACK INTO R0
047576  000207                         RTS     PC
```

```
                                        .SBTTL  CHECK HCE ROUTINE

                                        ;THIS IS A SUBROUTINE TO CHECK HEADER COMPARE ERROR
                                        ;(BIT #7) AND CRC ERROR (BIT #8)

                                        ;CALL JSR R0,@#HCCRCE

                                        ;       COM             ;COMMAND-READ HEADER AND DATA
                                                                ;       -WRITE DATA
                                        ;       C               ;CYLINDER
                                        ;       S               ;SECTOR
                                        ;       T               ;TRACK
                                        ;       -N.             ;WORD COUNT
                                        ;       B               ;RHBA BUFFER START
                                        ;       X               ;1=WRITE DATA 0=READ
                                        ;       H               ;H=1 HEADER CHECK, H=0 CRC CHECK

047600  010037  015132          HCCRCE: MOV     R0,@#PCJSR      ;SAVE PC OF JSR+4
047604  162737  000004  015132        SUB     #4,@#PCJSR      ;GET PC OF JSR
047612  004737  045770                JSR     PC,@#CLDISK     ;INIT AND SETUP GENERAL REG.
047616  004737  046024                JSR     PC,@#CHECKT     ;CHECK DVA,RDY,DPR,DRY = 1 AND OTHERS DON'T
047622  104401  005123                TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF THE
047626  000000                        HALT                    ;STOP THE TEST
047630  011037  001210                MOV     (R0),@#$TMP5    ;SAVE COMMAND
047634  012011                        MOV     (R0)+,@R1       ;COMMAND
047636  012077  145126                MOV     (R0)+,@RHCA     ;CYLINDER
047642  112046                        MOVB    (R0)+,-(SP)     ;SECTOR
047644  105720                        TSTB    (R0)+           ;UP DATE R0
047646  112066  000001                MOVB    (R0)+,1(SP)     ;TRACK
047652  105720                        TSTB    (R0)+           ;UPDATE R0
047654  012677  145102                MOV     (SP)+,@RHDST    ;TRACK SECTOR
047660  012077  145064                MOV     (R0)+,@RHWC     ;NO. OF DATA WORDS +4 HEADER
                                                                ;IF A READ HEADER AND DATA
047664  012077  145062                MOV     (R0)+,@RHBA     ;STARTING ADDRESS OF BUFFER
047670  012037  052664                MOV     (R0)+,@#X       ;X=0 READ HEADER AND DATA
                                                                ;X=1 WRITE DATA
047674  012777  014000  145064        MOV     #FMT22!ECI,@RHOF ;16 BITS PER WORD
                                                                ;ECC CORRECTION INHIBIT
047702  005037  015124                CLR     @#ERFLG$        ;CLEAR ERROR FLAG
047706  004737  052514                JSR     PC,@#COMHD      ;COMMAND

                                        ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
                                        ;FROM THE ''COMHD'' ROUTINE THAT MEANS SECTOR GAP,
                                        ;FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
                                        ;SYNC BYTE HAVE GONE BY AND SYNCS WERE CORRECTLY
                                        ;DETECTED
                                        ;HEADER AND DATA ARE TO BE CHECKED.

047712  004737  045470                JSR     PC,@#PUTREG     ;SAVE REGISTERS
047716  005737  015124                TST     @#ERFLG$        ;ANY ERRORS ALREADY THERE
047722  001034                        BNE     10$             ;BRANCH IF YES
047724  005737  052664                TST     @#X             ;IS THIS A READ
047730  001015                        BNE     3$              ;IF A WRITE DATA BRANCH

                                        ;NOW THE READ BUFFER WILL BE CHECKED
                                        ;HEADER SHOULD BE COMPLETELY READ AS WRITTEN
```

```
                                        ;NO DATA WORDS SHOULD BE READ
                                        ;REINTO BUFFER HAS BEEN FILLED WITH 0
                                        ;WRFROM BUFFER HAS BEEN FILLED WITH EXPECTED DATA

047732  004037  046664               JSR     R0,@#COMPAR     ;CHECK
047736  015220                       WRFROM                  ;GOOD DATA
047740  016264                       REINTO                  ;TEST BUFFER
047742  000400                       256.                    ;4 HEADER 252 DATA
047744  047752                       1$                      ;RETURN POINT FOR ERROR HEADER
047746  047756                       2$                      ;RETURN POINT FOR ERROR DATA
047750  050014                       10$                     ;RETURN FOR GOOD COMPARISON
047752  104004               1$:     ERROR   4               ;READ NEXT ERROR 5
047754  000207                       RTS     PC              ;RETURN TO COMPARISON SUBROUTINE
047756  104005               2$:     ERROR   5               ;WORD NO 1 THRU 4 ARE
                                                             ;HEADER WORDS AND HENCE
                                                             ;SHOULD BE READ AS WRITTEN ON
                                                             ;DISK, WORD NOS. 5 ONWARDS
                                                             ;SHOULD NOT BE READ AND HENCE
                                                             ;READ INTO BUFFER
                                                             ;SHOULD BE UNCHANGED
047760  000207                       RTS     PC              ;RETURN TO COMPARISON

047762  000414                       BR      10$             ;JUMP OUT

                                        ;NOW THE DISK WILL BE CHECKED
                                        ;NO DATA SHOULD BE WRITTEN
                                        ;REINTO BUFFER HAS BEEN FILLED WITH EXPECTED DATA
                                        ;DISK HAS BEEN FILLED WITH 177400
                                        ;WRFROM HAS BEEN FILLED WITH 125252

047764  004037  046664       3$:     JSR     R0,@#COMPAR     ;CHECK
047770  016264                       REINTO                  ;GOOD DATA BUFFER
047772  054572                       DISK                    ;TEST BUFFER
047774  000400                       256.
047776  050004                       4$                      ;RETURN POINT FOR ERROR HEADER
050000  050010                       5$                      ;RETURN POINT FOR ERROR DATA
050002  050014                       10$                     ;RETURN POINT FOR GOOD COMPARISON
050004  104004               4$:     ERROR   4               ;READ NEXT ERROR 5
050006  000207                       RTS     PC              ;RETURN TO COMPARISON SUBROUTINE
050010  104005               5$:     ERROR   5               ;WORD NO ARE ALL DATA
                                                             ;WORDS THE SHOULD NOT
                                                             ;HAVE BEEN CHANGED BY THE
                                                             ;WRITE COMMAND
050012  000207                       RTS     PC              ;RETURN TO COMPARISON SUBROUTINE
050014  005720               10$:    TST     (R0)+           ;IS THIS A HCRC ON HCE CHECK?
050016  001442                       BEQ     6$              ;BRANCH IF HCRC
050020  022737  000072  001210       CMP     #72,@#$TMP5     ;IS THIS A READ COMMAND
050026  001417                       BEQ     11$             ;BRANCH IF YES
050030  017737  144724  001126       MOV     @RHER1,@#$BDDAT ;TEST DATA
050036  022737  000200  001126       CMP     #HCE,@#$BDDAT   ;ONLY HEADER COMPARE BIT?
                                                             ;SHOULD BE SET
050044  001470                       BEQ     7$              ;BRANCH IF GOOD
050046  013737  014760  045534       MOV     @#RHER1,@#REGADR ;REGISTER ADDRESS RHER1
050054  012737  000200  001124       MOV     #HCE,@#$GDDAT   ;GOOD DATA
050062  104027                       ERROR   27              ;AFTER AN ERROR ON THE
                                                             ;HEADER ONLY HCE SHOULD
```

```
7777  050064  000460                          BR      7$              ;BE SET
7778  050066                          11$:
7779  050066  017737  144666  001126          MOV     @RHER1,@#$BDDAT ;TEST DATA
7780  050074  022737  100200  001126          CMP     #DCK!HCE,@#$BDDAT       ;ONLY HEADER COMPARE BIT?
7781                                                                  ;SHOULD BE SET
7782                                                                  ;DCK IS SET BECAUSE ECC IS NOT READ
7783  050102  001451                          BEQ     7$              ;BRANCH IF GOOD
7784  050104  013737  014760  045534          MOV     @#RHER1,@#REGADR ;REGISTER ADDRESS RHER1
7785  050112  012737  100200  001124          MOV     #DCK!HCE,@#$GDDAT       ;GOOD DATA
7786  050120  104027                          ERROR   27              ;AFTER AN ERROR ON THE
7787                                                                  ;HEADER ONLY HCE SHOULD
7788  050122  000441                          BR      7$              ;BE SET
7789  050124  022737  000072  001210  6$:     CMP     #72,@#$TMP5     ;IS THIS A READ COMMAND?
7790  050132  001417                          BEQ     12$             ;BRANCH IF A READ
7791  050134  017737  144620  001126          MOV     @RHER1,@#$BDDAT ;TEST DATA
7792  050142  022737  000400  001126          CMP     #HCRC,@#$BDDAT  ;ONLY CRC ERROR SHOULD BE THERE
7793  050150  001426                          BEQ     7$
7794  050152  013737  014760  045534          MOV     @#RHER1,@#REGADR ;REG. ADDR = RHER1
7795  050160  012737  000400  001124          MOV     #HCRC,@#$GDDAT  ;GOOD DATA
7796  050166  104027                          ERROR   27              ;AFTER A CRC ERROR ONLY CRC
7797                                                                  ;SHOULD BE SET
7798  050170  000416                          BR      7$              ;BRANCH OUT
7799  050172  017737  144562  001126  12$:    MOV     @RHER1,@#$BDDAT ;TEST DATA
7800
7801  050200  022737  100400  001126          CMP     #DCK!HCRC,@#$BDDAT;HCRC AND DCK SHOULD BE SET
7802                                                                  ;DCK IS SET BECAUSE ECC IS NOT READ
7803  050206  001407                          BEQ     7$              ;BRANCH IF GOOD
7804  050210  012737  100400  001124          MOV     #DCK!HCRC,@#$GDDAT;GOOD DATA
7805  050216  013737  014760  045534          MOV     @#RHER1,@#REGADR;FAILING REGISTER RHER1
7806  050224  104027                          ERROR   27              ;AFTER A CRC ERROR ON A READ
7807                                                                  ;DCK AND HCRC SHOULD BE SET
7808                                                                  ;DCK IS SET BECAUSE ECC IS NOT READ
7809  050226  000200                  7$:     RTS     R0              ;RETURN TO MAIN TEST
```

```
                                        .SBTTL  EXIT WRT HEADER & DATA ROUTINE

                                        ;THIS IS A SUBROUTINE TO LEAVE AT THE MIDDLE OF
                                        ;A WRITE HEADER AND DATA COMMAND
                                        ;IT TRYS TO GET SECTOR 10, TRACK 0, CYLINDER 0
                                        ;BUT COMES OUT AFTER ONE SECTOR
                                        ;THE COMMAND OS JSR PC,@#MIDDLE
                                        ;BAI IS SET

050230                          MIDDLE:
050230  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
050232  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
050234  013777  015172  144514          MOV     @#WRIFOR,@RHCS1 ;WRITE HEADER AND DATA=62
                                                                ;IN RHCS1
050242  012777  177766  144500          MOV     #-10.,@RHWC     ;10 WORDS
050250  012777  015220  144474          MOV     #WRFROM,@RHBA   ;BUS ADDRESS=WRFROM
050256  012777  000010  144476          MOV     #10,@RHDST      ;DESIRED TRACK=0 SECTOR=10
050264  052777  000010  144462          BIS     #BAI,@RHCS2     ;BUS ADDRESS INCREMENT INHIBIT
050272  012777  010000  144466          MOV     #FMT22,@RHOF    ;FORMAT 16 BIT WORDS
050300  005077  144464                  CLR     @RHCA           ;CYLINDER=0
050304  012737  000001  050332          MOV     #1,@#MID        ;SECTOR IS SET TO 1 SO THAT
                                                                ;WE CAN GET OUT AT THE
                                                                ;MIDDLE OF AN OPERATION
                                                                ;LOOKING FOR SECTOR 10
050312  012777  000001  144456          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
050320  052777  000001  144430          BIS     #GO,@RHCS1              ;GO TO RHCS1 WITH 62
050326  004137  056722                  JSR     R1,@#SEARCH
050332  000000                  MID:    .WORD   0               ;SECTOR
050334  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
050336  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
050340  000207                          RTS     PC
```

```
7841
7842                                            .SBTTL  JAM CURRENT CYLINDER ROUTINE
7843
7844                                    ;*THIS SUBROUTINE WILL CHANGE THE CURRENT CYLINDER REGISTER
7845                                    ;*THIS IS DONE BY GIVING A SEEK COMMAND THEN AN INIT
7846                                    ;*WHICH WILL LOAD THE CURRENT CYLINDER WITH THE DESIRED CYLINDER VALUE
7847                                    ;*
7848                                    ;*CALL IS:
7849                                    ;*      JSR     R0,@#MAKECYL
7850                                    ;*      XC                      ;DESIRED VALUE OF CURRENT CYLINDER
7851
7852  050342                            MAKECYL:
7853  050342  010546                           MOV     R5,-(SP)        ;;PUSH R5 ON STACK
7854  050344  010037  015132                   MOV     R0,@#PCJSR      ;PC OF JSR+4
7855  050350  162737  000004  015132           SUB     #4,@#PCJSR      ;SAVE PC OF JSR
7856  050356  012005                           MOV     (R0)+,R5        ;GETTING READY TO FILL DESIRED CYLINDER
7857  050360  010577  144404                   MOV     R5,@RHCA        ;FILL DESIRED CYLINDER REGISTER
7858  050364  005077  144372                   CLR     @RHDST          ;MAKE SURE DESIRED SECTOR TRACK IS NOT ILLEGAL
7859  050370  013777  015200  144360           MOV     @#SEECOM,@RHCS1 ;FILL SEEK COMMAND
7860  050376  012777  000001  144372           MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
7861  050404  052777  000001  144344           BIS     #GO,@RHCS1      ;GO TO SEEK
7862  050412  000240                           NOP                     ;ALLOW TIME FOR SEEK TO HANG UP
7863  050414  000240                           NOP                     ;ALLOW TIME FOR SEEK TO HANG UP
7864  050416  000240                           NOP                     ;ALLOW TIME FOR SEEK TO HANG UP
7865  050420  000240                           NOP                     ;ALLOW TIME FOR SEEK TO HANG UP
7866  050422  004737  045770                   JSR     PC,@#CLDISK     ;GIVE INIT
7867  050426  017737  144362  001126           MOV     @RHCC,@#$BDDAT  ;TEST DATA
7868  050434  020537  001126                   CMP     R5,@#$BDDAT     ;COMPARE CURRENT CYLINDER
7869  050440  001406                           BEQ     1$              ;BRANCH IF GOOD
7870  050442  010537  001124                   MOV     R5,@#$GDDAT     ;GOOD VALUE OF RHCC
7871  050446  013737  015014  045534           MOV     @#RHCC,@#REGADR ;FAILING REGISTER ADDRESS
7872  050454  104030                           ERROR   30              ;CURRENT CYLINDER DOES NOT MATCH DESIRED CYLINDER
7873                                                                   ;REGISTER AFTER A SEEK AND AN INIT
7874  050456                            1$:
7875  050456  012605                           MOV     (SP)+,R5        ;;POP STACK INTO R5
7876  050460  000200                           RTS     R0
7877
```

```
7878
7879                                    .SBTTL  ECC GENERATION AND COMPARISON ROUTINE
7880
7881                                    ;*THIS SUBROUTINE GENERATES AND TESTS ECC
7882                                    ;*CALL    JSR PC,ECTEST
7883
7884
7885
7886
7887            100000                  PIE1    =100000
7888            040000                  PIE2    =40000
7889            020000                  PIE3    =20000
7890            010000                  PIE4    =10000
7891            004000                  PIE5    =4000
7892            002000                  PIE6    =2000
7893            001000                  PIE7    =1000
7894            000400                  PIE8    =400
7895            000200                  PIE9    =200
7896            000100                  PIE10   =100
7897            000040                  PIE11   =40
7898            000020                  PIE12   =20
7899            000010                  PIE13   =10
7900            000004                  PIE14   =4
7901            000002                  PIE15   =2
7902            000001                  PIE16   =1
7903            100000                  PIE17   =100000
7904            040000                  PIE18   =40000
7905            020000                  PIE19   =20000
7906            010000                  PIE20   =10000
7907            004000                  PIE21   =4000
7908            002000                  PIE22   =2000
7909            001000                  PIE23   =1000
7910            000400                  PIE24   =400
7911            000200                  PIE25   =200
7912            000100                  PIE26   =100
7913            000040                  PIE27   =40
7914            000020                  PIE28   =20
7915            000010                  PIE29   =10
7916            000004                  PIE30   =4
7917            000002                  PIE31   =2
7918            000001                  PIE32   =1
7919
7920    050462  000000                  ECDATA: 0               ;DATA BIT FOR ECC
7921                                                            ;IF ALL ONES THEN CURRENT BIT IS A ONE
7922                                                            ;IF ZERO THEN CURRENT BIT IS A ZERO
7923
7924    050464  000000                  GECC1:  0               ;LOW ORDER ECC WORD TO BE GENERATED HERE
7925                                                            ;=R1
7926
7927    050466  000000                  GECC2:  0               ;HIGH ORDER ECC WORD TO BE GENERATED HERE
7928                                                            ;=R2
7929
7930    050470  000000                  TSECCG: 0               ;IF =177777 GENERATE AND TEST ECC FOR THIS BIT
7931                                                            ;IF =0 DO NOT GENERATE AND TEST ECC FOR THIS BIT
7932
7933    050472  113713                  NCODE:  38859.          ;N-CODE WORD
```

```
 7934  050474  000000                  NCOUNT: 0               ;TEMPORARY N CODE
 7935  050476  000000                  POSITI: 0               ;POSITION REGISTER
 7936  050500  010041                  HARDER: 4129.           ;HARD ERROR COUNT
 7937                                                          ;TRUE COUNT IS 4128 BUT AS COMPARES ARE
 7938                                                          ;DONE ONE STAGE LATER SO 4129
 7939  050502  000000                  DATENV: 0               ;DATA ENVELOPE FOR TYPE OUT
 7940                                                          ;MAX FOR WRITE IS 4096
 7941                                                          ;MAX FOR READ IS 4128
 7942  050504  000000                  ZCODE:  0               ;LEADING ZEROS ENVELOPE FOR TYPE OUT
 7943                                                          ;THIS IS SHUT OFF WHEN POSITION COUNTER
 7944                                                          ;IN ENABLED
 7945                                                          ;MAX COUNT IS 38859
 7946
 7947
 7948
 7949  050506  000000                  HADTMP: 0               ;TEMPORARY HARD ERROR COUNT
 7950  050510  000000                  P3:     0
 7951  050512  000000                  P12:    0
 7952  050514  000000                  P22:    0
 7953  050516  000000                  P24:    0
 7954
 7955
 7956
 7957
 7958
 7959  050520                          ECTEST:
 7960  050520  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 7961  050522  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 7962  050524  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 7963  050526  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 7964  050530  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
 7965  050532  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 7966  050534  013701  050464                  MOV     @#GECC1,R1              ;ECC1 WORD
 7967  050540  013702  050466                  MOV     @#GECC2,R2              ;ECC2 WORD
 7968  050544  005737  050462                  TST     @#ECDATA                ;IS CURRENT BIT A ONE
 7969  050550  001406                          BEQ     2$                      ;BRANCH IF CURRENT DATA D=0
 7970
 7971                                          ;IF CARRY IS NOT ZERO THEN D=1
 7972                                          ;INVERT X32 TO GIVE R0
 7973
 7974  050552  010103                  1$:     MOV     R1,R3
 7975  050554  052703  177776                  BIS     #^CPIE32,R3
 7976  050560  005103                          COM     R3
 7977  050562  010300                          MOV     R3,R0
 7978  050564  000404                          BR      3$
 7979
 7980                                          ;IF CARRY IS ZERO THEN D=0
 7981                                          ;X32 BECOMES R0
 7982  050566  010103                  2$:     MOV     R1,R3
 7983  050570  042703  177776                  BIC     #^CPIE32,R3
 7984  050574  010300                          MOV     R3,R0
 7985
 7986  050576  000241                  3$:     CLC
 7987  050600  006000                          ROR     R0
 7988  050602  006000                          ROR     R0
 7989  050604  005700                          TST     R0
```

```
7990  050606  001462                    BEQ     10$             ;BRANCH IF R0=0
7991                                    ;INVERT X2
7992
7993  050610  010203                    MOV     R2,R3
7994  050612  052703  137777            BIS     #^CPIE2,R3
7995  050616  005103                    COM     R3
7996  050620  010337  050510            MOV     R3,@#P3
7997  050624  006237  050510            ASR     @#P3
7998
7999                                    ;INVERT X11
8000
8001
8002  050630  010203                    MOV     R2,R3
8003  050632  052703  177737            BIS     #^CPIE11,R3
8004  050636  005103                    COM     R3
8005  050640  010337  050512            MOV     R3,@#P12
8006  050644  006237  050512            ASR     @#P12
8007
8008                                    ;INVERT X21
8009
8010  050650  010103                    MOV     R1,R3
8011  050652  052703  173777            BIS     #^CPIE21,R3
8012  050656  005103                    COM     R3
8013  050660  010337  050514            MOV     R3,@#P22
8014  050664  006237  050514            ASR     @#P22
8015
8016                                    ;INVERT X23
8017
8018  050670  010103                    MOV     R1,R3
8019  050672  052703  176777            BIS     #^CPIE23,R3
8020  050676  005103                    COM     R3
8021  050700  010337  050516            MOV     R3,@#P24
8022  050704  006237  050516            ASR     @#P24
8023
8024                                    ;NOW THAT R0 FOR POSITION 1
8025                                    ;         P3 FOR POSITION 3
8026                                    ;        P12 FOR POSITION 12
8027                                    ;        P22 FOR POSITION 22
8028                                    ;        P24 FOR POSITION 24
8029                                    ;ARE KNOWN THE ROTATE WILL BE DONE AND
8030                                    ;THESE BITS JAMED IN
8031
8032  050710  006002                    ROR     R2
8033  050712  006001                    ROR     R1
8034  050714  053700  050510            BIS     @#P3,R0
8035  050720  053700  050512            BIS     @#P12,R0
8036  050724  042702  120020            BIC     #PIE1!PIE3!PIE12,R2
8037  050730  050002                    BIS     R0,R2
8038
8039  050732  005000                    CLR     R0
8040  050734  053700  050514            BIS     @#P22,R0
8041  050740  053700  050516            BIS     @#P24,R0
8042  050744  042701  002400            BIC     #PIE22!PIE24,R1
8043  050750  050001                    BIS     R0,R1
8044  050752  000404                    BR      12$
8045
```

```
8046                                        ;THE PROGRAM COMES HERE IF R0=0
8047                                        ;SO AFTER ROTATE R0 GETS PUT INTO POSITION 1
8048
8049  050754  006002              10$:      ROR     R2
8050  050756  006001                        ROR     R1
8051  050760  042702  100000              BIC     #PIE1,R2
8052  050764  010137  050464      12$:    MOV     R1,@#GECC1      ;SAVE ECC1
8053  050770  010237  050466              MOV     R2,@#GECC2      ;SAVE ECC2
8054  050774  005737  050470              TST     @#TSECCG        ;IS HARDWARE TO BE CHECKED
8055                                                              ;IF =1777777 TEST HARDWARE
8056                                                              ;IF = 0 DO NOT TEST HARDWARE
8057  051000  001432                      BEQ     14$             ;BRANCH IF HARDWARE NOT TO BE CHECKED
8058
8059
8060                              ;*CHECK HARDWARE
8061  051002  032777  000400  130130      BIT     #SW8,@SWR       ;IS SWITCH 8 SET
8062  051010  001005                      BNE     15$             ;BRANCH IF SW8 IS SET
8063  051012  032777  000100  130120      BIT     #SW6,@SWR       ;IS SWITCH 6 SET
8064  051020  001401                      BEQ     15$             ;BRANCH IF SW6 IS NOT SET
8065  051022  000421                      BR      14$             ;IF SWITCH 8 IS NOT SET AND
8066                                                              ;SWITCH 6 IS SET THEN
8067                                                              ;DO NOT DO COMPARES
8068  051024  010146              15$:    MOV     R1,-(SP)        ;GOOD PATTERN REGISTER
8069  051026  042716  174000              BIC     #174000,(SP)    ;GET ONLY PATTERN BITS
8070  051032  022677  143752              CMP     (SP)+,@RHEC2    ;COMPARE PATTERN REGISTER
8071  051036  001404                      BEQ     13$             ;BRANCH IF GOOD
8072                                      ;TO SAVE TIME
8073  051040  004737  045470              JSR     PC,@#PUTREG     ;SAVE REGISTERS
8074  051044  104035                      ERROR   35              ;PATTERN REGISTER IN 11 BITS IN ERROR
8075  051046  000407                      BR      14$             ;BRANCH OUT
8076  051050  023777  050476  143730  13$:  CMP   @#POSITI,@RHEC1 ;COMPARE POISTION REGISTER
8077  051056  001403                      BEQ     14$             ;BRANCH IF GOOD
8078                                      ;TO SAVE TIME
8079  051060  004737  045470              JSR     PC,@#PUTREG     ;SAVE REGISTERS
8080  051064  104035                      ERROR   35              ;POSITION REGISTER IN ERROR
8081                                                              ;''DATA ENVLOP'' GIVES NUMBER OF CLOCK
8082                                                              ;PULSES FROM BEGINING OF COMMAND
8083                                                              ;THAT IS THE CLOCKS IN THE R/W DATA FIELD ENVELOPE
8084                                                              ;
8085                                                              ;IN A WRITE THERE ARE 10000 OCTAL CLOCKS
8086                                                              ;IN A READ THERE ARE 10040 OCTAL CLOCKS
8087                                                              ;
8088                                                              ;
8089                                                              ;''N-CODE ZEROS'' GIVE THE NUMBER OF CLOCKS
8090                                                              ; GIVEN FOR THE LEADING ZEROS FIELD
8091                                                              ;MAX COUNT IS 113713 OCTAL
8092                                                              ;
8093                                                              ;''GOOD POSITION'' GIVES NUMBER OF CLOCKS
8094                                                              ;GIVEN AFTER LEADING ZEROS WHICH IS FOR THE DATA
8095                                                              ;FIELD
8096                                                              ;MAX COUNT IS 10040 OR 10041 OCTAL
8097
8098
8099  051066                      14$:
8100  051066  012605                      MOV     (SP)+,R5        ;;POP STACK INTO R5
8101  051070  012604                      MOV     (SP)+,R4        ;;POP STACK INTO R4
```

```
 8102  051072  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
 8103  051074  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
 8104  051076  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
 8105  051100  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
 8106  051102  000207                        RTS     PC
 8107                                        .SBTTL  ECC GENERATION CONTROL ROUTINE
 8108
 8109                              ;*THIS SUBROUTINE WILL CONTROL THE ECC GENERATION ROUTINE
 8110                              ;*FOR ERROR CORRECTION PROCESS
 8111                              ;*CALL  JSR,    PC,@#ECORR
 8112                              ;*      XP      ;EXPECTED POSITION REGISTER WHEN CORRECTION IS COMPLETE
 8113
 8114
 8115  051104  000000             ERPOS:  0                       ;POSITION REG. WHEN CORRECTION IS COMPLETE
 8116
 8117
 8118
 8119  051106  010037  015132     ECORR:  MOV     R0,@#PCJSR      ;SAVE PC OF JSR + 4
 8120  051112  162737  000004  015132     SUB     #4,@#PCJSR      ;SAVE PC OF JSR
 8121  051120  012037  051104             MOV     (R0)+,@#ERPOS   ;GET POSITION REG. WHEN CORRECTION IS COMPLETE
 8122  051124  010146                     MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 8123  051126  013701  014776             MOV     @#RHMR,R1       ;MAIMTENANCE REGISTER
 8124  051132  012711  000001             MOV     #DMD,@R1        ;SET DIAGNOSTIC MODE BIT
 8125  051136  005037  050462             CLR     @#ECDATA        ;ECC DATA IS ZERO
 8126
 8127
 8128
 8129  051142  005737  050476     1$:     TST     @#POSITI        ;IS  SOFTWARE POSITION NON ZERO
 8130  051146  001007                     BNE     2$              ;BRANCH IF N-CODE S COMPLETE
 8131  051150  005337  050474             DEC     @#NCOUNT        ;DECREMENT N-CODE
 8132  051154  001001                     BNE     6$              ;BRANCH IF N-CODE IS NOT COMPLETE
 8133  051156  000403                     BR      2$              ;BRANCH  AS N-CODE IS COMPLETE
 8134  051160  005237  050504     6$:     INC     @#ZCODE         ;INCREMENT CLOCKS GIVEN FOR LEADING ZEROS
 8135  051164  000420                     BR      3$              ;BRANCH AS N-CODE IS NOT COMPLETE
 8136                                                             ;GO TO GIVE CLOCK AND TEST ECC
 8137  051166  005237  050476     2$:     INC     @#POSITI        ;INCREMENT SOFTWARE POSITION
 8138  051172  023737  051104  050476     CMP     @#ERPOS,@#POSITI;HAVE ENOUGH CLOCKS BEEN GIVEN TO DETECT ERROR
 8139  051200  103012                     BHIS    3$              ;BRANCH IF MORE CLOCKS TO BE GIVEN
 8140  051202  023737  050506  050476     CMP     @#HADTMP,@#POSITI;HAVE ENOUGH CLOCKS BEEN GIVEN FOR HARD ERROR
 8141                                                             ;THAT IS HAVE 4128 MORE CLOCKS BEEN GIVEN
 8142  051210  001415                     BEQ     5$              ;BRANCH IF YES
 8143  051212  032711  000400             BIT     #ZER,@R1        ;CHECK ZERO DETECT BIT IN RHMR
 8144  051216  001016                     BNE     4$              ;BRANCH IS ZER SET
 8145                                     ;TO SAVE TIME
 8146  051220  004737  045470             JSR     PC,@#PUTREG     ;SAVE REGISTERS
 8147  051224  104034                     ERROR   34              ;ZERO DETECT BIT NOT HIGH
 8148                                                             ;WHEN 21 BITS IN ECC 32 BIT REGISTER IS  0
 8149
 8150
 8151  051226  052711  000002     3$:     BIS     #MCLK,@R1       ;SET CLOCK
 8152  051232  042711  000002             BIC     #MCLK,@R1       ;CLEAR CLOCK
 8153  051236  004737  050520             JSR     PC,@#ECTEST     ;GO TO GENERATE AND TEST ECC
 8154  051242  000737                     BR      1$              ;CONTINUE
 8155
 8156                                     ;THIS EXTRA CLOCK IS TO BRING ECH HIGH
 8157                                     ;AFTER THIS CLOCK POSITION REGISTER MAY BE 10040 OR 10041 OCTAL
```

```
 8158
 8159  051244  052711  000002          5$:     BIS     #MCLK,@R1       ;SET CLOCK
 8160  051250  042711  000002                  BIC     #MCLK,@R1       ;CLEAR CLOCK
 8161
 8162  051254                          4$:
 8163  051254  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
 8164  051256  000200                          RTS     R0
```

```
8165                                    .SBTTL  SOFTWARE DISK DATA ECC GEN. ROUTINE
8166
8167                                    ;*THIS SUBROUTINE GENERATES THE ECC FOR WHAT IS ON DISK AND INSERTS THEM
8168                                    ;*ON LOCATIONS 'DISK+1000' AND 'DISK+1002'
8169
8170
8171  051260                            FILLEC:
8172  051260  010046                            MOV     R0,-(SP)        ;;PUSH R0 ON STACK
8173  051262  010146                            MOV     R1,-(SP)        ;;PUSH R1 ON STACK
8174  051264  010246                            MOV     R2,-(SP)        ;;PUSH R2 ON STACK
8175  051266  010346                            MOV     R3,-(SP)        ;;PUSH R3 ON STACK
8176  051270  010446                            MOV     R4,-(SP)        ;;PUSH R4 ON STACK
8177  051272  010546                            MOV     R5,-(SP)        ;;PUSH R5 ON STACK
8178  051274  005037  050476                    CLR     @#POSITI        ;CLEAR POSITION
8179  051300  005037  050464                    CLR     @#GECC1         ;CLEAR GECC1
8180  051304  005037  050466                    CLR     @#GECC2         ;CLEAR
8181  051310  012701  054572                    MOV     #DISK,R1        ;POINTER TO DATA FOR ECC GENERATION
8182  051314  012702  000400                    MOV     #256.,R2        ;COUNTER FOR NUMBER OF DATA WORDS
8183  051320  012703  000020            9$:     MOV     #16.,R3         ;COUNTER FOR NUMBER OF BITS PER WORD
8184  051324  012104                            MOV     (R1)+,R4        ;DATA IN R4
8185  051326  006004                    10$:    ROR     R4              ;GET ONE DATA BIT IN CARRY
8186  051330  103004                            BCC     11$             ;BRANCH IF DATA BIT IS ZERO
8187  051332  012737  177777  050462            MOV     #-1,@#ECDATA    ;ECC DATA BIT IS A ONE
8188  051340  000402                            BR      12$             ;BRANCH TO GENERATE ECC
8189  051342  005037  050462            11$:    CLR     @#ECDATA        ;ECC DATA BIT IS A ZERO
8190  051346  004737  050520            12$:    JSR     PC,@#ECTEST     ;GO TO GENERATE ECC
8191  051352  005303                            DEC     R3              ;DECREMENT BIT COUNT
8192  051354  001364                            BNE     10$             ;BRANCH IF 16 BITS NOT DONE
8193  051356  005302                            DEC     R2              ;DECREMENT WORD COUNT
8194  051360  001357                            BNE     9$              ;BRANCH IF 256 WORDS NOT DONE
8195  051362  013737  050464  055572            MOV     @#GECC1,@#DISK+<256.*2>;INSERT ECC1 ON DISK
8196  051370  013737  050466  055574            MOV     @#GECC2,@#DISK+<257.*2>;INSERT ECC2 ON DISK
8197  051376  012605                            MOV     (SP)+,R5        ;;POP STACK INTO R5
8198  051400  012604                            MOV     (SP)+,R4        ;;POP STACK INTO R4
8199  051402  012603                            MOV     (SP)+,R3        ;;POP STACK INTO R3
8200  051404  012602                            MOV     (SP)+,R2        ;;POP STACK INTO R2
8201  051406  012601                            MOV     (SP)+,R1        ;;POP STACK INTO R1
8202  051410  012600                            MOV     (SP)+,R0        ;;POP STACK INTO R0
8203  051412  000207                            RTS     PC
```

```
8204
8205                                    .SBTTL  RH BASE ADDRESS CHANGE ROUTINE
8206
8207                            ;*      THIS ROUTINE WILL ALLOW THE CHANGE OF THE BASE
8208                            ;*      ADDRESS FROM 176700 TO ANY TYPED VALUE
8209
8210  051414                    BASECH:
8211  051414  104401  051422            TYPE    ,65$                    ;;TYPE ASCIZ STRING
8212  051420  000425                    BR      64$                     ;;GET OVER THE ASCIZ
8213  051474  013746  014756            MOV     @#RHCS1,-(SP)           ;GET READY TO TYPE OLD BASE
8214  051500  104402                    TYPOC
8215  051502  104401  051510            TYPE    ,67$                    ;;TYPE ASCIZ STRING
8216  051506  000425                    BR      66$                     ;;GET OVER THE ASCIZ
8217  051562  004737  060352            JSR     PC,@#$TKINT             ;INITIALIZE THE TTY KEYBOARD
8218  051566  104412                    RDOCT
8219  051570  012700  014746            MOV     #RHDB,R0                ;GET STARTING ADDRESS OF REGISTERS
8220  051574  012701  000026            MOV     #22.,R1                 ;NUMBER OF REGISTERS
8221  051600  012737  052400  000004    MOV     #ADTIMO,@#4             ;SET UP TO TEST THIS ADDRESS
8222  051606  021637  014756            CMP     @SP,@#RHCS1             ;NEW CSR?
8223  051612  001431                    BEQ     1$                      ;NO, THE OLD ONE WAS RETYPED
8224  051614  005776  000000            TST     @0(SP)                  ;ACCESS THE NEW ADDRESS
8225  051620  163716  014756            SUB     @#RHCS1,@SP             ;GET THE ADDRESS OFFSET
8226  051624  061620            2$:     ADD     @SP,(R0)+               ;AND PLUG IT IN
8227  051626  005301                    DEC     R1                      ;ONE LESS ADDRESS TO CHANGE
8228  051630  001375                    BNE     2$                      ;BUT DO SOME MORE
8229  051632  104401  051640            TYPE    ,69$                    ;;TYPE ASCIZ STRING
8230  051636  000417                    BR      68$                     ;;GET OVER THE ASCIZ
8231  051676  013746  014744    1$:     MOV     @#RPVEC,-(SP)           ;GET READY TO TYPE OLD VECTOR ADDRESS
8232  051702  104402                    TYPOC
8233  051704  104401  051712            TYPE    ,71$                    ;;TYPE ASCIZ STRING
8234  051710  000437                    BR      70$                     ;;GET OVER THE ASCIZ
8235  052010  104412                    RDOCT
8236  052012  012637  014744            MOV     (SP)+,@#RPVEC           ;SETUP VECTOR ADDRESS
8237  052016  104401  052024            TYPE    ,73$                    ;;TYPE ASCIZ STRING
8238  052022  000416                    BR      72$                     ;;GET OVER THE ASCIZ
8239  052060  013746  014756            MOV     @#RHCS1,-(SP)
8240  052064  104402                    TYPOC
8241  052066  104401  052074            TYPE    ,75$                    ;;TYPE ASCIZ STRING
8242  052072  000416                    BR      74$                     ;;GET OVER THE ASCIZ
8243  052130  013746  014744            MOV     @#RPVEC,-(SP)
8244  052134  104402                    TYPOC
8245  052136  104401  052144            TYPE    ,77$                    ;;TYPE ASCIZ STRING
8246  052142  000417                    BR      76$                     ;;GET OVER THE ASCIZ
8247  052202  104401  052210            TYPE    ,79$                    ;;TYPE ASCIZ STRING
8248  052206  000402                    BR      78$                     ;;GET OVER THE ASCIZ
8249  052214  104401  052222            TYPE    ,81$                    ;;TYPE ASCIZ STRING
8250  052220  000424                    BR      80$                     ;;GET OVER THE ASCIZ
8251  052272  104401  052300            TYPE    ,83$                    ;;TYPE ASCIZ STRING
8252  052276  000426                    BR      82$                     ;;GET OVER THE ASCIZ
8253  052354  012746  000200            MOV     #RA,-(SP)
8254  052360  104402                    TYPOC
8255  052362  104401  052370            TYPE    ,85$                    ;;TYPE ASCIZ STRING
8256  052366  000402                    BR      84$                     ;;GET OVER THE ASCIZ
8257  052374  000137  017356            JMP     @#BEGIN                 ;ALL DONE, NOW START OVER!
8258  052400  022626            ADTIMO: CMP     (SP)+,(SP)+
8259  052402  104401  052410            TYPE    ,65$                    ;;TYPE ASCIZ STRING
```

```
 8260  052406  000424                    BR      64$             ;;GET OVER THE ASCIZ
 8261  052460  000137  051414            JMP     @#BASECH
 8262
 8263
 8264
 8265                                    ;*THIS IS A LITTLE ROUTINE THAT TESTS NED BIT 11 IN RHCS2
 8266                                    ;*THIS LOOPS HERE FOR EVER
 8267                                    ;*TO BE USED ONLY IF DRIVES PRESENT LOOKING AT NED DOES NOT AGREE
 8268                                    ;*WITH WHAT IS REALY THERE
 8269  052464  000000                    ERUNIT: 0                       ;UNIT UNDER MANUAL TEST
 8270  052466  004737  045770            ERSTART:JSR     PC,@#CLDISK     ;SET GENERAL REG.
 8271  052472  013712  052464                    MOV     @#ERUNIT,@R2    ;SELECT UNIT
 8272  052476  005714                    1$:     TST     @R4             ;TEST RHER1
 8273  052500  032712  010000                    BIT     #NED,@R2        ;TEST NED
 8274  052504  001401                            BEQ     2$              ;BRANCH IF GOOD
 8275  052506  000773                            BR      1$              ;NED NOT SET
 8276  052510  000772                    2$:     BR      1$              ;NED SET
```

```
8277                                    .SBTTL  DISK SIMULATION
8278                                ;*IN A WRITE HEADER AND DATA COMMAND FILL THE FOLLOWING
8279                                ;*WCLY=WITH CYLINDER TO BE ON DISK
8280                                ;*WSECTR=WITH SECTOR AND TRACK TO BE ON DISK
8281                                ;*WKEY1= WITH KEY1 TO BE ON DISK
8282                                ;*WKEY2= WITH KEY2 TO BE ON DISK
8283                                ;*FNWORD= NO OF DATA WORDS TO BE WRITTEN ON DISK
8284                                ;*THE COMMAND THEN IS   JSR PC,COMWHD
8285                                ;*
8286                                ;*
8287                                ;*
8288                                ;*
8289                                ;*IN A WRITE DATA COMMAND FILL THE FOLLOWING
8290                                ;*CYL=WITH CYLINDER TO BE FOUND ON DISK
8291                                ;*SECOTR= WITH SECTOR AND TRACK TO BE FOUND ON DISK
8292                                ;*KEY1= WITH KEY1 TO BE FOUND ON DISK
8293                                ;*KEY2= WITH KEY2 TO BE FOUND ON DISK
8294                                ;*X= 1  MUST BE ONE
8295                                ;*NOWORD= WITH NUMBER OF DATA WORDS TO BE WRITTEN
8296                                ;*THE COMMAND THEN IS JSR PC,COMHD
8297                                ;*
8298                                ;*
8299                                ;*
8300                                ;*
8301                                ;*
8302                                ;*
8303                                ;*IN A READ HEADER AND DATA COMMAND FILL THE FOLLOWING
8304                                ;*CYL= WITH CYLINDER TO BE FOUND ON DISK
8305                                ;*SECOTR= WITH SECTOR AND TRACK TO BE FOUND ON DISK
8306                                ;*KEY1= WITH KEY1 TO BE FOUND ON DISK
8307                                ;*KEY2=WITH KEY2 TO BE FOUND ON DISK
8308                                ;*DAWORD= WITH NUMBER OF WORDS TO BE FOUND ON DISK
8309                                ;*;X=0 MUST BE ZERO
8310                                ;*THE COMMAND THEN IS JSR PC,COMHD
8311                                ;*
8312                                ;*
8313                                ;*
8314                                ;*
8315                                ;*
8316                                ;*
8317                                ;*
8318                                ;*
8319                                ;*
8320                                ;*
8321                                ;*IN A READ DATA COMMAND FILL THE FOLLOWING
8322                                ;*CYL= WITH CYLINDER TO BE FOUND ON DISK
8323                                ;*SECOTR= WITH SECTOR AND TRACK TO BE FOUND ON DISK
8324                                ;*KEY1= WITH KEY1 TO BE FOUND ON DISK
8325                                ;*KEY2=WITH KEY2 TO BE FOUND ON DISK
8326                                ;*DAWORD= WITH NUMBER OF WORDS TO BE FOUND ON DISK
8327                                ;*;X=0 MUST BE ZERO
8328                                ;*THE COMMAND THEN IS JSR PC,COMHD
8329                                ;*
```

```
8330
8331
8332
8333
8334
8335
8336                                    ;*WRITE DATA COMMAND
8337                                    ;*OR READ COMMAND, I.E DATA ONLY, OR HEADER AND DATA
8338
8339
8340
8341
8342
8343                                    ;*THIS SUBROUTINE IS THE FIRST IN A SERIES OF NESTED SUBROUTINES
8344
8345                                    ;*IT ISSUES DIAGNOSTIC MODE, AND EXTRA DIAGNOSTIC INDEX, AND THE
8346                                    ;*'GO' BIT
8347
8348                                    ;*IT THEN CALLS THREE OTHER SUBROUTINES, WHICH IN TURN CALL
8349                                    ;*OTHER SUBROUTINES.  THE SUBROUTINES CALLED HERE ARE:
8350
8351                                    ;*      SEARCH          ;ISSUES SECTOR CLOCKS TO SET SECTOR FOUND FLOP
8352                                    ;*      RDHEAD          ;READS THE SECTOR HEADER
8353                                    ;*      WRDATA          ;WRITES THE SECTOR DATA (WRITE OPERATION)
8354                                    ;*      REDATA          ;READS THE SECTOR DATA (READ OPERATION)
8355
8356
8357
8358
8359
8360  052512  000000                    RUNCTR: .WORD   0
8361  052514  011637  015132            COMHD:  MOV     (SP),@#PCJSR    ;SAVE PC OF JSR + 4
8362  052520  162737  000004  015132            SUB     #4,@#PCJSR      ;SAVE PC OF JSR
8363  052526  010046                            MOV     R0,-(SP)        ;;PUSH R0 ON STACK
8364  052530  010146                            MOV     R1,-(SP)        ;;PUSH R1 ON STACK
8365  052532  010246                            MOV     R2,-(SP)        ;;PUSH R2 ON STACK
8366  052534  010346                            MOV     R3,-(SP)        ;;PUSH R3 ON STACK
8367  052536  010446                            MOV     R4,-(SP)        ;;PUSH R4 ON STACK
8368  052540  010546                            MOV     R5,-(SP)        ;;PUSH R5 ON STACK
8369
8370  052542  012777  000001  142226            MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
8371  052550  052777  000004  142220            BIS     #MINX,@RHMR     ;SET DIAGNOSTIC INDEX
8372  052556  042777  000004  142212            BIC     #MINX,@RHMR     ;CLEAR DIAGNOSTIC INDEX
8373  052564  052777  000001  142164            BIS     #GO,@RHCS1      ;ISSUE 'GO' BIT & STALL 'TILL 'RUN'
8374                                                                    ;FUNCTION CODE WAS ISSUED BY THE TEST
8375  052572  012737  000113  052512  RUNWAT: MOV     #75.,@#RUNCTR   ;LOAD STALL COUNT = APPROX. 450US FOR 11/50 CPU
8376  052600  005337  052512          1$:     DEC     @#RUNCTR        ;COUNT DOWN ONE
8377  052604  001375                          BNE     1$              ;CONTINUE UNTIL = 0
8378
8379  052606  013746  052656                  MOV     SECOTR, -(SP)   ;GET DESIRED SECTOR/TRACK
8380  052612  042716  177740                  BIC     #177740,(SP)    ;MAKE ONLY SECTOR
8381  052616  012637  052626                  MOV     (SP)+,@#TRK     ;SAVE SECTOR
8382  052622  004137  056722                  JSR     R1,@#SEARCH     ;ISSUE SECTOR CLOCKS <---------------->
8383
8384  052626  000000                    TRK:    .WORD   0
8385  052630  012701  000240                  MOV     #+NOP,R1        ;GOING TO MOVE NOPS
```

```
8386  052634  010137  052666                  MOV     R1,@#SSYN       ;NOP INTO SSYN
8387  052640  010137  052670                  MOV     R1,@#HEDGAP     ;NOP INTO HEDGAP
8388  052644  010137  052672                  MOV     R1,@#HEDSYN     ;NOP INTO HEDSYN
8389  052650  004137  052776                  JSR     R1,@#RDHEAD     ;READ THE HEADER <------------>
8390
8391  052654  000000              CYL:    .WORD   0               ;CYLINDER ADDRESS
8392  052656  000000              SECOTR: .WORD   0               ;SECTOR/TRACK ADDRESS
8393  052660  000000              KEY1:   .WORD   0               ;KEY1 WORD
8394  052662  000000              KEY2:   .WORD   0               ;KEY2 WORD
8395  052664  000000              X:      .WORD   0               ;X=1 WRITE COMMAND
8396                                                              ;X=0 READ COMMAND
8397
8398
8399                                  ;DUMMY ERROR CALL LOCATIONS FOR THE READ HEADER OPERATION
8400
8401  052666  000240              SSYN:   NOP                     ;IF 'ERROR 2' INSERTED BY RDHEAD
8402                                                              ;SUBROUTINE, THEN THE FIRST SYNC
8403                                                              ;IS NOT DETECTED.  NO BAD DATA
8404                                                              ;IS GIVEN BECAUSE SYNC=144000
8405                                                              ;CANNOT BE READ.  WORD NUMBER
8406                                                              ;IS '1' BECAUSE THIS IS THE FIRST
8407                                                              ;WORD TESTED.
8408
8409  052670  000240              HEDGAP: NOP                     ;IF 'ERROR 3' INSERTED BY
8410                                                              ;RDHEAD SUBROUTINE, THEN THE
8411                                                              ;HEADER GAP 0'S WERE NOT
8412                                                              ;WRITTEN RIGHT.
8413
8414                                                              ;IF 'WORD NO' CONTAINS, SAY
8415                                                              ;3(8), THEN IT IS THE THIRD
8416                                                              ;WORD OF A 5 WORD HEADER
8417                                                              ;GAP THAT IS WRONG.
8418
8419                                                              ;'BAD DATA' CONTAINS WHAT IS
8420                                                              ;GOING ON THE DISK.
8421
8422  052672  000240              HEDSYN: NOP
8423                                                              ;IF 'ERROR 3' INSERTED BY RDHEAD
8424                                                              ;SUBROUTINE, THEN THE HEADER SYNC
8425                                                              ;GENERATED BY DCL IS WRONG,
8426                                                              ;OR THE LAST BYTE
8427                                                              ;OF THE HEADER GAP 0'S IS WRONG.
8428
8429                                                              ;IN EITHER CASE WORD NO=6,
8430                                                              ;RIGHT BYTE IS HEADER 0,
8431                                                              ;LEFT BYTE IS SYNC.
8432
8433                                                              ;'BAD DATA' HAS WHAT IS GOING
8434                                                              ;ON DISK.
8435
8436  052674  005737  015124          TST     @#ERFLG$        ;WERE ANY ERRORS DETECTED ?
8437  052700  001017                  BNE     OUT             ;IF YES, EXIT ---------------------->
8438
8439  052702  005737  052664          TST     @#X             ;IS IT A DATA WRITE OPERATION ?
8440  052706  001410                  BEQ     DAREAD          ;NO...THEN DO A DATA READ
8441  052710  005737  052770          TST     @#NOSYNC        ;IS THIS FORCED HEADER ERROR COMMAND ?
```

```
8442                                                    ;IF YES  NOSYNC=-1 THEN WRITE OR READ
8443                                                    ;IS SHUT OFF, SO BRANCH OUT
8444                                                    ;IF  NOSYNC=0 THEN CONTINUE
8445  052714  001011                    BNE     OUT     ;EXIT IF SET ---------------------->
8446
8447  052716  004137  054242            JSR     R1,@#WRDATA     ;WRITE DATA <---------------------->
8448  052722  000000            NOWORD: .WORD   0       ;NO OF WORDS TO BE WRITTEN
8449  052724  000000            Y:      .WORD   0       ;
8450  052726  000404                    BR      OUT     ;EXIT ---------------------------->
8451
8452  052730  004137  057176    DAREAD: JSR     R1,@#REDATA     ;READ DATA <---------------------->
8453  052734  000000    DAWORD: .WORD   0       ;NO OF WORDS TO BE READ
8454  052736  000000            .WORD   0
8455
8456  052740            OUT:
8457  052740  012605            MOV     (SP)+,R5        ;;POP STACK INTO R5
8458  052742  012604            MOV     (SP)+,R4        ;;POP STACK INTO R4
8459  052744  012603            MOV     (SP)+,R3        ;;POP STACK INTO R3
8460  052746  012602            MOV     (SP)+,R2        ;;POP STACK INTO R2
8461  052750  012601            MOV     (SP)+,R1        ;;POP STACK INTO R1
8462  052752  012600            MOV     (SP)+,R0        ;;POP STACK INTO R0
8463  052754  000207            RTS     PC              ;EXIT
8464
```

```
 8465
 8466
 8467                                   ;*THE DISK SECTOR IS DEVIDED AS FOLLOWS
 8468
 8469                                   ;*19 WORDS OF 0, ONE WORD 144000
 8470                                   ;*THESE MAKE 39 BYTES FOR SECTOR GAP AND ONE SYNC. BYTE
 8471
 8472
 8473  052756  014400                   RSYNC:  14400
 8474  052760  000000                   RCYL:   0
 8475  052762  000000                   RSETR:  0
 8476  052764  000000                   RKEY1:  0
 8477  052766  000000                   RKEY2:  0
 8478
 8479
 8480                                   ;*5 WORDS OF 0'S, ONE WORD 144000
 8481                                   ;*THESE MAKE 11 BYTES FOR HEADER GAP AND ONE SYNC. BYTE
 8482                                   ;*THESE ARE DCL GENERATED
 8483
 8484
 8485
 8486                                   ;*THERE ARE 256 WORDS OF DATA
 8487                                   ;*THERE ARE 2 WORDS FOR ECC GENERATED BY DCL
 8488                                   ;*15 WORDS OF 0'S FOR DATA GAP AND TOLERANCE GAP
 8489
 8490
 8491
 8492
 8493
 8494
```

```
8495                                    ;*READ DISK HEADER
8496
8497
8498
8499
8500
8501  052770  000000                    NOSYNC: 0                       ;FORCED HEADER ERROR = -1
8502                                                                    ;NORMAL = 0
8503  052772  000000                    TY:     0                       ;ERROR TYPE NO.
8504  052774  000000                    ERWORD: 0                       ;ERROR WORD NO.
8505
8506
8507
8508
8509  052776  012137  052760            RDHEAD: MOV     (R1)+,@#RCYL    ;STORE CYLINDER ADDRESS
8510  053002  012137  052762                    MOV     (R1)+,@#RSETR   ;STORE SECTOR AND TRACK ADDRESS
8511  053006  012137  052764                    MOV     (R1)+,@#RKEY1   ;STORE KEY1
8512  053012  012137  052766                    MOV     (R1)+,@#RKEY2   ;STORE KEY2
8513  053016  012137  053566                    MOV     (R1)+,@#COMPA   ;STORE COMPARE OR NOT
8514  053022  010146                            MOV     R1,-(SP)        ;;PUSH R1 ON STACK
8515
8516  053024  013700  014776                    MOV     @#RHMR,R0       ;R0 CONTAINS MAINTANENCE REG.
8517  053030  012705  000002                    MOV     #2,R5           ;R5 IS A COUNTER FOR WORDS
8518  053034  012710  000001                    MOV     #DMD,@R0        ;SET DIAG. MODE
8519  053040  052710  000010                    BIS     #MSTCK,@R0      ;SET SECTOR CLOCK FOR FIRST WORD
8520  053044  052710  000002                    BIS     #MCLK,@R0       ;SET MAINT.CLOCK FOR FIRST WORD
8521  053050  042710  000012                    BIC     #MSTCK!MCLK,@R0 ;RESET THEM
8522
8523  053054  000404                            BR      2$              ;DON'T GIVE SECTOR CLOCK FIRST TIME
8524  053056  012710  000013            1$:     MOV     #MSTCK!MCLK!DMD,@R0;SET SECTOR, CLOCK, DIAG. MODE, RESET INDEX
8525  053062  042710  000012                    BIC     #MSTCK!MCLK,@R0 ;RESET SECTOR & MAINT.CLOCK
8526
8527  053066  012702  000007            2$:     MOV     #7,R2           ;LOAD BYTE COUNTER
8528  053072  052710  000002            3$:     BIS     #MCLK,@R0       ;SET MAINT. CLOCK
8529  053076  042710  000002                    BIC     #MCLK,@R0       ;RESET IT
8530  053102  005302                            DEC     R2              ;BYTE COUNTER
8531  053104  001372                            BNE     3$              ;CONTINUE IF BYTE NOT COMPLETE
8532  053106  005305                            DEC     R5              ;WORD COUNTER
8533  053110  001362                            BNE     1$              ;CONTINUE IF WORD NOT COMPLETE
8534
8535  053112  012702  000022                    MOV     #18.,R2         ;LOAD NO OF WORDS OF ZEROS
8536  053116  005037  053564            4$:     CLR     @#WORD          ;DO 0'S
8537  053122  004737  053570                    JSR     PC,@#READ       ;READ 0'S <------------------->
8538  053126  005302                            DEC     R2              ;COUNT DOWN WORDS
8539  053130  001372                            BNE     4$              ;CONTINUE
8540
8541  053132  013737  052756  053564          MOV     @#RSYNC,@#WORD  ;SYNC. WORD
8542  053140  004737  053570                    JSR     PC,@#READ       ;READ IT <---------------------->
8543  053144  032710  001000                    BIT     #DTSY,  @R0     ;SYNC. BYTE DETECTED?
8544  053150  001012                            BNE     5$              ;CONTINUE IF SYNC DETECTED
8545  053152  012737  000001  052774          MOV     #1,@#ERWORD     ;ERROR WORD NO
8546  053160  013737  052756  001124          MOV     @#RSYNC,@#$GDDAT;SYNC WORD
8547  053166  012737  104002  052666          MOV     #104002,@#SSYN  ;INSERT 'ERROR 2'' IN SSYN
8548  053174  000571                            BR      13$             ;BRANCH OUT ---------------------->
8549
8550  053176  013737  052760  053564  5$:     MOV     @#RCYL, @#WORD  ;SETUP CYLINDER
```

```
8551  053204  004737  053570                      JSR     PC,@#READ       ;READ IT <---------------------->
8552  053210  013737  052762  053564              MOV     @#RSETR,@#WORD  ;SETUP SECTOR/TRACK
8553  053216  004737  053570                      JSR     PC,@#READ       ;READ THEM <--------------------->
8554  053222  013737  052764  053564              MOV     @#RKEY1,@#WORD  ;SETUP KEY1
8555  053230  004737  053570                      JSR     PC,@#READ       ;READ IT <---------------------->
8556  053234  013737  052766  053564              MOV     @#RKEY2,@#WORD  ;SETUP KEY2
8557  053242  004737  053570                      JSR     PC,@#READ       ;READ IT <---------------------->
8558  053246  013737  054554  053564              MOV     @#WCRC,@#WORD   ;SETUP CRC
8559  053254  004737  053570                      JSR     PC,@#READ       ;READ IT <---------------------->
8560
8561  053260  005737  015144                      TST     @#TESDTE        ;IS THIS A DRIVE TIMING ERROR ?
8562  053264  001135                              BNE     13$             ;BRANCH OUT IF YES ------------>
8563  053266  005737  053566                      TST     @#COMPA         ;IS THIS A READ OR WRITE COMMAND ?
8564  053272  001472                              BEQ     11$             ;DO READ IF = 0
8565
8566                                  ;*CONTINUE WITH DIAGNOSTIC WRITE COMMAND
8567
8568  053274  012705  054556                      MOV     #HEGAP, R5      ;POINTER FOR HEADER GAP
8569  053300  012702  000005                      MOV     #5,R2           ;NO OF WORDS OF ZEROS
8570  053304  012737  000006  052774  6$:         MOV     #6,@#ERWORD     ;ERROR WORD NO SET
8571  053312  004737  054022                      JSR     PC,@#WRITE      ;FOR HEADER GAP
8572  053316  005737  054020                      TST     @#WWORD         ;TEST WRITTEN WORD
8573  053322  001413                              BEQ     7$              ;CONTINUE IF GOOD, THAT IS = 0
8574  053324  160237  052774                      SUB     R2,@#ERWORD     ;WORD NO IN ERROR
8575  053330  005037  001124                      CLR     @#$GDDAT        ;GOOD WORD SHOULD BE 0
8576  053334  013737  054020  001126              MOV     @#WWORD,$BDDAT  ;BAD DATA
8577  053342  012737  104003  052670              MOV     #104003,@#HEDGAP;''ERROR 2'' GOES IN HEDGAP
8578  053350  000503                              BR      13$             ;BRANCH OUT ---------------------->
8579
8580  053352  013725  054020          7$:         MOV     @#WWORD,(R5)+   ;SAVE HEADER GAP
8581  053356  005302                              DEC     R2
8582  053360  001351                              BNE     6$
8583  053362  004737  054022                      JSR     PC,@#WRITE      ;WRITE HEADER (DATA) GAP SYNC
8584  053366  023737  052756  054020              CMP     @#RSYNC,@#WWORD
8585  053374  001426                              BEQ     10$
8586  053376  005737  052770                      TST     @#NOSYNC        ;IS THIS FORCED HEADER ERROR COMMAND ?
8587                                                                      ;IF YES  NOSYNC=-1 THEN WRITE OR READ
8588                                                                      ;IS SHUT OFF SO BRANCH OUT
8589                                                                      ;IF NO NOSYNC=0 THEN CONTINUE
8590  053402  001406                              BEQ     14$             ;PRINT IT IF TRUE ERROR
8591
8592  053404  005737  054020                      TST     @#WWORD         ;IS IT = 0 ?
8593  053410  001420                              BEQ     10$             ;CONTINUE IF GOOD
8594  053412  005037  001124                      CLR     @#$GDDAT        ;IT SHOULD BE ZERO
8595  053416  000403                              BR      15$             ;BRANCH TO TYPE ERROR
8596  053420  013737  052756  001124  14$:         MOV     @#RSYNC,@#$GDDAT;GOOD DATA
8597  053426  013737  054020  001126  15$:         MOV     @#WWORD,@#$BDDAT;BAD DATA
8598  053434  012737  000006  052774              MOV     #6,@#ERWORD
8599  053442  012737  104003  052672              MOV     #104003,@#HEDSYN
8600  053450  000443                              BR      13$             ;BRANCH OUT ---------------------->
8601
8602  053452  013725  054020          10$:        MOV     @#WWORD,(R5)+   ;SAVE DATA SYNC.
8603  053456  000440                              BR      13$             ;EXIT --------------------------->
8604
8605                                  ;*READ COMMAND START FROM HERE
8606
```

```
 8607  053460  012702  000005          11$:    MOV     #5,R2
 8608  053464  005037  053564          12$:    CLR     WORD
 8609  053470  004737  053570                  JSR     PC,@#READ       ;READ HEADER GAP <---------------->
 8610  053474  005302                          DEC     R2              ;ARE 5 HEADER GAP ZEROS COMPLETE ?
 8611  053476  001372                          BNE     12$             ;IF NOT CONTINUE
 8612  053500  013737  052756  053564          MOV     @#RSYNC,@#WORD  ;SYNC WORD
 8613  053506  004737  053570                  JSR     PC,@#READ       ;READ HEADER (DATA) SYNC)
 8614  053512  005737  052770                  TST     @#NOSYNC        ;FORCED SYNC ERROR ?
 8615  053516  001404                          BEQ     16$             ;IF NOT ERROR COMMAND CONTINUE
 8616  053520  032710  001000                  BIT     #DTSY,@R0       ;SYNC. DETECTED
 8617  053524  001415                          BEQ     13$             ;IF ZERO BRANCH OUT ------------>
 8618  053526  000403                          BR      17$             ;IF NOT ZERO BRANCH TO ERROR
 8619
 8620  053530  032710  001000          16$:    BIT     #DTSY,  @R0     ;SYNC. DETECTED ?
 8621  053534  001011                          BNE     13$             ;EXIT IF YES ------------------->
 8622  053536  012737  000006  052774  17$:    MOV     #6,@#ERWORD     ;ERROR WORD NO.
 8623  053544  013737  052756  001124          MOV     @#RSYNC,@#$GDDAT;SYNC WORD
 8624  053552  012737  104002  052672          MOV     #104002,@#HEDSYN;MOVE ''ERROR 2''
 8625  053560                          13$:
 8626  053560  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
 8627  053562  000201                          RTS     R1              ;EXIT ------------------------->
 8628
 8629
 8630
 8631
 8632
 8633
 8634
 8635
```

```
8636                                    ;*READ ONE WORD IN 'WORD''
8637
8638
8639
8640
8641  053564  000000                    WORD:   0
8642  053566  000000                    COMPA:  0
8643
8644
8645
8646
8647  053570                            READ:
8648  053570  010246                            MOV     R2,-(SP)        ;;PUSH R2 ON STACK
8649  053572  012705  000002                    MOV     #2,R5           ;WORD COUNTER
8650  053576  012710  000001                    MOV     #DMD,@R0        ;SET DIAG. MODE
8651  053602  006037  053564                    ROR     @#WORD          ;CHECKING IF THERE IS A ONE
8652  053606  103002                            BCC     1$              ;IF NO ONE BRANCH
8653  053610  052710  000020                    BIS     #MRD,@R0        ;SET BIT 4 IF DATA HAS ONE
8654  053614  012702  000007            1$:     MOV     #7,R2           ;BYTE COUNTER
8655  053620  052710  000012                    BIS     #MSTCK!MCLK,@R0 ;SET CLOCK,DATA IF ANY, SECTOR
8656  053624  005737  050470                    TST     @#TSECCG        ;IS THIS BIT TO GENERATE AND TEST ECC ?
8657  053630  001411                            BEQ     6$              ;BRANCH IF NO
8658  053632  032710  000020                    BIT     #MRD,@R0        ;IS DATA BIT A ONE ?
8659  053636  001404                            BEQ     5$              ;BRANCH IF DATA BIT IS 0
8660  053640  012737  177777  050462           MOV     #-1,@#ECDATA    ;ECC DATA BIT IS A ONE
8661  053646  000402                            BR      6$              ;BRANCH
8662  053650  005037  050462            5$:     CLR     @#ECDATA        ;ECC DATA BIT IS A 0
8663  053654  012746  000001            6$:     MOV     #DMD,-(SP)      ;KEEP ONLY DIAG. MODE
8664  053660  006037  053564                    ROR     @#WORD          ;CHECKING IF THERE IS A ONE
8665  053664  103002                            BCC     2$              ;IF NO ONE BRANCH
8666  053666  012716  000021                    MOV     #MRD!DMD,(SP)   ;KEEP DATA AND DIAG. MODE
8667  053672  012610                    2$:     MOV     (SP)+,@R0       ;PUT IN DATA,RESET CLOCK, SECTOR
8668  053674  005737  050470                    TST     @#TSECCG        ;IS ECC TO BE GENERATED FOR THIS BIT
8669  053700  001404                            BEQ     3$              ;BRANCH IF NO
8670  053702  005237  050502                    INC     @#DATENV        ;NUMBER OF CLOCKS GIVEN FOR DATA ENVELOPE
8671  053706  004737  050520                    JSR     PC,@#ECTEST     ;GO TO GENERATE AND TEST ECC
8672  053712  052710  000002            3$:     BIS     #MCLK,@R0       ;SET CLOCK
8673  053716  005737  050470                    TST     @#TSECCG        ;IS THIS BIT TO GENERATE ECC
8674  053722  001411                            BEQ     8$              ;BRANCH IF NO
8675  053724  032710  000020                    BIT     #MRD,@R0        ;IS DATA BIT A ONE
8676  053730  001404                            BEQ     7$              ;BRANCH IF DATA BIT IS = 0
8677  053732  012737  177777  050462           MOV     #-1,@#ECDATA    ;ECC DATA BIT IS A ONE
8678  053740  000402                            BR      8$              ;BRANCH
8679  053742  005037  050462            7$:     CLR     @#ECDATA        ;ECC DATA BIT IS = 0
8680  053746  012746  000001            8$:     MOV     #DMD,-(SP)      ;KEEP DIAG. MODE
8681  053752  006037  053564                    ROR     @#WORD          ;CHECKING IF THERE IS A ONE
8682  053756  103002                            BCC     4$              ;BRANCH IF NO ONE
8683  053760  012716  000021                    MOV     #MRD!DMD,(SP)   ;KEEP DIAG. MODE AND DATA
8684  053764  012610                    4$:     MOV     (SP)+,@R0       ;SET DATA, DIAG. MODE, CLEAR CLOCK
8685  053766  005737  050470                    TST     @#TSECCG        ;IS THIS BIT TO GENERATE ECC
8686  053772  001404                            BEQ     9$              ;BRANCH IF NO
8687  053774  005237  050502                    INC     @#DATENV        ;NUMBER OF CLOCKS FOR DATA ENVELOPE
8688  054000  004737  050520                    JSR     PC,@#ECTEST     ;GO TO GENERATE AND TEST ECC
8689
8690  054004  005302                    9$:     DEC     R2              ;BYTE COUNTER
8691  054006  001341                            BNE     3$              ;CONTINUE IF ONE BYTE NOT COMPLETE
```

```
8692  054010  005305                    DEC     R5              ;WORD COUNTER
8693  054012  001300                    BNE     1$              ;CONTINUE IF ONE WORD NOT COMPLETE
8694  054014  012602                    MOV     (SP)+,R2        ;;POP STACK INTO R2
8695  054016  000207                    RTS     PC              ;EXIT ------------------------------>
8696
8697
8698
8699
8700
8701
```

```
8702                                    ;*WRITE ONE WORD WHICH COMES BACK IN 'WWORD'
8703
8704
8705
8706
8707
8708
8709  054020  000000                    WWORD:  0
8710
8711
8712
8713
8714  054022                            WRITE:
8715  054022  010046                            MOV     R0,-(SP)          ;;PUSH R0 ON STACK
8716  054024  010246                            MOV     R2,-(SP)          ;;PUSH R2 ON STACK
8717  054026  010346                            MOV     R3,-(SP)          ;;PUSH R3 ON STACK
8718  054030  010546                            MOV     R5,-(SP)          ;;PUSH R5 ON STACK
8719  054032  012705  000002                    MOV     #2,R5             ;WORD COUNTER
8720  054036  012710  000001                    MOV     #1,@R0            ;SET DIAG. MODE
8721  054042  012702  000007            1$:     MOV     #7,R2             ;BYTE COUNTER
8722  054046  012710  000013                    MOV     #MSTCK!MCLK!DMD,@R0;SET SECTOR & MANT. CLOCKS
8723  054052  032710  000040                    BIT     #MWR,@R0          ;CHECK WRITEBIT IN MAINT. REG.
8724  054056  001406                            BEQ     2$                ;BRANCH IF ZERO
8725  054060  012737  177777  050462            MOV     #-1,@#ECDATA      ;ECC DATA BIT IS A ONE
8726  054066  000261                            SEC                       ;SET CARRY
8727  054070  006003                            ROR     R3                ;MOVE 1 FORWARD
8728  054072  000404                            BR      3$
8729  054074  005037  050462            2$:     CLR     @#ECDATA          ;ECC DATA BIT IS = 0
8730  054100  000241                            CLC                       ;CLEAR CARRY
8731  054102  006003                            ROR     R3                ;MOVE 0 FOR WWORD
8732  054104  012710  000001            3$:     MOV     #DMD,@R0          ;CLEAR SECTOR AND CLOCK
8733  054110  005737  050470                    TST     @#TSECCG          ;IS THIS BIT TO GENERATE ECC ?
8734  054114  001404                            BEQ     4$                ;BRANCH IF NO
8735  054116  005237  050502                    INC     @#DATENV          ;NUMBER OF CLOCKS FOR DATA ENVELOPE
8736  054122  004737  050520                    JSR     PC,@#ECTEST       ;GO TO GENERATE AND TEST ECC <-------->
8737
8738  054126  052710  000002            4$:     BIS     #MCLK,@R0         ;SET CLOCK
8739  054132  032710  000040                    BIT     #MWR,@R0          ;CHECK WRITE BIT IN MAINT. REG.
8740  054136  001406                            BEQ     5$                ;BRANCH IF ZERO
8741  054140  012737  177777  050462            MOV     #-1,@#ECDATA      ;ECC DATA BIT IS A ONE
8742  054146  000261                            SEC                       ;SET CARRY
8743  054150  006003                            ROR     R3                ;MOVE 1 FOR WWORD
8744  054152  000404                            BR      6$
8745  054154  005037  050462            5$:     CLR     @#ECDATA          ;ECC DATA BIT IS ZERO
8746  054160  000241                            CLC                       ;CLEAR CARRY
8747  054162  006003                            ROR     R3                ;MOVE 0 FOR WWORD
8748  054164  012710  000001            6$:     MOV     #DMD,@R0          ;CLEAR CLOCK
8749  054170  005737  050470                    TST     @#TSECCG          ;IS THIS BIT TO GENERATE ECC ?
8750  054174  001404                            BEQ     7$                ;BRANCH IF NO
8751  054176  005237  050502                    INC     @#DATENV          ;NUMBER OF CLOCKS FOR DATA ENVELOPE
8752  054202  004737  050520                    JSR     PC,@#ECTEST       ;GO TO GENERATE AND TEST ECC <-------->
8753
8754  054206  005302            7$:     DEC     R2                ;COUNT FOR BYTE END
8755  054210  001346                    BNE     4$                ;IF NOT BYTE END CONTINUE
8756  054212  005305                    DEC     R5                ;COUNT FOR WORD END
8757  054214  001312                    BNE     1$                ;IF NOT WORD END CONTINUE
```

```
 8758  054216  010337  054020          MOV     R3,@#WWORD      ;STORE THE WORD
 8759  054222  012605                  MOV     (SP)+,R5        ;;POP STACK INTO R5
 8760  054224  012603                  MOV     (SP)+,R3        ;;POP STACK INTO R3
 8761  054226  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
 8762  054230  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
 8763  054232  000207                  RTS     PC              ;EXIT ---------------------------------->
 8764
 8765
 8766
 8767
 8768
 8769
```

```
8770                                    ;*WRITE DATA HOUSEKEEPING ROUTINE
8771
8772
8773
8774
8775
8776
8777
8778  054234  000000              COUNTD: 0
8779  054236  000400              FORMAT: 256.
8780  054240  000000              ZWORDS: 0
8781  054242                      WRDATA:
8782  054242  011137  054234              MOV     (R1),@#COUNTD   ;STORE NO. OF WORDS TO BE WRITTEN
8783  054246  012102                      MOV     (R1)+,R2        ;SAME IN R2
8784  054250  012137  053566              MOV     (R1)+,@#COMPA   ;COMPARE OR NOT
8785  054254  010046                      MOV     R0,-(SP)        ;;PUSH R0 ON STACK
8786  054256  010146                      MOV     R1,-(SP)        ;;PUSH R1 ON STACK
8787  054260  010246                      MOV     R2,-(SP)        ;;PUSH R2 ON STACK
8788  054262  010346                      MOV     R3,-(SP)        ;;PUSH R3 ON STACK
8789  054264  010446                      MOV     R4,-(SP)        ;;PUSH R4 ON STACK
8790  054266  012701  000016              MOV     #14.,R1         ;NO. OF TOLERANCE GAP WORDS
8791  054272  012703  055600              MOV     #TOLGAP,R3      ;START OF TOLERANCE GAP TABLE
8792  054276  012723  177777      1$:     MOV     #-1,(R3)+       ;MAKE IT 177777
8793  054302  005301                      DEC     R1              ;IS 14 COMPLETED ?
8794  054304  001374                      BNE     1$              ;IF NOT, CONTINUE
8795
8796  054306  013700  014776              MOV     @#RHMR,R0       ;R0 CONTAINS MAINTANENCE REG.
8797  054312  013746  054236              MOV     @#FORMAT,-(SP)
8798  054316  163716  054234              SUB     @#COUNTD,(SP)
8799  054322  011637  054240              MOV     (SP),@#ZWORDS   ;NO. OF ZERO WORDS TO BE WRITTEN
8800  054326  012604                      MOV     (SP)+,R4
8801  054330  005737  015142              TST     @#TSECC         ;IS THIS AN ECC TEST
8802  054334  001403                      BEQ     7$              ;BRANCH IF NO
8803  054336  012737  177777  050470      MOV     #-1,@#TSECCG    ;THESE BITS ARE TO GENERATE ECC
8804  054344  012703  054572      7$:     MOV     #DISK,R3        ;SIMULATED DISK AREA
8805  054350  004737  054022      2$:     JSR     PC,@#WRITE      ;WRITE ON SIMULATED DISK
8806  054354  013723  054020              MOV     @#WWORD,(R3)+   ;STORE ON SIMULATED DISK
8807  054360  005302                      DEC     R2
8808  054362  001372                      BNE     2$
8809  054364  005704                      TST     R4              ;ANY ZEROS TO BE WRITTEN ?
8810  054366  001406                      BEQ     4$              ;BRANCH IF NONE TO BE WRITTEN
8811
8812  054370  004737  054022      3$:     JSR     PC,@#WRITE      ;WRITE ZEROS ON SIMULATED DISK
8813  054374  013723  054020              MOV     @#WWORD,(R3)+   ;STORE THEM
8814  054400  005304                      DEC     R4
8815  054402  001372                      BNE     3$
8816  054404  005037  050470      4$:     CLR     @#TSECCG        ;NO MORE ECC TO BE GENERATED
8817  054410  012701  000002              MOV     #2,R1
8818  054414  004737  054022      5$:     JSR     PC,@#WRITE      ;WRITE ECC1 AND ECC2 ON SIMULATED DISK
8819  054420  013723  054020              MOV     @#WWORD,(R3)+   ;STORE IN WEEC1 AND WEEC2
8820  054424  005301                      DEC     R1
8821  054426  001372                      BNE     5$
8822  054430  004737  054022              JSR     PC,@#WRITE      ;WRITE DATA GAP
8823  054434  013723  054020              MOV     @#WWORD,(R3)+   ;STORE IT
8824  054440  012701  000016              MOV     #14.,R1
8825  054444  004737  054022      6$:     JSR     PC,@#WRITE      ;WRITE TOLERANCE GAP ZEROS
```

```
 8826  054450  013723  054020                MOV     @#WWORD,(R3)+   ;STORE THEM
 8827  054454  005301                        DEC     R1
 8828  054456  001372                        BNE     6$
 8829  054460  012604                        MOV     (SP)+,R4        ;;POP STACK INTO R4
 8830  054462  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
 8831  054464  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
 8832  054466  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
 8833  054470  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
 8834  054472  000201                        RTS     R1              ;EXIT ---------------------------->
 8835
 8836
 8837
 8838
 8839
 8840
 8841
 8842
```

```
8843
8844                                    ;*THIS IS THE SIMULATED DISK
8845                                    ;*ONLY ONE SECTOR OF SPACE IS ALLOCATED
8846
8847
8848
8849
8850
8851
8852
8853  054474  000023          SECGAP: .BLKW   19.          ;SECTOR GAP 38 BYTES OF 0
8854  054542  000001          WSSYNC: .BLKW   1           ;SECTOR GAP 1 BYTE OF 0 ONE SYNC BYTE
8855  054544  000004          HEADER: .BLKW   4           ;HEADER = CYL, SECTOR/TRACK, KEY1, KEY2
8856  054554  000001          WCRC:   .BLKW   1           ;CRC
8857  054556  000005          HEGAP:  .BLKW   5           ;HEADER GAP 10 BYTES OF 0
8858  054570  000001          HDWSYN: .BLKW   1           ;HEADER GAP 1 BYTE OF 0 ONE SYNC. BYTE
8859  054572                  SILOTB:                     ;(ALSO USED IN SILO TEST AS SILO TABLE)
8860  054572  000400          DISK:   .BLKW   256.        ;DATA SPACE
8861  055572  000001          WECC1:  .BLKW   1           ;ECC1
8862  055574  000001          WECC2:  .BLKW   1           ;ECC2
8863  055576  000001          DTAGAP: .BLKW   1           ;DATA GAP 2 BYTES OF 0
8864  055600  000016          TOLGAP: .BLKW   14.         ;TOLERANCE GAP 28 BYTES OF 0
8865
8866
8867
8868
8869
8870
```

```
 8871                                   ;*WRITE HEADER AND DATA
 8872
 8873
 8874
 8875
 8876
 8877                                   ;*THIS SUBROUTINE IS THE FIRST IN A SERIES OF NESTED SUBROUTINES
 8878
 8879                                   ;*IT ISSUES DIAGNOSTIC MODE, AN EXTRA DIAGNOSTIC INDEX, AND THE
 8880                                   ;*'GO' BIT
 8881
 8882                                   ;*IT THEN CALLS THREE OTHER SUBROUTINES, WHICH IN TURN CALL OTHER
 8883                                   ;*SUBROUTINES.  THE THREE SUBROUTINES CALLED HERE ARE:
 8884
 8885                                   ;*      SEARCH          ;ISSUES SECTOR CLOCKS TO SET SECTOR FOUND FLOP
 8886                                   ;*      WRHEAD          ;WRITES THE SECTOR HEADER
 8887                                   ;*      WRDATA          ;WRITES THE ACTUAL SECTOR DATA
 8888
 8889                                   ;*ALL OF THE ABOVE MENTIONED 'WRITING'' IS ACTUALLY DONE INTO A CORE
 8890                                   ;*BUFFER AREA CALLED 'DISK' VIA THE MAINTENANCE REGISTER (RHMR)
 8891
 8892
 8893
 8894
 8895
 8896
 8897  055634  000000                  RNCTR1: .WORD   0               ;'RUN' LINE STALL COUNTER
 8898
 8899  055636  011637  015132          COMWHD: MOV     (SP),@#PCJSR    ;SAVE PC OF JSR + 4
 8900  055642  162737  000004  015132          SUB     #4,@#PCJSR      ;SAVE PC OF JSR
 8901  055650  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 8902  055652  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 8903  055654  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 8904  055656  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 8905  055660  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
 8906  055662  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 8907
 8908  055664  012777  000001  137104          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
 8909  055672  052777  000004  137076          BIS     #MINX,@RHMR     ;SET DIAGNOSTIC INDEX
 8910  055700  042777  000004  137070          BIC     #MINX,@RHMR     ;CLEAR DIAGNOSTIC INDEX
 8911  055706  052777  000001  137042          BIS     #GO,@RHCS1      ;SET 'GO' BIT & STALL 'TILL 'RUN'
 8912                                                                  ;(FUNCTION CODE WAS SET UP BY THE TEST)
 8913  055714  012737  000113  055634  RNWAT1: MOV     #75.,@#RNCTR1   ;LOAD STALL COUNTER = APPROX 450US
 8914                                                                  ;FOR 11/50 CPU
 8915  055722  005337  055634          1$:     DEC     @#RNCTR1        ;COUNT DOWN 1 TIME
 8916  055726  001375                          BNE     1$              ;CONTINUE UNTIL = 0
 8917
 8918  055730  013746  056014                  MOV     @#WSECTR,-(SP)  ;GET DESIRED SECTOR/TRACK
 8919  055734  042716  177740                  BIC     #177740,(SP)    ;MAKE ONLY SECTOR
 8920  055740  012637  055750                  MOV     (SP)+,@#WTRK    ;SAVE SECTOR
 8921  055744  004137  056722          2$:     JSR     R1,@#SEARCH     ;ISSUE SECTOR CLOCKS TO GET TO
 8922                                                                  ;DESIRED SECTOR <------------------->
 8923
 8924  055750  000000                  WTRK:   .WORD   0               ;SECTOR NO.
 8925  055752  012701  000240                  MOV     #+NOP,R1        ;GOING TO MOVE NOPS
 8926  055756  010137  056024                  MOV     R1,@#SEGPER     ;NOP INTO SEGAP
```

```
8927  055762  010137  056026                  MOV     R1,@#FSYNER     ;NOP INTO FSYNER
8928  055766  010137  056030                  MOV     R1,@#ERHEAD     ;NOP INTO ERHEAD
8929  055772  010137  056032                  MOV     R1,@#ERCRC      ;NOPINTO ERCRC
8930  055776  010137  056034                  MOV     R1,@#ERHDGP     ;NOP INTO ERHDGAP
8931  056002  010137  056036                  MOV     R1,@#HDESYN     ;NOP INTO HDESYN
8932
8933  056006  004137  056106                  JSR     R1,@#WRHEAD     ;WRITE THE HEADER <---------------->
8934
8935  056012  000000              WCYL:   0               ;CYLINDER
8936  056014  000000              WSECTR: 0               ;SECTOR AND TRACK
8937  056016  000000              WKEY1:  0               ;KEY1
8938  056020  000000              WKEY2:  0               ;KEY2
8939  056022  000000              GCRC:   0               ;GOOD CRC
8940
8941
8942                                  ;DUMMY ERROR CALL LOCATIONS FOR THE WRITE HEADER OPERATION
8943
8944  056024  000240              SEGPER: NOP             ;IF 'ERROR 6'' INSERTED BY
8945                                                      ;WRHEAD SUBROUTINE, THEN
8946                                                      ;SECTOR GAP GOING ON DISK
8947                                                      ;IS NOT RIGHT.
8948
8949                                                      ;WORD NO. CONTAINS WHICH
8950                                                      ;WORD IS WRONG, THAT IS
8951                                                      ;FIRST OF TENTH, OR WHAT EVER.
8952
8953                                                      ;BAD WORD IS WHAT IS GOING ON DISK
8954
8955  056026  000240              FSYNER: NOP             ;IF 'ERROR 6'' INSERTED BY
8956                                                      ;WRHEAD SUBROUTINE, THEN
8957                                                      ;THE LAST 0 BYTE OF SECTOR
8958                                                      ;GAP OF FIRST SYNC. BYTE
8959                                                      ;AFTER SECTOR GAP IS IN
8960                                                      ;ERROR.
8961
8962                                                      ;WORD NO. CONTAINS 20
8963                                                      ;RIGHT BYTE IS SECTOR GAP
8964                                                      ;LEFT BYTE IS SYNC. BYTE.
8965
8966                                                      ;BAD WORD IS WHAT IS GOING ON
8967                                                      ;DISK
8968
8969  056030  000240              ERHEAD: NOP
8970                                                      ;IF 'ERROR 6'' INSERTED BY
8971                                                      ;WRHEAD SUBROUTINE, THEN
8972                                                      ;HEADER GOING ON DISK
8973                                                      ;IS WRONG.
8974
8975                                                      ;WORD NO 1 = CYLINDER NO
8976                                                      ;WORD NO 2 = SECTOR/TRACK
8977                                                      ;WORD NO 3 = KEY1
8978                                                      ;WORD NO 4 = KEY2
8979
8980                                                      ;BAD WORD IS WHAT IS GOING ON
8981                                                      ;DISK
8982
```

```
 8983  056032  000240                ERCRC:  NOP
 8984                                                ;IF 'ERROR 6'' INSERTED BY
 8985                                                ;WRHEAD SUBROUTINE, THEN CRC WRITTEN
 8986                                                ;ON DISK IS IN ERROR.
 8987
 8988                                                ;GOOD DATA IS WHAT SHOULD BE ON DISK
 8989                                                ;BAD DATA IS WHAT IS GOING ON DISK.
 8990
 8991                                                ;WORD NO IS 5
 8992
 8993  056034  000240                ERHDGP: NOP
 8994                                                ;IF 'ERROR 6'' INSERTED BY
 8995                                                ;WRHEAD SUBROUTINE, THEN HEADER
 8996                                                ;GAP GOING ON DISK IS WRONG.
 8997
 8998                                                ;WORD NO GIVES WHICH OF
 8999                                                ;THE HEADER GAP WORDS
 9000                                                ;ARE WRONG. FOR EXAMPLE:
 9001
 9002                                                ;WORD NO 1 = FIRST HEADER
 9003                                                ;            GAP WORD
 9004                                                ;BAD WORD IS WHAT IS GOING ON DISK
 9005
 9006  056036  000240                HDESYN: NOP
 9007                                                ;IF 'ERROR 6'' INSERTED BY
 9008                                                ;WRHEAD SUBROUTINE, THEN LAST
 9009                                                ;HEADER GAP BYTE OR HEADER
 9010                                                ;SYNC BYTE GOING ON DISK IS WRONG.
 9011
 9012                                                ;WORD NO = 5
 9013
 9014                                                ;BAD DATA IS WHAT IS GOING
 9015                                                ;ON DISK, RIGHT BYTE IS HEADER
 9016                                                ;GAP 0'S BYTE, LEFT BYTE IS HEADER
 9017                                                ;GAP SYNC.
 9018
 9019
 9020  056040  005737  015124                TST     @#ERFLG$        ;ARE ANY ERRORS DETECTED ?
 9021  056044  001004                        BNE     FOUT            ;IF YES BRANCH
 9022  056046  004137  054242                JSR     R1,@#WRDATA     ;WRITE THE DATA
 9023
 9024  056052  000000        FNWORD: .WORD   0               ;FORMAT COMMAND NO. OF DATA
 9025  056054  000000                .WORD   0
 9026
 9027  056056                FOUT:
 9028  056056  012605                MOV     (SP)+,R5        ;;POP STACK INTO R5
 9029  056060  012604                MOV     (SP)+,R4        ;;POP STACK INTO R4
 9030  056062  012603                MOV     (SP)+,R3        ;;POP STACK INTO R3
 9031  056064  012602                MOV     (SP)+,R2        ;;POP STACK INTO R2
 9032  056066  012601                MOV     (SP)+,R1        ;;POP STACK INTO R1
 9033  056070  012600                MOV     (SP)+,R0        ;;POP STACK INTO R0
 9034  056072  000207                RTS     PC              ;EXIT
```

```
 9035
 9036                                   ;*WRITE HEADER
 9037
 9038
 9039
 9040
 9041
 9042                                   ;*R0 = MAINT.REG.
 9043                                   ;*R1 = SIMULATED DISK
 9044                                   ;*R2 = BYTE COUNT
 9045                                   ;*R3 = WRITE WORD
 9046                                   ;*R5 = WORD COUNT
 9047
 9048
 9049
 9050
 9051  056074  000000                   SCYL:   0                       ;
 9052  056076  000000                   SSECTR: 0                       ;
 9053  056100  000000                   SKEY1:  0                       ;
 9054  056102  000000                   SKEY2:  0                       ;
 9055  056104  000000                   SCRC:   0                       ;
 9056
 9057
 9058  056106  012137  056074           WRHEAD: MOV     (R1)+,@#SCYL
 9059  056112  012137  056076                   MOV     (R1)+,@#SSECTR
 9060  056116  012137  056100                   MOV     (R1)+,@#SKEY1
 9061  056122  012137  056102                   MOV     (R1)+,@#SKEY2
 9062  056126  012137  056104                   MOV     (R1)+,@#SCRC
 9063  056132  010146                           MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 9064
 9065  056134  012701  054474                   MOV     #SECGAP,R1      ;SIMULATED DISK INDICATOR
 9066  056140  013700  014776                   MOV     @#RHMR, R0      ;R0 NOW HAS MAINT. REG. ADDR.
 9067  056144  012710  000001                   MOV     #DMD,@R0        ;SET DIAG. MODE
 9068  056150  012705  000002                   MOV     #2,R5           ;WORD COUNTER
 9069  056154  052710  000010                   BIS     #MSTCK,@R0      ;SET SECTOR FOR FIRST BYTE
 9070  056160  012710  000013           1$:     MOV     #MSTCK!MCLK!DMD,@R0;SET SECTOR, CLOCK, DIAG.
 9071                                                                   ;MODE, RESET INDEX
 9072  056164  032710  000040                   BIT     #MWR,@R0        ;CHECK WRITE BIT IN MAINT. REG.
 9073  056170  001403                           BEQ     2$
 9074  056172  000261                           SEC                     ;SET CARRY
 9075  056174  006003                           ROR     R3              ;MOVE ONE FORWARD
 9076  056176  000402                           BR      3$              ;
 9077  056200  000241           2$:     CLC                             ;CLEAR CARRY
 9078  056202  006003                           ROR     R3              ;MOVE ZERO FORWARD
 9079  056204  012710  000001           3$:     MOV     #DMD,@R0        ;CLEAR CLOCK, SECTOR
 9080  056210  012702  000007                   MOV     #7,R2           ;BYTE COUNTER
 9081  056214  052710  000002           4$:     BIS     #MCLK,@R0       ;SET BIT CLOCK
 9082  056220  032710  000040                   BIT     #MWR,@R0        ;CHECK WRITE BIT IN MAINT.REG.
 9083  056224  001403                           BEQ     5$              ;BRANCH IF ZERO
 9084
 9085  056226  000261                           SEC                     ;SET CARRY
 9086  056230  006003                           ROR     R3              ;MOVE ONE FORWARD
 9087  056232  000402                           BR      6$
 9088  056234  000241           5$:     CLC
 9089  056236  006003                           ROR     R3
 9090  056240  012710  000001           6$:     MOV     #DMD,@R0
```

```
9091  056244  005302                        DEC     R2
9092  056246  001362                        BNE     4$
9093  056250  005305                        DEC     R5
9094  056252  001342                        BNE     1$
9095  056254  010321                        MOV     R3,(R1)+
9096  056256  005703                        TST     R3
9097  056260  001414                        BEQ     7$
9098
9099  056262  012737  000001  052774        MOV     #1,@#ERWORD
9100  056270  005037  001124                CLR     @#$GDDAT
9101  056274  010337  001126                MOV     R3,@#$BDDAT
9102  056300  012737  104006  056024        MOV     #104006,@#SEGPER
9103  056306  000137  056714                JMP     @#17$           ;BRANCH OUT ---------------------->
9104
9105  056312  012702  000022        7$:     MOV     #18.,R2         ;COUNT NO. OF SECTOR GAP
9106  056316  012737  000024  052774  10$:  MOV     #20.,@#ERWORD   ;COUNT TO GIVE ERROR WORD
9107  056324  004737  054022                JSR     PC,@#WRITE      ;WRITE SECTOR GAP
9108  056330  013721  054020                MOV     @#WWORD,(R1)+   ;STORE SECTOR GAP WORD
9109  056334  001413                        BEQ     11$
9110  056336  160237  052774                SUB     R2,@#ERWORD     ;IF NOT GET ERROR WORD NO.
9111  056342  005037  001124                CLR     @#$GDDAT        ;GOOD WORD
9112  056346  013737  054020  001126        MOV     @#WWORD,@#$BDDAT;BAD WORD
9113  056354  012737  104006  056024        MOV     #104006,@#SEGPER;STORE 'ERROR 6'' IN SEGPER
9114  056362  000554                        BR      17$             ;BRANCH OUT
9115  056364  005302              11$:      DEC     R2              ;HAVE 18 WORDS OF ZEROS BEEN WRITTEN ?
9116  056366  001353                        BNE     10$             ;IF NOT CONTINUE
9117
9118                                        ;AT THIS POINT THE SECTOR FOUND FLOP SHOULD
9119                                        ;BE HIGH,  SO THAT THE HEADER SYNC BYTE CAN BE GIVEN.
9120
9121                                        ;HOWEVER, IN THE DRIVE TIMING ERROR TEST THE REST OF THE ROUTINE
9122                                        ;IS ABORTED
9123
9124  056370  005737  015144                TST     @#TESDTE        ;IS THIS A DRIVE TIMING ERROR TEST ?
9125  056374  001147                        BNE     17$             ;BRANCH OUT IF YES ------------------->
9126
9127  056376  004737  054022                JSR     PC,@#WRITE      ;WRITE ONE SECTOR GAP 0 BYTE
9128                                                                ;AND ONE SYNC. BYTE = 230
9129  056402  013711  054020                MOV     @#WWORD,(R1)    ;SAVE 0 BYTE AND SYNC BYTE
9130  056406  023721  052756                CMP     @#RSYNC,(R1)+   ;IF SYNC. BYTE RIGHT
9131  056412  001414                        BEQ     12$             ;IF YES CONTINUE OPERATION
9132  056414  012737  000024  052774        MOV     #20.,@#ERWORD   ;IF NOT GET READY FOR ERROR PRINT
9133
9134  056422  013737  052756  001124        MOV     @#RSYNC,@#$GDDAT;GOOD WORD
9135  056430  014137  001126                MOV     -(R1),@#$BDDAT  ;BAD WORD
9136  056434  012737  104006  056026        MOV     #104006,@#FSYNER;INSERT 'ERROR 6'' IN FSYNER
9137  056442  000524                        BR      17$             ;BRANCH OUT ------------------------->
9138
9139  056444  012702  000004        12$:    MOV     #4,R2           ;FOUR HEADER WORDS
9140  056450  012703  056074                MOV     #SCYL,R3        ;POINTER FOR HEADER TABLE
9141  056454  012737  000005  052774  13$:  MOV     #5,@#ERWORD     ;ERROR WORD NO SET
9142  056462  004737  054022                JSR     PC,@#WRITE      ;WRITE 4 HEADER WORDS
9143  056466  013711  054020                MOV     @#WWORD,(R1)    ;STORE WRITTEN WORD
9144  056472  022321                        CMP     (R3)+,(R1)+     ;IS IT CORRECT ?
9145  056474  001412                        BEQ     14$             ;IF GOOD CONTINUE OPERATION
9146                                                                ;IF NOT GET READY FOR ERROR PRINT
```

```
9147
9148  056476  160237  052774                  SUB     R2,@#ERWORD     ;WORD NO
9149  056502  014337  001124                  MOV     -(R3),@#$GDDAT  ;GOOD DATA
9150  056506  014137  001126                  MOV     -(R1),@#$BDDAT  ;BAD DATA
9151  056512  012737  104006  056030          MOV     #104006,@#ERHEAD;INSERT 'ERROR 6''
9152  056520  000475                          BR      17$             ;BRANCH OUT ---------->
9153
9154  056522  005302                14$:     DEC     R2              ;ARE 4 HEADER WORDS DONE?
9155  056524  001353                         BNE     13$             ;IF NOT CONTINE
9156  056526  004737  054022                 JSR     PC,@#WRITE      ;WRITE CRC
9157  056532  013711  054020                 MOV     @#WWORD,(R1)    ;STORE CRC
9158  056536  022137  056022                 CMP     (R1)+,@#GCRC    ;COMPARE GOOD CRC
9159  056542  001414                         BEQ     20$             ;IF GOOD CONTINUE OPERATION
9160                                                                 ;IF NOT GET READY FOR ERROR PRINT
9161
9162  056544  014137  001126                  MOV     -(R1),@#$BDDATA ;BAD CRC WRITTEN
9163  056550  013737  056022  001124          MOV     @#GCRC,@#$GDDAT ;GOOD CRC
9164  056556  012737  000005  052774          MOV     #5,@#ERWORD     ;ERROR WORD NO
9165  056564  012737  104006  056032          MOV     #104006,@#ERCRC ;INSERT ERROR 6
9166  056572  000450                          BR      17$             ;EXIT ---------------------------->
9167
9168  056574  012702  000005                20$:     MOV     #5,R2           ;NO OF HEADER GAP
9169  056600  012737  000006  052774  15$:    MOV     #6,@#ERWORD     ;ERROR WORD NO SET
9170  056606  004737  054022                 JSR     PC,@#WRITE      ;WRITE HEADER GAP
9171  056612  013721  054020                 MOV     @#WWORD,(R1)+   ;STORE
9172  056616  001412                         BEQ     16$             ;IF GOOD CONTINUE OPERATION
9173                                                                 ;IF NOT GET READY FOR PRINT
9174
9175  056620  160237  052774                  SUB     R2,@#ERWORD     ;ERROR WORD NO
9176  056624  005037  001124                  CLR     @#$GDDAT        ;GOOD DATA
9177  056630  014137  001126                  MOV     -(R1),@#$BDDAT  ;BAD DATA
9178  056634  012737  104006  056034          MOV     #104006,@#ERHDGP;STORE 'ERROR 6''
9179  056642  000424                          BR      17$             ;BRANCH OUT ---------->
9180
9181  056644  005302                16$:     DEC     R2              ;ARE 5 HEADER GAP ZEROS DONE ?
9182  056646  001354                         BNE     15$             ;IF NOT CONTINUE
9183  056650  004737  054022                 JSR     PC,@#WRITE
9184  056654  013711  054020                 MOV     @#WWORD,(R1)
9185  056660  023721  052756                 CMP     @#RSYNC,(R1)+
9186  056664  001413                         BEQ     17$             ;A-OK, EXIT ---------->
9187
9188  056666  012737  000005  052774          MOV     #5,@#ERWORD     ;IF NOT GET READY FOR ERROR PRINT
9189  056674  014137  001126                  MOV     -(R1),@#$BDDAT
9190  056700  013737  052756  001124          MOV     @#RSYNC,@#$GDDAT
9191  056706  012737  104006  056036          MOV     #104006,@#HDESYN
9192
9193  056714                        17$:
9194  056714  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
9195  056716  000201                          RTS     R1              ;EXIT ---------------------------->
9196
9197
9198
9199
9200
9201
9202
```

```
                                                                H 1
9203
```

```
                                        ;*SEARCH SECTOR
                                        ;*       R0=RHMR ADDRESS
                                        ;*       R1=PASSED ARGUMENT (SECTOR SEARCHED FOR)
                                        ;*       R2=CLOCK COUNT (PER BYTE)
                                        ;*       R3=SECTOR COUNTER FROM R1
                                        ;*       R5=BYTES PER WORD COUNT
                                        ;*BEFORE INDEX IS GIVEN TWO SECTOR CLOCKS ARE GIVEN TO RESET
                                        ;*SECTOR PULSE IN CASE IT IS SET.
                                        ;*AT THE BEGINNING OF EACH SECTOR, ONE SECTOR CLOCK HAS TO RISE
                                        ;*BEFORE MAINT. CLOCK, THEN EVERY EIGHT MAINT.CLOCKS, ONE SECTOR CLOCK
                                        ;*IS IDENTICAL WITH THE MAINT. CLOCK
                                        ;*THE SECTOR CLOCKS ARE NUMBERED AS FOLLOWS:
                                        ;*THE SECTOR CLOCK UNDER INDEX - 0
                                        ;*THE NEXT - 1
                                        ;*THE NEXT - 2
                                        ;*ETC.
                                        ;*THEN THE LAST SECTOR CLOCK IN ONE SECTOR HAS NUMBER - 608
                                        ;*THE NEXT SECTOR WILL HAVE 608 SECTOR CLOCKS
                                        ;*THE NEXT SECTOR THEN HAS ANOTHER 608 SECTOR CLOCKS
                                        ;*AND SO ON

056720  000000          SECTR:  0                       ;SECTOR SEARCHED FOR

056722  012137  056720  SEARCH: MOV     (R1)+,@#SECTR ;SAVE SECTOR SEARCHED FOR
056726  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
056730  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
056732  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
056734  010346                  MOV     R3,-(SP)        ;;PUSH R3 ON STACK
056736  010446                  MOV     R4,-(SP)        ;;PUSH R4 ON STACK
056740  010546                  MOV     R5,-(SP)        ;;PUSH R5 ON STACK
056742  013700  014776          MOV     @#RHMR, R0      ;NOW R0 HAS MAINTENANCE REG. ADR.
056746  013703  056720          MOV     @#SECTR,R3      ;LOAD SECTOR COUNTER

056752  012710  000001          MOV     #DMD,@R0        ;SET DIAGNOSTIC MODE
056756  052710  000010          BIS     #MSTCK, @R0     ;SET SECTOR CLOCK
056762  042710  000010          BIC     #MSTCK, @R0     ;CLEAR SECTOR CLOCK
056766  052710  000010          BIS     #MSTCK, @R0     ;SET SECTOR CLOCK
056772  042710  000010          BIC     #MSTCK, @R0     ;CLEAR SECTOR CLOCK
                                                        ;THE ABOVE TWO SECTOR CLOCKS ARE GIVEN FOR
                                                        ;RESETING SECTOR PULSE
                                                        ;IN CASE IT STARTS SET
056776  052710  000014          BIS     #MINX!MSTCK,@R0 ;SET INDEX AND SECTOR CLOCK
057002  042710  000014          BIC     #MINX!MSTCK,@R0 ;RESET INDEX AND SECTOR CLOCK
057006  005703                  TST     R3              ;TEST FOR SECTOR 0
057010  001461                  BEQ     7$              ;IF SECTOR 0 IS REQUIRED, EXIT ------>
```

```
                                        ;NOW 304 WORDS WILL START (608 BYTES)

                                    ;*FOR FIRST BYTE MAINT. SECTOR CLOCK WILL GO HIGH, THEN MAINT. CLOCK WILL GO HIGH
                                    ;*BOTH WILL COME DOWN TOGETHER, THEN SEVEN MAINT. CLOCKS WILL BE GIVEN
                                    ;*FOR SECOND BYTE, AND ALL OTHER BYTES TILL NEXT SECTOR, SECTOR CLOCK
                                    ;*WILL BE IDENTICAL WITH ONE MAINT. CLOCK


                                        ;ONE WORD ONLY

057012  012702  000010         1$:     MOV     #8.,    R2      ;BYTE COUNTER
057016  012705  000002                 MOV     #2,     R5      ;BYTES PER WORD
057022  052710  000010                 BIS     #MSTCK,@R0      ;SET SECTOR CLOCK
057026  052710  000002                 BIS     #MCLK,@R0       ;SET MAINT. CLOCK
057032  000402                         BR      3$              ;BRANCH TO CLEAR SECTOR AND CLOCK
057034  052710  000012         2$:     BIS     #MSTCK!MCLK,@R0 ;SET BOTH CLOCKS
057040  042710  000012         3$:     BIC     #MSTCK!MCLK,@R0 ;CLEAR BOTH CLOCKS
057044  052710  000002         8$:     BIS     #MCLK,@R0       ;SET MAINT. CLOCK
057050  042710  000002                 BIC     #MCLK,@R0       ;CLEAR MAINT.CLOCK
057054  005302                         DEC     R2              ;BYTE COUNTER
057056  001372                         BNE     8$              ;CONTINUE IF BYTE NOT COMPLETE

057060  012702  000007                 MOV     #7,R2           ;SETUP FOR SECOND BYTE
057064  005305                         DEC     R5              ;IS WORD COMPLETE?
057066  001362                         BNE     2$              ;CONTINUE IF NOT COMPLETE
                                                                ;TO GIVE SECTOR CLOCK AND MAINT. CLOCK


                                        ;NOW 303 WORDS ARE LEFT AND ALL ARE IDENTICAL

057070  012701  000457                 MOV     #303.,R1        ;WORDS PER SECTOR COUNTER
057074  012705  000002         4$:     MOV     #2,R5           ;BYTES PER WORD COUNTER
057100  012702  000007         5$:     MOV     #7,R2           ;BIT COUNTER (MAINT.CLOCK COUNTER)
057104  052710  000012                 BIS     #MSTCK!MCLK,@R0 ;SET SECTOR CLOCK AND MAINT. CLOCK
057110  042710  000012                 BIC     #MSTCK!MCLK,@R0 ;CLEAR CLOCKS
057114  052710  000002         6$:     BIS     #MCLK,@R0       ;SET MAINT. CLOCK
057120  042710  000002                 BIC     #MCLK,@R0       ;RESET MAINT. CLOCK
057124  005302                         DEC     R2              ;IS BYTE COMPLETE ?
057126  001372                         BNE     6$              ;CONTINUE IF NOT COMPLETE

057130  005305                         DEC     R5              ;IS WORD COMPLETE ?
057132  001362                         BNE     5$              ;CONTINUE IF NOT

057134  005301                         DEC     R1              ;IS SECTOR COMPLETE ?
057136  001356                         BNE     4$              ;CONTINUE IF NOT

057140  052710  000010                 BIS     #MSTCK,@R0      ;SET SECTOR CLOCK 1 MORE TIME (FOR 609)
057144  042710  000010                 BIC     #MSTCK,@R0      ;CLEAR SECTOR CLOCK
057150  005303                         DEC     R3              ;IS REQUIRED NO OF SECTORS COMPLETE ?
057152  001317                         BNE     1$              ;CONTINUE IF NOT

057154                          7$:
057154  012605                         MOV     (SP)+,R5        ;;POP STACK INTO R5
057156  012604                         MOV     (SP)+,R4        ;;POP STACK INTO R4
057160  012603                         MOV     (SP)+,R3        ;;POP STACK INTO R3
```

```
 9316  057162  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
 9317  057164  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
 9318  057166  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
 9319  057170  000201                        RTS     R1
 9320
 9321
 9322                                  ;*READ ONE SECTOR OF DATA
 9323
 9324  057172  000000                  RNO:  0                       ;NO. OF WORDS READ
 9325  057174  000000                  RCOM: 0                       ;EXTRA STORAGE
 9326
 9327
 9328
 9329  057176  012137  057172          REDATA: MOV     (R1)+,@#RNO     ;SAVE NO. OF WORDS ONLY FOR INFORMATION
 9330  057202  012137  057174                  MOV     (R1)+,@#RCOM    ;EXTRA WORD ONLY FOR INFORMATION
 9331  057206  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 9332  057210  005737  015142                  TST     @#TSECC         ;IS THIS AN ECC TEST
 9333  057214  001403                          BEQ     1$              ;BRANCH IF NO
 9334  057216  012737  177777  050470          MOV     #-1,@#TSECCG    ;THESE BITS ARE TO GENERATE ECC
 9335  057224  012702  000402          1$:     MOV     #258.,R2        ;256 WORDS PER SECTOR
 9336                                                                  ;PLUS 2 ECC WORDS
 9337  057230  012703  054572                  MOV     #DISK,R3        ;POINTE TO DISK SIMULATION
 9338  057234  012337  053564          2$:     MOV     (R3)+,@#WORD    ;READY TO READ CONTENTS
 9339  057240  004737  053570                  JSR     PC,@#READ               ;READ
 9340  057244  005302                          DEC     R2              ;IS 256 WORDS DONE?
 9341  057246  001372                          BNE     2$              ;IF NOT BRANCH
 9342  057250  005737  015142                  TST     @#TSECC         ;IS THIS AN ECC TEST
 9343  057254  001012                          BNE     4$              ;BRANCH OUT IF YES
 9344  057256  005037  050470                  CLR     @#TSECCG        ;NO MORE ECC BITS ARE TO BE GENERATED
 9345  057262  012702  000017                  MOV     #15.,R2         ;ONE DATA GAP, 14 TOLERANCE GAP
 9346  057266  012337  053564          3$:     MOV     (R3)+,@#WORD    ;READY TO READ CONTENTS OF WORD
 9347  057272  004737  053570                  JSR     PC,@#READ       ;READ
 9348  057276  005302                          DEC     R2              ;COUNT
 9349  057300  001372                          BNE     3$              ;BRANCH IF 14 NOT DONE
 9350  057302                          4$:
 9351  057302  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
 9352  057304  000201                          RTS     R1              ;RETURN
```

```
 9353  057306                                RPVECT:
 9354  057306  104401  057314                      TYPE    ,65$            ;;TYPE ASCIZ STRING
 9355  057312  000423                            BR      64$             ;;GET OVER THE ASCIZ
 9356  057362  104402                            TYPOC                           ;TYPE FROM PC
 9357  057364  012777  057306  135352            MOV     #RPVECT,@RPVEC          ;RESTORE TRAP RP04 VECTOR
 9358  057372  000000                            HALT                            ;CHANGE TO CONTINUE
 9359
```

```
 9360                                          .SBTTL
 9361                                          .SBTTL  ***SYSMAC LIBRARY ROUTINES***
 9362                                          .SBTTL
 9363
 9364
 9365
 9366  057374  104407                         CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
 9367  057376  005037  052770                 CLR     @#NOSYNC        ;CLEAR FLAG FOR HEADER ERROR COMMANDS
 9368  057402  005037  015142                 CLR     @#TSECC         ;CLEAR FLAG FOR ECC TEST
 9369  057406  005037  050470                 CLR     @#TSECCG        ;EVEN IN AN ECC TEST EVERY CLOCK
 9370  057412  005037  015144                 CLR     @#TESDTE        ;DRIVE TIMING ERROR TEST
 9371  057416  032777  040000  121514  1$:    BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
 9372  057424  001111                         BNE     $OVER           ;;YES IF SW14=1
 9373  057426  000416                 $XTSTR: BR      6$              ;;IF RUNNING ON THE 'XOR' TESTER CHANGE
 9374  057430  013746  000004                 MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
 9375  057434  012737  057454  000004         MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
 9376  057442  005737  177060                 TST     @#177060        ;;TIME OUT ON XOR?
 9377  057446  012637  000004                 MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
 9378  057452  000463                         BR      $SVLAD          ;;GO TO THE NEXT TEST
 9379  057454  022626                 5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
 9380  057456  012637  000004                 MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
 9381  057462  000423                         BR      7$              ;;LOOP ON THE PRESENT TEST
 9382  057464  032777  000400  121446         BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
 9383  057472  001404                         BEQ     2$              ;;BR IF NO
 9384  057474  127737  121440  001102         CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
 9385  057502  001462                         BEQ     $OVER           ;;BR IF YES
 9386  057504  105737  001103         2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
 9387  057510  001421                         BEQ     3$              ;;BR IF NO
 9388  057512  123737  001115  001103         CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
 9389  057520  101015                         BHI     3$              ;;BR IF NO
 9390  057522  032777  001000  121410         BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
 9391  057530  001404                         BEQ     4$              ;;BR IF NO
 9392  057532  013737  001110  001106  7$:    MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
 9393  057540  000443                         BR      $OVER
 9394  057542  105037  001103         4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
 9395  057546  005037  001212                 CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
 9396  057552  000415                         BR      1$              ;;ESCAPE TO THE NEXT TEST
 9397  057554  032777  004000  121356  3$:    BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
 9398  057562  001011                         BNE     1$              ;;BR IF YES
 9399  057564  005737  001100                 TST     $PASS           ;;IF FIRST PASS OF PROGRAM
 9400  057570  001406                         BEQ     1$              ;;      INHIBIT ITERATIONS
 9401  057572  005237  001104                 INC     $ICNT           ;;INCREMENT ITERATION COUNT
 9402  057576  023737  001212  001104         CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
 9403  057604  002021                         BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
 9404  057606  012737  000001  001104  1$:    MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
 9405  057614  013737  057664  001212         MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
 9406  057622  105237  001102         $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
 9407  057626  011637  001106                 MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
 9408  057632  011637  001110                 MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
 9409  057636  005037  001214                 CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
 9410  057642  112737  000001  001115         MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
 9411  057650  013777  001102  121264  $OVER: MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
 9412  057656  013716  001106                 MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
 9413  057662  000002                         RTI                     ;;FIXES PS
 9414  057664  000004                 $MXCNT: 4                       ;;MAX. NUMBER OF ITERATIONS
```

```
 9415  057666  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 9416  057670  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 9417  057672  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 9418  057674  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 9419  057676  010546                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 9420  057700  012746  020200                MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
 9421  057704  016605  000020                MOV     20(SP),R5       ;;GET THE INPUT NUMBER
 9422  057710  100004                        BPL     1$              ;;BR IF INPUT IS POS.
 9423  057712  005405                        NEG     R5              ;;MAKE THE BINARY NUMBER POS.
 9424  057714  112766  000055  000001        MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
 9425  057722  005000                1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
 9426  057724  012703  060102                MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
 9427  057730  112723  000040                MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
 9428  057734  005002                2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
 9429  057736  016001  060072                MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
 9430  057742  160105                3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
 9431  057744  002402                        BLT     4$              ;;BR IF DONE
 9432  057746  005202                        INC     R2              ;;INCREASE THE BCD DIGIT BY 1
 9433  057750  000774                        BR      3$
 9434  057752  060105                4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
 9435  057754  005702                        TST     R2              ;;CHECK IF BCD DIGIT=0
 9436  057756  001002                        BNE     5$              ;;FALL THROUGH IF 0
 9437  057760  105716                        TSTB    (SP)            ;;STILL DOING LEADING 0'S?
 9438  057762  100407                        BMI     7$              ;;BR IF YES
 9439  057764  106316                5$:     ASLB    (SP)            ;;MSD?
 9440  057766  103003                        BCC     6$              ;;BR IF NO
 9441  057770  116663  000001  177777        MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
 9442  057776  052702  000060          6$:   BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
 9443  060002  052702  000040          7$:   BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
 9444  060006  110223                        MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
 9445  060010  005720                        TST     (R0)+           ;;JUST INCREMENTING
 9446  060012  020027  000010                CMP     R0,#10          ;;CHECK THE TABLE INDEX
 9447  060016  002746                        BLT     2$              ;;GO DO THE NEXT DIGIT
 9448  060020  003002                        BGT     8$              ;;GO TO EXIT
 9449  060022  010502                        MOV     R5,R2           ;;GET THE LSD
 9450  060024  000764                        BR      6$              ;;GO CHANGE TO ASCII
 9451  060026  105726                8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
 9452  060030  100003                        BPL     9$              ;;BR IF NO
 9453  060032  116663  177777  177776        MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
 9454  060040  105013                9$:     CLRB    (R3)            ;;SET THE TERMINATOR
 9455  060042  012605                        MOV     (SP)+,R5        ;;POP STACK INTO R5
 9456  060044  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
 9457  060046  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
 9458  060050  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
 9459  060052  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
 9460  060054  104401  060102                TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
 9461  060060  016666  000002  000004        MOV     2(SP),4(SP)     ;;ADJUST THE STACK
 9462  060066  012616                        MOV     (SP)+,(SP)
 9463  060070  000002                        RTI                     ;;RETURN TO USER
 9464  060072  023420                $DTBL:  10000.
 9465  060074  001750                        1000.
 9466  060076  000144                        100.
 9467  060100  000012                        10.
```

```
 9468  060112  105737  001157          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
 9469  060116  100002                  BPL     1$              ;;BR IF YES
 9470  060120  000000                  HALT                    ;;HALT HERE IF NO TERMINAL
 9471  060122  000407                  BR      3$              ;;LEAVE
 9472  060124  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
 9473  060126  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
 9474  060132  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
 9475  060134  001005                  BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
 9476  060136  005726                  TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
 9477  060140  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
 9478  060142  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
 9479  060146  000002                  RTI                     ;;RETURN
 9480  060150  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
 9481  060154  001430                  BEQ     8$
 9482  060156  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
 9483  060162  001006                  BNE     5$
 9484  060164  005726                  TST     (SP)+           ;;POP  <CR><LF> EQUIV
 9485  060166  104401                  TYPE                    ;;TYPE A CR AND LF
 9486  060170  001223                  $CRLF
 9487  060172  105037  060326                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
 9488  060176  000755                  BR      2$              ;;GET NEXT CHARACTER
 9489  060200  004737  060262          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
 9490  060204  123726  001156          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
 9491  060210  001350                  BNE     2$              ;;IF NO GO GET NEXT CHAR.
 9492  060212  013746  001154                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
 9493  060216  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
 9494  060222  002770                  BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
 9495  060224  004737  060262                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
 9496  060230  105337  060326                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
 9497  060234  000770                  BR      7$              ;;LOOP
 9498  060236  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
 9499  060242  004737  060262          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
 9500  060246  132737  000007  060326          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
 9501  060254  001372                  BNE     9$              ;;TAB STOP
 9502  060256  005726                  TST     (SP)+           ;;POP SPACE OFF STACK
 9503  060260  000724                  BR      2$              ;;GET NEXT CHARACTER
 9504  060262  105777  120662          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
 9505  060266  100375                  BPL     $TYPEC
 9506  060270  116677  000002  120654          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
 9507  060276  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
 9508  060304  001003                  BNE     1$              ;;BRANCH IF NO
 9509  060306  105037  060326                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
 9510  060312  000406                  BR      $TYPEX          ;;EXIT
 9511  060314  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
 9512  060322  001402                  BEQ     $TYPEX          ;;BRANCH IF YES
 9513  060324  105227                  INCB    (PC)+           ;;COUNT THE CHARACTER
 9514  060326  000000          $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
 9515  060330  000207          $TYPEX: RTS     PC
```

```
 9516  060332  000000                $TKCNT: .WORD   0               ;;NUMBER OF ITEMS IN QUEUE
 9517  060334  000000                $TKQIN: .WORD   0               ;;INPUT POINTER
 9518  060336  000000                $TKQOUT: .WORD  0               ;;OUTPUT POINTER
 9519  060352  005037  060332        $TKINT: CLR     $TKCNT          ;;CLEAR COUNT OF ITEMS IN QUEUE
 9520  060356  012737  060340  060334        MOV     #$TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
 9521  060364  013737  060334  060336        MOV     $TKQIN,$TKQOUT  ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
 9522  060372  012737  060422  000060        MOV     #$TKSRV,@#TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
 9523  060400  012737  000200  000062        MOV     #200,@#TKVEC+2  ;;'BR'' LEVEL 4
 9524  060406  005777  120534                TST     @$TKB           ;;CLEAR DONE FLAG
 9525  060412  012777  000100  120524        MOV     #100,@$TKS      ;;ENABLE TTY KEYBOARD INTERRUPT
 9526  060420  000207                        RTS     PC              ;;RETURN TO CALLER
 9527  060422  117746  120520        $TKSRV: MOVB    @$TKB,-(SP)     ;;PICKUP THE CHARACTER
 9528  060426  042716  177600                BIC     #^C177,(SP)     ;;STRIP THE JUNK
 9529  060432  021627  000003                CMP     (SP),#3         ;;IS IT A CONTROL C?
 9530  060436  001007                        BNE     1$              ;;BRANCH IF NO
 9531  060440  104401  061411                TYPE    ,$CNTLC         ;;TYPE A CONTROL-C (^C)
 9532  060444  004737  060352                JSR     PC,$TKINT       ;;INIT THE KEYBOARD
 9533  060450  005726                        TST     (SP)+           ;;CLEAN UP STACK
 9534  060452  000137  044710                JMP     OPERSEL         ;;CONTROL C RESTART
 9535  060456  021627  000007        1$:     CMP     (SP),#7         ;;IS IT A CONTROL G?
 9536  060462  001004                        BNE     2$              ;;BRANCH IF NO
 9537  060464  022737  000176  001140        CMP     #SWREG,SWR      ;;IS SOFT-SWR SELECTED?
 9538  060472  001500                        BEQ     6$              ;;GO TO SWR CHANGE
 9539  060474  022737  000011  060332        CMP     #9.,$TKCNT      ;;IS THE QUEUE FULL?
 9540  060502  001004                        BNE     3$              ;;BRANCH IF NO
 9541  060504  104401  001216                TYPE    ,$BELL          ;;RING THE TTY BELL
 9542  060510  005726                        TST     (SP)+           ;;CLEAN CHARACTER OFF OF STACK
 9543  060512  000451                        BR      5$              ;;EXIT
 9544  060514  021627  000023        3$:     CMP     (SP),#23        ;;IS IT A CONTROL-S?
 9545  060520  001021                        BNE     32$             ;;BRANCH IF NO
 9546  060522  005077  120416                CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
 9547  060526  005726                        TST     (SP)+           ;;CLEAN CHAR OFF STACK
 9548  060530  105777  120410        31$:    TSTB    @$TKS           ;;WAIT FOR A CHAR
 9549  060534  100375                        BPL     31$             ;;LOOP UNTIL ITS THERE
 9550  060536  117746  120404                MOVB    @$TKB,-(SP)     ;;GET THE CHARACTER
 9551  060542  042716  177600                BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
 9552  060546  022627  000021                CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
 9553  060552  001366                        BNE     31$             ;;BRANCH IF NO
 9554  060554  012777  000100  120362        MOV     #100,@$TKS      ;;REENABLE TTY KEYBOARD INTERRUPTS
 9555  060562  000002                        RTI                     ;;RETURN
 9556  060564  005237  060332        32$:    INC     $TKCNT          ;;COUNT THIS CHARACTER
 9557  060570  021627  000140                CMP     (SP),#140       ;;IS IT UPPER CASE?
 9558  060574  002405                        BLT     4$              ;;BRANCH IF YES
 9559  060576  021627  000175                CMP     (SP),#175       ;;IS IT A SPECIAL CHAR?
 9560  060602  003002                        BGT     4$              ;;BRANCH IF YES
 9561  060604  042716  000040                BIC     #40,(SP)        ;;MAKE IT UPPER CASE
 9562  060610  112677  177520        4$:     MOVB    (SP)+,@$TKQIN   ;;AND PUT IT IN QUEUE
 9563  060614  005237  060334                INC     $TKQIN          ;;UPDATE THE POINTER
 9564  060620  023727  060334  060351        CMP     $TKQIN,#$TKQEND ;;GO OFF THE END?
 9565  060626  001003                        BNE     5$              ;;BRANCH IF NO
 9566  060630  012737  060340  060334        MOV     #$TKQSRT,$TKQIN ;;RESET THE POINTER
 9567  060636  000002                5$:     RTI                     ;;RETURN
 9568  060640  022737  000176  001140 $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED
 9569  060646  001124                        BNE     15$             ;;EXIT IF NOT
 9570  060650  105777  120270                TSTB    @$TKS           ;;IS A CHAR WAITING?
 9571  060654  100121                        BPL     15$             ;;IF NOT, EXIT
```

```
 9572  060656  117746  120264                    MOVB    @$TKB,-(SP)     ;;YES
 9573  060662  042716  177600                    BIC     #^C177,(SP)     ;;MAKE IT  7-BIT ASCII
 9574  060666  021627  000007                    CMP     (SP),#7         ;;IS IT A CONTROL-G?
 9575  060672  001300                            BNE     2$              ;;IF NOT, PUT IT IN THE TTY QUEUE
 9576  060674  123727  001134  000001  6$:      CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
 9577  060702  001674                            BEQ     2$              ;;BRANCH IF YES
 9578  060704  005726                            TST     (SP)+           ;;CLEAR CONTROL-G OFF STACK
 9579  060706  004737  060352                    JSR     PC,$TKINT       ;;FLUSH THE TTY INPUT QUEUE
 9580  060712  005077  120226                    CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
 9581  060716  112737  000001  001135          MOVB    #1,$INTAG       ;;SET INTERRUPT MODE INDICATOR
 9582  060724  104401  061423                    TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
 9583  060730  104401  061430          $GTSWR:   TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
 9584  060734  013746  000176                    MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
 9585  060740  104402                            TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 9586  060742  104401  061441                    TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
 9587  060746  005046                  19$:      CLR     -(SP)           ;;CLEAR COUNTER
 9588  060750  005046                            CLR     -(SP)           ;;THE NEW SWR
 9589  060752  105777  120166          7$:       TSTB    @$TKS           ;;CHAR THERE?
 9590  060756  100375                            BPL     7$              ;;IF NOT TRY AGAIN
 9591  060760  117746  120162                    MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
 9592  060764  042716  177600                    BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
 9593  060770  021627  000003                    CMP     (SP),#3         ;;IS IT A CONTROL-C?
 9594  060774  001015                            BNE     9$              ;;BRANCH IF NOT
 9595  060776  104401  061411                    TYPE    ,$CNTLC         ;;YES, ECHO CONTROL-C (^C)
 9596  061002  062706  000006                    ADD     #6,SP           ;;CLEAN UP STACK
 9597  061006  123727  001135  000001          CMPB    $INTAG,#1       ;;REENABLE TTY KEYBOARD INTERRUPTS?
 9598  061014  001003                            BNE     8$              ;;BRANCH IF NO
 9599  061016  012777  000100  120120          MOV     #100,@$TKS      ;;ALLOW TTY KEYBOARD INTERRUPTS
 9600  061024  000137  044710          8$:       JMP     OPERSEL         ;;CONTROL-C RESTART
 9601  061030  021627  000025          9$:       CMP     (SP),#25        ;;IS IT A CONTROL-U?
 9602  061034  001005                            BNE     10$             ;;BRANCH IF NOT
 9603  061036  104401  061416                    TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
 9604  061042  062706  000006          20$:      ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
 9605  061046  000737                            BR      19$             ;;LET'S TRY IT AGAIN
 9606  061050  021627  000015          10$:      CMP     (SP),#15        ;;IS IT A <CR>?
 9607  061054  001022                            BNE     16$             ;;BRANCH IF NO
 9608  061056  005766  000004                    TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
 9609  061062  001403                            BEQ     11$             ;;BRANCH IF YES
 9610  061064  016677  000002  120046          MOV     2(SP),@SWR      ;;SAVE NEW SWR
 9611  061072  062706  000006          11$:      ADD     #6,SP           ;;CLEAR UP STACK
 9612  061076  104401  001223          14$:      TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
 9613  061102  123727  001135  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
 9614  061110  001003                            BNE     15$             ;;BRANCH IF NOT
 9615  061112  012777  000100  120024          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
 9616  061120  000002                  15$:      RTI                     ;;RETURN
 9617  061122  004737  060262          16$:      JSR     PC,$TYPEC       ;;ECHO CHAR
 9618  061126  021627  000060                    CMP     (SP),#60        ;;CHAR < 0?
 9619  061132  002420                            BLT     18$             ;;BRANCH IF YES
 9620  061134  021627  000067                    CMP     (SP),#67        ;;CHAR > 7?
 9621  061140  003015                            BGT     18$             ;;BRANCH IF YES
 9622  061142  042726  000060                    BIC     #60,(SP)+       ;;STRIP-OFF ASCII
 9623  061146  005766  000002                    TST     2(SP)           ;;IS THIS THE FIRST CHAR
 9624  061152  001403                            BEQ     17$             ;;BRANCH IF YES
 9625  061154  006316                            ASL     (SP)            ;;NO, SHIFT PRESENT
 9626  061156  006316                            ASL     (SP)            ;;   CHAR OVER TO MAKE
 9627  061160  006316                            ASL     (SP)            ;;   ROOM FOR NEW ONE.
```

```
 9628  061162  005266  000002          17$:    INC     2(SP)               ;;KEEP COUNT OF CHAR
 9629  061166  056616  177776                  BIS     -2(SP),(SP)         ;;SET IN NEW CHAR
 9630  061172  000667                          BR      7$                  ;;GET THE NEXT ONE
 9631  061174  104401  001222          18$:    TYPE    ,$QUES              ;;TYPE ?<CR><LF>
 9632  061200  000720                          BR      20$                 ;;SIMULATE CONTROL-U
 9633  061202  011646                $RDCHR: MOV     (SP),-(SP)          ;;PUSH DOWN THE PC AND
 9634  061204  016666  000004  000002        MOV     4(SP),2(SP)         ;;THE PS
 9635  061212  005066  000004                CLR     4(SP)               ;;GET READY FOR A CHARACTER
 9636  061216  005046                        CLR     -(SP)               ;;PUT NEW PS ON STACK
 9637  061220  012746  061226                MOV     #64$,-(SP)          ;;PUT NEW PC ON  STACK
 9638  061224  000002                        RTI                         ;;POP NEW PC AND PS
 9639  061226  005737  060332          1$:     TST     $TKCNT              ;;WAIT ON A CHARACTER
 9640  061232  001775                        BEQ     1$
 9641  061234  005337  060332                DEC     $TKCNT              ;;DECREMENT THE COUNTER
 9642  061240  117766  177072  000004        MOVB    @$TKQOUT,4(SP)      ;;GET ONE CHARACTER
 9643  061246  005237  060336                INC     $TKQOUT             ;;UPDATE THE POINTER
 9644  061252  023727  060336  060351        CMP     $TKQOUT,#$TKQEND    ;;DID IT GO OFF OF THE END?
 9645  061260  001003                        BNE     2$                  ;;BRANCH IF NO
 9646  061262  012737  060340  060336        MOV     #$TKQSRT,$TKQOUT    ;;RESET THE POINTER
 9647  061270  000002          2$:     RTI                         ;;RETURN
 9648  061272  010346                $RDLIN: MOV     R3,-(SP)            ;;SAVE R3
 9649  061274  012703  061400          1$:     MOV     #$TTYIN,R3          ;;GET ADDRESS
 9650  061300  022703  061411          2$:     CMP     #$TTYIN+9.,R3       ;;BUFFER FULL?
 9651  061304  101405                        BLOS    4$                  ;;BR IF YES
 9652  061306  104410                        RDCHR                       ;;GO READ ONE CHARACTER FROM THE TTY
 9653  061310  112613                        MOVB    (SP)+,(R3)          ;;GET CHARACTER
 9654  061312  122713  000177          10$:    CMPB    #177,(R3)           ;;IS IT A RUBOUT
 9655  061316  001003                        BNE     3$                  ;;SKIP IF NOT
 9656  061320  104401  001222          4$:     TYPE    ,$QUES              ;;TYPE A '?'
 9657  061324  000763                        BR      1$                  ;;CLEAR THE BUFFER AND LOOP
 9658  061326  111337  061376          3$:     MOVB    (R3),9$             ;;ECHO THE CHARACTER
 9659  061332  104401  061376                TYPE    ,9$
 9660  061336  122723  000015                CMPB    #15,(R3)+           ;;CHECK FOR RETURN
 9661  061342  001356                        BNE     2$                  ;;LOOP IF NOT RETURN
 9662  061344  105063  177777                CLRB    -1(R3)              ;;CLEAR RETURN (THE 15)
 9663  061350  104401  001224                TYPE    ,$LF                ;;TYPE A LINE FEED
 9664  061354  012603                        MOV     (SP)+,R3            ;;RESTORE R3
 9665  061356  011646                        MOV     (SP),-(SP)          ;;ADJUST THE STACK AND PUT ADDRESS OF THE
 9666  061360  016666  000004  000002        MOV     4(SP),2(SP)         ;;      FIRST ASCII CHARACTER ON IT
 9667  061366  012766  061400  000004        MOV     #$TTYIN,4(SP)
 9668  061374  000002                        RTI                         ;;RETURN
 9669  061376     000          9$:     .BYTE   0                   ;;STORAGE FOR ASCII CHAR. TO TYPE
 9670  061377     000                  .BYTE   0                   ;;TERMINATOR
 9671  061411     136  006503  000012  $CNTLC: .ASCIZ  /^C/<15><12>    ;;CONTROL ''C''
 9672  061416  052536  005015     000  $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL 'U''
 9673  061423     136  006507  000012  $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL 'G''
 9674  061430  005015  053523  020122  $MSWR:  .ASCIZ  <15><12>/SWR = /
 9675  061436  020075     000
 9676  061441     040  047040  053505  $MNEW:  .ASCIZ  /  NEW = /
 9677  061446  036440  000040
 9678                                                        ;FROM THE TTY
```

```
 9679  061452  011646                    $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
 9680  061454  016666  000004  000002            MOV     4(SP),2(SP)     ;;INPUT NUMBER
 9681  061462  010046                            MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 9682  061464  010146                            MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 9683  061466  010246                            MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 9684  061470  104411                    1$:     RDLIN                   ;;READ AN ASCIZ LINE
 9685  061472  012600                            MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
 9686  061474  010037  061600                    MOV     R0,5$           ;;AND SAVE IT
 9687  061500  005001                            CLR     R1              ;;CLEAR DATA WORD
 9688  061502  005002                            CLR     R2
 9689  061504  112046                    2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
 9690  061506  001420                            BEQ     3$              ;;IF ZERO GET OUT
 9691  061510  122716  000060                    CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
 9692  061514  003026                            BGT     4$              ;;IS AN OCTAL DIGIT
 9693  061516  122716  000067                    CMPB    #'7,(SP)
 9694  061522  002423                            BLT     4$
 9695  061524  006301                            ASL     R1              ;;*2
 9696  061526  006102                            ROL     R2
 9697  061530  006301                            ASL     R1              ;;*4
 9698  061532  006102                            ROL     R2
 9699  061534  006301                            ASL     R1              ;;*8
 9700  061536  006102                            ROL     R2
 9701  061540  042716  177770                    BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
 9702  061544  062601                            ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
 9703  061546  000756                            BR      2$              ;;LOOP
 9704  061550  005726                    3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
 9705  061552  010166  000012                    MOV     R1,12(SP)       ;;SAVE THE RESULT
 9706  061556  010237  061610                    MOV     R2,$HIOCT
 9707  061562  012602                            MOV     (SP)+,R2        ;;POP STACK INTO R2
 9708  061564  012601                            MOV     (SP)+,R1        ;;POP STACK INTO R1
 9709  061566  012600                            MOV     (SP)+,R0        ;;POP STACK INTO R0
 9710  061570  000002                            RTI                     ;;RETURN
 9711  061572  005726                    4$:     TST     (SP)+           ;;CLEAN PARTIAL FROM STACK
 9712  061574  105010                            CLRB    (R0)            ;;SET A TERMINATOR
 9713  061576  104401                            TYPE                    ;;TYPE UP THRU THE BAD CHAR.
 9714  061600  000000                    5$:     .WORD   0
 9715  061602  104401  001222                    TYPE    ,$QUES          ;;''?'' ''CR'' & ''LF''
 9716  061606  000730                            BR      1$              ;;TRY AGAIN
 9717  061610  000000            $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
```

```
 9718  061612  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
 9719  061614  012737  177777  015124          MOV     #-1,@#ERFLG$    ;SET ERROR FLAG
 9720  061622  105237  001103          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
 9721  061626  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
 9722  061630  013777  001102  117304          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
 9723  061636  032777  002000  117274          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
 9724  061644  001402                          BEQ     1$              ;;NO - SKIP
 9725  061646  104401  001216                  TYPE    ,$BELL          ;;RING BELL
 9726  061652  005237  001112          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
 9727  061656  011637  001116                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
 9728  061662  162737  000002  001116          SUB     #2,$ERRPC
 9729  061670  117737  117222  001114          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
 9730  061676  032777  020000  117234          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
 9731  061704  001004                          BNE     20$             ;;SKIP TYPEOUTS
 9732  061706  004737  061760                  JSR     PC,$ERRTYP      ;;GO TO USER ERROR ROUTINE
 9733  061712  104401  001223                  TYPE    ,$CRLF
 9734  061716  005777  117216          2$:     TST     @SWR            ;;HALT ON ERROR
 9735  061722  100002                          BPL     3$              ;;SKIP IF CONTINUE
 9736  061724  000000                          HALT                    ;;HALT ON ERROR!
 9737  061726  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
 9738  061730  032777  001000  117202  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
 9739  061736  001402                          BEQ     4$              ;;BR IF NO
 9740  061740  013716  001110                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
 9741  061744  005737  001214          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
 9742  061750  001402                          BEQ     5$              ;;BR IF NONE
 9743  061752  013716  001214                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
 9744  061756  000002                          RTI                     ;;RETURN
```

```
 9745  061760  104401  001223                TYPE    ,$CRLF          ;;''CARRIAGE RETURN'' & 'LINE FEED''
 9746  061764  010046                        MOV     R0,-(SP)        ;;SAVE R0
 9747  061766  005000                        CLR     R0              ;;PICKUP THE ITEM INDEX
 9748  061770  153700  001114                BISB    @#$ITEMB,R0
 9749  061774  001004                        BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
 9750  061776  013746  001116                MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
 9751  062002  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 9752  062004  000445                        BR      10$             ;;GET OUT
 9753  062006  005300                1$:     DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
 9754  062010  006300                        ASL     R0              ;;      WORK FOR THE ERROR TABLE
 9755  062012  006300                        ASL     R0
 9756  062014  006300                        ASL     R0
 9757  062016  062700  001226                ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
 9758  062022  012037  062032                MOV     (R0)+,2$        ;;PICKUP ''ERROR MESSAGE'' POINTER
 9759  062026  001404                        BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
 9760  062030  104401                        TYPE                    ;;TYPE THE ''ERROR MESSAGE''
 9761  062032  000000                2$:     .WORD   0               ;;''ERROR MESSAGE'' POINTER GOES HERE
 9762  062034  104401  001223                TYPE    ,$CRLF          ;;''CARRIAGE RETURN'' & 'LINE FEED''
 9763  062040  012037  062050        3$:     MOV     (R0)+,4$        ;;PICKUP ''DATA HEADER'' POINTER
 9764  062044  001404                        BEQ     5$              ;;SKIP TYPEOUT IF 0
 9765  062046  104401                        TYPE                    ;;TYPE THE ''DATA HEADER''
 9766  062050  000000                4$:     .WORD   0               ;;''DATA HEADER'' POINTER GOES HERE
 9767  062052  104401  001223                TYPE    ,$CRLF          ;;''CARRIAGE RETURN'' & 'LINE FEED''
 9768  062056  010146                5$:     MOV     R1,-(SP)        ;;SAVE R1
 9769  062060  012001                        MOV     (R0)+,R1        ;;PICKUP ''DATA TABLE'' POINTER
 9770  062062  001415                        BEQ     9$              ;;BR IF NO DATA TO BE TYPED
 9771  062064  012000                        MOV     (R0)+,R0        ;;PICKUP ''DATA FORMAT'' POINTER
 9772  062066  105720                6$:     TSTB    (R0)+           ;;''OCTAL'' OR ''DECIMAL''
 9773  062070  001003                        BNE     7$              ;;BR IF DECIMAL
 9774  062072  013146                        MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
 9775  062074  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 9776  062076  000402                        BR      8$
 9777  062100  013146                7$:     MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
 9778  062102  104405                        TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
 9779  062104  005711                8$:     TST     (R1)            ;;IS THERE ANOTHER NUMBER?
 9780  062106  001403                        BEQ     9$              ;;BR IF NO
 9781  062110  104401  062130                TYPE    ,11$            ;;TYPE TWO(2) SPACES
 9782  062114  000764                        BR      6$              ;;LOOP
 9783  062116  012601                9$:     MOV     (SP)+,R1        ;;RESTORE R1
 9784  062120  012600                10$:    MOV     (SP)+,R0        ;;RESTORE R0
 9785  062122  104401  001223                TYPE    ,$CRLF          ;;''CARRIAGE RETURN'' & 'LINE FEED''
 9786  062126  000207                        RTS     PC              ;;RETURN
 9787  062130  020040    000         11$:    .ASCIZ  /  /            ;;TWO(2) SPACES
```

```
 9788  062134  017646  000000          $TYPOS: MOV     @(SP),-(SP)       ;;PICKUP THE MODE
 9789  062140  116637  000001  062357          MOVB    1(SP),$OFILL      ;;LOAD ZERO FILL SWITCH
 9790  062146  112637  062361                  MOVB    (SP)+,$OMODE+1    ;;NUMBER OF DIGITS TO TYPE
 9791  062152  062716  000002                  ADD     #2,(SP)           ;;ADJUST RETURN ADDRESS
 9792  062156  000406                          BR      $TYPON
 9793  062160  112737  000001  062357  $TYPOC: MOVB    #1,$OFILL         ;;SET THE ZERO FILL SWITCH
 9794  062166  112737  000006  062361          MOVB    #6,$OMODE+1       ;;SET FOR SIX(6) DIGITS
 9795  062174  112737  000005  062356  $TYPON: MOVB    #5,$OCNT          ;;SET THE ITERATION COUNT
 9796  062202  010346                          MOV     R3,-(SP)          ;;SAVE R3
 9797  062204  010446                          MOV     R4,-(SP)          ;;SAVE R4
 9798  062206  010546                          MOV     R5,-(SP)          ;;SAVE R5
 9799  062210  113704  062361                  MOVB    $OMODE+1,R4       ;;GET THE NUMBER OF DIGITS TO TYPE
 9800  062214  005404                          NEG     R4
 9801  062216  062704  000006                  ADD     #6,R4             ;;SUBTRACT IT FOR MAX. ALLOWED
 9802  062222  110437  062360                  MOVB    R4,$OMODE         ;;SAVE IT FOR USE
 9803  062226  113704  062357                  MOVB    $OFILL,R4         ;;GET THE ZERO FILL SWITCH
 9804  062232  016605  000012                  MOV     12(SP),R5         ;;PICKUP THE INPUT NUMBER
 9805  062236  005003                          CLR     R3                ;;CLEAR THE OUTPUT WORD
 9806  062240  006105                  1$:     ROL     R5                ;;ROTATE MSB INTO "C"
 9807  062242  000404                          BR      3$                ;;GO DO MSB
 9808  062244  006105                  2$:     ROL     R5                ;;FORM THIS DIGIT
 9809  062246  006105                          ROL     R5
 9810  062250  006105                          ROL     R5
 9811  062252  010503                          MOV     R5,R3
 9812  062254  006103                  3$:     ROL     R3                ;;GET LSB OF THIS DIGIT
 9813  062256  105337  062360                  DECB    $OMODE            ;;TYPE THIS DIGIT?
 9814  062262  100016                          BPL     7$                ;;BR IF NO
 9815  062264  042703  177770                  BIC     #177770,R3        ;;GET RID OF JUNK
 9816  062270  001002                          BNE     4$                ;;TEST FOR 0
 9817  062272  005704                          TST     R4                ;;SUPPRESS THIS 0?
 9818  062274  001403                          BEQ     5$                ;;BR IF YES
 9819  062276  005204                  4$:     INC     R4                ;;DON'T SUPPRESS ANYMORE 0'S
 9820  062300  052703  000060                  BIS     #'0,R3            ;;MAKE THIS DIGIT ASCII
 9821  062304  052703  000040          5$:     BIS     #' ,R3            ;;MAKE ASCII IF NOT ALREADY
 9822  062310  110337  062354                  MOVB    R3,8$             ;;SAVE FOR TYPING
 9823  062314  104401  062354                  TYPE    ,8$               ;;GO TYPE THIS DIGIT
 9824  062320  105337  062356          7$:     DECB    $OCNT             ;;COUNT BY 1
 9825  062324  003347                          BGT     2$                ;;BR IF MORE TO DO
 9826  062326  002402                          BLT     6$                ;;BR IF DONE
 9827  062330  005204                          INC     R4                ;;INSURE LAST DIGIT ISN'T A BLANK
 9828  062332  000744                          BR      2$                ;;GO DO THE LAST DIGIT
 9829  062334  012605                  6$:     MOV     (SP)+,R5          ;;RESTORE R5
 9830  062336  012604                          MOV     (SP)+,R4          ;;RESTORE R4
 9831  062340  012603                          MOV     (SP)+,R3          ;;RESTORE R3
 9832  062342  016666  000002  000004          MOV     2(SP),4(SP)       ;;SET THE STACK FOR RETURNING
 9833  062350  012616                          MOV     (SP)+,(SP)
 9834  062352  000002                          RTI                       ;;RETURN
 9835  062354     000                  8$:     .BYTE   0                 ;;STORAGE FOR ASCII DIGIT
 9836  062355     000                          .BYTE   0                 ;;TERMINATOR FOR TYPE ROUTINE
 9837  062356     000          $OCNT:  .BYTE   0                 ;;OCTAL DIGIT COUNTER
 9838  062357     000          $OFILL: .BYTE   0                 ;;ZERO FILL SWITCH
 9839  062360  000000          $OMODE: .WORD   0                 ;;NUMBER OF DIGITS TO TYPE
```

```
 9840  062362  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
 9841  062364  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
 9842  062370  005740                          TST     -(R0)           ;;BACKUP BY 2
 9843  062372  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
 9844  062374  006300                          ASL     R0              ;;POSITION FOR INDEXING
 9845  062376  016000  062416                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
 9846  062402  000200                          RTS     R0              ;;GO TO ROUTINE
 9847  062404  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
 9848  062406  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
 9849  062414  000002                          RTI                     ;;RESTORE THE PSW
 9850  062416  062404                  $TRPAD: .WORD   $TRAP2
 9851  062420  060112                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
 9852  062422  062160                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
 9853  062424  062134                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
 9854  062426  062174                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
 9855  062430  057666                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
 9856  062432  060730                          $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING
 9857  062434  060640                          $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
 9858  062436  061202                          $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
 9859  062440  061272                          $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
 9860  062442  061452                          $RDOCT  ;;CALL=RDOCT    TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
 9861  062444  045752                          T.SCOP  ;;CALL=SCOP1    TRAP+13(104413) MY LOCAL SCOPES
 9862  062446  046024                          CHECKT  ;;CALL=CHECKD   TRAP+14(104414) CHECK DVA,RDY,DPR,DRY
 9863  062450  046342                          WAIT.T  ;;CALL=WAT      TRAP+15(104415) WAIT LOOP
```

```
 9864  062452  012737  062616  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
 9865  062460  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
 9866  062466  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 9867  062470  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 9868  062472  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 9869  062474  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 9870  062476  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
 9871  062500  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 9872  062502  017746  116432                  MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
 9873  062506  010637  062622                  MOV     SP,$SAVR6       ;;SAVE SP
 9874  062512  012737  062524  000024          MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
 9875  062520  000000                          HALT
 9876  062522  000776                          BR      .-2             ;;HANG UP
 9877  062524  012737  062616  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
 9878  062532  013706  062622                  MOV     $SAVR6,SP       ;;GET SP
 9879  062536  005037  062622                  CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
 9880  062542  005237  062622          1$:     INC     $SAVR6          ;;WAIT FOR THE INC
 9881  062546  001375                          BNE     1$              ;;OF  WORD
 9882  062550  012677  116364                  MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
 9883  062554  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
 9884  062556  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
 9885  062560  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
 9886  062562  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
 9887  062564  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
 9888  062566  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
 9889  062570  012737  062452  000024          MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
 9890  062576  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
 9891  062604  104401                          TYPE                    ;REPORT THE POWER FAILURE
 9892  062606  062624          $PWRMG: .WORD   $POWER                  ;;POWER FAIL MESSAGE POINTER
 9893  062610  012716                  MOV     (PC)+,(SP)              ;;RESTART AT BEGIN
 9894  062612  017356          $PWRAD: .WORD   BEGIN                   ;;RESTART ADDRESS
 9895  062614  000002                  RTI
 9896  062616  000000          $ILLUP: HALT                            ;;THE POWER UP SEQUENCE WAS STARTED
 9897  062620  000776                  BR      .-2                     ;; BEFORE THE POWER DOWN WAS COMPLETE
 9898  062622  000000          $SAVR6: 0                               ;;PUT THE SP HERE
 9899  062624  005015  047520  042527  $POWER: .ASCIZ  <15><12>'POWER'
 9900  062632  000122
 9901
 9902          000001                  .END
```

```
ACL    = 000040            2491#
ACU    = 100000            2444#
ADDMOD   000204             506#  2797
ADTIMO   052400            8221   8258#
AOE    = 001000            2360#
AS       015050            1944   1947   2569#  4330*  4342   4504*  4515   4640*  4650
ATA    = 100000            2348#  4327   4501   4637
ATABLE   017336            2669#  3012
ATTENT   015134            2604#  2974*  3012*  4330   4504   4640
AT0    = 000001            2382#
AT1    = 000002            2383#
AT2    = 000004            2384#
AT3    = 000010            2385#
AT4    = 000020            2386#
AT5    = 000040            2387#
AT6    = 000100            2388#
AT7    = 000200            2389#
A16    = 000400            2323#  4181
A17    = 001000            2324#  4181
BA       015026            1940   2558#  4323*  4489*  4494*  4626*  4630*
BAI    = 000010            2298#  4183   7827
BASECH   051414             506   8210#  8261
BEGIN    017356             504   2679#  8257   9894
BEGIN2   017346             508   2677#
BITST    045536            2831   7136#
BIT0   = 000001             494#  7571   7606   7608
BIT00  = 000001             494#
BIT01  = 000002             494#
BIT02  = 000004             494#
BIT03  = 000010             494#
BIT04  = 000020             494#
BIT05  = 000040             494#
BIT06  = 000100             494#
BIT07  = 000200             494#
BIT08  = 000400             494#  9382
BIT09  = 001000             494#  9390   9738
BIT1   = 000002             494#  7585
BIT10  = 002000             494#  9723
BIT11  = 004000             494#  9397
BIT12  = 010000             494#
BIT13  = 020000             494#  7601   7603   9730
BIT14  = 040000             494#  7578   9371
BIT15  = 100000             494#  7573   7580   7587   7596   7598
BIT2   = 000004             494#
BIT3   = 000010             494#
BIT4   = 000020             494#
BIT5   = 000040             494#
BIT6   = 000100             494#
BIT7   = 000200             494#
BIT8   = 000400             494#
BIT9   = 001000             494#
BLT1     045564            7142#  7147
BLT2     045606            7142   7143   7149#
BLT3     045616            7150   7151#
BPTVEC=  000014             494#
CA       015044            2567#
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
CC       015070          2577#
CHECKD=  104414          9862#
CHECKE   046214          4795  4924  5091  5256  5428  5565  5733  5893  6067  6194  6355  6516  6688
                         6865  7291#
CHECKT   046024          4177  4249  4409  4567  4786  4890  5044  5213  5376  5531  5686  5851  6013
                         6161  6308  6473  6636  6811  7236#  7425  7519  7685  9862
CKSWR =  104407          9366  9718  9737  9857#
CLAREA   045706          4149  4932  4939  5574  5581  6203  6210  7182#
CLDISK   045770          2973  3071  3094  3149  4131  4170  4241  4401  4559  4677  4845  5032  5201
                         5365  5486  5674  5840  6001  6116  6296  6461  6625  6799  6914  6943  6967
                         7218#  7414  7507  7684  7866  8270
CLR    = 000040          2300#  3169  3208  3229  3256  3280  3308  3333  3373  3414  3473  3514  3550
                         3607  3651  3707  3738  3765  3794  3822  3834  3852  3864  3886  3913  3949
                         3959  3975  3985  4005  4033  4062  4098  7223
COMHD    052514          4185  4273  4432  5050  5219  5381  5692  5856  6019  6314  6479  6641  6817
                         7430  7523  7704  8361#
COMPA    053566          8513*  8563  8642#  8784*
COMPAR   046664          3128  4346  4363  4521  4655  4800  4950  5115  5284  5457  5592  5754  5919
                         6093  6221  6375  6543  6718  6892  7445#  7725  7751
COMWHD   055636          4590  4790  4899  5540  6170  8899#
COUNTD   054234          8778#  8782*  8798
CPHALT   005123          1482#  4178  4250  4410  4568  4787  4796  4891  4925  5045  5092  5214  5257
                         5377  5429  5532  5566  5687  5734  5852  5894  6014  6068  6162  6195  6309
                         6356  6474  6517  6637  6689  6812  6866  7426  7520  7686
CR     = 000015           494#  9507  9516
CRC      047176          4163  4235  4395  4553  4730  4769  4869  5023  5183  5348  5510  5665  5822
                         5983  6140  6287  6437  6606  6782  7408  7501  7556#  7657
CRLF   = 000200           494#  9482  9516
CSF    = 000002          2431#
CSU    = 000010          2433#
CS1      015032          1929  1932  1940  1947  1954  1964  1967  1971  1987  2562#  4324*  4498*  4632*
                         6932  7239  7243  7250  7294  7302
CS2      015030          1929  1932  1940  1954  1987  2559#  4490*  4495*  4496*  4633*  4634*
CYL      052654          4156*  4164  4228*  4236  4347  4388*  4396  5015*  5024  5175*  5184  5340*  5349
                         5657*  5666  5814*  5823  5975*  5984  6279*  6288  6429*  6438  6598*  6607  6774*
                         6783  7401*  7409  7493*  7502  7651*  7658  8391#
DAREAD   052730          8440  8452#
DATENV   050502          1983  4855*  5009*  5169*  5334*  5496*  5651*  5808*  5969*  6126*  6273*  6423*  6592*
                         6768*  7939#  8670*  8687*  8735*  8751*
DAWORD   052734          5021*  5181*  5346*  5663*  5820*  5981*  6285*  6435*  6604*  6780*  7406*  7498*  7656*
                         8453#
DB       015022          1940  2556#
DCK    = 100000          2366#  5237  5399  5419  5874  6037  6057  6497  6659  6679  6835  6855  7780
                         7785  7801  7804
DCL    = 000100          2492#
DCLEAR   015156          2631#
DDISP  = 177570           494#  542  2705
DENVL  = 000200          2376#  4500
DE1    = 000040          2338#
DFF20  = 000002          2334#
DF1      012516           582  701  708  1994#
DF11     012534           690  2001#
DF14     012543           721  2004#
DF15     012553           730  2007#
DF16     012556           741  2008#
DF17     012563           754  2010#
```

```
DF2     012523  1996#
DF20    012572   767  2013#
DF21    012601   778  2017#
DF22    012606   795   809  2019#
DF24    012612   824   837  2021#
DF26    012621   850  2024#
DF27    012631   865  2027#
DF3     012527   670   677  1998#
DF30    012637   878  2029#
DF31    012645   633   641  1008  1016  2032#
DF32    012655   617   658  2035#
DF33    012666   600  2038#
DF34    012676   896  2041#
DF35    012705   915   987  2044#
DF36    012715   930   947  2047#
DF37    012723   970  2049#
DF40    012734  1033  2052#
DF42    014730  1039  1046  1073  2272#
DF44    014733  1053  1059  1066  1080  1087  1094  1101  1115  1122  1129  1136  1143  1150
                1157  1164  1171  1178  1199  2273#
DF5   = 000001  2333#
DF54    014740  1108  1185  1192  2275#
DH1     005451   577   696  1523#
DH11    005701   682  1551#
DH14    006056   713  1570#
DH15    006255   727  1592#
DH16    006305   735  1597#
DH17    006443   746  1614#
DH2     005567   664  1537#
DH20    006620   759  1633#
DH21    006774   772  1651#
DH22    007110   789  1665#
DH24    007204   817   830  1676#
DH26    007362   841  1695#
DH27    007560   857  1717#
DH30    007716   871  1734#
DH31    010050   623   998  1750#
DH32    010246   606   647  1772#
DH33    010463   591  1796#
DH34    010662   886  1818#
DH35    011035   906   977  1836#
DH36    011232   921   938  1858#
DH37    011370   958  1875#
DH40    011607  1025  1900#
DH42    014410  1037  1044  1071  2228#
DH44    014471  1051  1057  1064  1078  1085  1092  1099  1113  1120  1127  1134  1141  1148
                1155  1162  1169  1176  1197  2238#
DH54    014603  1106  1183  1190  2252#
DIGB  = 000004  2335#
DISK    054572  1974  4150  4198  4802  4952  4985  5144  5193*  5310  5357*  5594  5627  5783
                5832*  5945  5993*  6223  6250  6398  6568  6616*  6618*  6744  6792*  6793*  7638
                7753  8181  8195*  8196*  8804  8860#  9337
DISPLA  001142   542#  2705*  2712*  9411*  9722*
DISPRE  000174   495#  2712
DLT   = 100000  2310#
DL64  = 000020  2337#
```

```
DMD    = 000001   2370# 3096  3150  4132  4299  4458  4500  7834  7860  8124  8370  8518  8524
                  8650  8663  8666  8680  8683  8722  8732  8748  8908  9067  9070  9079  9090
                  9248
DPR    = 000400   2341# 7264  7267  7314
DRY    = 000200   2340# 7264  7272  7309
DST      015036   1951  2564# 4329* 4503* 4639*
DSWR   = 177570    494#  541  2704
DS1      015054   1929  1932  1944  1947  1954  1964  1967  1971  1987  2571# 4327* 4501* 4637*
                  7262  7267  7272  7309  7314
DT       015056   2572#
DTAGAP   055576   8863#
DTE    = 010000   2363# 4325  4499  4635
DTSY   = 001000   2378# 8543  8616  8620
DT1      011764    581   700   707  1922#
DT11     012026    689  1929#
DT14     012046    720  1932#
DT15     012070    729  1935#
DT16     012100    740  1937#
DT17     012116    753  1940#
DT2      012000   1924#
DT20     012136    766  1944#
DT21     012156    777  1947#
DT22     012172    794   808  1949#
DT24     012204    823   836  1951#
DT26     012224    849  1954#
DT27     012246    864  1957#
DT3      012012    669   676  1926#
DT30     012264    877  1960#
DT31     012302    632   640  1007  1015  1964#
DT32     012324    616   657  1967#
DT33     012350    599  1971#
DT34     012372    894  1974#
DT35     012412    913   985  1977#
DT36     012434    928   945  1980#
DT37     012452    968  1983#
DT40     012476   1032  1987#
DT42     014672   1038  1045  1072  2266#
DT44     014702   1052  1058  1065  1079  1086  1093  1100  1114  1121  1128  1135  1142  1149
                  1156  1163  1170  1177  1198  2268#
DT54     014716   1107  1184  1191  2270#
DVA    = 004000   2326# 6932  7239  7243  7302
ECDATA   050462   7920# 7968  8125* 8187* 8189* 8660* 8662* 8677* 8679* 8725* 8729* 8741* 8745*
ECH    = 000100   2357# 5419  6057  6679  6855
ECI    = 004000   2457# 4246  4564  7421  7515  7701
ECORR    051106   5250  5412  5887  6050  6510  6672  6848  8119#
ECTEST   050520   7959# 8153  8190  8671  8688  8736  8752
EC1      015062   1977  1980  1983  2574#
EC2      015064   1977  1980  1983  2575# 4480* 5076  5246  5408  5718  5883  6046  6340  6506
                  6668  6844
EMTVEC=  000030    494# 2691* 2692* 2780* 2815*
EM1      002136    576   855  1210#
EM11     002307    681   694   839  1229#
EM14     002342    712  1234#
EM15     002362    725  1237#
EM16     002432    734  1244#
EM17     002453    745  1247#
```

```
EM2     002221   589    604    621    1219#
EM20    002477   758   1251#
EM21    002520   771   1254#
EM22    002542   782   1257#
EM24    003143   814   1304#
EM25    003236   827   1314#
EM30    003276   870   1320#
EM31    003420   883   1334#
EM32    003543   900   1349#
EM33    004032   920   1382#
EM34    004104   933   1389#
EM35    004177   952   1399#
EM36    004476   975   1434#
EM37    004664   995   1454#
EM40    004752  1022   1463#
EM42    012744  1036   2057#
EM43    013027  1043   2067#
EM44    013073  1050   2075#
EM45    013130  1056   2081#
EM46    013157  1063   2086#
EM47    013206  1070   2091#
EM50    013243  1077   2097#
EM51    013301  1084   2104#
EM52    013322  1091   2108#
EM53    013362  1098   2115#
EM54    013422  1105   2122#
EM55    013461  1112   2129#
EM56    013474  1119   2132#
EM57    013511  1126   2136#
EM6     002250   645    663    1223#
EM60    013540  1133   2141#
EM61    013627  1140   2152#
EM62    013677  1147   2160#
EM63    013754  1154   2169#
EM64    014032  1161   2178#
EM65    014066  1168   2184#
EM66    014150  1175   2194#
EM67    014222  1182   2202#
EM70    014307  1189   2212#
EM71    014361  1196   2221#
ERCRC   056032  8929*  8983#  9165*
ERFLG$  015124  2593#  4184*  4195   4206   4269*  4428*  4586*  4784*  4792   4896*  4907   4945*  5049*
                5065   5108*  5218*  5234   5278*  5380*  5396   5451*  5537*  5548   5587*  5691*  5707
                5749*  5855*  5871   5914*  6018*  6034   6088*  6167*  6178   6216*  6313*  6329   6370*
                6478*  6494   6538*  6640*  6656   6713*  6816*  6832   6887*  7466   7703*  7714   8436
                9020   9719*
ERHDGP  056034  8930*  8993#  9178*
ERHEAD  056030  8928*  8969#  9151*
ERPOS   051104  8115#  8121*  8138
ERR     =040000 2347#  4327   4501   4637
ERRVEC= 000004   494#  2702   2703*  2713*  2782*  2789*  2817*  9374   9375*  9377*  9380*
ERSTAR  052466  8270#
ERUNIT  052464  8269#  8271
ERWORD  052774  1924   1926   1964   1967   1971   3135   4200*  4205*  4371   4529   4663   7459*  7465*
                8504#  8545*  8570*  8574*  8598*  8622*  9099*  9106*  9110*  9132*  9141*  9148*  9164*
                9169*  9175*  9188*
```

```
ER1     015034          1929  1932  1944  1954  1964  1967  1971  1977  1987  2563#  4325*  4499*  4635*
                        5068  5237  5399  5419  5710  5874  6037  6057  6332  6497   6659   6679   6835
                        6855
ER2     015040          1944  2565#
ER3     015046          1944  2568#
EXT1  = 000001          2411#
EXT10 = 000010          2414#
EXT2  = 000002          2412#
EXT20 = 000020          2415#
EXT4  = 000004          2413#
EXT40 = 000040          2416#
E00     024202          3174  3187#
E11     025610          3458  3466#
E12     025736          3496  3503#
E44     030442          4078  4086#
FEN   = 000200          2437#
FER   = 000020          2355#
FILLEC  051260          4998  5157  5323  5640  5796  5958  6262  6411  6581  6757  8171#
FIRST   017332          2657#  2720  2750*
FMT22 = 010000          2458#  4156  4175  4228  4246  4388  4406  4547  4564  4708  4726  4763  4781
                        4863  4879  4897  5015  5038  5095  5175  5207  5260  5340  5371  5432  5504
                        5520  5538  5657  5680  5737  5814  5846  5897  5975  6007  6071  6134  6150
                        6168  6279  6302  6359  6429  6467  6520  6598  6631  6692  6774  6805  6869
                        7401  7421  7493  7515  7651  7701  7828
FNWORD  056052          4552*  4768*  4868*  5509*  6139*  9024#
FORMAT  054236          8779#  8797
FOUT    056056          9021  9027#
FSYNER  056026          8927*  8955#  9136*
FUTABL  015150          2627#
GCRC    056022          4555  4771  4871  5512  6142  8939#  9158  9163
GECC1   050464          1974  1977  1983  4853*  4914  4937  5007*  5074  5167*  5244  5332*  5406  5494*
                        5555  5579  5649*  5716  5806*  5881  5967*  6044  6124*  6184  6208  6271*  6338
                        6421*  6504  6590*  6666  6766*  6842  7924#  7966  8052*  8179*  8195
GECC2   050466          1974  1977  1983  4854*  4918  4938  5008*  5168*  5333*  5495*  5559  5580  5650*
                        5807*  5968*  6125*  6188  6209  6272*  6422*  6591*  6767*  7927#  7967  8053*  8180*
                        8196
GNS   = ****** U         495  2729  2732  2735  2737  2739  2741  2743  2745  2747  2749  2768  2797
                        2810  2815  2866  2891  2893  2895  2897  2940  2944  2948  3015  3019  3023
                        3054  3058  3062  3077  3079  3081  3086  3428  3430  3432  3434  6972  6976
                        7066  7070  7075  7077  7079  7084  7086  8213  8217  8231  8235  8239  8243
                        8247  8249  8251  8253  8257  8261  9356  9851  9852  9853  9854  9855  9856
                        9857  9858  9859  9860  9861  9862  9863
GO    = 000001          2320#  3107  7835  7861  8373  8911
GRV   = 000010          2336#
GTSWR = 104406          2756  9856#
HADTMP  050506          4852*  5006*  5166*  5331*  5493*  5648*  5805*  5966*  6123*  6270*  6420*  6589*  6765*
                        7949#  8140
HARDER  050500          4852  5006  5166  5331  5493  5648  5805  5966  6123  6270  6420  6589  6765
                        7936#
HCCRCE  047600          7682#
HCE   = 000200          2358#  7770  7774  7780  7785
HCI   = 002000          2456#
HCRC  = 000400          2359#  7792  7795  7801  7804
HDESYN  056036          8931*  9006#  9191*
HDWSYN  054570          8858#
HEADER  054544          8855#
```

```
HEDGAP  052670          8387* 8409# 8577*
HEDSYN  052672          8388* 8422# 8599* 8624*
HEGAP   054556          8568  8857#
HT    = 000011           494# 9480  9516
IAE   = 002000          2361#
IE    = 000100          2321# 3719  3749  4046  4074  4081  6919  6948
ILF   = 000001          2351#
ILLEGL  015212          2645#
ILR   = 000002          2352#
IOTVEC= 000020           494# 2689* 2690*
IR    = 000100          2301# 4490  4495  4633
IXE   = 004000          2441#
I00     024156          3178# 3184
I02     024262          3218#
I03     024356          3241#
I04     024456          3267#
I05     024556          3293#
I06     024656          3320#
I07     024756          3346#
I14     026130          3560#
I15     026332          3619#
I16     026474          3665#
I24     027060          3772#
KEY1    052660          4159* 4232* 4392* 5019* 5179* 5344* 5661* 5818* 5979* 6283* 6433* 6602* 6778*
                        7404* 7496* 7654* 8393#
KEY2    052662          4160* 4233* 4393* 5020* 5180* 5345* 5662* 5819* 5980* 6284* 6434* 6603* 6779*
                        7405* 7497* 7655* 8394#
KIPAR0= 172340           520#
KIPAR1= 172342           520#
KIPAR2= 172344           520#
KIPAR3= 172346           520#
KIPAR4= 172350           520#
KIPAR5= 172352           520#
KIPAR6= 172354           520#
KIPAR7= 172356           520#
KIPDR0= 172300           520#
KIPDR1= 172302           520#
KIPDR2= 172304           520#
KIPDR3= 172306           520#
KIPDR4= 172310           520#
KIPDR5= 172312           520#
KIPDR6= 172314           520#
KIPDR7= 172316           520#
LA      015066         2576# 4328* 4502* 4638*
LAD     045750         7150* 7200# 7204
LERR    045532         7133# 7139* 7162
LF    = 000012          494# 9511  9516
LST   = 002000         2343#
LST14A  026160         3556  3570#
LST15A  026362         3613  3632#
LST16A  026532         3657  3682#
L00     024152         3176#
L02     024260         3216#
L03     024352         3237#
L04     024450         3264#
L05     024546         3288#
```

```
L06      024650          3316#
L07      024746          3341#
L10      025102          3389#
L11      025546          3450#
L12      025674          3488#
L13      026006          3523#
L14      026124          3558#   3568
L15      026324          3615#   3628
L16      026464          3659#   3678
L21      026656          3719#
L22      026742          3744#
L30      027170          3802#
L31      027270          3830#
L34      027374          3860#
L35      027510          3893#
L36      027624          3924#
L40      027740          3957#
L41      030042          3983#
L42      030152          4015#
L43      030266          4043#
L44      030376          4071#
L45      030520          4107#
MAKECY   050342          3151    4133    7852#
MASK     045530          7132#   7136*   7152*   7153    7154*   7157*   7158    7159*
MCLK   = 000002          2371#   4300    4301    4303    4304    4459    4460    4462    4463    5087    5088    5729    5730
                         6351    6352    8151    8152    8159    8160    8520    8521    8524    8525    8528    8529    8655
                         8672    8722    8738    9070    9081    9274    9276    9277    9278    9279    9294    9295    9296
                         9297
MCPE   = 020000          2327#
MHS    = 001000          2439#
MID      050332          7830*   7837#
MIDDLE   050230          7819#
MINX   = 000004          2372#   8371    8372    8909    8910    9256    9257
MMVEC  = 000250           520#
MOL    = 010000          2345#   3072    3082
MPE    = 000400          2303#
MR       015052          1980    2570#   4326*   4500*   4636*
MRD    = 000020          2374#   8653    8658    8666    8675    8683
MSE    = 000020          2434#
MSTCK  = 000010          2373#   4300    4301    4313    4314    4459    4460    4472    4473    4614    4615    8519    8521
                         8524    8525    8655    8722    9069    9070    9249    9250    9251    9252    9256    9257    9273
                         9276    9277    9294    9295    9307    9308
MWR    = 000040          2375#   8723    8739    9072    9082
MXF    = 001000          2304#   3770    3772    3776
NCODE    050472          4851    5005    5165    5330    5492    5647    5804    5965    6122    6269    6419    6588    6764
                         7933#
NCOUNT   050474          4851*   5005*   5165*   5330*   5492*   5647*   5804*   5965*   6122*   6269*   6419*   6588*   6764*
                         7934#   8131*
NED    = 010000          2307#   3446    3455    3493    3534    4119    8273
NEM    = 004000          2306#
NHS    = 002000          2440#
NOPERA   015150          2628#
NOSYNC   052770          7090*   8441    8501#   8586    8614    9367*
NOUNIT   015114          2586#   2911*   2953*   2960    2996*   6990*
NOWORD   052722          4161*   8448#
NUNIT    015116          2588#   2960*   2961*
```

```
N11      025576            3443    3460#
N12      025724            3482    3498#
N14      026152            3564    3567#
N36      027664            3923    3937#
OCYL   = 100000            2494#
OF       015042            2566#
OFREV  = 000200            2455#
OFSETC   015202            2641#
OF100  = 000004            2450#
OF200  = 000010            2451#
OF25   = 000001            2448#
OF400  = 000020            2452#
OF50   = 000002            2449#
OF800  = 000040            2453#
OPERSE   044710            7062#   9534    9600
OPI    = 020000            2364#
OR     = 000200            2302#   4490    4496    4634
OUT      052740            8437    8445    8450    8456#
O00      024204            3185    3189#
O10      025120            3387    3397#
O21      026674            3717    3726#
O43      030314            4041    4053#
O44      030420            4069    4080#
PAR    = 000010            2354#
PAT    = 000020            2299#
PCJSR    015132            1954    1957    1960    1967    1971    1980    2602#   7236*   7237*   7291*   7292*   7682*   7683*
                           7854*   7855*   8119*   8120*   8361*   8362*   8899*   8900*
PGE    = 002000            2305#
PIE1   = 100000            7887#   8036    8051
PIE10  = 000100            7896#
PIE11  = 000040            7897#   8003
PIE12  = 000020            7898#   8036
PIE13  = 000010            7899#
PIE14  = 000004            7900#
PIE15  = 000002            7901#
PIE16  = 000001            7902#
PIE17  = 100000            7903#
PIE18  = 040000            7904#
PIE19  = 020000            7905#
PIE2   = 040000            7888#   7994
PIE20  = 010000            7906#
PIE21  = 004000            7907#   8011
PIE22  = 002000            7908#   8042
PIE23  = 001000            7909#   8019
PIE24  = 000400            7910#   8042
PIE25  = 000200            7911#
PIE26  = 000100            7912#
PIE27  = 000040            7913#
PIE28  = 000020            7914#
PIE29  = 000010            7915#
PIE3   = 020000            7889#   8036
PIE30  = 000004            7916#
PIE31  = 000002            7917#
PIE32  = 000001            7918#   7975    7983
PIE4   = 010000            7890#
PIE5   = 004000            7891#
```

```
PIE6   = 002000          7892#
PIE7   = 001000          7893#
PIE8   = 000400          7894#
PIE9   = 000200          7895#
PIP    = 020000          2346#
PIRQ   = 177772           494#
PIRQVE= 000240            494#
PKACK    015206          2643#  3098
PLU    = 020000          2443#
POSITI   050476          1977   1983   4850*  5004*  5164*  5329*  5491*  5646*  5803*  5964*  6121*  6268*  6418*
                         6587*  6763*  7935#  8076   8129   8137*  8138   8140   8178*
PRE    = 000020          2490#
PROG   = 001000          2342#  7263
PR0    = 000000           494#
PR1    = 000040           494#
PR2    = 000100           494#
PR3    = 000140           494#
PR4    = 000200           494#
PR5    = 000240           494#
PR6    = 000300           494#
PR7    = 000340           494#
PS     = 177776           494#  2717*  4013*  4044*  4072*  6918*  6947*  6969*  7063*
PSEL   = 002000          2325#  3745   3957   3962
PSU    = 000001          2487#
PSW    = 177776           494#
PUTREG   045470          4194   4279   4438   4596   5067   5109   5236   5279   5398   5418   5452   5709   5750
                         5873   5913   6036   6054   6087   6331   6371   6496   6537   6658   6678   6712   6834
                         6854   6888   6925   6931   6957   7112#  7238   7293   7713   8073   8079   8146
PWRVEC= 000024            494#  2695*  2696*  9864*  9865*  9874*  9877*  9889*  9890*
P12      050512          7951#  8005*  8006*  8035
P22      050514          7952#  8013*  8014*  8040
P24      050516          7953#  8021*  8022*  8041
P3       050510          7950#  7996*  7997*  8034
RA       000200           504#  8253
RCOM     057174          9325#  9330*
RCYL     052760          8474#  8509*  8550
RDCHR  = 104410          9652   9858#
RDHEAD   052776          8389   8509#
RDLIN  = 104411          9684   9859#
RDOCT  = 104412          2768   7079   7086   8218   8235   9860#
RDY    = 000200          2322#  4046   6919   6932   6948   7239   7250   7294
READ     053570          8537   8542   8551   8553   8555   8557   8559   8609   8613   8647#  9339   9347
READAT   015174          2638#
READIN   015210          2644#
RECALI   015154          2630#
REDATA   057176          8452   9329#
REFOR    015176          2639#  4252   5048   5217   5379   5690   5854   6017   6312   6477   6639   6815
REGADR   045534          1922   1957   1960   2266   2268   2270   2852*  3137*  3179*  3213*  3234*  3261*  3285*
                         3313*  3338*  3382*  3438*  3478*  3531*  3555*  3612*  3656*  3712*  3741*  3769*  3799*
                         3827*  3857*  3896*  3927*  3954*  3980*  4010*  4036*  4065*  4116*  4373*  4531*  4665*
                         7134#  7138*  7773*  7784*  7794*  7805*  7871*
REGSA1   061622          2780   9720#
REINTO   016264          2653#  3103   3110*  3123   3129   4243   4323   4348   4511   4515*  4523   4646   4650*
                         4657   4933   4934   4937*  4938*  4940   4941   4951   5034   5117   5203   5286   5367
                         5459   5575   5576   5579*  5580*  5582   5583   5593   5676   5756   5842   5921   6003
                         6095   6204   6205   6208*  6209*  6211   6212   6222   6298   6377   6463   6545   6627
```

```
                    6720    6801    6894    7417    7511    7727    7752
RELEAS  015160      2632#
RESVEC= 000010       494#
RETCL   015204      2642#
RHAS    014774      2536#   2846    2866    3525    4109
RHBA    014752      2527#   3554    3611    3655    4172*   4243*   4403*   4561*   4776*   4877*   5034*   5203*   5367*
                    5518*   5676*   5842*   6003*   6148*   6298*   6463*   6627*   6801*   7417*   7511*   7698*   7825*
RHBAE   015020      2549#   2806
RHCA    014770      2534#   4176*   4248*   4408*   4566*   4782*   4898*   5042*   5211*   5375*   5539*   5684*   5850*
                    6011*   6169*   6306*   6471*   6635*   6809*   7424*   7518*   7690*   7829*   7857*
RHCC    015014      2544#   7118    7867    7871
RHCS1   014756      2529#   3098*   3107*   3171    3448    3486    3521    3711    3740    3895    3926    3953    3979
                    4009    4035    4064    4105    4276    4435    4593    4778*   7218    7822*   7835*   7859*   7861*
                    8213    8222    8225    8239    8373*   8911*
RHCS2   014754      2528#   2830    2867    2917*   3169*   3208*   3229*   3256*   3280*   3308*   3333*   3373*   3414*
                    3437    3473*   3477    3514*   3518    3530    3550*   3607*   3651*   3707*   3738*   3765*   3768
                    3794*   3798    3822*   3826    3852*   3856    3886*   3890    3913*   3917    3949*   3959*   3975*
                    3985*   4005*   4033*   4062*   4098*   4102    4115    4183*   7219    7827*
RHCS3   015016      2548#   3377
RHDB    014746      2525#   2784    8219
RHDST   014762      2531#   4174*   4244*   4404*   4562*   4777*   4888*   5037*   5206*   5370*   5529*   5679*   5845*
                    6006*   6159*   6301*   6466*   6630*   6804*   7420*   7514*   7695*   7826*   7858*
RHDS1   015000      2538#   7220
RHDT    015002      2539#   2918    2920    2924    2926    2929    2931    2944    3023    3030    3032    3035    3037
                    3040    3042    3064    3066
RHEC1   015006      2541#   8076
RHEC2   015010      2542#   3183    3380    4480    8070
RHER1   014760      2530#   2870    7221    7769    7773    7779    7784    7791    7794    7799    7805
RHER2   014764      2532#
RHER3   014772      2535#
RHLA    015012      2543#
RHMR    014776      2537#   3096*   3150*   4132*   4298    4457    5087*   5088*   5729*   5730*   6351*   6352*   7834*
                    7860*   8123    8370*   8371*   8372*   8516    8796    8908*   8909*   8910*   9066    9245
RHOF    014766      2533#   4175*   4246*   4406*   4564*   4781*   4897*   5038*   5207*   5371*   5538*   5680*   5846*
                    6007*   6168*   6302*   6467*   6631*   6805*   7421*   7515*   7701*   7828*
RHSN    015004      2540#   3019    3063    3065
RHWC    014750      2526#   3102    3116    3137    3212    3233    3260    3284    3312    3337    4171*   4242*   4257
                    4337    4373    4402*   4417    4510    4531    4560*   4575    4645    4665    4775*   4875*   5033*
                    5202*   5366*   5516*   5675*   5841*   6002*   6146*   6297*   6462*   6626*   6800*   7116    7118
                    7416*   7510*   7696*   7824*
RH70    017334      2660#   2762*   2805*   2807*   2826    3375    3762    3790    3882    4145    4485    4622
RH70CK  020720      2760#
RKEY1   052764      8476#   8511*   8554
RKEY2   052766      8477#   8512*   8556
RMR     = 000004    2353#
RNCTR1  055634      8897#   8913*   8915*
RNO     057172      9324#   9329*
RNWAT1  055714      8913#
RPTRP1  044220      6916    6930#
RPTRP2  044324      6945    6956#
RPVEC   014744      2516#   2718*   3716    3717*   3726*   4040    4041*   4053*   4068    4069*   4088*   6915    6944
                    8231    8236*   8243    9357*
RPVECT  057306      2718    9353#   9357
RSETR   052762      8475#   8510*   8552
RSYNC   052756      4725    4741    4748    8473#   8541    8546    8584    8596    8612    8623    9130    9134    9185
                    9190
```

```
RTN      021362              2810    2815#
RUNCTR   052512              8360#   8375*   8376*
RUNWAT   052572              8375#
R11      025530              3445#
R15      026354              3623    3627#
R16      026524              3673    3677#
R44      030444              4084    4088#
SAVDT    015126              2595#   3066*
SAVER    046462              3101    3115    4256    4336    4416    4509    4574    4644    7376#
SAVSN    015130              2598#   3065*
SC     = 100000              2329#   4277    4324    4436    4498    4594    4632
SCOP1  = 104413              7163    9861#
SCRC     056104              9055#   9062*
SCYL     056074              9051#   9058*   9140
SC1    = 000100              2417#
SC10   = 001000              2420#
SC2    = 000200              2418#
SC20   = 002000              2421#
SC4    = 000400              2419#
SEARCH   056722              7836    8382    8921    9238#
SECGAP   054474              4697    4840    5481    6111    8853#   9065
SECOTR   052656              4158*   4230*   4390*   5017*   5018*   5177*   5178*   5342*   5343*   5659*   5660*   5816*   5817*
                             5977*   5978*   6281*   6282*   6431*   6432*   6600*   6601*   6776*   6777*   7402*   7403*   7494*
                             7495*   7652*   7653*   8379    8392#
SECTR    056720              9236#   9238*   9246
SEECOM   015200              2640#   7859
SEGPER   056024              8926*   8944#   9102*   9113*
SELECT   015120              2590#   2677*   2679*   2764    2964    2983    6980
SELTST   022626              2885    2964#
SERCH    015162              2633#
SETCK1   031340              4276#
SETCK2   032034              4435#
SETCK3   032560              4593#
SETDSK   047452              7632#
SILOTB   054572              8859#
SKEY1    056100              9053#   9060*
SKEY2    056102              9054#   9061*
SKI    = 040000              2493#
SN       015060              2573#
SND1     020662              2724    2750#
SR0    = 177572               520#
SR1    = 177574               520#
SR2    = 177576               520#
SR3    = 172516               520#
SSECTR   056076              9052#   9059*
SSYN     052666              8386*   8401#   8547*
SS1      033140              4696#   4828
SS10     033306              4689*   4750#   4752    4819*
SS12     033336              4690*   4765#   4820*
SS13     033344              4691*   4766#   4821*
SS14     033352              4692*   4767#   4822*
SS15     033412              4693*   4777#   4823*
SS2      033576              4825#
SS3      033166              4685*   4710#   4815*
SS4      033172              4686*   4711#   4816*
SS5      033176              4687*   4712#   4817*
```

```
SS7      033224          4688* 4727# 4818*
STACK  = 001000           494# 2688  2778  2824  3091  3147  3176  3210  3231  3258  3282  3310  3335
                         3389  3435  3475  3516  3552  3609  3653  3709  3744  3767  3796  3824  3854
                         3888  3915  3951  3977  4007  4043  4071  4100  4129  4143  4223  4384  4542
                         4675  4837  4975  5134  5300  5478  5617  5773  5935  6108  6241  6390  6559
                         6734  6912  6941
START    017362          2678  2682#
STARTA   017604          2717#
STKLMT=  177774           494#
SWR      001140           541# 2686  2704* 2706  2711* 2754  2760  2793  2862  2883  4215  7202  7472
                         8061  8063  9371  9382  9384  9390  9397  9537  9568  9610* 9723  9730  9734
                         9738  9872  9882*
SWREG    000176           496# 2711  2754  9537  9568  9584
SW0    = 000001           494#
SW00   = 000001           494#
SW01   = 000002           494#
SW02   = 000004           494#
SW03   = 000010           494#
SW04   = 000020           494#
SW05   = 000040           494#
SW06   = 000100           494#
SW07   = 000200           494# 4217  7474
SW08   = 000400           494#
SW09   = 001000           494# 7202
SW1    = 000002           494#
SW10   = 002000           494#
SW11   = 004000           494#
SW12   = 010000           494# 2760
SW13   = 020000           494# 2793  2862  2883
SW14   = 040000           494#
SW15   = 100000           494#
SW2    = 000004           494#
SW3    = 000010           494#
SW4    = 000020           494#
SW5    = 000040           494#
SW6    = 000100           494# 8063
SW7    = 000200           494#
SW8    = 000400           494# 8061
SW9    = 001000           494#
S11      025524          3442# 3464
S12      025660          3481# 3502
S36      027620          3922# 3939
TAGDTE   015146          2617# 4679* 4825*
TBITVE=  000014           494#
TDF    = 000040          2435#
TESDTE   015144          2613# 4268* 4427* 4585* 7099* 8561  9124  9370*
TESTAD   044706          7060#
TIMCNT   046340          6920  6949  7329# 7339  7345
TIMOT  = 000004          3157# 3173  3174* 3189* 3386  3387* 3397*
TKVEC  = 000060           494# 9522* 9523*
TMPILL   015140          2607# 4173* 4197  4203
TOLGAP   055600          4151  8791  8864#
TOTALA   015136          2605# 2876* 2878* 3416  3440  3479  3920
TPVEC  = 000064           494#
TRAPVE=  000034           494# 2693* 2694*
TRE    = 040000          2328# 3891  3899  3918  3930  4324  4498  4632
```

```
TRK      052626            8381*   8384#
TRK1   = 004000            2422#
TRK10  = 040000            2425#
TRK2   = 010000            2423#
TRK20  = 100000            2426#
TRK4   = 020000            2424#
TRP      021322            2804    2811#
TRTVEC= 000014             494#
TSECC    015142            2609#   4849*   5003*   5163*   5328*   5490*   5645*   5802*   5963*   6120*   6267*   6417*   6586*
                           6762*   7091*   8801    9332    9342    9368*
TSECCG   050470            7095*   7930#   8054    8656    8668    8673    8685    8733    8749    8803*   8816*   9334*   9344*
                           9369*
TSTNM    017330            1922    1924    1926    1929    1932    1935    1937    1940    1944    1947    1949    1951    1954
                           1957    1960    1964    1967    1971    1974    1977    1980    1983    1987    2266    2268    2270
                           2656#   2779*   2825*   2844*   3070*   3092*   3148*   3168*   3207*   3228*   3255*   3279*   3307*
                           3332*   3372*   3413*   3472*   3513*   3549*   3606*   3650*   3706*   3737*   3761*   3789*   3821*
                           3851*   3881*   3912*   3948*   3974*   4004*   4032*   4061*   4097*   4130*   4144*   4224*   4385*
                           4543*   4676*   4839*   4977*   5136*   5302*   5480*   5619*   5775*   5937*   6110*   6242*   6391*
                           6560*   6736*   6911*   6940*   7066
TST1     021034            2765    2776#   7030
TST10    024052            3146#
TST11    024106            3166#
TST12    024212            3205#
TST13    024304            3222    3226#
TST14    024402            3248    3253#
TST15    024500            3272    3277#
TST16    024602            3300    3305#
TST17    024700            3325    3330#
TST2     021376            2822#
TST20    025002            3353    3370#
TST21    025126            3411#
TST22    025612            3463    3470#
TST23    025740            3500    3511#
TST24    026060            3537    3547#
TST25    026260            3569    3604#
TST26    026420            3629    3648#
TST27    026572            3679    3704#
TST3     021454            2828    2842#
TST30    026702            3735#
TST31    027000            3752    3759#
TST32    027110            3764    3779    3787#
TST33    027224            3792    3810    3819#
TST34    027330            3840    3849#
TST35    027434            3870    3879#
TST36    027550            3884    3902    3910#
TST37    027674            3933    3935    3946#
TST4     021526            2850    2857#
TST40    027776            3965    3972#
TST41    030100            3991    4002#
TST42    030206            4023    4030#
TST43    030322            4059#
TST44    030452            4095#
TST45    030572            4122    4128#
TST46    030626            4141#
TST47    031160            4147    4196    4218    4222#
TST5     022642            2965    2969#   6988    6998
```

```
TST50   031652          4282    4383#
TST51   032376          4442    4541#
TST52   033036          4599    4674#
TST53   033612          4793    4827    4836#
TST54   034240          4909    4956    4974#
TST55   034756          5066    5121    5133#
TST56   035502          5235    5290    5299#
TST57   036244          5397    5463    5477#
TST6    023346          3069#
TST60   036670          5550    5598    5616#
TST61   037406          5708    5760    5772#
TST62   040132          5872    5925    5934#
TST63   040674          6035    6099    6107#
TST64   041320          6180    6227    6240#
TST65   042036          6330    6381    6389#
TST66   042572          6495    6549    6558#
TST67   043350          6657    6724    6733#
TST7    023710          3073    3090#
TST70   044126          6833    6898    6909#
TST71   044240          6928    6933    6938#
TST72   044334          6954    6965#
TTNO  = 000071          2779#   2825#   2844#   3070#   3092#   3148#   3168#   3207#   3228#   3255#   3279#   3307#   3332#
                        3372#   3413#   3472#   3513#   3549#   3606#   3650#   3706#   3737#   3761#   3789#   3821#   3851#
                        3881#   3912#   3948#   3974#   4004#   4032#   4061#   4097#   4130#   4144#   4224#   4385#   4543#
                        4676#   4839#   4977#   5136#   5302#   5480#   5619#   5775#   5937#   6110#   6242#   6391#   6560#
                        6736#   6911#   6940#
TUF   = 000100          2436#
TY      052772          8503#
TYPDS = 104405          2941    2952    3016    6973    6977    6986    7020    9778    9855#
TYPE  = 104401          2727    2730    2733    2735    2737    2739    2741    2743    2745    2747    2766    2795    2808
                        2813    2864    2889    2891    2893    2895    2938    2942    2946    2950    3013    3017    3021
                        3052    3056    3060    3075    3077    3079    3084    3426    3428    3430    3432    4178    4250
                        4410    4568    4787    4796    4891    4925    5045    5092    5214    5257    5377    5429    5532
                        5566    5687    5734    5852    5894    6014    6068    6162    6195    6309    6356    6474    6517
                        6637    6689    6812    6866    6970    6974    6984    6987    7018    7021    7064    7068    7072
                        7073    7075    7077    7082    7084    7426    7520    7686    8211    8215    8229    8233    8237
                        8241    8245    8247    8249    8251    8255    8259    9354    9460    9485    9531    9541    9582
                        9583    9586    9595    9603    9612    9631    9656    9659    9663    9713    9715    9725    9733
                        9745    9760    9762    9765    9767    9781    9785    9823    9851#   9891
TYPOC = 104402          2800    2945    3020    3024    7067    7071    8214    8232    8240    8244    8254    9356    9585
                        9751    9775    9852#
TYPON = 104404          9854#
TYPOS = 104403          9853#
T.SCOP  045752          7202#   9861
UN      021444          2830*   2832#
UNIT    015112          2585#   2770*   2959*   2966*   2978    2995*   2997    3015    6972    6992    6997*   7224
UNITS   015072          2584#   2905    2909    2959    2990    6993
UNITSL  015122          2591#   2771*   2966
UNLOAD  015152          2629#
UNS   = 040000          2365#
UPE   = 020000          2308#   3800    3807    3830    3837    3893
US1   = 000001          2295#   3455    3493    3519    3523    3534    3860    3867    4103    4107    4119
US2   = 000002          2296#   3455    3493    3519    3523    3534    3860    3867    4103    4107    4119
US4   = 000004          2297#   3455    3493    3519    3523    3534    3860    3867    4103    4107    4119
UWR   = 000010          2489#
U11     025502          3417    3435#
```

```
VUF    = 000002          2488#
VU30   = 010000          2442#
VV     = 000100          2339#  3110   7263
WAIT.T   046342          7331#  9863
WAT    = 104415          9863#
WC       015024          1940   2557#  4258   4322*  4364   4418   4488*  4493*  4522   4576   4625*  4629*  4656
                         7117
WCE    = 040000          2309#
WCF    = 000040          2356#
WCRC     054554          4165   4237   4397   5025   5185   5350   5667   5824   5985   6289   6439   6608   6784
                         7410   7503   7659   8558   8856#
WCU    = 000001          2430#
WCYL     056012          4547*  4554   4763*  4770   4863*  4870   5504*  5511   6134*  6141   8935#
WECC1    055572          1974   4914   5555   6184   8861#
WECC2    055574          1974   4918   5559   6188   6448   6451*  6452*  8862#
WKEY1    056016          4550*  4766*  4866*  5507*  6137*  8937#
WKEY2    056020          4551*  4767*  4867*  5508*  6138*  8938#
WLE    = 004000          2362#
WORD     053564          8536*  8541*  8550*  8552*  8554*  8556*  8558*  8608*  8612*  8641#  8651*  8664*  8681*
                         9338*  9346*
WRCHDA   047026          7493#
WRCHDT   015166          2635#  7428
WRCHEK   015164          2634#  7522
WRCHHD   046514          7401#
WRDATA   054242          8447   8781#  9022
WRFROM   015220          2652#  3117   3123*  3130   4338   4342*  4365   4403   4489   4494   4561   4626   4630
                         4705   4731   4732   4776   4801   4876   5094   5116   5259   5274*  5285   5431   5447*
                         5458   5517   5736   5755   5896   5912*  5920   6070   6086*  6094   6147   6358   6376
                         6519   6544   6691   6707*  6709*  6719   6868   6884*  6885*  6893   7726   7825
WRHEAD   056106          8933   9058#
WRIDAT   015170          2636#  4180   4412
WRIFOR   015172          2637#  4570   4778   4894   5535   6165   7822
WRITE    054022          8571   8583   8714#  8805   8812   8818   8822   8825   9107   9127   9142   9156   9170
                         9183
WRL    = 004000          2344#
WRU    = 000400          2438#
WSECTR   056014          4549*  4765*  4865*  5506*  6136*  8918   8936#
WSSYNC   054542          8854#
WSU    = 000004          2432#
WTRK     055750          8920*  8924#
WWORD    054020          8572   8576   8580   8584   8592   8597   8602   8709#  8758*  8806   8813   8819   8823
                         8826   9108   9112   9129   9143   9157   9171   9184
X        052664          4162*  4234*  4394*  5022*  5182*  5347*  5664*  5821*  5982*  6286*  6436*  6605*  6781*
                         7407*  7499*  7699*  7716   8395#  8439
XE2      022270          2881   2903#
X11      025604          3423   3462#
Y        052724          8449#
Y11      025176          3422   3425#
ZCODE    050504          1983   4856*  5010*  5170*  5335*  5497*  5652*  5809*  5970*  6127*  6274*  6424*  6593*
                         6769*  7942#  8134*
ZER    = 000400          2377#  8143
ZWORDS   054240          8780#  8799*
$AUTOB   001134           538#  2758*  9576   9678
$BDADR   001122           533#  1929   1932   1954   1987   7245*  7252*  7256*  7269*  7274*  7277*  7297*  7305*
                         7311*  7316*
$BDDAT   001126           535#  1922   1926   1932   1937   1949   1951   1957   1960   1964   1967   2268   2848*
```

```
                    2849    3219*   3221    3245*   3247    3269*   3271    3297*   3299    3322*   3324    3350*   3352
                    3383*   3454*   3455*   3457    3492*   3493*   3495    3533*   3534*   3536    3561*   3563    3620*
                    3622    3669*   3672    3713*   3748*   3749*   3751    3774*   3776*   3778    3806*   3807*   3809
                    3836*   3837*   3839    3866*   3867*   3869    3898*   3899*   3901    3929*   3930*   3932    3961*
                    3962*   3964    3987*   3988*   3990    4019*   4020*   4022    4037*   4080*   4081*   4083    4118*
                    4119*   4121    4204*   7156*   7158*   7160    7350*   7464*   7769*   7770    7779*   7780    7791*
                    7792    7799*   7801    7867*   7868    8576*   8597*   9101*   9112*   9135*   9150*   9162*   9177*
                    9189*
$BELL   001216       566#   9541    9671    9725    9745
$CHARC  060326      9487*   9496*   9500    9509*   9514#
$CKSWR  060640      9568#   9857
$CMTAG  001100       522#   2684    2691    2697    2698
$CM1  = 000006       552#    553#    554#    555#    556#    557#    558#
$CM2  = 000014       552#    553#    554#    555#    556#    557#    558#
$CM3  = 000006       551#    552
$CM4  = 000006       558#    559#    560#    561#    562#    563#    564#
$CNTLC  061411      9531    9595    9671#
$CNTLG  061423      9582    9673#
$CNTLU  061416      9603    9672#
$CRLF   001223       568#   2950    7072    9486    9516    9612    9671    9718    9733    9745    9762    9767    9785
$DBLK   060102      9426    9460    9468#
$DOAGN  044662      7014    7023    7029#
$DTBL   060072      9429    9464#
$ENDAD  044652       498    7025#
$ENDCT  044620      7016#
$ENDMG  044671      6984    7018    7032#
$ENULL  044666      6987    7021    7031#
$EOP    044564      2802    2898    6991    7007#
$EOPCT  044612      7013#   7017
$ERFLG  001103       524#   9366    9386    9388    9394*   9415    9720*   9745
$ERMAX  001115       530#   2699*   9388    9410*   9415
$ERROR  061612      2691    2815    9718#
$ERRPC  001116       531#   1922    1924    1926    1929    1932    1935    1937    1940    1944    1947    1949    1951
                    1954    1957    1960    1964    1967    1971    1974    1977    1980    1983    1987    2266    2268
                    2270    9727*   9728*   9729    9745    9750
$ERRTB  001226       570#   9757
$ERRTY  061760      9732    9745#
$ERTTL  001112       528#   6976    6978*   9726*   9745
$ESCAP  001214       565#   2698*   9409*   9741    9743    9745
$FILLC  001156       549#   9490    9516
$FILLS  001155       548#   9516
$GDADR  001120       532#
$GDDAT  001124       534#   1922    1924    1926    1949    1957    1960    1964    1967    1971    2268    2270    2851*
                    3214*   3221    3235*   3247    3262*   3271    3286*   3299    3314*   3324    3339*   3352    3384*
                    3445*   3446*   3457    3484*   3495    3519*   3536    3558*   3560    3563    3617*   3619    3622
                    3661*   3665    3670*   3672    3714*   3722*   3742*   3751    3770*   3778    3800*   3802    3809
                    3828*   3839    3858*   3869    3891*   3901    3918*   3932    3955*   3964    3981*   3990    4011*
                    4022    4038*   4050*   4066*   4083    4103*   4121    4203*   7151*   7153*   7155    7160    7463*
                    7774*   7785*   7795*   7804*   7870*   8546*   8575*   8594*   8596*   8623*   9100*   9111*   9134*
                    9149*   9163*   9176*   9190*
$GET42  044642      7022#
$GTSWR  060730      9583#   9856
$HD   = 000000       474
$HIOCT  061610      9706*   9717#
$ICNT   001104       525#   9401*   9402    9404*   9414
$ILLUP  062616      9864    9877    9896#
```

```
$INTAG  001135          539#  9581*  9597  9613  9678
$ITEMB  001114          529#  9729*  9745  9748
$LF     001224          569#  9516  9663  9671  9718  9745
$LPADR  001106          526#  2700*  2971*  7081*  7088  9392*  9407*  9412  9414
$LPERR  001110          527#  2701*  7070  7087*  9392  9408*  9414  9740
$MAIL = ****** U       2714  2754  9407  9474  9734
$MNEW   061441         9586  9676#
$MSWR   061430         9583  9674#
$MXCNT  057664         9405  9414#
$NULL   001154          547#  9492  9516
$NWTST= 000001         2776#  2822#  2842#  2857#  2969#  3069#  3090#  3146#  3166#  3205#  3226#  3253#  3277#
                       3305#  3330#  3370#  3411#  3470#  3511#  3547#  3604#  3648#  3704#  3735#  3759#  3787#
                       3819#  3849#  3879#  3910#  3946#  3972#  4002#  4030#  4059#  4095#  4128#  4141#  4222#
                       4383#  4541#  4674#  4836#  4974#  5133#  5299#  5477#  5616#  5772#  5934#  6107#  6240#
                       6389#  6558#  6733#  6909#  6938#  6965#
$OCNT   062356         9795*  9824*  9837#
$OMODE  062360         9790*  9794*  9799  9802*  9813*  9839#
$OVER   057650         9372  9385  9393  9403  9411#
$PASS   001100          522#  3419  6983*  6985  7010*  7011*  7019  7031  9399  9415
$POWER  062624         9892  9899#
$PWRAD  062612         9894#
$PWRDN  062452         2695  9864#  9889
$PWRMG  062606         9892#
$PWRUP  062524         9874  9877#
$QUES   001222          567#  9516  9631  9656  9671  9715  9718  9745
$RDCHR  061202         9633#  9858
$RDDEC= ****** U       9861
$RDLIN  061272         9648#  9859
$RDOCT  061452         9679#  9860
$RDSZ = 000011         9648#
$REGAD  001160          551#
$REG0   001162          552#
$REG1   001164          553#
$REG2   001166          554#
$REG3   001170          555#
$REG4   001172          556#
$REG5   001174          557#
$RTNAD  044664         7030#
$R2A  = ****** U       9861
$SAVRE= ****** U       9861
$SAVR6  062622         9873*  9878  9879*  9880*  9898#
$SCOPE  057374         2689  9366#
$SETUP= 000117         2683#  2689  2691  2693  2695  2697  2698  2700  2752  7008  9366  9535  9539
                       9568  9678  9718  9737  9744
$SS1  = 000000         2715#
$STUP = 177777         2683#
$SVLAD  057622         9378  9406#
$SVPC = 000200          498#  500
$SWR  = 167770          474#  492  564  565  566  2697  2698  2700  2701  2777  2823  2843  2858
                       2970  3070  3091  3147  3167  3206  3227  3254  3278  3306  3331  3371  3412
                       3471  3512  3548  3605  3649  3705  3736  3760  3788  3820  3850  3880  3911
                       3947  3973  4003  4031  4060  4096  4129  4142  4223  4384  4542  4675  4837
                       4975  5134  5300  5478  5617  5773  5935  6108  6241  6390  6559  6734  6910
                       6939  6966  7007  7009  7024  7029  7031  9366  9371  9381  9382  9383  9386
                       9387  9388  9395  9396  9397  9408  9411  9414  9718  9723  9730  9734  9738
                       9745  9895
```

```
$SWRMK= 000000   492  9366  9384
$TIMES  001212   564# 2697* 2777* 2823* 2843* 2858* 2970* 3167* 3206* 3227* 3254* 3278* 3306*
                3331* 3371* 3412* 3471* 3512* 3548* 3605* 3649* 3705* 3736* 3760* 3788* 3820*
                3850* 3880* 3911* 3947* 3973* 4003* 4031* 4060* 4096* 6966* 7009* 9395* 9402
                9405* 9414
$TKB    001146   544# 9516  9524  9527  9550  9572  9591
$TKCNT  060332  9516# 9519* 9539  9556* 9639  9641*
$TKINT  060352  2719  2860  8217  9519# 9532  9579
$TKQEN= 060351  9519# 9564  9644
$TKQIN  060334  9517# 9520* 9521  9562* 9563* 9564  9566*
$TKQOU  060336  9518# 9521* 9642  9643* 9644  9646*
$TKQSR  060340  9519# 9520  9566  9646
$TKS    001144   543# 4172  4173  9516  9525* 9546* 9548  9554* 9570  9580* 9589  9599* 9615*
$TKSRV  060422  9522  9527#
$TMP0   001176   558# 1937  2913* 2914* 3173* 3189  3386* 3397  3452* 3490* 3527* 3716* 3726
                4017* 4040* 4053  4068* 4088  4276* 4277  4435* 4436  4593* 4594  7335* 7341
                7346  7350  7454* 7468  7565* 7611*
$TMP1   001200   559# 1935  1937  1951  2791* 6920* 6921* 6949* 6950* 7336* 7341  7346  7455*
                7470
$TMP2   001202   560# 1951  7339* 7343* 7345* 7348* 7567* 7609*
$TMP3   001204   561# 1937  1951  7333* 7334* 7566* 7568  7569*
$TMP4   001206   562# 7568* 7592*
$TMP5   001210   563# 7564* 7570* 7576  7688* 7767  7789
$TN   = 000073   474# 2765  2776  2777# 2779  2822  2823# 2825  2828  2842  2843# 2844  2850
                2857  2858# 2965  2969  2970# 3069  3070# 3073  3090  3091# 3092  3146  3147#
                3148  3166  3167# 3205  3206# 3207  3222  3226  3227# 3228  3248  3253  3254#
                3255  3272  3277  3278# 3279  3300  3305  3306# 3307  3325  3330  3331# 3332
                3353  3370  3371# 3372  3411  3412# 3413  3462  3470  3471# 3472  3500  3511
                3512# 3513  3537  3547  3548# 3549  3569  3604  3605# 3606  3629  3648  3649#
                3650  3679  3704  3705# 3706  3735  3736# 3737  3752  3759  3760# 3761  3764
                3779  3787  3788# 3789  3792  3810  3819  3820# 3821  3840  3849  3850# 3851
                3870  3879  3880# 3881  3884  3902  3910  3911# 3912  3933  3935  3946  3947#
                3948  3965  3972  3973# 3974  3991  4002  4003# 4004  4023  4030  4031# 4032
                4059  4060# 4061  4095  4096# 4097  4122  4128  4129# 4130  4141  4142# 4144
                4147  4196  4218  4222  4223# 4224  4282  4383  4384# 4385  4442  4541  4542#
                4543  4599  4674  4675# 4676  4793  4827  4836  4837# 4839  4909  4956  4974
                4975# 4977  5066  5121  5133  5134# 5136  5235  5290  5299  5300# 5302  5397
                5463  5477  5478# 5480  5550  5598  5616  5617# 5619  5708  5760  5772  5773#
                5775  5872  5925  5934  5935# 5937  6035  6099  6107  6108# 6110  6180  6227
                6240  6241# 6242  6330  6381  6389  6390# 6391  6495  6549  6558  6559# 6560
                6657  6724  6733  6734# 6736  6833  6898  6909  6910# 6911  6928  6933  6938
                6939# 6940  6954  6965  6966#
$TPB    001152   546# 9506* 9516
$TPFLG  001157   550# 9468  9516
$TPS    001150   545# 9504  9516
$TRAP   062362  2693  9840#
$TRAP2  062404  9847# 9850
$TRP  = 000016  9850# 9852# 9853# 9854# 9855# 9856# 9857# 9858# 9859# 9860# 9861# 9862# 9863#
                9864#
$TRPAD  062416  9845  9850#
$TSTNM  001102   523# 6979* 7008* 9366  9384  9406* 9411  9415  9722  9745
$TTYIN  061400  9649  9650  9667  9671#
$TYPBN= ****** U  9856
$TYPDS  057666  9415# 9855
$TYPE   060112  9468# 9850  9851
$TYPEC  060262  9489  9495  9499  9504# 9505  9617
```

```
$TYPEX  060330          9510    9512    9515#
$TYPOC  062160          9793#   9852
$TYPON  062174          9792    9795#   9854
$TYPOS  062134          9788#   9853
$XTSTR  057426          9373#
$$GET4= 000000          7024#
$OFILL  062357          9789*   9793*   9803    9838#
$40CAT= ****** U        9371    9732
.       = 062634         495#    498#    499#    500#    503#    521#    522#    570    2056#   2264#   2584#   2651#   2652#
                        2653#   2687    2700    2701    2729#   2732#   2737#   2739#   2745#   2810#   2815#   2866#   2891#
                        2893#   2940#   2944#   2948#   3015#   3054#   3058#   3062#   3077#   3079#   3086#   3432#   6972#
                        6976#   7031    7035    7075#   7079#   7084#   7086#   8231#   8247#   8253#   8261#   8853#   8854#
                        8855#   8856#   8857#   8858#   8860#   8861#   8862#   8863#   8864#   9356#   9414    9415    9468#
                        9516    9519#   9671#   9678    9718    9745    9788#   9876    9897

. ABS.  062634      000


 ERRORS DETECTED:  0

 CZRJHC,CZRJHC.SEQ/SOL/NL:TOC:MC:ME:CND=SYSMAC.SML[400,1066],CZRJHC.P11[400,1424],CZRJHC.P12[400,1424]
 RUN-TIME: 33 39 3 SECONDS
 RUN-TIME RATIO: 498/75=6.5
 CORE USED:  37K  (73 PAGES)

 DOCUMENT PAGES:  250
```