RP04/5/6

DISKLESS CONTROLLER 2

CZRJHB0

AH-9215B-MC

FICHE 1 OF 2

JAN 1978

digital

MADE IN USA

RP04/5/6

DISKLESS CONTROLLER 2
CZRJHB0

AH-9215B-MC

FICHE 2 OF 2

JAN 1978
digital
MADE IN USA

.REM @

IDENTIFICATION

PRODUCT CODE: AC-9213B-MC
PRODUCT NAME: CZRJHB0 RP04/5/6 DISKLESS CONTROLLER TEST-PART II
DATE CREATED: DECEMBER 1977
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: PETE BLACKSTONE

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.

```
440
441                                     WHEN THIS OPERATOR SELECTABLE SCOPE LOOP IS USED THEN THE POINT
442                                     THE PROGRAM GOES BACK TO CAN BE CHANGED.
443                                     THE RESTRICTIONS TO THE POINT WHERE THE PROGRAM CAN GO ARE: -
444                                     1.  IT MUST BE WITHIN THE TEST UNDER CONSIDERATION
445                                     2.  LOOP ON ERROR SWITCH MUST BE SET
446                                     3.  THE ERROR MUST OCCUR WITHIN THE TEST UNDER CONSIDERATION
447                                     IF THE ERROR DOES NOT OCCUR WITHIN THE TEST UNDER CONSIDERATION
448                                     THE PROGRAM WILL REVERT TO NORMAL OPERATION.  HOWEVER, IF LOOP ON
449                                     TEST SWITCH IS SET AND THIS OPERATOR SELECTABLE SCOPE LOOP IS USED
450                                     THEN THE PROGRAM WILL LOOP BACK TO THE SELECTED POINT WHEN IT
451                                     COMES TO THE END OF THE TEST UNDER CONSIDERATION.
452
453                                     AFTER LOOPING FOR SOME TIME IF THE LOOP SWITCH IS PUT DOWN THEN
454                                     NORMAL OPERATION WILL CONTINUE.
455
456                             8.4     PROGRAM REVISION HISTORY
457
458
459                             9.0     PROGRAM DESCRIPTION
460
461                                     THE FOLLOWING SECTIONS DESCRIBE EACH TEST AND SUBROUTINES
462                                     IN DETAIL AND CAN ALSO BE USED AS AN INDEX TO THE LISTING.
463                                     THE LEFT MOST COLUMN IS THE LINE NUMBER WITHIN THE LISTING
464                                     WHERE THAT ITEM WILL BE FOUND.
465                                     @
```

```
;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.
                ;*
```

```
487
```

```
                                        .SBTTL  STARTING ADDRESSES

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

        000204  000137  051410          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
```

```
509
510          001110                          .=1110                          ;
```

```
                                ;ITEM1
001226  002136                          EM1             ;WRONG DATA IN READING OR WRITING HARDWARE REGISTER
001230  005451                          DH1             ;PC
                                                        ;REG. ADDR.
                                                        ;GOOD DATA
                                                        ;RECEIVED DATA
001232  011764                          DT1             ;$ERRPC,$TSTNM,REGADR,$GDDAT,$BDDAT
001234  012516                          DF1             ;0,0,0,0,0



                                ;ITEM2
001236  002221                          EM2             ;ERROR ON  DATA COMMAND
001240  010463                          DH33            ;PC
                                                        ;PC OF JSR
                                                        ;TEST NO
                                                        ;WORD NO.
                                                        ;GOOD DATA
                                                        ;CONTENTS OF RHCS1
                                                        ;CONTENTS OF RHDS1
                                                        ;CONTENTS OF RHER1
001242  012350                          DT33            ;$ERRPC,PCJSR,$TSTNM,ERWORD,$GDDAT,CS1,DS1,ER1
001244  012666                          DF33            ;0,0,0,1,0,0,0,0


                                ;ITEM3
001246  002221                          EM2             ;ERROR ON  DATA COMMAND

001250  010246                          DH32            ;PC
                                                        ;PC OF JSR
                                                        ;TEST NO
                                                        ;WORD NO.
                                                        ;GOOD DATA
                                                        ;BAD DATA
                                                        ;CONTENTS OF RHCS1
                                                        ;CONTENTS OF RHDS1
                                                        ;CONTENTS OF RHER1

001252  012324                          DT32            ;$ERRPC,PCJSR,$TSTNM,ERWORD,$GDDAT,$BDDAT,CS1,DS1,ER1
001254  012655                          DF32            ;0,0,0,1,0,0,0,0,0,


                                ;ITEM4
001256  002221                          EM2             ;ERROR ON  DATA COMMAND

001260  010050                          DH31            ;PC
                                                        ;TEST NO
                                                        ;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
                                                              ;
```

```
                                ;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,
```

```
                                ;ITEM41
001626  004752                          EM40            ;THERE WAS A READ/WRITE HEADER & DATA
                                                        ;ERROR DURING 'DTE' TEST SETUP - THE
                                                        ;TEST IS ABORTED AT THAT POINT
001630  011607                          DH40            ;PC
                                                        ;TEST NO
                                                        ;FAILING REGISTER ADDRESS
                                                        ;CONTENTS OF RHCS1
                                                        ;CONTENTS OF RHCS2
                                                        ;CONTENTS OF RHDS1
                                                        ;CONTENTS OF RHER1
001632  012476                          DT40            ;$ERRPC,$TSTNM,$BDADR,CS1,CS2,DS1,ER1
001634  012734                          DF40            ;0,0,0,0,0,0
```

```
                                ;ITEM 42
001636  012744                          EM42
001640  014410                          DH42
001642  014672                          DT42    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS.
001644  014730                          DF42

                                ;ITEM 43
001646  013027                          EM43
001650  014410                          DH42
001652  014672                          DT42    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS.
001654  014730                          DF42

                                ;ITEM 44
001656  013073                          EM44
001660  014471                          DH44
001662  014702                          DT44    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS CORRECT, ACTUAL.
001664  014733                          DF44

                                ;ITEM 45
001666  013130                          EM45
001670  014471                          DH44
001672  014702                          DT44    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS CORRECT, ACTUAL.
001674  014733                          DF44

                                ;ITEM 46
001676  013157                          EM46
001700  014471                          DH44
001702  014702                          DT44    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS CORRECT, ACTUAL.
001704  014733                          DF44

                                ;ITEM 47
001706  013206                          EM47
001710  014410                          DH42
001712  014672                          DT42    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS.
001714  014730                          DF42

                                ;ITEM 50
001716  013243                          EM50
001720  014471                          DH44
001722  014702                          DT44    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS CORRECT, ACTUAL.
001724  014733                          DF44

                                ;ITEM 51
001726  013301                          EM51
001730  014471                          DH44
001732  014702                          DT44    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS CORRECT, ACTUAL.
001734  014733                          DF44

                                ;ITEM 52
```

```
1031
1032  001736  013322                    EM52
1033  001740  014471                    DH44
1034  001742  014702                    DT44
1035  001744  014733                    DF44
1036
1037                              ;ITEM 53
1038
1039  001746  013362                    EM53
1040  001750  014471                    DH44
1041  001752  014702                    DT44    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS CORRECT, ACTUAL.
1042  001754  014733                    DF44
1043
1044                              ;ITEM 54
1045
1046  001756  013422                    EM54
1047  001760  014603                    DH54
1048  001762  014716                    DT54    ;ERROR PC, TEST NUMBER.
1049  001764  014740                    DF54
1050
1051                              ;ITEM 55
1052
1053  001766  013461                    EM55
1054  001770  014471                    DH44
1055  001772  014702                    DT44    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS CORRECT, ACTUAL.
1056  001774  014733                    DF44
1057
1058                              ;ITEM 56
1059
1060  001776  013474                    EM56
1061  002000  014471                    DH44
1062  002002  014702                    DT44    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS CORRECT, ACTUAL.
1063  002004  014733                    DF44
1064
1065                              ;ITEM 57
1066
1067  002006  013511                    EM57
1068  002010  014471                    DH44
1069  002012  014702                    DT44    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS CORRECT, ACTUAL.
1070  002014  014733                    DF44
1071
1072                              ;ITEM 60
1073
1074  002016  013540                    EM60
1075  002020  014471                    DH44
1076  002022  014702                    DT44    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS CORRECT, ACTUAL.
1077  002024  014733                    DF44
1078
1079                              ;ITEM 61
1080
1081  002026  013627                    EM61
1082  002030  014471                    DH44
1083  002032  014702                    DT44    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS CORRECT, ACTUAL.
1084  002034  014733                    DF44
1085
1086                              ;ITEM 62
```

```
1087
1088  002036  013677                    EM62
1089  002040  014471                    DH44
1090  002042  014702                    DT44    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS CORRECT, ACTUAL.
1091  002044  014733                    DF44
1092
1093                              ;ITEM 63
1094
1095  002046  013754                    EM63
1096  002050  014471                    DH44
1097  002052  014702                    DT44    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS CORRECT, ACTUAL.
1098  002054  014733                    DF44
1099
1100                              ;ITEM 64
1101
1102  002056  014032                    EM64
1103  002060  014471                    DH44
1104  002062  014702                    DT44    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS CORRECT, ACTUAL.
1105  002064  014733                    DF44
1106
1107                              ;ITEM 65
1108
1109  002066  014066                    EM65
1110  002070  014471                    DH44
1111  002072  014702                    DT44    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS CORRECT, ACTUAL.
1112  002074  014733                    DF44
1113
1114                              ;ITEM 66
1115
1116  002076  014150                    EM66
1117  002100  014471                    DH44
1118  002102  014702                    DT44    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS CORRECT, ACTUAL.
1119  002104  014733                    DF44
1120
1121                              ;ITEM 67
1122
1123  002106  014222                    EM67
1124  002110  014603                    DH54
1125  002112  014716                    DT54    ;ERROR PC, TEST NUMBER.
1126  002114  014740                    DF54
1127
1128                              ;ITEM 70
1129
1130  002116  014307                    EM70
1131  002120  014603                    DH54
1132  002122  014716                    DT54    ;ERROR PC, TEST NUMBER.
1133  002124  014740                    DF54
1134
1135                              ;ITEM 71
1136
1137  002126  014361                    EM71
1138  002130  014471                    DH44
1139  002132  014702                    DT44    ;ERROR PC,TEST NUMBER, REGISTER ADDRESS CORRECT, ACTUAL.
1140  002134  014733                    DF44
```

```
1141
1142                            ;*****************************************************************************
1143                            ;
1144                            ;ERROR AND MESSAGE TABLE CONDIMENTS
1145                            ;
1146                            ;*****************************************************************************
1147
1148
1149
1150
1151  002136  051127  047117  020107  EM1:    .ASCIZ  /WRONG DATA IN READING OR WRITING HARDWARE REGISTER/
1152  002144  040504  040524  044440
1153  002152  020116  042522  042101
1154  002160  047111  020107  051117
1155  002166  053440  044522  044524
1156  002174  043516  044040  051101
1157  002202  053504  051101  020105
1158  002210  042522  044507  052123
1159  002216  051105     000
1160  002221     105  051122  051117  EM2:    .ASCIZ  /ERROR ON  DATA COMMAND/
1161  002226  047440  020116  042040
1162  002234  052101  020101  047503
1163  002242  046515  047101  000104
1164  002250  051105  047522  020122  EM6:    .ASCIZ  /ERROR ON WRITE HEADER AND DATA/
1165  002256  047117  053440  044522
1166  002264  042524  044040  040505
1167  002272  042504  020122  047101
1168  002300  020104  040504  040524
1169  002306     000
1170  002307     103  047117  051124  EM11:   .ASCIZ  /CONTROLLER OR DRIVE STATUS/
1171  002314  046117  042514  020122
1172  002322  051117  042040  044522
1173  002330  042526  051440  040524
1174  002336  052524  000123
1175  002342  042522  044507  052123  EM14:   .ASCIZ  /REGISTER FAILED/
1176  002350  051105  043040  044501
1177  002356  042514  000104
1178  002362  047516  020116  054105  EM15:   .ASCIZ  /NON EXISTENT REGISTER, PROGRAM ABORTED./
1179  002370  051511  042524  052116
1180  002376  051040  043505  051511
1181  002404  042524  026122  050040
1182  002412  047522  051107  046501
1183  002420  040440  047502  052122
1184  002426  042105  000056
1185  002432  040527  052111  046040  EM16:   .ASCIZ  /WAIT LOOP FAILED/
1186  002440  047517  020120  040506
1187  002446  046111  042105     000
1188  002453     127  044522  042524  EM17:   .ASCIZ  /WRITE CHECK FAILING/
1189  002460  041440  042510  045503
1190  002466  043040  044501  044514
1191  002474  043516     000
1192  002477     122  043505  051511  EM20:   .ASCIZ  /REGISTER FAILING/
1193  002504  042524  020122  040506
1194  002512  046111  047111  000107
1195  002520  047111  042524  051122  EM21:   .ASCIZ  /INTERRUPT FAILING/
1196  002526  050125  020124  040506
```

```
1197  002534  046111  047111  000107
1198  002542  051105  047522  020122  EM22:   .ASCII  /ERROR ON DRIVE PRESENT/<15><12>
1199  002550  047117  042040  044522
1200  002556  042526  050040  042522
1201  002564  042523  052116  005015
1202  002572  044124  020105  047125          .ASCII  /THE UNIT NO'S FOUND BY SETTING RHAS/<15><12>
1203  002600  052111  047040  023517
1204  002606  020123  047506  047125
1205  002614  020104  054502  051440
1206  002622  052105  044524  043516
1207  002630  051040  040510  006523
1208  002636     012
1209  002637     104  020117  047516          .ASCII  /DO NOT AGREE WITH THE UNIT NO. FOUND FROM/<15><12>
1210  002644  020124  043501  042522
1211  002652  020105  044527  044124
1212  002660  052040  042510  052440
1213  002666  044516  020124  047516
1214  002674  020056  047506  047125
1215  002702  020104  051106  046517
1216  002710  005015
1217  002712  044122  051503  026462          .ASCII  /RHCS2-'NED' BIT #12/<15><12>
1218  002720  047047  042105  020047
1219  002726  044502  020124  030443
1220  002734  006462     012
1221  002737     061  033467  033467          .ASCII  /177777-MEANS NO UNIT FOUND/<15><12>
1222  002744  026467  042515  047101
1223  002752  020123  047516  052440
1224  002760  044516  020124  047506
1225  002766  047125  006504     012
1226  002773     116  052117  035105          .ASCII  /NOTE: ON DUAL PORT SYSTEM, DRIVE ON OTHER PORT WILL NOT GIVE/<15><12>
1227  003000  047440  020116  052504
1228  003006  046101  050040  051117
1229  003014  020124  054523  052123
1230  003022  046505  020054  051104
1231  003030  053111  020105  047117
1232  003036  047440  044124  051105
1233  003044  050040  051117  020124
1234  003052  044527  046114  047040
1235  003060  052117  043440  053111
1236  003066  006505     012
1237  003071     047  042516  023504          .ASCIZ  /'NED', HENCE THERE WILL BE AN EXTRA DRIVE/
1238  003076  020054  042510  041516
1239  003104  020105  044124  051105
1240  003112  020105  044527  046114
1241  003120  041040  020105  047101
1242  003126  042440  052130  040522
1243  003134  042040  044522  042526
1244  003142     000
1245  003143     114  047517  020113  EM24:   .ASCIZ  /LOOK AHEAD REGISTER AT THE BEGINNING OF SECTOR IS IN ERROR/
1246  003150  044101  040505  020104
1247  003156  042522  044507  052123
1248  003164  051105  040440  020124
1249  003172  044124  020105  042502
1250  003200  044507  047116  047111
1251  003206  020107  043117  051440
1252  003214  041505  047524  020122
```

```
1253   003222  051511  044440  020116
1254   003230  051105  047522  000122
1255   003236  047514  045517  040440  EM25:   .ASCIZ  /LOOK AHEAD REGISTER IS IN ERROR/
1256   003244  042510  042101  051040
1257   003252  043505  051511  042524
1258   003260  020122  051511  044440
1259   003266  020116  051105  047522
1260   003274  000122
1261   003276  052503  051122  047105  EM30:   .ASCII  /CURRENT CYLINDER DOES NOT MATCH DESIRED CYLINDER REGIHSTER/<15><12>
1262   003304  020124  054503  044514
1263   003312  042116  051105  042040
1264   003320  042517  020123  047516
1265   003326  020124  040515  041524
1266   003334  020110  042504  044523
1267   003342  042522  020104  054503
1268   003350  044514  042116  051105
1269   003356  051040  043505  044111
1270   003364  052123  051105  005015
1271   003372  043101  042524  020122          .ASCIZ  /AFTER A SEEK AND INIT/
1272   003400  020101  042523  045505
1273   003406  040440  042116  044440
1274   003414  044516  000124
1275   003420  041505  020103  042507  EM31:   .ASCII  /ECC GENERATED IS INCORRECT/<15><12>
1276   003426  042516  040522  042524
1277   003434  020104  051511  044440
1278   003442  041516  051117  042522
1279   003450  052103  005015
1280   003454  053105  051105  020131          .ASCIZ  /EVERY WORD ON THIS SECTOR IS THAT GIVEN IN "DATA USED"/
1281   003462  047527  042122  047440
1282   003470  020116  044124  051511
1283   003476  051440  041505  047524
1284   003504  020122  051511  052040
1285   003512  040510  020124  044507
1286   003520  042526  020116  047111
1287   003526  021040  040504  040524
1288   003534  052440  042523  021104
1289   003542     000
1290   003543     117  020116  042522  EM32:   .ASCII  /ON READ COMMAND, AFTER DATA AND ECC HAVE BEEN READ,/<15><12>
1291   003550  042101  041440  046517
1292   003556  040515  042116  020054
1293   003564  043101  042524  020122
1294   003572  040504  040524  040440
1295   003600  042116  042440  041503
1296   003606  044040  053101  020105
1297   003614  042502  047105  051040
1298   003622  040505  026104  005015
1299   003630  041505  020103  042522          .ASCII  /ECC REGISTERS OR RHER1 ARE IN ERROR/<15><12>
1300   003636  044507  052123  051105
1301   003644  020123  051117  051040
1302   003652  042510  030522  040440
1303   003660  042522  044440  020116
1304   003666  051105  047522  006522
1305   003674     012
1306   003675     117  046116  020131          .ASCII  /ONLY LOWER 11 BITS OF PATTERN REG. CAN BE READ/<15><12>
1307   003702  047514  042527  020122
1308   003710  030461  041040  052111
```

```
1309  003716  020123  043117  050040
1310  003724  052101  042524  047122
1311  003732  051040  043505  020056
1312  003740  040503  020116  042502
1313  003746  051040  040505  006504
1314  003754     012
1315  003755     124  044510  020123          .ASCIZ  /THIS SHOULD MATCH LOWER 11 BITS OF GOOD ECC1/
1316  003763  044123  052517  042114
1317  003770  046440  052101  044103
1318  003776  046040  053517  051105
1319  004004  030440  020061  044502
1320  004012  051524  047440  020106
1321  004020  047507  042117  042440
1322  004026  041503  000061
1323  004032  044510  044107  041440  EM33:   .ASCIZ  /HIGH COUNT BIT NOT SET AFTER 38859 CLOCKS/
1324  004040  052517  052116  041040
1325  004046  052111  047040  052117
1326  004054  051440  052105  040440
1327  004062  052106  051105  031440
1328  004070  034070  034465  041440
1329  004076  047514  045503  000123
1330  004104  042532  047522  042040  EM34:   .ASCIZ  /ZERO DETECT BIT NOT HIGH WHEN 32 BIT ECC REG. HAS 21 ZEROS/
1331  004112  052105  041505  020124
1332  004120  044502  020124  047516
1333  004126  020124  044510  044107
1334  004134  053440  042510  020116
1335  004142  031063  041040  052111
1336  004150  042440  041503  051040
1337  004156  043505  020056  040510
1338  004164  020123  030462  055040
1339  004172  051105  051517     000
1340  004177     120  051517  052111  EM35:   .ASCII  /POSITION REGISTER OR 11 BITS OF PATTERN REGISTER INCORRECT/<15><12>
1341  004204  047511  020116  042522
1342  004212  044507  052123  051105
1343  004220  047440  020122  030461
1344  004226  041040  052111  020123
1345  004234  043117  050040  052101
1346  004242  042524  047122  051040
1347  004250  043505  051511  042524
1348  004256  020122  047111  047503
1349  004264  051122  041505  006524
1350  004272     012
1351  004273     114  053517  051105          .ASCII  /LOWER 11 BITS OF PATTERN REGISTER SHOULD MATCH LOWER/<15><12>
1352  004300  030440  020061  044502
1353  004306  051524  047440  020106
1354  004314  040520  052124  051105
1355  004322  020116  042522  044507
1356  004330  052123  051105  051440
1357  004336  047510  046125  020104
1358  004344  040515  041524  020110
1359  004352  047514  042527  006522
1360  004360     012
1361  004361     061  020061  044502          .ASCII  /11 BITS OF GOOD ECC1/<15><12>
1362  004366  051524  047440  020106
1363  004374  047507  042117  042440
1364  004402  041503  006461     012
```

```
1365  004407     104  052101  042440          .ASCIZ  /DAT ENVLOP GOOD POSITION AND N-CODE ZEROS ARE IN OCTAL/
1366  004414  053116  047514  020120
1367  004422  047507  042117  050040
1368  004430  051517  052111  047511
1369  004436  020116  047101  020104
1370  004444  026516  047503  042504
1371  004452  055040  051105  051517
1372  004460  040440  042522  044440
1373  004466  020116  041517  040524
1374  004474  000114
1375  004476  047117  051040  040505  EM36:   .ASCII  /ON READ COMMAND WITH NON-CORRECTABLE ERROR DCK AND ECH SHOULD BE SET/<1
1376  004504  020104  047503  046515
1377  004512  047101  020104  044527
1378  004520  044124  047040  047117
1379  004526  041455  051117  042522
1380  004534  052103  041101  042514
1381  004542  042440  051122  051117
1382  004550  042040  045503  040440
1383  004556  042116  042440  044103
1384  004564  051440  047510  046125
1385  004572  020104  042502  051440
1386  004600  052105  005015
1387  004604  043111  050040  051517          .ASCIZ  /IF POSITION REGISTER =10040 OR 10041 IT IS GOOD/
1388  004612  052111  047511  020116
1389  004620  042522  044507  052123
1390  004626  051105  036440  030061
1391  004634  032060  020060  051117
1392  004642  030440  030060  030464
1393  004650  044440  020124  051511
1394  004656  043440  047517  000104
1395  004664  051127  052111  047111  EM37:   .ASCIZ  /WRITING WITH BUS ADDRESS HIGHER THAN 28K CAUSED ERROR/
1396  004672  020107  044527  044124
1397  004700  041040  051525  040440
1398  004706  042104  042522  051523
1399  004714  044040  043511  042510
1400  004722  020122  044124  047101
1401  004730  031040  045470  041440
1402  004736  052501  042523  020104
1403  004744  051105  047522  000122
1404  004752  044124  051105  020105  EM40:   .ASCII  /THERE WAS A READ-WRITE HEADER & DATA ERROR DURING 'DTE'/<15><12>
1405  004760  040527  020123  020101
1406  004766  042522  042101  053455
1407  004774  044522  042524  044040
1408  005002  040505  042504  020122
1409  005010  020046  040504  040524
1410  005016  042440  051122  051117
1411  005024  042040  051125  047111
1412  005032  020107  042047  042524
1413  005040  006447     012
1414  005043     124  051505  020124          .ASCIZ  /TEST SETUP - THE TEST WAS ABORTED AT THAT POINT/
1415  005050  042523  052524  020120
1416  005056  020055  044124  020105
1417  005064  042524  052123  053440
1418  005072  051501  040440  047502
1419  005100  052122  042105  040440
1420  005106  020124  044124  052101
```

```
 1421  005114  050040  044517  052116
 1422  005122     000
```

```
1423  005123     106  052101  046101  CPHALT: .ASCII  /FATAL ERROR - SEE DOCUMENT FOR BEST COURSE OF ACTION/<15><12>
1424  005130  042440  051122  051117
1425  005136  026440  051440  042505
1426  005144  042040  041517  046525
1427  005152  047105  020124  047506
1428  005160  020122  042502  052123
1429  005166  041440  052517  051522
1430  005174  020105  043117  040440
1431  005202  052103  047511  006516
1432  005210     012
1433  005211     040  005015  177607          .ASCII  / /<15><12><207><377><377><207><377><377><207><377><377>
1434  005216  103777  177777  177607
1435  005224     377
1436  005225     124  042510  050040          .ASCII  /THE PROGRAM HAS HALTED DURING A TEST BECAUSE CONTROLLER/<15><12>
1437  005232  047522  051107  046501
1438  005240  044040  051501  044040
1439  005246  046101  042524  020104
1440  005254  052504  044522  043516
1441  005262  040440  052040  051505
1442  005270  020124  042502  040503
1443  005276  051525  020105  047503
1444  005304  052116  047522  046114
1445  005312  051105  005015
1446  005316  051117  042040  053105          .ASCII  /OR DEVICE HAS LOST 'READY', BECOME UNAVAILABLE,/<15><12>
1447  005324  041511  020105  040510
1448  005332  020123  047514  052123
1449  005340  023440  042522  042101
1450  005346  023531  020054  042502
1451  005354  047503  042515  052440
1452  005362  040516  040526  046111
1453  005370  041101  042514  006454
1454  005376     012
1455  005377     107  047117  020105          .ASCIZ  /GONE OFFLINE, OR CANNOT CLEAR STATUS BITS/
1456  005404  043117  046106  047111
1457  005412  026105  047440  020122
1458  005420  040503  047116  052117
1459  005426  041440  042514  051101
1460  005434  051440  040524  052524
1461  005442  020123  044502  051524
1462  005450     000
1463
1464  005451     120  020103  020040  DH1:    .ASCII  /PC      TEST    REG.    GOOD    ACTUAL /<15><12>
1465  005456  020040  052040  051505
1466  005464  020124  020040  051040
1467  005472  043505  020056  020040
1468  005500  043440  047517  020104
1469  005506  020040  040440  052103
1470  005514  040525  004514  005015
1471  005522  020040  020040  020040          .ASCIZ  /       NO      ADDR.   DATA    DATA/
1472  005530  020040  047516  020040
1473  005536  020040  020040  042101
1474  005544  051104  020056  020040
1475  005552  040504  040524  020040
1476  005560  020040  040504  040524
1477  005566     000
1478  005567     120  020103  020040  DH2:    .ASCII  /PC      TEST    WORD    GOOD    BAD/<15><12>
```

```
1479  005574  020040  052040  051505
1480  005602  020124  020040  053440
1481  005610  051117  020104  020040
1482  005616  043440  047517  020104
1483  005624  020040  041040  042101
1484  005632  005015
1485  005634  020040  020040  020040          .ASCIZ  /       NO      NO      DATA    DATA/
1486  005642  020040  047516  020040
1487  005650  020040  020040  047516
1488  005656  020040  020040  020040
1489  005664  040504  040524  020040
1490  005672  020040  040504  040524
1491  005700      000
1492  005701      120 020103  020040  DH11:   .ASCII  /PC      TEST    FAILING CONT.   CONT.   CONT.   CONT./<15><12>
1493  005706  020040  052040  051505
1494  005714  020124  020040  043040
1495  005722  044501  044514  043516
1496  005730  041440  047117  027124
1497  005736  020040  041440  047117
1498  005744  027124  020040  041440
1499  005752  047117  027124  020040
1500  005760  041440  047117  027124
1501  005766  005015
1502  005770  020040  020040  020040          .ASCIZ  /       NO      REG ADR RHCS1   RHCS2   RHDS1   RHER1/
1503  005776  020040  047516  020040
1504  006004  020040  020040  042522
1505  006012  020107  042101  020122
1506  006020  044122  051503  020061
1507  006026  020040  044122  051503
1508  006034  020062  020040  044122
1509  006042  051504  020061  020040
1510  006050  044122  051105  000061
1511  006056  041520  020040  020040  DH14:   .ASCII  /PC      TEST    FAILING CONT.   CONT.   CONT.   CONT.   CONT.  /<15><12
1512  006064  020040  042524  052123
1513  006072  020040  020040  040506
1514  006100  046111  047111  020107
1515  006106  047503  052116  020056
1516  006114  020040  047503  052116
1517  006122  020056  020040  047503
1518  006130  052116  020056  020040
1519  006136  047503  052116  020056
1520  006144  020040  047503  052116
1521  006152  004456  005015
1522  006156  020040  020040  020040          .ASCIZ  /       NO      REG ADR BAD REG RHCS1   RHCS2   RHDS1   RHER1   /
1523  006164  020040  047516  020040
1524  006172  020040  020040  042522
1525  006200  020107  042101  020122
1526  006206  040502  020104  042522
1527  006214  020107  044122  051503
1528  006222  020061  020040  044122
1529  006230  051503  020062  020040
1530  006236  044122  051504  020061
1531  006244  020040  044122  051105
1532  006252  004461      000
1533  006255      120 020103  020040  DH15:   .ASCIZ  /PC      TEST    REG ADR/
1534  006262  020040  052040  051505
```

```
1535  006270  020124  020040  051040
1536  006276  043505  040440  051104
1537  006304     000
1538  006305     120  020103  020040  DH16:   .ASCII  /PC     TEST    WAT     BIT     REG     REG/<15><12>
1539  006312  020040  052040  051505
1540  006320  020124  020040  053440
1541  006326  052101  020040  020040
1542  006334  041040  052111  020040
1543  006342  020040  051040  043505
1544  006350  020040  020040  051040
1545  006356  043505  005015
1546  006362  020040  020040  020040          .ASCIZ  /       NO      PC      WANTED  ADDRESS CONTENTS/
1547  006370  020040  047516  020040
1548  006376  020040  020040  041520
1549  006404  020040  020040  020040
1550  006412  040527  052116  042105
1551  006420  020040  042101  051104
1552  006426  051505  020123  047503
1553  006434  052116  047105  051524
1554  006442     000
1555  006443     120  020103  020040  DH17:   .ASCII  /PC     TEST    CONT.   CONT.   CONT.   CONT.   CONT./<15><12>
1556  006450  020040  052040  051505
1557  006456  020124  020040  041440
1558  006464  047117  027124  020040
1559  006472  041440  047117  027124
1560  006500  020040  041440  047117
1561  006506  027124  020040  041440
1562  006514  047117  027124  020040
1563  006522  041440  047117  027124
1564  006530  005015
1565  006532  020040  020040  020040          .ASCIZ  /       NO      RHBA    RHDB    RHWC    RHCS1   RHCS2/
1566  006540  020040  047516  020040
1567  006546  020040  020040  044122
1568  006554  040502  020040  020040
1569  006562  044122  041104  020040
1570  006570  020040  044122  041527
1571  006576  020040  020040  044122
1572  006604  051503  020061  020040
1573  006612  044122  051503  000062
1574  006620  041520  020040  020040  DH20:   .ASCII  /PC     TEST    CONT    CONT    CONT    CONT    CONT/<15><12>
1575  006626  020040  042524  052123
1576  006634  020040  020040  047503
1577  006642  052116  020040  020040
1578  006650  047503  052116  020040
1579  006656  020040  047503  052116
1580  006664  020040  020040  047503
1581  006672  052116  020040  020040
1582  006700  047503  052116  005015
1583  006706  020040  020040  020040          .ASCIZ  /       NO      RHER1   RHER2   RHER3   RHAS    RHDS1/
1584  006714  020040  047516  020040
1585  006722  020040  020040  044122
1586  006730  051105  020061  020040
1587  006736  044122  051105  020062
1588  006744  020040  044122  051105
1589  006752  020063  020040  044122
1590  006760  051501  020040  020040
```

```
1591   006766  044122  051504  000061
1592   006774  041520  020040  020040  DH21:   .ASCII  /PC      TEST    CONT    CONT    CONT/<15><12>
1593   007002  020040  042524  052123
1594   007010  020040  020040  047503
1595   007016  052116  020040  020040
1596   007024  047503  052116  020040
1597   007032  020040  047503  052116
1598   007040  005015
1599   007042  020040  020040  020040          .ASCIZ  /       NO      RHCS1   RHAS    RHDS1/
1600   007050  020040  047516  020040
1601   007056  020040  020040  044122
1602   007064  051503  020061  020040
1603   007072  044122  051501  020040
1604   007100  020040  044122  051504
1605   007106  000061
1606   007110  041520  020040  020040  DH22:   .ASCII  /PC      TEST    RHAS    RHCS2/<15><12>
1607   007116  020040  042524  052123
1608   007124  020040  020040  044122
1609   007132  051501  020040  020040
1610   007140  044122  051503  006462
1611   007146     012
1612   007147     040  020040  020040          .ASCIZ  /       NO      UNIT    UNIT/
1613   007154  020040  047040  020117
1614   007162  020040  020040  052440
1615   007170  044516  020124  020040
1616   007176  052440  044516  000124
1617   007204  041520  020040  020040  DH24:   .ASCII  /PC      TEST    RHDST   BAD     GOOD    SECTOR  SECTOR/<15><12>
1618   007212  020040  042524  052123
1619   007220  020040  020040  044122
1620   007226  051504  020124  020040
1621   007234  040502  020104  020040
1622   007242  020040  047507  042117
1623   007250  020040  020040  042523
1624   007256  052103  051117  020040
1625   007264  042523  052103  051117
1626   007272  005015
1627   007274  020040  020040  020040          .ASCIZ  /       NO      CONT.   RHLA    RHLA    NO      CLOCK/
1628   007302  020040  047516  020040
1629   007310  020040  020040  047503
1630   007316  052116  020056  020040
1631   007324  044122  040514  020040
1632   007332  020040  044122  040514
1633   007340  020040  020040  047516
1634   007346  020040  020040  020040
1635   007354  046103  041517  000113
1636   007362  041520  020040  020040  DH26:   .ASCII  /PC      TEST    PC OF   FAILING CONT.   CONT.   CONT.   CONT./<15><12>
1637   007370  020040  042524  052123
1638   007376  020040  020040  041520
1639   007404  047440  020106  020040
1640   007412  040506  046111  047111
1641   007420  020107  047503  052116
1642   007426  020056  020040  047503
1643   007434  052116  020056  020040
1644   007442  047503  052116  020056
1645   007450  020040  047503  052116
1646   007456  006456     012
```

```
1647  007461     040  020040  020040          .ASCIZ  /       NO      JSR     REG ADD RHCS1   RHCS2   RHDS1   RHER1   /
1648  007466  020040  047040  020117
1649  007474  020040  020040  045040
1650  007502  051123  020040  020040
1651  007510  051040  043505  040440
1652  007516  042104  051040  041510
1653  007524  030523  020040  051040
1654  007532  041510  031123  020040
1655  007540  051040  042110  030523
1656  007546  020040  051040  042510
1657  007554  030522  000011
1658  007560  041520  020040  020040  DH27:   .ASCII  /PC      TEST    PC OF   FAILING GOOD    ACTUAL /<15><12>
1659  007566  020040  042524  052123
1660  007574  020040  020040  041520
1661  007602  047440  020106  020040
1662  007610  040506  046111  047111
1663  007616  020107  047507  042117
1664  007624  020040  020040  041501
1665  007632  052524  046101  006411
1666  007640     012
1667  007641     040  020040  020040          .ASCIZ  /       NO      JSR     REG.    DATA    DATA/
1668  007646  020040  047040  020117
1669  007654  020040  020040  045040
1670  007662  051123  020040  020040
1671  007670  051040  043505  020056
1672  007676  020040  042040  052101
1673  007704  020101  020040  042040
1674  007712  052101  000101
1675  007716  041520  020040  020040  DH30:   .ASCII  /PC      TEST    PC OF   REG.    GOOD    BAD/<15><12>
1676  007724  020040  042524  052123
1677  007732  020040  020040  041520
1678  007740  047440  020106  020040
1679  007746  042522  027107  020040
1680  007754  020040  047507  042117
1681  007762  020040  020040  040502
1682  007770  006504     012
1683  007773     040  020040  020040          .ASCIZ  /       NO      JSR     ADDR    DATA    DATA/
1684  010000  020040  047040  020117
1685  010006  020040  020040  045040
1686  010014  051123  020040  020040
1687  010022  040440  042104  020122
1688  010030  020040  042040  052101
1689  010036  020101  020040  042040
1690  010044  052101  000101
1691  010050  041520  020040  020040  DH31:   .ASCII  /PC      TEST    WORD    GOOD    BAD     CONT.   CONT.   CONT./<15><12>
1692  010056  020040  042524  052123
1693  010064  020040  020040  047527
1694  010072  042122  020040  020040
1695  010100  047507  042117  020040
1696  010106  020040  040502  020104
1697  010114  020040  020040  047503
1698  010122  052116  020056  020040
1699  010130  047503  052116  020056
1700  010136  020040  047503  052116
1701  010144  006456     012
1702  010147     040  020040  020040          .ASCIZ  /       NO      NO      DATA    DATA    RHCS1   RHDS1   RHER1   /
```

```
1703  010154  020040  047040  020117
1704  010162  020040  020040  047040
1705  010170  020117  020040  020040
1706  010176  042040  052101  020101
1707  010204  020040  042040  052101
1708  010212  020101  020040  051040
1709  010220  041510  030523  020040
1710  010226  051040  042110  030523
1711  010234  020040  051040  042510
1712  010242  030522  000011
1713  010246  041520  020040  020040  DH32:  .ASCII  /PC      TEST    PC OF   WORD    GOOD    BAD     CONT.   CONT.   CONT./
1714  010254  020040  042524  052123
1715  010262  020040  020040  041520
1716  010270  047440  020106  020040
1717  010276  047527  042122  020040
1718  010304  020040  047507  042117
1719  010312  020040  020040  040502
1720  010320  020104  020040  020040
1721  010326  047503  052116  020056
1722  010334  020040  047503  052116
1723  010342  004456  020040  047503
1724  010350  052116  006456     012
1725  010355     040  020040  020040          .ASCIZ  /       NO      JSR     NO      DATA    DATA    RHCS1   RHDS1   RHER1/
1726  010362  020040  047040  020117
1727  010370  020040  020040  045040
1728  010376  051123  020040  020040
1729  010404  047040  020117  020040
1730  010412  020040  042040  052101
1731  010420  020101  020040  042040
1732  010426  052101  020101  020040
1733  010434  051040  041510  030523
1734  010442  020040  051040  042110
1735  010450  030523  020040  051040
1736  010456  042510  030522     000
1737  010463     120  020103  020040  DH33:  .ASCII  /PC      TEST    PC OF   WORD    GOOD    CONT.   CONT.   CONT.   /<15><12
1738  010470  020040  052040  051505
1739  010476  020124  020040  050040
1740  010504  020103  043117  020040
1741  010512  053440  051117  020104
1742  010520  020040  043440  047517
1743  010526  020104  020040  041440
1744  010534  047117  027124  020040
1745  010542  041440  047117  027124
1746  010550  020040  041440  047117
1747  010556  027124  006411     012
1748  010563     040  020040  020040          .ASCIZ  /       NO      JSR     NO      DATA    RHCS1   RHDS1   RHER1   /
1749  010570  020040  047040  020117
1750  010576  020040  020040  045040
1751  010604  051123  020040  020040
1752  010612  047040  020117  020040
1753  010620  020040  042040  052101
1754  010626  020101  020040  051040
1755  010634  041510  030523  020040
1756  010642  051040  042110  030523
1757  010650  020040  051040  042510
1758  010656  030522  000011
```

```
 1759  010662  041520  020040  020040  DH34:   .ASCII  /PC      TEST    GOOD    GOOD    WRITTEN WRITTEN DATA/<15><12>
 1760  010670  020040  042524  052123
 1761  010676  020040  020040  047507
 1762  010704  042117  020040  020040
 1763  010712  047507  042117  020040
 1764  010720  020040  051127  052111
 1765  010726  042524  020116  051127
 1766  010734  052111  042524  020116
 1767  010742  040504  040524  005015
 1768  010750  020040  020040  020040          .ASCIZ  /       NO      ECC1    ECC2    ECC1    ECC2    USED/
 1769  010756  020040  047516  020040
 1770  010764  020040  020040  041505
 1771  010772  030503  020040  020040
 1772  011000  041505  031103  020040
 1773  011006  020040  041505  030503
 1774  011014  020040  020040  041505
 1775  011022  031103  020040  020040
 1776  011030  051525  042105     000
 1777  011035     120  020103  020040  DH35:   .ASCII  /PC      TEST    GOOD    GOOD    PATTERN POSITON GOOD    RHER1   /<15><12
 1778  011042  020040  052040  051505
 1779  011050  020124  020040  043440
 1780  011056  047517  020104  020040
 1781  011064  043440  047517  020104
 1782  011072  020040  050040  052101
 1783  011100  042524  047122  050040
 1784  011106  051517  052111  047117
 1785  011114  043440  047517  020104
 1786  011122  020040  051040  042510
 1787  011130  030522  006411     012
 1788  011135     040  020040  020040          .ASCIZ  /       NO      ECC1    ECC2    REG.    REG.    POSITON REG./
 1789  011142  020040  047040  020117
 1790  011150  020040  020040  042440
 1791  011156  041503  020061  020040
 1792  011164  042440  041503  020062
 1793  011172  020040  051040  043505
 1794  011200  020056  020040  051040
 1795  011206  043505  020056  020040
 1796  011214  050040  051517  052111
 1797  011222  047117  051040  043505
 1798  011230  000056
 1799  011232  041520  020040  020040  DH36:   .ASCII  /PC      TEST    PC OF   RHMR    POSITON PATTERN/<15><12>
 1800  011240  020040  042524  052123
 1801  011246  020040  020040  041520
 1802  011254  047440  020106  020040
 1803  011262  044122  051115  020040
 1804  011270  020040  047520  044523
 1805  011276  047524  020116  040520
 1806  011304  052124  051105  006516
 1807  011312     012
 1808  011313     040  020040  020040          .ASCIZ  /       NO      JSR     CONT.   REG.    REG./
 1809  011320  020040  047040  020117
 1810  011326  020040  020040  045040
 1811  011334  051123  020040  020040
 1812  011342  041440  047117  027124
 1813  011350  020040  051040  043505
 1814  011356  020056  020040  051040
```

```
1815  011364  043505  000056
1816  011370  041520  020040  020040  DH37:   .ASCII  /PC      TEST    POSITON POSITON GOOD    GOOD    PATTERN DATA    N-CODE/
1817  011376  020040  042524  052123
1818  011404  020040  020040  047520
1819  011412  044523  047524  020116
1820  011420  047520  044523  047524
1821  011426  020116  047507  042117
1822  011434  020040  020040  047507
1823  011442  042117  020040  020040
1824  011450  040520  052124  051105
1825  011456  020116  040504  040524
1826  011464  020040  020040  026516
1827  011472  047503  042504  005015
1828  011500  020040  020040  020040          .ASCIZ  /       NO      ECC     GOOD    ECC1    ECC2    ECC     ENVLOPE ZEROS
1829  011506  020040  047516  020040
1830  011514  020040  020040  041505
1831  011522  020103  020040  020040
1832  011530  047507  042117  020040
1833  011536  020040  041505  030503
1834  011544  020040  020040  041505
1835  011552  031103  020040  020040
1836  011560  041505  020103  020040
1837  011566  020040  047105  046126
1838  011574  050117  020105  042532
1839  011602  047522  004523     000
1840
1841  011607     120  020103  020040  DH40:   .ASCII  /PC      TEST    FAILING CONT.   CONT.   CONT.   CONT./<15><12>
1842  011614  020040  052040  051505
1843  011622  020124  020040  043040
1844  011630  044501  044514  043516
1845  011636  041440  047117  027124
1846  011644  020040  041440  047117
1847  011652  027124  020040  041440
1848  011660  047117  027124  020040
1849  011666  041440  047117  027124
1850  011674  005015
1851  011676  020040  020040  020040          .ASCIZ  /       NO      REG ADR RHCS1   RHCS2   RHDS1   RHER1/
1852  011704  020040  047516  020040
1853  011712  020040  020040  042522
1854  011720  020107  042101  020122
1855  011726  044122  051503  020061
1856  011734  020040  044122  051503
1857  011742  020062  020040  044122
1858  011750  051504  020061  020040
1859  011756  044122  051105  000061
1860
1861                                    .EVEN
1862
1863  011764  001116  017330  045530  DT1:    .WORD   $ERRPC,TSTNM,REGADR,$GDDAT,$BDDAT,0
1864  011772  001124  001126  000000
1865  012000  001116  017330  052770  DT2:    .WORD   $ERRPC,TSTNM,ERWORD,$GDDAT,0
1866  012006  001124  000000
1867  012012  001116  017330  052770  DT3:    .WORD   $ERRPC,TSTNM,ERWORD,$GDDAT,$BDDAT,0
1868  012020  001124  001126  000000
1869
1870  012026  001116  017330  001122  DT11:   .WORD   $ERRPC,TSTNM,$BDADR,CS1,CS2,DS1,ER1,0
```

```
1871  012034  015032  015030  015054
1872  012042  015034  000000
1873  012046  001116  017330  001122  DT14:   .WORD   $ERRPC,TSTNM,$BDADR,$BDDAT,CS1,CS2,DS1,ER1,0
1874  012054  001126  015032  015030
1875  012062  015054  015034  000000
1876  012070  001116  017330  001200  DT15:   .WORD   $ERRPC,TSTNM,$TMP1,0
1877  012076  000000
1878  012100  001116  017330  001204  DT16:   .WORD   $ERRPC,TSTNM,$TMP3,$TMP1,$TMP0,$BDDAT,0
1879  012106  001200  001176  001126
1880  012114  000000
1881  012116  001116  017330  015026  DT17:   .WORD   $ERRPC,TSTNM,BA,DB,WC,CS1,CS2,0
1882  012124  015022  015024  015032
1883  012132  015030  000000
1884
1885  012136  001116  017330  015034  DT20:   .WORD   $ERRPC,TSTNM,ER1,ER2,ER3,AS,DS1,0
1886  012144  015040  015046  015050
1887  012152  015054  000000
1888  012156  001116  017330  015032  DT21:   .WORD   $ERRPC,TSTNM,CS1,AS,DS1,0
1889  012164  015050  015054  000000
1890  012172  001116  017330  001124  DT22:   .WORD   $ERRPC,TSTNM,$GDDAT,$BDDAT,0
1891  012200  001126  000000
1892  012204  001116  017330  015036  DT24:   .WORD   $ERRPC,TSTNM,DST,$BDDAT,$TMP1,$TMP2,$TMP3,0
1893  012212  001126  001200  001202
1894  012220  001204  000000
1895  012224  001116  017330  015132  DT26:   .WORD   $ERRPC,TSTNM,PCJSR,$BDADR,CS1,CS2,DS1,ER1,0
1896  012232  001122  015032  015030
1897  012240  015054  015034  000000
1898  012246  001116  017330  015132  DT27:   .WORD   $ERRPC,TSTNM,PCJSR,REGADR,$GDDAT,$BDDAT,0
1899  012254  045530  001124  001126
1900  012262  000000
1901  012264  001116  017330  015132  DT30:   .WORD   $ERRPC,TSTNM,PCJSR,REGADR,$GDDAT,$BDDAT,0
1902  012272  045530  001124  001126
1903  012300  000000
1904
1905  012302  001116  017330  052770  DT31:   .WORD   $ERRPC,TSTNM,ERWORD,$GDDAT,$BDDAT,CS1,DS1,ER1,0
1906  012310  001124  001126  015032
1907  012316  015054  015034  000000
1908  012324  001116  017330  015132  DT32:   .WORD   $ERRPC,TSTNM,PCJSR,ERWORD,$GDDAT,$BDDAT,CS1,DS1,ER1,0
1909  012332  052770  001124  001126
1910  012340  015032  015054  015034
1911  012346  000000
1912  012350  001116  017330  015132  DT33:   .WORD   $ERRPC,TSTNM,PCJSR,ERWORD,$GDDAT,CS1,DS1,ER1,0
1913  012356  052770  001124  015032
1914  012364  015054  015034  000000
1915  012372  001116  017330  050460  DT34:   .WORD   $ERRPC,TSTNM,GECC1,GECC2,WECC1,WECC2,DISK,0
1916  012400  050462  055566  055570
1917  012406  054566  000000
1918  012412  001116  017330  050460  DT35:   .WORD   $ERRPC,TSTNM,GECC1,GECC2,EC2,EC1,POSITI,ER1,0
1919  012420  050462  015064  015062
1920  012426  050472  015034  000000
1921  012434  001116  017330  015132  DT36:   .WORD   $ERRPC,TSTNM,PCJSR,MR,EC1,EC2,0
1922  012442  015052  015062  015064
1923  012450  000000
1924  012452  001116  017330  015062  DT37:   .WORD   $ERRPC,TSTNM,EC1,POSITI,GECC1,GECC2,EC2,DATENV,ZCODE,0
1925  012460  050472  050460  050462
1926  012466  015064  050476  050500
```

```
1927  012474  000000
1928  012476  001116  017330  001122  DT40:   .WORD   $ERRPC,TSTNM,$BDADR,CS1,CS2,DS1,ER1,0
1929  012504  015032  015030  015054
1930  012512  015034  000000
1931
1932
1933
1934
1935  012516      000      000      000  DF1:    .BYTE   0,0,0,0,0
1936  012521      000      000
1937  012523      000      000      001  DF2:    .BYTE   0,0,1,0
1938  012526      000
1939  012527      000      000      001  DF3:    .BYTE   0,0,1,0,0
1940  012532      000      000
1941
1942  012534      000      000      000  DF11:   .BYTE   0,0,0,0,0,0,0
1943  012537      000      000      000
1944  012542      000
1945  012543      000      000      000  DF14:   .BYTE   0,0,0,0,0,0,0,0
1946  012546      000      000      000
1947  012551      000      000
1948  012553      000      000      000  DF15:   .BYTE   0,0,0
1949  012556      000      000      000  DF16:   .BYTE   0,0,0,0,0
1950  012561      000      000
1951  012563      000      000      000  DF17:   .BYTE   0,0,0,0,0,0,0
1952  012566      000      000      000
1953  012571      000
1954  012572      000      000      000  DF20:   .BYTE   0,0,0,0,0,0,0
1955  012575      000      000      000
1956  012600      000
1957
1958  012601      000      000      000  DF21:   .BYTE   0,0,0,0,0
1959  012604      000      000
1960  012606      000      000      000  DF22:   .BYTE   0,0,0,0
1961  012611      000
1962  012612      000      000      000  DF24:   .BYTE   0,0,0,0,0,0,0
1963  012615      000      000      000
1964  012620      000
1965  012621      000      000      000  DF26:   .BYTE   0,0,0,0,0,0,0,0
1966  012624      000      000      000
1967  012627      000      000
1968  012631      000      000      000  DF27:   .BYTE   0,0,0,0,0,0
1969  012634      000      000      000
1970  012637      000      000      000  DF30:   .BYTE   0,0,0,0,0,0
1971  012642      000      000      000
1972
1973  012645      000      000      001  DF31:   .BYTE   0,0,1,0,0,0,0,0
1974  012650      000      000      000
1975  012653      000      000
1976  012655      000      000      000  DF32:   .BYTE   0,0,0,1,0,0,0,0,0
1977  012660      001      000      000
1978  012663      000      000      000
1979  012666      000      000      000  DF33:   .BYTE   0,0,0,1,0,0,0,0
1980  012671      001      000      000
1981  012674      000      000
1982  012676      000      000      000  DF34:   .BYTE   0,0,0,0,0,0,0
```

```
1983  012701  000     000     000
1984  012704  000
1985  012705  000     000     000     DF35:   .BYTE   0,0,0,0,0,0,0,0
1986  012710  000     000     000
1987  012713  000     000
1988  012715  000     000     000     DF36:   .BYTE   0,0,0,0,0,0
1989  012720  000     000     000
1990  012723  000     000     000     DF37:   .BYTE   0,0,0,0,0,0,0,0,0
1991  012726  000     000     000
1992  012731  000     000     000
1993  012734  000     000     000     DF40:   .BYTE   0,0,0,0,0,0,0
1994  012737  000     000     000
1995  012742  000
1996
1997          012744                          .EVEN
1998  012744  044122  030461  051040  EM42:   .ASCIZ  /RH11 REGISTER FAILED TO RESPOND TO A "TST" COMMAND/
1999  012752  043505  051511  042524
2000  012760  020122  040506  046111
2001  012766  042105  052040  020117
2002  012774  042522  050123  047117
2003  013002  020104  047524  040440
2004  013010  021040  051524  021124
2005  013016  041440  046517  040515
2006  013024  042116     000
2007
2008  013027     122  030510  020061  EM43:   .ASCIZ  /RH11 ILLEGAL REGISTER RESPONSE TEST/
2009  013034  046111  042514  040507
2010  013042  020114  042522  044507
2011  013050  052123  051105  051040
2012  013056  051505  047520  051516
2013  013064  020105  042524  052123
2014  013072     000
2015
2016  013073     115  053117  020105  EM44:   .ASCIZ  /MOVE BYTE TO WORD COUNT TEST/
2017  013100  054502  042524  052040
2018  013106  020117  047527  042122
2019  013114  041440  052517  052116
2020  013122  052040  051505  000124
2021
2022  013130  044502  020123  054502  EM45:   .ASCIZ  /BIS BYTE TO WORD COUNT/
2023  013136  042524  052040  020117
2024  013144  047527  042122  041440
2025  013152  052517  052116     000
2026
2027  013157     102  041511  041040  EM46:   .ASCIZ  /BIC BYTE TO WORD COUNT/
2028  013164  052131  020105  047524
2029  013172  053440  051117  020104
2030  013200  047503  047125  000124
2031
2032  013206  042522  044507  052123  EM47:   .ASCIZ  /REGISTER ADDRESS SELECT TEST/
2033  013214  051105  040440  042104
2034  013222  042522  051523  051440
2035  013230  046105  041505  020124
2036  013236  042524  052123     000
2037
2038  013243     116  047117  042440  EM50:   .ASCIZ  /NON EXISTANT DRIVE (NED) TEST/
```

```
2039  013250  044530  052123  047101
2040  013256  020124  051104  053111
2041  013264  020105  047050  042105
2042  013272  020051  042524  052123
2043  013300     000
2044
2045  013301     101  020123  042522  EM51:   .ASCIZ  /AS REGISTER TEST/
2046  013306  044507  052123  051105
2047  013314  052040  051505  000124
2048
2049  013322  052502  051523  040440  EM52:   .ASCIZ  /BUSS ADDRESS REGISTER DATA TEST/
2050  013330  042104  042522  051523
2051  013336  051040  043505  051511
2052  013344  042524  020122  040504
2053  013352  040524  052040  051505
2054  013360  000124
2055
2056  013362  052502  051523  040440  EM53:   .ASCIZ  /BUSS ADDRESS REGISTER MOVE BYTE/
2057  013370  042104  042522  051523
2058  013376  051040  043505  051511
2059  013404  042524  020122  047515
2060  013412  042526  041040  052131
2061  013420  000105
2062
2063  013422  044122  030461  044440  EM54:   .ASCIZ  /RH11 INTERRUPT FAILED TO OCCUR/
2064  013430  052116  051105  052522
2065  013436  052120  043040  044501
2066  013444  042514  020104  047524
2067  013452  047440  041503  051125
2068  013460     000
2069
2070  013461     122  051505  052105  EM55:   .ASCIZ  /RESET TEST/
2071  013466  052040  051505  000124
2072
2073  013474  054115  020106  044502  EM56:   .ASCIZ  /MXF BIT TEST/
2074  013502  020124  042524  052123
2075  013510     000
2076
2077  013511     125  044516  052502  EM57:   .ASCIZ  /UNIBUS PARITY BIT TEST/
2078  013516  020123  040520  044522
2079  013524  054524  041040  052111
2080  013532  052040  051505  000124
2081
2082  013540  047504  051505  051440  EM60:   .ASCIZ  /DOES SELECTING THE RH11 CLEAR THE UNIT SELECT REGISTER/
2083  013546  046105  041505  044524
2084  013554  043516  052040  042510
2085  013562  051040  030510  020061
2086  013570  046103  040505  020122
2087  013576  044124  020105  047125
2088  013604  052111  051440  046105
2089  013612  041505  020124  042522
2090  013620  044507  052123  051105
2091  013626     000
2092
2093  013627     104  042517  020123  EM61:   .ASCIZ  /DOES TRE GET SET BY UNIBUS PARITY ERROR/
2094  013634  051124  020105  042507
```

```
2095  013642  020124  042523  020124
2096  013650  054502  052440  044516
2097  013656  052502  020123  040520
2098  013664  044522  054524  042440
2099  013672  051122  051117     000
2100
2101  013677     116  047117  042440  EM62:   .ASCIZ  /NON EXISTANT DRIVE (NED) FAILED TO SET (TRE)/
2102  013704  044530  052123  047101
2103  013712  020124  051104  053111
2104  013720  020105  047050  042105
2105  013726  020051  040506  046111
2106  013734  042105  052040  020117
2107  013742  042523  020124  052050
2108  013750  042522  000051
2109
2110  013754  047516  020116  054105  EM63:   .ASCIZ  /NON EXISTANT MEMORY (NEM) FAILED TO SET (TRE)/
2111  013762  051511  040524  052116
2112  013770  046440  046505  051117
2113  013776  020131  047050  046505
2114  014004  020051  040506  046111
2115  014012  042105  052040  020117
2116  014020  042523  020124  052050
2117  014026  042522  000051
2118
2119  014032  047520  052122  051440  EM64:   .ASCIZ  /PORT SELECT FAILED TO CLEAR/
2120  014040  046105  041505  020124
2121  014046  040506  046111  042105
2122  014054  052040  020117  046103
2123  014062  040505  000122
2124
2125  014066  047503  052116  047522  EM65:   .ASCIZ  /CONTROLLER CLEAR FAILED TO CLEAR COMMAND REGISTER/
2126  014074  046114  051105  041440
2127  014102  042514  051101  043040
2128  014110  044501  042514  020104
2129  014116  047524  041440  042514
2130  014124  051101  041440  046517
2131  014132  040515  042116  051040
2132  014140  043505  051511  042524
2133  014146  000122
2134
2135  014150  042522  042523  042514  EM66:   .ASCIZ  /RESELECT FAILED TO CLEAR COMMAND REGISTER/
2136  014156  052103  043040  044501
2137  014164  042514  020104  047524
2138  014172  041440  042514  051101
2139  014200  041440  046517  040515
2140  014206  042116  051040  043505
2141  014214  051511  042524  000122
2142
2143  014222  047111  042524  051122  EM67:   .ASCIZ  /INTERRUPT CAUSED BY  RDY!IE BITS SET FAILED TO OCCUR/
2144  014230  050125  020124  040503
2145  014236  051525  042105  041040
2146  014244  020131  051040  054504
2147  014252  044441  020105  044502
2148  014260  051524  051440  052105
2149  014266  043040  044501  042514
2150  014274  020104  047524  047440
```

```
2151   014302   041503   051125      000
2152
2153   014307      111   020105   040506   EM70:   .ASCIZ  /IE FAILED TO CLEAR FOLLOWING AN INTERRUPT/
2154   014314   046111   042105   052040
2155   014322   020117   046103   040505
2156   014330   020122   047506   046114
2157   014336   053517   047111   020107
2158   014344   047101   044440   052116
2159   014352   051105   052522   052120
2160   014360      000
2161
2162   014361      101   020123   042522   EM71:   .ASCIZ  /AS REGISTER WRITE TEST/
2163   014366   044507   052123   051105
2164   014374   053440   044522   042524
2165   014402   052040   051505   000124
2166
2167
2168
2169   014410   041520   020040   020040   DH42:   .ASCII  /PC      TEST    ILLEGAL/<15><12>
2170   014416   020040   042524   052123
2171   014424   020040   020040   046111
2172   014432   042514   040507   006514
2173   014440      012
2174   014441      040   020040   020040           .ASCIZ  /        NO      ADDRESS/
2175   014446   020040   047040   020117
2176   014454   020040   020040   040440
2177   014462   042104   042522   051523
2178   014470      000
2179   014471      120   020103   020040   DH44:   .ASCII  /PC      TEST    REGISTR CONT    CONT/<15><12>
2180   014476   020040   052040   051505
2181   014504   020124   020040   051040
2182   014512   043505   051511   051124
2183   014520   041440   047117   020124
2184   014526   020040   041440   047117
2185   014534   006524      012
2186   014537      040   020040   020040           .ASCIZ  /        NO      ADDRESS GOOD    BAD/
2187   014544   020040   047040   020117
2188   014552   020040   020040   040440
2189   014560   042104   042522   051523
2190   014566   043440   047517   020104
2191   014574   020040   041040   042101
2192   014602      000
2193   014603      120   020103   020040   DH54:   .ASCII  /PC      TEST    REGISTR CONT/<15><12>
2194   014610   020040   052040   051505
2195   014616   020124   020040   051040
2196   014624   043505   051511   051124
2197   014632   041440   047117   006524
2198   014640      012
2199   014641      040   020040   020040           .ASCIZ  /        NO      ADDRESS/
2200   014646   020040   047040   020117
2201   014654   020040   020040   040440
2202   014662   042104   042522   051523
2203   014670      000
2204
2205            014672                    .EVEN
2206
```

```
 2207  014672  001116  017330  045530  DT42:   .WORD   $ERRPC,TSTNM,REGADR,0
 2208  014700  000000
 2209  014702  001116  017330  045530  DT44:   .WORD   $ERRPC,TSTNM,REGADR,$GDDAT,$BDDAT,0
 2210  014710  001124  001126  000000
 2211  014716  001116  017330  045530  DT54:   .WORD   $ERRPC,TSTNM,REGADR,$GDDAT,0
 2212  014724  001124  000000
```

```
2213  014730  000  000  000  DF42:  .BYTE  0,0,0
2214  014733  000  000  000  DF44:  .BYTE  0,0,0,0,0
2215  014736  000  000
2216  014740  000  000  000  DF54:  .BYTE  0,0,0,0
2217  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          DFS=    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)

                        EXT1=   1                       ;EXTENSION 1 (BIT #0)
                        EXT2=   2                       ;EXTENSION 2 (BIT #1)
                        EXT4=   4                       ;EXTENSION 3 (BIT #2)
                        EXT10=  10                      ;EXTENSION 4 (BIT #3)
                        EXT20=  20                      ;EXTENSION 5 (BIT #4)
                        EXT40=  40                      ;EXTENSION 6 (BIT #5)
                        SC1=    100                     ;SECTOR COUNT FIELD 0 (BIT #6)
                        SC2=    200                     ;SECTOR COUNT FIELD 1 (BIT #7)
                        SC4=    400                     ;SECTOR COUNT FIELD 2 (BIT #8)
                        SC10=   1000                    ;SECTOR COUNT FIELD 3 (BIT #9)
                        SC20=   2000                    ;SECTOR COUNT FIELD 4 (BIT #10)
                        TRK1=   4000                    ;TRACK FIELD 1 (BIT #11)
                        TRK2=   10000                   ;TRACK FIELD 2 (BIT #12)
                        TRK4=   20000                   ;TRACK FIELD 3 (BIT #13)
                        TRK10=  40000                   ;TRACK FIELD 4 (BIT #14)
```

```
                100000          TRK20=  100000                  ;TRACK FIELD 5 (BIT #15)

                                ;ERROR REGISTER #2 (RHER2) (#10)

                000001          WCU=    1                       ;WRITE CURRENT UNSAFE (BIT #0)
                000002          CSF=    2                       ;CURRENT SINK FAILURE (BIT #1)
                000004          WSU=    4                       ;WRITE SELECT UNSAFE (BIT #2)
                000010          CSU=    10                      ;CURRENT SWITCH UNSAFE (BIT #3)
                000020          MSE=    20                      ;MOTOR SEQUENCE ERROR (BIT #4)
                000040          TDF=    40                      ;TRANSITIONS DETECTOR FAILURE (BIT #5)
                000100          TUF=    100                     ;TRANSITIONS UNSAFE (BIT #6)
                000200          FEN=    200                     ;FAILSAFE ENABLED (BIT #7)
                000400          WRU=    400                     ;WRITE READY UNSAFE (BIT #8)
                001000          MHS=    1000                    ;MULTIPLE HEAD SELECT (BIT #9)
                002000          NHS=    2000                    ;NO HEAD SELECTION (BIT #10)
                004000          IXE=    4000                    ;INDEX ERROR (BIT #11)
                010000          VU30=   10000                   ;30VOLT UNSAFE (BIT #12)
                020000          PLU=    20000                   ;PLO UNSAFE (BIT #13)
                100000          ACU=    100000                  ;ACUNSAFE (BIT #15)

                                ;OFFSET REGISTER (RHOF) (#11)

                000001          OF25=   1                       ;OFFSET 25 MICRO INCHES (BIT #0)
                000002          OF50=   2                       ;OFFSET 50 MICRO INCHES (BIT #1)
                000004          OF100=  4                       ;OFFSET 100 MICRO INCHES (BIT #2)
                000010          OF200=  10                      ;OFFSET 200 MICRO INCHES (BIT #3)
                000020          OF400=  20                      ;OFFSET 400 MICRO INCHES (BIT #4)
                000040          OF800=  40                      ;OFFSET 800 MICRO INCHES (BIT #5)

                000200          OFREV=  200                     ;OFFSET NEGATIVE (REVERSE) (BIT #7)
                002000          HCI=    2000                    ;HEADER COMPARE INHIBIT (BIT #10)
                004000          ECI=    4000                    ;ERROR CORRECTION CODE INHIBIT (BIT #11)
                010000          FMT22=  10000                   ;FORMAT BIT (BIT #12


                                ;DESIRED CYLINDER ADDRESS (RHCA) (#12)
                                ;EACH BIT IS CALLED BY BIT NUMBER.



                                ;CURRENT CYLINDER ADDRESS (RHCC) (#13)
                                ;EACH BIT IS CALLED BY BIT NUMBER



                                ;SERIAL NUMBER REGISTER (RHSN) (#14)
                                ;EACH IS CALLED BY BIT NUMBER
```

```
                                ;ERROR REGISTER #03 (RHER3) (#15)

000001                          PSU=    1               ;PACK SPEED UNSAFE (BIT #0)
000002                          VUF=    2               ;VELOCITY UNSAFE (BIT #1)
000010                          UWR=    10              ;ANY UNSAFE EXCEPT READ/WRITE (BIT #3)
000020                          PRE=    20              ;DISK PACK ROTATION ERROR (BIT #4)
000040                          ACL=    40              ;AC LOW (BIT #5)
000100                          DCL=    100             ;DC LOW (BIT #6)
040000                          SKI=    40000           ;SEEK INCOMPLETE (BIT #14)
100000                          OCYL=   100000          ;OFF CYLINDER (BIT #15)

                                ;ECC POSITION REGISTER (RHEC1) (#16)
                                ;EACH BIT IS CALLED BY BIT NUMBER



                                ;ECC PATTERN REGISTER (RHEC2) (#17)
                                ;EACH BIT IS CALLED BY BIT NUMBER
```

```
                                .SBTTL  REGISTER ADDRESSES


                                ;RP04/5/6 VECTOR ADDRESS
014744  000254                  RPVEC:  254                     ;RP04/5/6 VECTOR ADDRESS


                                ;NOTE:  THE CONTENTS OF THESE LOCATIONS WILL BE DIFFRENT
                                ;       IF THE "CHANGE BASE ADDRESS" ROUTINE IS USED.
                                ;       THIS ROUTINE STARTS AT LOCATION TAGGED "BASECH"

014746  176722                  RHDB:   176722                  ;DATA BUFFER
014750  176702                  RHWC:   176702                  ;WORD COUNT
014752  176704                  RHBA:   176704                  ;BUS ADDRESS
014754  176710                  RHCS2:  176710                  ;CONTROL AND STATUS
014756  176700                  RHCS1:  176700                  ;CONTROL AND STATUS 1 SEE NOTE ABOVE
014760  176714                  RHER1:  176714                  ;ERROR #1 SEE NOTE ABOVE
014762  176706                  RHDST:  176706                  ;DESIRED SECTOR / TRACK ADDRESS
014764  176740                  RHER2:  176740                  ;ERROR #2
014766  176732                  RHOF:   176732                  ;OFFSET
014770  176734                  RHCA:   176734                  ;DESIRED CYLINDER ADDRESS
014772  176742                  RHER3:  176742                  ;ERROR #3
014774  176716                  RHAS:   176716                  ;ATTENTION SUMMARY SEE NOTE ABOVE
014776  176724                  RHMR:   176724                  ;MAINTAINABILITY
015000  176712                  RHDS1:  176712                  ;DRIVE STATUS
015002  176726                  RHDT:   176726                  ;DRIVE TYPE
015004  176730                  RHSN:   176730                  ;SERIAL NUMBER SEE NOTE ABOVE
015006  176744                  RHEC1:  176744                  ;ECC POSITION
015010  176746                  RHEC2:  176746                  ;ECC PATTERN
015012  176720                  RHLA:   176720                  ;LOOK AHEAD
015014  176736                  RHCC:   176736                  ;CURRENT CYLINDER ADDRESS

                                ;ADDITIONAL REGISTERS LOCATED IN THE RH70 CONTROLLER

015016  176752                  RHCS3:  176752                  ;CONTROL AND STATUS REG #3
015020  176750                  RHBAE:  176750                  ;BUS ADDRESS EXTENSION REGISTER
```

```
2491
2492                                ;THE FOLLOWING LOCATIONS ARE RESERVED FOR REGISTER SAVES
2493                                ;ANY TIME THERE IS AN ERROR ALL THESE WILL BE FILLED
2494                                ;ONLY SOME MAY BE PRINTED BUT ALL WILL BE FILLED TRUE
2495                                ;FOR THE TIME JUST AFTER THE "ERROR" ERROR COMMAND
2496
2497  015022  000000                DB:     0                       ;DATA BUFFER
2498  015024  000000                WC:     0                       ;WORD COUNT
2499  015026  000000                BA:     0                       ;BUS ADDRESS
2500  015030  000000                CS2:    0                       ;CONTROL AND STATUS 2
2501
2502
2503  015032  000000                CS1:    0                       ;CONTROL AND STATUS 1
2504  015034  000000                ER1:    0                       ;ERROR #1
2505  015036  000000                DST:    0                       ;DESIRED SECTOR/TRACK ADDRESS
2506  015040  000000                ER2:    0                       ;ERROR #2
2507  015042  000000                OF:     0                       ;OFFSET
2508  015044  000000                CA:     0                       ;DESIRED CYLINDER ADDRESS
2509  015046  000000                ER3:    0                       ;ERROR #3
2510  015050  000000                AS:     0                       ;ATTENTION SUMMARY
2511  015052  000000                MR:     0                       ;MAINTAINABILITY
2512  015054  000000                DS1:    0                       ;DRIVE STATUS
2513  015056  000000                DT:     0                       ;DRIVE TYPE
2514  015060  000000                SN:     0                       ;SERIAL NUMBER
2515  015062  000000                EC1:    0                       ;ECC POSITION
2516  015064  000000                EC2:    0                       ;ECC PATTERN
2517  015066  000000                LA:     0                       ;LOOK-AHEAD
2518  015070  000000                CC:     0                       ;CURRENT CYLINDER ADDRESS
2519
2520
```

```
                        ;FLAGS AND INTERNAL PROGRAM CONTROL WORDS

015072  000010  UNITS:  .BLKW   8.          ;THIS IS FILLED WITH -1
015112  000000  UNIT:   .WORD   0           ;UNIT UNDER TEST
015114  000000  NOUNIT: .WORD   0           ;NUMBER OF UNITS PRESENT
                                            ;USED TO KEEP TRACK OF UNIT UNDER TEST
015116  000000  NUNIT:  .WORD   0           ;USED TO DETERMIN IF THERE ARE MORE
                                            ;THAN ONE UNIT
015120  000000  SELECT: .WORD   0           ;ALL ONES INDICATE UNIT TO BE SELECTED
015122  000000  UNITSL: .WORD   0           ;UNIT NO. SELECTED

015124  000000  ERFLG$: 0                   ;ERROR FLAG

015126  000000  SAVDT:  0                   ;SAVE DRIVE TYPE REGISTER
                                            ;FOR COMPARISON IN DRIVE CLEAR TEST
                                            ;AND RH INIT TEST
015130  000000  SAVSN:  0                   ;SAVE SERIAL NUMBER REGISTER
                                            ;FOR COMPARISON IN DRIVE CLEAR TEST
                                            ;AND RH INIT TEST

015132  000000  PCJSR:  0                   ;SAVE PC OF JSR WHICH GAVE THE ERROR

015134  000000  ATTENT: 0                   ;ATTENTION BIT FOR PRESENT UNIT
015136  000000  TOTALAT:       0           ;TOTAL ATTENTION BITS

015140  000000  TMPILL: 0                   ;TEMPORARY ILLEGAL FUNCTION

015142  000000  TSECC:  0                   ;FLAG TO SAY IF ECC TEST OR NOT
                                            ;WHEN =177777 IT IS AN ECC TEST
                                            ;WHEN =0IT IS NOT AN ECC TEST

015144  000000  TESDTE: 0                   ;FLAG TO SAY IF DRIVE TIMING ERROR OR NOT
                                            ;WHEN = 177777 IT IS A DTE TEST
                                            ;WHEN = 0      IT IS NOT A DTE TEST

015146  000000  TAGDTE: 0                   ;TEMPORARY TAG USED IN DRIVE TIMING
                                            ;ERROR TEST
```

```
                                        ;FUNCTION EQUATES
                                        ;TABLE OF COMMAND FUNCTIONS FOR RHCS1
                                        ;THEN "GO" BIT HAS TO BE SET
015150                          FUTABL:
015150  000000                  NOPERA: 0                       ;NO OPERATION
015152  000002                  UNLOAD: 2                       ;UNLOAD (STAND BY)
015154  000006                  RECALI: 6                       ;RECALIBRATE
015156  000010                  DCLEAR: 10                      ;DRIVE CLEAR
015160  000012                  RELEAS: 12                      ;RELEASE (DUAL-PORT OPERATION)
015162  000030                  SERCH:  30                      ;SEARCH COMMAND
015164  000050                  WRCHEK: 50                      ;WRITE CHECK DATA
015166  000052                  WRCHDT: 52                      ;WRITE CHECK HEADER AND DATA
015170  000060                  WRIDAT: 60                      ;WRITE DATA
015172  000062                  WRIFOR: 62                      ;WRITE HEADER AND DATA (FORMAT)
015174  000070                  READAT: 70                      ;READ DATA
015176  000072                  REFOR:  72                      ;READ HEADER AND DATA
015200  000004                  SEECOM: 4                       ;SEEK COMMAND
015202  000014                  OFSETC: 14                      ;OFFSET COMMAND
015204  000016                  RETCL:  16                      ;RETURNTO CENTERLINE
015206  000022                  PKACK:  22                      ;PACK ACKNOWLEDGE
015210  000020                  READIN: 20                      ;READ IN
015212  000000                  ILLEGL: .WORD                   ;COMPUTED ILLEGAL FUNCTION
                                        ;DATA BUFFER FOR READ WRITE
015220  000422                  WRFROM: .BLKW   274.            ;WRITE FROM THIS BUFFER
016264  000422                  REINTO: .BLKW   274.            ;READ INTO THIS BUFFER
017330  000000                  TSTNM:  0                       ;TEST NUMBER
017332  000000                  FIRST:  0                       ;IF ZERO WILL TYPE HEADER
                                                                ;IF ONES WILL NOT TYPE HEADER
017334  000000                  RH70:   0                       ;FLAG = 1 FOR RH70 CONTROLLER
                                                                ;FLAG = 0 FOR RH11
                                        ;TABLE FOR ATTENTION BITS
                                        ;ATTENTION TABLE
017336    001     002     004   ATABLE: .BYTE   1,2,4,10,20,40,100,200
017341    010     020     040
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


017604  012737  000000  177776  STARTA: MOV     #0,PS           ;SET PROCESSOR STATUS TO 0
017612  012777  057302  175124          MOV     #RPVECT,@RPVEC  ;THIS IS FOR UNTIMELY DRIVE INTERRUPTS
017620  004737  060346                  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$:


020662  012737  177777  017332  SND1:   MOV     #-1,@#FIRST             ;NEXT TIME DO NOT GIVE HEADER

020720  032777  010000  160212  RH70CK: BIT     #SW12,@SWR              ;LOOK TO SEE IF USING RH70
020726  001403                  BEQ     3$              ;IF SW12 = 0, SKIP NEXT
020730  012737  000001  017334          MOV     #1,@#RH70               ;IF SW12 = 1, CU IS AN RH70

020736  005737  015120          3$:     TST     @#SELECT                ;200 START?
021016  104412                  RDOCT
021020  042716  177770                  BIC     #177770,(SP)            ;ONLY KEEP LAST 3 BITS
021024  011637  015112                  MOV     (SP),@#UNIT             ;SAVE UNIT TO BE TESTED
021030  012637  015122                  MOV     (SP)+,@#UNITSL          ;SAVE UNIT TO BE TESTED
```

```
2656
2657  021044  012706  001000                MOV     #STACK, SP          ;SET UP STACK POINTER
2658  021056  012737  061616  000030        MOV     #REGSA1,@#EMTVEC    ;ERROR VECTOR SO THAT
2659                                                                    ;NO REGISTERS ARE SAVED
2660  021064  012737  021112  000004        MOV     #2$,@#ERRVEC        ;SET UP FOR BUS TIMEOUT
2661  021072  012700  000024                MOV     #24,R0              ;THERE ARE 24 REG TO TEST
2662  021076  012701  014746                MOV     #RHDB,R1            ;R1 NOW HAS ADDR OF ADDR OF FIRST REG.
2663  021102  013102                1$:     MOV     @(R1)+,R2           ;READ HARDWARE REG.
2664  021104  005300                        DEC     R0                  ;COUNT DOWN
2665  021106  001375                        BNE     1$                  ;BRANCH IF 24 NOT DONE
2666  021110  000454                        BR      3$                  ;BRANCH IF 24 DONE
2667  021112  012737  000006  000004  2$:   MOV     #ERRVEC+2,@#ERRVEC ;RESTORE TRAP CATCHER
2668  021120  022626                        CMP     (SP)+,(SP)+         ;CLEAN STACK
2669  021122  016137  177776  001200        MOV     -2(R1), $TMP1       ;STORE FAILING REG ADDR
2670  021130  104015                        ERROR   15                  ;REGISTER NON EXISTANT
2671  021132  032777  020000  160000        BIT     #SW13,@SWR          ;INHIBIT ERROR PRINTOUT ?
2672  021140  001036                        BNE     4$                  ;BRANCH IF YES
2673  021226  012746  000204                MOV     #ADDMOD,-(SP)       ;GET READY TO TYPE STARTING ADDRESS
2674
2675                                                                    ;OF "CHANGE OF BASE ADDRESS" ROUTINE
2676  021232  104402                        TYPOC
2677  021234  000000                        HALT                        ;STOP TO FORCE THE RESTART!
2678  021236  000137  044560        4$:     JMP     @#$EOP              ;GO TO END OF PROGRAM ---------------->
2679
2680  021242  012737  021316  000004  3$:   MOV     #TRP,@#4            ;INITIALIZE VECTOR
2681  021250  005737  015020                TST     @#RHBAE             ;ADDRESS RPBAE (RH11/RH70?)
2682  021254  005237  017334                INC     @#RH70              ;FOUND AN RH70
2683  021314  000420                        BR      RTN                 ;SET MASK AND GET OUT
2684  021316  005726                TRP:    TST     (SP)+               ;NORMALIZE THE
2685  021320  005726                        TST     (SP)+               ;STACK
2686  021356  012737  061606  000030  RTN:   MOV     #$ERROR,@#EMTVEC    ;RESTORE ERROR VECTOR
2687                                                                    ;SO THAT REGISTERS ARE SAVED
2688  021364  012737  000006  000004        MOV     #ERRVEC+2,@#ERRVEC ;RESTORE TRAP CATCHER
2689
2690
2691
```

```
2692
2693  021402  012706  001000                  MOV     #STACK,SP       ;RESET STACK
2694
2695  021426  013737  014754  021442          MOV     @#RHCS2,@#UN+2
2696  021434  004537  045532                  JSR     R5,@#BITST      ;TEST BITS IN REGISTER
2697  021440  020017                  UN:     20017                   ;ONLY THESE BITS ARE TEST READ/WRITE
2698  021442  000000                          .WORD   0               ;ADDRESS OF REG. BEING TESTED
2699  021444  104001                          ERROR   1               ;IN CORRECT DATA RECEIVED
2700  021446  000207                          RTS     PC              ;RETURN TO BLT3 ROUTINE
2701
2702
2703
2704
2705
2706
2707
2708  021466  013701  014774                  MOV     @#RHAS,R1       ;R1 HAS ADDRESS OF RHAS
2709  021472  012711  177777                  MOV     #-1,@R1         ;THIS CLEARS RHAS (SURPRISED!)
2710  021476  011137  001126                  MOV     @R1,@#$BDDAT    ;TEST DATA
2711  021502  105737  001126                  TSTB    @#$BDDAT
2712  021510  005037  001124                  CLR     @#$GDDAT        ;GOOD DATA
2713  021514  010137  045530                  MOV     R1,@#REGADR     ;FAILING REG. RHAS
2714  021520  104001                          ERROR   1               ;RHAS DOES NOT CLEAR
2715                                                                  ;WITH ONES MOVED INTO IT
2716
```

```
                                        RESET                   ;START WITH AN INIT
021534  004737  060346                  JSR     PC,@#$TKINT     ;INITILIZE TTY KEYBOARD

021540  032777  020000  157372          BIT     #SW13,@SWR      ;INHIBIT ERROR TYPEOUT ?
021546  001026                          BNE     4$              ;SKIP NEXT IF SO
021624  013701  014774          4$:     MOV     @#RHAS,R1       ;R1 HAS ADDR. OF RHAS
021630  013702  014754                  MOV     @#RHCS2,R2      ;R2 HAS ADDR. OF RHCS2
021634  005012                          CLR     @R2             ;CLEAR RHCS2
021636  012700  000010                  MOV     #8.,R0          ;COUNT
021642  013704  014760                  MOV     @#RHER1,R4      ;R4 HAS ADDR. OF RHER1

021646  012714  177777          1$:     MOV     #-1,@R4         ;MOVE ERRORS INTO RHER1
021652  005212                          INC     @R2             ;INCREMENT UNIT NO.
021654  005300                          DEC     R0              ;COUNT
021656  001373                          BNE     1$              ;BRANCH IF 8 NOT DONE
021660  111137  015136                  MOVB    @R1,@#TOTALAT   ;SAVE TOTAL ATTENTION
                                                                ;USED IN DRIVE CLEAR TEST
021664  105037  015137                  CLRB    @#TOTALAT+1     ;CLEAR UPPER BYTE
021670  105711                          TSTB    @R1             ;TEST FOR ANY DRIVES PRESENT
021672  001402                          BEQ     2$              ;NONE RESPONDING - TYPE THE MESSAGE
021674  000137  022264                  JMP     XE2             ;SOME THERE - GO FILL "UNITS" TABLE

021700  032777  020000  157232  2$:     BIT     #SW13,@SWR      ;INHIBIT ERROR TYPE OUT?
021706  001402                          BEQ     3$              ;"NO DRIVES" MESSAGE IF NO
021710  000137  022622                  JMP     SELTST          ;CHECK FOR SELECTED UNIT START AND LOAD
                                                                ;"UNITS" TABLE WITH DESIRED DRIVE IF SO

021714                          3$:

022260  000137  044560                  JMP     @#$EOP          ;GO OUT---------------------------------->


                                        ;*SET UP UNITS TABLE

022264                          XE2:
022264  012700  000010          2$:     MOV     #8.,R0          ;COUNTER
022270  012703  015072                  MOV     #UNITS,R3       ;POINTER
022274  012723  177777          3$:     MOV     #-1,(R3)+       ;PRESET BLOCK TO ALL ONES
022300  005300                          DEC     R0              ;COUNT
022302  001374                          BNE     3$              ;BRANCH IF 8 NOT DONE
022304  012703  015072                  MOV     #UNITS,R3       ;POINTER
022310  005005                          CLR     R5
022312  005037  015114                  CLR     @#NOUNIT        ;NO. OF UNITS PRESENT
022316  012700  000010                  MOV     #8.,R0          ;COUNTER
022322  011137  001176                  MOV     @R1,@#$TMPO     ;TEMPORARY STORAGE
022326  006037  001176          4$:     ROR     @#$TMPO         ;SET CARRY IF ONE IN  0 BIT

022332  103120                          BCC     5$
022334  010577  172414                  MOV     R5,@RHCS2       ;INSERT UNIT NUMBER
022340  022777  024020  172434          CMP     #24020,@RHDT    ;IS THIS A DUAL PORT RP04 ?
022346  001503                          BEQ     6$              ;TYPE DRIVE NO. IF SO
022350  022777  020020  172424          CMP     #20020,@RHDT    ;IS THIS A SINGLE PORT RP04 ?
022356  001477                          BEQ     6$              ;TYPE DRIVE NO.IF YES
```

```
2773  022360  022777  024021  172414          CMP     #24021,@RHDT    ;IS THIS A DUAL PORT RP05 ?
2774  022366  001473                          BEQ     6$              ;TYPE UNIT NO. OUT
2775  022370  022777  020021  172404          CMP     #20021,@RHDT    ;IS THIS A SINGLE PORT RP05 ?
2776  022376  001467                          BEQ      6$             ;TYPE UNIT NO. IF SO
2777
2778  022400  022777  024022  172374          CMP     #24022,@RHDT    ;IS THIS A DUAL PORT RP06 ?
2779  022406  001463                          BEQ     6$              ;TYPE THE NO IF SO
2780  022410  022777  020022  172364          CMP     #20022,@RHDT    ;IS THIS A SINGLE PORT RP06 ?
2781  022416  001457                          BEQ     6$              ;TYPE THE NO IF SO
2782
2783
2784                                          ;*NO...IT'S NOT AN RP04/RP05/RP06 DEVICE
2785                                          ;*SO TYPE OUT THE DEVICE TYPE
2786
2787  022446  010546                          MOV     R5,-(SP)        ;GET READY TO TYPE UNIT NUMBER
2788  022450  104405                          TYPDS
2789  022474  017746  172302                  MOV     @RHDT,-(SP)     ;GET READY TO TYPE RHDT
2790  022500  104402                          TYPOC
2791  022554  000407                          BR      5$              ;NO RP04/5/6 FOUND SO BRANCH
2792  022556  010523                  6$:     MOV     R5,(R3)+
2793  022560  104401  001223                  TYPE    ,$CRLF
2794  022564  010546                          MOV     R5,-(SP)        ;PUT DRIVE NO. ON STACK
2795  022566  104405                          TYPDS                   ;TYPE DRIVE NO.
2796  022570  005237  015114                  INC     @#NOUNIT        ;INCR TOTAL NO. OF UNITS
2797
2798  022574  005205                  5$:     INC     R5              ;'RHCS2' UNIT ADDRESS
2799  022576  005300                          DEC     R0              ;DRIVE COUNTER DOWN ONE
2800  022600  001252                          BNE     4$              ;TEST AND DO NEXT UNIT IF 8 NOT DONE
2801
2802  022602  013737  015072  015112          MOV     @#UNITS,@#UNIT  ;SET UNIT NO. TO FIRST ONE FOUND/OR 0
2803  022610  013737  015114  015116          MOV     @#NOUNIT,@#NUNIT  ;SAVE NO. OF UNITS
2804  022616  005337  015116                  DEC     @#NUNIT         ;IF NUNIT = 0 THEN ONLY ONE UNIT
2805                                                                  ;IF NUNIT > 0 THEN MORE THAN ONE UNIT
2806
2807  022622  005737  015120          SELTST: TST     @#SELECT        ;STARTING ADDRESS 200 ?
2808  022630  013737  015122  015112          MOV     @#UNITSL,@#UNIT ;CHANGE UNIT NUMBER TO SELECTED ONE
2809
```

```
2810
2811
2812  022654  004737  045764              JSR     PC,@#CLDISK     ;FILL UNIT NO.
2813  022660  005037  015134              CLR     @#ATTENT        ;CLEAR
2814
2815                                      ;*TEST FOR UNIT #0
2816
2817  022664  005737  015112              TST     @#UNIT          ;IS UNIT #0 NEXT IN THE UNITS TABLE ?
2818  022670  001022                      BNE     10$             ;IF NOT, TEST THIS UNIT
2819  022672  012700  000041              MOV     #41,R0          ;IF SO, CHECK THE LOAD MEDIA LOCATION
2820  022676  122710  000011              CMPB    #11,(R0)        ;WAS IS AN RP04/5/6 ?
2821  022702  001015                      BNE     10$             ;NO... GO AHEAD AND TEST UNIT #0
2822  022704  005737  015120              TST     @#SELECT        ;WAS UNIT #0 SELECTED ?
2823                                                              ;(IE. WAS IT A 210 START ?)
2824  022710  001012                      BNE     10$             ;IF SO...TEST UNIT #0
2825
2826                                      ;*INCREMENT THE UNITS TABLE TO NEXT DRIVE (IF ANY)
2827                                      ;* & DECREMENT THE "NOUNITS" PRESENT (TO BE TESTED)
2828
2829  022712  012700  015072              MOV     #UNITS,R0       ;LOAD UNITS TABLE POINTER
2830  022716  005720                      TST     (R0)+           ;SELECT THE NEXT UNIT IN THE TABLE
2831                                                              ;(DOUBLE INCREMENT THE POINTER)
2832  022720  022710  177777              CMP     #-1,(R0)        ;IS THERE ANOTHER TABLE ENTRY PRESENT ?
2833  022724  001404                      BEQ     10$             ;IF NOT (LOC = -1) ...MUST USE UNIT #0
2834  022726  011037  015112              MOV     (R0),@#UNIT     ;SET UP TO BE THE UNIT UNDER TEST
2835  022732  005337  015114              DEC     @#NOUNITS       ;DECREMENT BECAUSE UNIT #0 WON'T BE TESTED
2836  022736  013700  015112      10$:    MOV     @#UNIT,R0       ;R0 CONTAINS THE UNIT UNDER TEST
2837
2838
2839
2840                              ;       CLR     @#RP06          ;CLEAR RP06 DEVICE TYPE FLAG
2841                              ;       CMP     #24022,@RHDT    ;DUAL PORT RP06 ?
2842                              ;       BEQ     2$              ;YES...SET THE FLAG
2843                              ;       CMP     #20022,@RHDT    ;SINGLE PORT RP06 ?
2844                              ;       BEQ     2$              ;YES...SET FLAG
2845                              ;       BR      3$              ;DON'T SET THE RP06 FLAG
2846                              ;2$:    MOV     #-1,@#RP06      ;SET THE FLAG
2847
2848                              ;3$:                            ;ASSUME THE NEXT UNIT IS AN RP04
2849
2850
2851  022742  116037  017336  015134      MOVB    ATABLE(R0),@#ATTENT  ;SET APPROPRIATE ATTENTION BIT
2852  023006  013746  015112              MOV     @#UNIT,-(SP)    ;UNIT NO. TO STACK
2853  023012  104405                      TYPDS                   ;TYPE  DRIVE NO.
2854
2855
2856
2857
2858
2859  023104  022777  024020  171670      CMP     #24020,@RHDT    ;DUAL PORT RP04 ?
2860  023112  001425                      BEQ     4$              ;TYPE ASCII MSG OUT
2861  023114  022777  020020  171660      CMP     #20020,@RHDT    ;SINGLE PORT RP04 ?
2862  023122  001421                      BEQ     4$              ;TYPE THE MESSAGE
2863
2864  023124  022777  024021  171650      CMP     #24021,@RHDT    ;DUAL PORT RP05 ?
2865  023132  001434                      BEQ     5$              ;TYPE THE MESSAGE
```

```
2866  023134  022777  020021  171640          CMP     #20021,@RHDT    ;SINGLE PORT RP05 ?
2867  023142  001430                          BEQ     5$              ;TYPE THE MESSAGE
2868
2869  023144  022777  024022  171630          CMP     #24022,@RHDT    ;DUAL PORT RP06 ?
2870  023152  001443                          BEQ     6$              ;TYPE THE MESSAGE
2871  023154  022777  020022  171620          CMP     #20022,@RHDT    ;SINGLE PORT RP06 ?
2872  023162  001437                          BEQ     6$              ;TYPE IT OUT
2873  023164  000454                          BR      1$              ;DRIVE IS NOT RP04/5/6 - SO
2874                                                                  ;DO NOT TYPE ANY MESSAGE OUT
2875
2876                                                                  ;-SHOULD NEVER HAPPEN AT THIS POINT
2877                                                                  ;UNLESS DRIVE GOT SICK WHILE TESTING
2878                                                                  ;WAS IN PROGRESS
2879
2880  023166                          4$:
2881  023222  000435                          BR      1$              ;SKIP NEXT ONES
2882  023224                          5$:
2883  023260  000416                          BR      1$              ;SKIP NEXT
2884  023262                          6$:
2885
2886  023316  005777  171462          1$:     TST     @RHSN           ;READ SERIAL NO. AND DRIVE TYPE
2887  023322  005777  171454                  TST     @RHDT           ;THESE TWO ARE TO HELP SCOPE LOOPS
2888  023326  017737  171452  015130          MOV     @RHSN,@#SAVSN   ;SAVE TO CHECK IF CLR RHCS2 BIT 5 CLEARS ANY BITS
2889  023334  017737  171442  015126          MOV     @RHDT,@#SAVDT   ;SAVE TO CHECK IF CLR RHCS2 BIT 5 CLEARS ANY BITS
```

```
2890
2891
2892  023352  004737  045764          JSR     PC,@#CLDISK     ;GIVE INITILIZE
2893  023356  032713  010000          BIT     #MOL,@R3        ;CHECK MOL IN RHDS1
2894
2895
2896  023600  032713  010000    1$:   BIT     #MOL,@R3        ;CHECK MOL IN RHDS1
2897  023604  001375                  BNE     1$              ;BRANCH IF MOL IS HIGH
2898
2899
2900
```

```
2901
2902  023706  012706  001000                MOV     #STACK,SP       ;RESET STACK
2903
2904  023720  004737  045764                JSR     PC,@#CLDISK     ;INIT AND SET UP GENERAL REG.
2905                                                                  ;AND UNIT NUMBER
2906  023724  012777  000001  171044        MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
2907
2908  023732  013777  015206  171016        MOV     @#PKACK,@RHCS1  ;LOAD PACK ACKNOWLEDGE COMMAND INTO RHCS1
2909
2910                                        ;SAVE REGISTERS FOR COMPARISON AFTER GO
2911  023740  004037  046456                JSR     R0,@#SAVER      ;SAVE
2912  023744  014750                        RHWC                    ;FROM
2913  023746  016264                        REINTO                  ;TO
2914  023750  000023                        19.                     ;NUMBER OF REGISTERS SAVED
2915
2916                                        ;GIVE GO TO PACK ACKNOWLEDGE COMMAND
2917  023752  052777  000001  170776        BIS     #GO,@RHCS1      ;GO TO PACK ACKNOWLEDGE COMMAND
2918
2919                                        ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
2920  023760  052737  000100  016314        BIS     #VV,@#REINTO+30 ;SAVED RHDS1
2921
2922                                        ;AFTER GO HAS BEEN GIVEN TO PACK ACKNOWLEDGE COMMAND
2923                                        ;SAVE REGISTERS AGAIN SO THAT COMPARISONS CAN
2924                                        ;BE DONE
2925  023766  004037  046456                JSR     R0,@#SAVER      ;SAVE
2926  023772  014750                        RHWC                    ;FROM
2927  023774  015220                        WRFROM
2928  023776  000023                        19.                     ;NUMBER OF REGISTERS SAVED
2929
2930                                        ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
2931                                        ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
2932                                        ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
2933  024000  113737  016311  015245        MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
2934
2935
2936                                        ;COMPARE REGISTERS BEFORE PACK ACKNOWLEDGE COMMAND
2937                                        ;WITH AFTER GO
2938  024006  004037  046660                JSR     R0,@#COMPAR     ;COMPARE
2939  024012  016264                        REINTO                  ;GOOD BUFFER
2940  024014  015220                        WRFROM                  ;TEST BUFFER
2941  024016  000023                        19.                     ;NUMBER
2942  024020  024026                        1$                      ;RETURN FOR ERROR
2943  024022  024026                        1$                      ;SAME
2944  024024  024046                        2$                      ;RETURN FOR GOOD COMPARISON
2945  024026  013705  052770          1$:   MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
2946  024032  060505                        ADD     R5,R5           ;DOUBLE ERROR WORD
2947  024034  016537  014746  045530        MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
2948
2949  024042  104001                        ERROR   1               ;IMPROPER REGISTER CHANGE
2950                                                                  ;AFTER PACK ACKNOWLEDGE COMMAND
2951                                                                  ;WITH GO IS GIVEN
2952  024044  000207                        RTS     PC              ;RETURN TO COMPARISION
2953
2954  024046                          2$:
2955
2956
```

```
2957                                    .SBTTL
2958                                    .SBTTL  ***DIAGNOSTIC CODE***
2959                                    .SBTTL
```

```
2960            000004                          TIMOT=4
2961                                            .REM %
2962                                            THE FOLLOWING TESTS WILL ATTEMPT TO CATCH THOSE BUGS WHICH FAULT
2963                                            INSERTION HAS SHOWN US WE MISSED IN THE FIRST PART OF THIS TEST.
2964
2965                                             THIS TEST WILL ASCERTAIN THAT THE ALL RH11 REGISTERS WIIL
2966                                            RESPOND TO I.E. NOT CAUSE A NON-EXISTANT MEMORY ERROR WHEN ACCESSED
2967                                            BY A "TST" INSRUCTION.
2968                                            %
2969  024120  012777  000040  170626              MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
2970
2971  024126  012705  014756                      MOV     #RHCS1,R5       ;R5=LIST POINTER.
2972
2973  024132  013737  000004  001176              MOV     @#TIMOT,$TMPO   ;SAVE TIMEOUT VECTOR.
2974  024140  012737  024176  000004              MOV     #E00,@#TIMOT    ;SET VECTOR TO E00
2975
2976  024146  012706  001000          L00:        MOV     #STACK,SP       ;SET STACK POINTER.
2977
2978  024152  011502                  I00:        MOV     (R5),R2         ;R2=RH11 ADDRESS.
2979  024154  010237  045530                      MOV     R2,REGADR
2980  024160  005712                              TST     (R2)            ;DOES THE RH11 REGISTER RESPOND?
2981
2982  024162  062705  000002                      ADD     #2,R5           ;UPDATE ADDRESS
2983  024166  020527  015010                      CMP     R5,#RHEC2       ;AT THE END OF LEGAL RH11 REGISTERS?
2984  024172  101767                              BLOS    I00             ;NOPE
2985  024174  000401                              BR      000             ;YES! GOTO 000.
2986
2987  024176  104042          E00:        ERROR   42
2988
2989  024200  013737  001176  000004  000:        MOV     $TMPO,@#TIMOT   ;RESTORE TIMEOUT VECTOR.
```

2990

```
                                        .REM %
                                        THE FOLLOWING 6 TESTS WILL TEST THE BYTE LOGIC FOR THE RH11 REGISTERS
                                        NO ATTEMPT IS MADE TO DETERMINE IF ALL POSSIBLE DATA PATTERNS CAN BE
                                        LOADED, ONLY THAT THE RH11 HARDWARE WILL ALLOW THE PROGRAM TO SELECTIVLY
                                        MANIPULATE BYTES USING THE FOLLOWING COMMANDS:

                                                1). MOVE BYTE BOTH TO AND FROM THE WORD COUNT REGISTER

                                                2). BIT SET INTO THE WORD COUNT REGISTER.

                                                3). BIT CLEAR INTO THE WORD COUNT REGISTER.

                                        %

024224  012777  000040  170522            MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER

024232  012706  001000                    MOV     #STACK,SP       ;SET STACK POINTER.

024236  013702  014750                    MOV     RHWC,R2         ;R2=RH11 WC ADDRESS.
024242  010237  045530                    MOV     R2,REGADR       ;REGISTER ADDRESS TO REGADR FOR TYPING.
024246  012737  000377  001124            MOV     #377,$GDDAT     ;$GDDAT=S/B.

024254  005012                  LO2:      CLR     (R2)            ;CLEAR RH11 WC.

024256  112712  000377          IO2:      MOVB    #377,(R2)       ;SET WC TO LO BYTE.
024262  011237  001126                    MOV     (R2),$BDDAT     ;$BDDAT=ACTUAL WAS

024266  023737  001126  001124            CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS
024276  104044                            ERROR   44
```

```
3021
3022  024316  012777  000040  170430          MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3023
3024  024324  012706  001000                  MOV     #STACK,SP       ;SET STACK POINTER.
3025
3026  024330  013702  014750                  MOV     RHWC,R2         ;R2=RH11 WC HI BYTE ADDRESS.
3027  024334  010237  045530                  MOV     R2,REGADR
3028  024340  012737  177400  001124          MOV     #177400,$GDDAT  ;$GDDAT=S/B.
3029
3030  024346  005012                  L03:    CLR     (R2)            ;CLEAR RH11 WC.
3031
3032  024350  005202                          INC     R2              ;R2=HI BYTE ADDRESS.
3033
3034  024352  112712  000377          I03:    MOVB    #377,(R2)       ;SET WC HI BYTE.
3035
3036  024356  005302                          DEC     R2              ;R2=FULL WORD ADDRESS.
3037
3038  024360  011237  001126                  MOV     (R2),$BDDAT     ;$BDDAT=ACTUAL.
3039
3040  024364  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS.
3041  024374  104044                          ERROR   44
3042
3043
```

```
3044
3045  024414  012777  000040  170332          MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3046
3047  024422  012706  001000                  MOV     #STACK,SP       ;SET STACK POINTER.
3048
3049  024426  013702  014750                  MOV     RHWC,R2         ;R2=RH11 WC ADDRESS
3050  024432  010237  045530                  MOV     R2,REGADR
3051  024436  012737  000377  001124          MOV     #377,$GDDAT     ;$GDDAT=S/B.
3052
3053  024444  005012                  L04:    CLR     (R2)            ;CLEAR RH11 WC.
3054
3055  024446  012712  000252                  MOV     #252,(R2)       ;SET UP WC
3056  024452  152712  000125          I04:    BISB    #125,(R2)       ;DO A BISB
3057
3058  024456  011237  001126                  MOV     (R2),$BDDAT     ;$BDDAT=WAS.
3059
3060  024462  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS
3061  024472  104045                          ERROR   45
3062
3063
```

```
3064
3065  024512  012777  000040  170234          MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3066
3067  024520  012706  001000                  MOV     #STACK,SP       ;SET STACK POINTER.
3068
3069  024524  013702  014750                  MOV     RHWC,R2         ;R2=RH11 WC ADDRESS.
3070  024530  010237  045530                  MOV     R2,REGADR
3071  024534  012737  177400  001124          MOV     #177400,$GDDAT  ;$GDDAT=S/B.
3072
3073  024542  005012                  LO5:    CLR     (R2)            ;CLEAR RH11 WC.
3074
3075  024544  005202                          INC     R2              ;R2 =HI BYTE.
3076
3077  024546  112712  000125                  MOVB    #125,(R2)       ;SET UP RH11 WC.
3078  024552  152712  000252          IO5:    BISB    #252,(R2)       ;DO A BISB.
3079
3080  024556  005302                          DEC     R2              ;R2=FULL WORD ADDRESS.
3081
3082  024560  011237  001126                  MOV     (R2),$BDDAT     ;$BDDAT=WAS.
3083
3084  024564  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS.
3085  024574  104045                          ERROR   45
3086
3087
```

```
3088
3089  024614  012777  000040  170132          MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3090
3091  024622  012706  001000                  MOV     #STACK,SP       ;SET STACK POINTER.
3092
3093  024626  013702  014750                  MOV     RHWC,R2         ;R2=RH11 WC ADDRESS.
3094  024632  010237  045530                  MOV     R2,REGADR
3095  024636  012737  000252  001124          MOV     #252,$GDDAT     ;$GDDAT=S/B.
3096
3097  024644  005012                  L06:    CLR     (R2)            ;CLEAR RH11 WC.
3098
3099  024646  012712  000377                  MOV     #377,(R2)       ;SET WC=0000377
3100
3101  024652  142712  000125          I06:    BICB    #125,(R2)       ;DO A BICB
3102
3103  024656  011237  001126                  MOV     (R2),$BDDAT     ;$BDDAT=WAS.
3104
3105  024662  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS.
3106  024672  104046                          ERROR   46
3107
3108
```

```
3109
3110  024712  012777  000040  170034          MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3111
3112  024720  012706  001000                  MOV     #STACK,SP       ;SET STACK POINTER.
3113
3114  024724  013702  014750                  MOV     RHWC,R2         ;R2=RH11 WC ADDRESS.
3115  024730  010237  045530                  MOV     R2,REGADR
3116  024734  012737  125000  001124          MOV     #125000,$GDDAT  ;$GDDAT=S/B.
3117
3118  024742  005012                  LO7:    CLR     (R2)            ;CLEAR RH11 WC.
3119
3120  024744  005202                          INC     R2              ;R2=HI BYTE.
3121
3122  024746  112712  000377                  MOVB    #377,(R2)       ;SET WC=177400
3123  024752  142712  000125          IO7:    BICB    #125,(R2)       ;DO A BICB
3124
3125  024756  005302                          DEC     R2              ;R2=FULL WORD.
3126
3127  024760  011237  001126                  MOV     (R2),$BDDAT     ;$BDDAT=WAS.
3128
3129  024764  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS.
3130  024774  104046                          ERROR   46
3131
3132
```

```
                        .REM    %
                        THIS TEST ATTEMPT TO ACCESS AN ILLEGAL REGISTER WITHIN THE RANGE
                        OF ADDRESSES SELECTED BY THE XOR'S. THE RH11 ADDRESS DECODE LOGIC WILL ALLOW UP TO 32
                        CONTIGUOUS REGISTERS TO EXIST, IN OUR CONFIGURATION ONLY 16 WILL REALLY
                        EXIST, THIS TEST ATTEMPTS TO ACCESS THE 20(8) REGISTER - IF IT RESPONDS
                        THE TEST WILL TYPE OUT AN ERROR.

                        FOR THE CASE OF THE RH70, THE CONFIGURATION ALLOWS 18 OR 32 REGISTERS, SO
                        WE WILL ATTEMPT TO ADDRESS REGISTER 23(8), 33(8) OR AS BEFORE THE LAST
                        REGISTER + 2.
                                %

025014  012777  000040  167732          MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER

025022  005737  017334                  TST     @#RH70          ;CHECK TO SEE IF RUNNING WITH RH70
025026  001403                          BEQ     1$              ;IF NOT...SKIP NEXT & DO FOLLOWING
025030  013702  015016                  MOV     RHCS3,R2        ;R2 = LAST LEGAL RH70 REG ADDRESS
025034  000402                          BR      2$              ;SKIP NEXT

025036  013702  015010          1$:     MOV     RHEC2,R2        ;R2 = LAST LEGAL RH11 REG ADDRESS.
025042  062702  000002          2$:     ADD     #2,R2           ;R2 = FIRST ILLEGAL ADDRESS.
025046  010237  045530                  MOV     R2,REGADR
025052  005037  001126                  CLR     $BDDAT          ;$BDDAT=WAS.
025056  005037  001124                  CLR     $GDDAT          ;$GDDAT=S/B.

025062  013737  000004  001176          MOV     @#TIMOT,$TMP0   ;SAVE TIMEOUT VECTOR.
025070  012737  025114  000004          MOV     #010,@#TIMOT    ;SET TIMEOUT VECTOR TO 010.

025076  012706  001000          L10:    MOV     #STACK,SP       ;SET THE STACK POINTER.

025102  005712                          TST     (R2)            ;TEST ADDRESS.

025104  062702  000024                  ADD     #24,R2          ;THIS MIGHT BE THE CASE OF 32 REGISTERS
025110  005712                          TST     (R2)            ;TEST IT AGAIN
025112  104047                          ERROR   47

025114  013737  001176  000004  010:    MOV     $TMP0,@#TIMOT   ;RESTORE TIMEOUT VECTOR.
```

```
                                        ;
                                        ;NOW WE ATTACK BTCB
                                        ;
                                        .REM    %
                                        THE FOLLOWING TWO TESTS DETERMINE IF ALL NON DRIVES SET THE
                                        NED BIT AND THAT ALL EXISTING DRIVES CLEAR THE NED BIT.
                                        THE TESTS LOOK AT TOTALAT TO DETERMINE WHICH DRIVES EXIST AND TEST THAT
                                        THESE DRIVES WILL CLEAR THE NED BIT. AND ALSO THAT NON EXISTANT DRIVES
                                        WILL SET THE NED BIT.
                                        ON FIRST PASS ONLY, IF ALL DRIVES ARE ON LINE A MESSAGE WILL SO ADVISE THE
                                        OPERATOR. THE TEST WILL CONTINUE REGARLESS OF THE OPERATORS ACTION.
                                                %
025140  012777  000040  167606          MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER

025146  123727  015136  000377          CMPB    @#TOTALAT,#377          ;ARE ALL DRIVES ON LINE.
025154  001150                          BNE     U11             ;NO GO RUN THE TEST.

025156  023727  001100  000000          CMP     $PASS,#0        ;IF PASS #0 THEN TYPE MESSAGE
                                                                ;ADVISING OPERATOR
                                                                ;THAT THIS TEST WILL NOT BE RUN
025164  001002                          BNE     Y11
025166  000137  025600                  JMP     X11

025172                          Y11:

025476  012706  001000          U11:    MOV     #STACK,SP       ;SET STACK POINTER.

025502  013702  014754                  MOV     RHCS2,R2        ;R2=RH11 RHCS2 ADDRESS.
025506  010237  045530                  MOV     R2,REGADR

025512  013700  015136                  MOV     @#TOTALAT,R0
025516  005001                          CLR     R1
025520  006000                  S11:    ROR     R0
025522  103423                          BCS     N11

025524  010137  001124          R11:    MOV     R1,$GDDAT       ;$GDDAT=S/B.
025530  052737  010000  001124          BIS     #NED,$GDDAT

025536  013705  014756                  MOV     RHCS1,R5        ;ADDRESS OF A DEVICE REGISTER.

025542  010112                  L11:    MOV     R1,(R2)         ;LOAD A NON-EXISTANT DRIVE.

025544  011537  001176                  MOV     (R5),$TMP0      ;ATTEMPT TO READ FROM DEVICE REGISTER.

025550  011237  001126                  MOV     (R2),$BDDAT     ;$BDDAT=WAS.
025554  042737  167770  001126          BIC     #^C<NED!US4!US2!US1>,$BDDAT;$BDDAT=SAVED DATA.

025562  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS.
025570  001005                          BNE     E11

025572  005201                  N11:    INC     R1
025574  020127  000010                  CMP     R1,#8.          ;TESTED ALL DRIVES YET.
025600                          X11:
025602  000746                          BR      S11

025604  104050                  E11:    ERROR   50
```

```
3228
3229
```

```
3230
3231  025624  012777  000040  167122          MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3232
3233  025632  012706  001000                  MOV     #STACK,SP       ;SET STACK POINTER.
3234
3235  025636  013702  014754                  MOV     RHCS2,R2        ;R2=RH11 RHCS2 ADDRESS.
3236  025642  010237  045530                  MOV     R2,REGADR
3237  025646  013700  015136                  MOV     @#TOTALAT,R0
3238  025652  005001                          CLR     R1
3239  025654  006000                  S12:    ROR     R0
3240  025656  103020                          BCC     N12
3241
3242  025660  010137  001124                  MOV     R1,$GDDAT
3243
3244  025664  013705  014756                  MOV     RHCS1,R5        ;ADDRESS OF A DEVICE REGISTER.
3245
3246  025670  010112                  L12:    MOV     R1,(R2)         ;SELECT UNIT.
3247
3248  025672  011537  001176                  MOV     (R5),$TMP0      ;ATTEMPT TO READ FROM DEVICE.
3249
3250  025676  011237  001126                  MOV     (R2),$BDDAT     ;$BDDAT=WAS.
3251  025702  042737  167770  001126          BIC     #^C<NED!US4!US2!US1>,$BDDAT     ;$BDDAT=SAVED DATA.
3252
3253  025710  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS.
3254  025716  001005                          BNE     E12
3255
3256  025720  005201                  N12:    INC     R1
3257  025722  020127  000010                  CMP     R1,#8.          ;TESTED ALL DRIVES.
3258
3259  025730  000751                          BR      S12
3260  025732  104050                  E12:    ERROR   50
3261
3262
```

```
3263
3264                                        .REM    %
3265                                        TEST TO SEE IF WE CAN READ FROM THE "AS" REGISTER AND NOT CAUSE
3266                                        A NED NON-EXISTANT DRIVE ERROR
3267                                        %
3268  025752  012777  000040  166774        MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3269
3270  025760  012706  001000                MOV     #STACK,SP       ;SET STACK POINTER.
3271
3272  025764  013702  014754                MOV     RHCS2,R2        ;R2=RH11 CS2 ADDRESS.
3273  025770  012737  000007  001124        MOV     #US4!US2!US1,$GDDAT;$GDDAT=S/B.
3274
3275  025776  013705  014756                MOV     RHCS1,R5        ;ADDRESS OF A DEVICE REGISTER.
3276
3277  026002  012712  000007        L13:    MOV     #US4!US2!US1,(R2);LOAD A NON-EXISTANT DEVICE.
3278
3279  026006  013702  014774                MOV     RHAS,R2         ;R2= "AS".
3280
3281  026012  011237  001176                MOV     (R2),$TMP0      ;ATTEMPT TO READ FROM DEVICE AS.
3282
3283  026016  005012                        CLR     (R2)            ;CLEAR AS REGISTER.
3284  026020  013702  014754                MOV     RHCS2,R2        ;R2=RH11 CS2 ADDRESS.
3285  026024  010237  045530                MOV     R2,REGADR
3286
3287  026030  011237  001126                MOV     (R2),$BDDAT     ;$BDDAT=WAS.
3288  026034  042737  167770  001126        BIC     #↑C<NED!US4!US2!US1>,$BDDAT;SAVE NED AND DEVICE SELECTED.
3289
3290  026042  023737  001126  001124        CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS.
3291  026052  104051                        ERROR   51
3292
3293
```

```
                                        .REM    %
                                        THE NEXT THREE TESTS WILL TEST THE BUS ADDRESS REGISTER IN BOTH WORD
                                        AND BYTE MODE. THE PATTERNS ARE CHOSEN TO PICK UP ANY DROPPED OR STUCK
                                        BITS.
                                                %

026072  012777  000040  166654            MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER

026100  012706  001000                    MOV     #STACK,SP       ;SET STACK POINTER.

026104  013702  014752                    MOV     RHBA,R2         ;R2=RH11 BA ADDRESS.
026110  010237  045530                    MOV     R2,REGADR
026114  012705  026154                    MOV     #LST14A,R5      ;R5=TEST LIST ADDRESS.

026120  012537  001124          L14:      MOV     (R5)+,$GDDAT    ;$GDDAT=S/B.

026124  013712  001124          I14:      MOV     $GDDAT,(R2)     ;SET BUS ADDRESS REGISTER.
026130  011237  001126                    MOV     (R2),$BDDAT     ;READ BUS ADDRESS REGISTER.

026134  023737  001126  001124            CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS.
026142  001401                            BEQ     N14             ;GET NEXT TEST DATA.
026144  104052                            ERROR   52

026146  005715                  N14:      TST     (R5)            ;AT END OF LIST
026150  001363                            BNE     L14             ;NO!
026154  000002                  LST14A:   2
026156  000004                            4
026160  000010                            10
026162  000020                            20
026164  000040                            40
026166  000100                            100
026170  000200                            200
026172  000400                            400
026174  001000                            1000
026176  002000                            2000
026200  004000                            4000
026202  010000                            10000
026204  020000                            20000
026206  040000                            40000
026210  100000                            100000
026212  177776                            177776
026214  177774                            177774
026216  177772                            177772
026220  177766                            177766
026222  177756                            177756
026224  177736                            177736
026226  177676                            177676
026230  177576                            177576
026232  177376                            177376
026234  176776                            176776
026236  175776                            175776
026240  173776                            173776
026242  167776                            167776
026244  157776                            157776
026246  137776                            137776
026250  077776                            077776
```

```
3350  026252  000000                    0
3351
```

```
3352
3353  026272  012777  000040  166454          MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3354
3355  026300  012706  001000                  MOV     #STACK,SP       ;RESET STACK.
3356
3357  026304  013702  014752                  MOV     RHBA,R2         ;R2=RH11 BA ADDRESS.
3358  026310  010237  045530                  MOV     R2,REGADR
3359  026314  012705  026356                  MOV     #LST15A,R5      ;R5=TEST LIST ADDRESS.
3360
3361  026320  005012                  L15:    CLR     (R2)            ;CLEAR RH11 BA REGISTER.
3362
3363  026322  012537  001124                  MOV     (R5)+,$GDDAT    ;$GDDAT=S/B.
3364
3365  026326  113712  001124          I15:    MOVB    $GDDAT,(R2)     ;SET BUS ADDRESS REGISTER.
3366  026332  011237  001126                  MOV     (R2),$BDDAT     ;READ BUS ADDRESS REGISTER.
3367
3368  026336  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS.
3369  026344  001401                          BEQ     R15
3370
3371  026346  104053                          ERROR   53
3372
3373  026350  005715                  R15:    TST     (R5)            ;AT END OF TEST LIST.
3374  026352  001362                          BNE     L15             ;NO!
3375                                  ;THIS LIST WILL BE USED TO LOAD THE LOWER BYTE OF THE BA REGISTER.
3376                                  ;
3377  026356  000002                  LST15A: 2
3378  026360  000004                          4
3379  026362  000010                          10
3380  026364  000020                          20
3381  026366  000040                          40
3382  026370  000100                          100
3383  026372  000200                          200
3384  026374  000376                          376
3385  026376  000372                          372
3386  026400  000366                          366
3387  026402  000356                          356
3388  026404  000336                          336
3389  026406  000276                          276
3390  026410  000176                          176
3391  026412  000000                          0
```

```
3392
3393  026432  012777  000040  166314          MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3394
3395  026440  012706  001000                  MOV     #STACK,SP       ;RESET STACK.
3396
3397  026444  013702  014752                  MOV     RHBA,R2         ;R2=RH11 BA ADDRESS.
3398  026450  010237  045530                  MOV     R2,REGADR
3399  026454  012705  026526                  MOV     #LST16A,R5      ;R5=TEST LIST ADDRESS.
3400
3401  026460  005012                  L16:    CLR     (R2)            ;CLEAR BA REGISTER.
3402
3403  026462  012537  001124                  MOV     (R5)+,$GDDAT    ;$GDDAT=S/B.
3404
3405  026466  005202                          INC     R2              ;R2=HI BYTE ADDRESS.
3406
3407  026470  113712  001124          I16:    MOVB    $GDDAT,(R2)     ;SET BUS ADDRESS REGISTER HI-BYTE.
3408
3409  026474  005302                          DEC     R2              ;R2=FULL WORD ADDRESS.
3410
3411  026476  011237  001126                  MOV     (R2),$BDDAT     ;READ BUS ADDRESS.
3412  026502  000337  001124                  SWAB    $GDDAT          ;ADJUST DATA FOR HI BYTE.
3413
3414  026506  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS.
3415  026514  001401                          BEQ     R16             ;OKAY
3416
3417  026516  104053                          ERROR   53
3418
3419  026520  005715                  R16:    TST     (R5)            ;AT END OF TEST LIST.
3420  026522  001356                          BNE     L16             ;NOPE
3421                                  ;THIS LIST WILL BE USED TO LOAD THE UPPER BYTE OF THE BA REGISTER.
3422                                  ;
3423  026526  000002                  LST16A: 2
3424  026530  000004                          4
3425  026532  000010                          10
3426  026534  000020                          20
3427  026536  000040                          40
3428  026540  000100                          100
3429  026542  000200                          200
3430  026544  000376                          376
3431  026546  000374                          374
3432  026550  000376                          376
3433  026552  000366                          366
3434  026554  000356                          356
3435  026556  000336                          336
3436  026560  000276                          276
3437  026562  000176                          176
3438  026564  000000                          0
3439
```

```
3440                                    .REM    %
3441                                    THIS TEST CAUSE AN RH11 INTERRUPT VIA THE SPECIAL COMMAND SEQUENCE
3442                                    BIS     #IE,@RHCS2. THIS TEST PROVES ONLY THAT THE HARDWARE CAN HANDLE
3443                                    INTERRUPTS PROPERLY
3444                                    %
3445  026604  012777  000040  166142            MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3446
3447  026612  012706  001000                    MOV     #STACK,SP       ;SET STACK POINTER.
3448
3449  026616  013702  014756                    MOV     RHCS1,R2        ;R2=RH11 ADDRESS.
3450  026622  010237  045530                    MOV     R2,REGADR
3451  026626  005037  001126                    CLR     $BDDAT          ;$BDDAT=WAS.
3452  026632  005037  001124                    CLR     $GDDAT          ;$GDDAT=S/B.
3453
3454  026636  017737  166102  001176            MOV     @RPVEC,$TMP0    ;SAVE RH11 INTERRUPT VECTOR.
3455  026644  012777  026670  166072            MOV     #021,@RPVEC     ;SET RH11 INTERRUPT VECTOR TO 021.
3456
3457  026652  052712  000100          L21:      BIS     #IE,(R2)        ;CAUSE INTERRUPT.
3458  026656  000240                            NOP                     ;WAIT FOR INTERRUPT.
3459  026660  000240                            NOP
3460  026662  011237  001124                    MOV     (R2),$GDDAT     ;SAVE CONTENTS OF REGISTER.
3461
3462  026666  104054                            ERROR   54
3463
3464  026670  013777  001176  166046  021:      MOV     $TMP0,@RPVEC    ;RESTORE RH11 INTERRUPT VECTOR.
3465
3466
```

```
3467
3468                                    ;TEST THAT RESET CAN GENERATE THE SIGNAL CLR L.
3469                                    .REM    %
3470                                            THE PROGRAM SETS THE PORT SELECT BIT .THEN DOES A RESET
3471                                            IF THE SIGNAL CLR L WAS GENERATED THE PORT SELECT BIT WILL CLEAR.
3472                                            %
3473  026714  012777  000040  166032            MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3474
3475  026722  013702  014756                    MOV     RHCS1,R2        ;R2=RH11 CS1 ADDRESS
3476  026726  010237  045530                    MOV     R2,REGADR
3477  026732  005037  001124                    CLR     $GDDAT          ;$GDDAT=S/B
3478
3479  026736  012706  001000            L22:    MOV     #STACK,SP       ;SET STACK POINTER.
3480  026742  012712  002000                    MOV     #PSEL,(R2)      ;SET PORT SELECT FLOP.
3481  026746  000005                            RESET                   ;DO A BUSA INIT.
3482
3483  026750  011237  001126                    MOV     (R2),$BDDAT     ;$BDDAT=WAS.
3484  026754  042737  177677  001126            BIC     #^C<IE>,$BDDAT  ;SAVE ONLY I.E. BIT.
3485
3486  026762  023737  001126  001124            CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS.
3487  026772  104055                            ERROR   55
3488
3489
```

```
3490                                    .REM    %
3491                                    SET THE MXF FLOP AND READ IT BACK.
3492                                    %
3493  027024  012777  000040  165722            MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3494
3495  027032  012706  001000                    MOV     #STACK,SP       ;SET STACK POINTER.
3496  027036  013702  014754                    MOV     RHCS2,R2        ;R2=RH11 CS2 ADDRESS.
3497  027042  010237  045530                    MOV     R2,REGADR
3498  027046  012737  001000  001124            MOV     #MXF,$GDDAT     ;$GDDAT=S/B.
3499
3500  027054  012712  001000          I24:      MOV     #MXF,(R2)       ;LOAD MXF
3501
3502  027060  011237  001126                    MOV     (R2),$BDDAT     ;$BDDAT=WAS.
3503
3504  027064  042737  176777  001126            BIC     #^C<MXF>,$BDDAT ;SAVE ONLY MXF
3505
3506  027072  023737  001126  001124            CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS
3507  027102  104056                            ERROR   56
3508
3509
```

```
3510                                        .REM    %
3511                                        SET THE UNIBUS PARITY ERROR FLOP DIRECTLY VIA A MOV #UPE TO RHCS2
3512                                        ATTEMPT TO READ IT BACK.
3513                                        %
3514
3515  027134  012777  000040  165612        MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3516
3517  027142  012706  001000                MOV     #STACK,SP       ;SET STACK POINTER.
3518
3519  027146  013702  014754                MOV     RHCS2,R2        ;R2=RHCS2 ADDRESS.
3520  027152  010237  045530                MOV     R2,REGADR
3521  027156  012737  020000  001124        MOV     #UPE,$GDDAT     ;$GDDAT=S/B.
3522
3523  027164  013712  001124        L30:    MOV     $GDDAT,(R2)     ;ATTEMPT TO SET UPE.
3524  027170  000240                        NOP
3525  027172  000240                        NOP
3526
3527  027174  011237  001126                MOV     (R2),$BDDAT     ;$BDDAT=ACTUAL DATA.
3528  027200  042737  157777  001126        BIC     #^C<UPE>,$BDDAT ;SAVE ONLY UPE.
3529
3530  027206  023737  001126  001124        CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS.
3531  027216  104057                        ERROR   57
3532
3533
```

```
3534
3535                                    .REM    %
3536                                    SET THE UNIBUS PARITY ERROR FLOP AND ATTEMPT TO CLEAR IT WITH A
3537                                    CONTROLLER CLEAR.
3538                                    %
3539  027236  012777  000040  165510            MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3540
3541  027244  012706  001000                    MOV     #STACK,SP       ;SET STACK POINTER.
3542
3543  027250  013702  014754                    MOV     RHCS2,R2        ;R2=RHCS2 ADDRESS.
3544  027254  010237  045530                    MOV     R2,REGADR
3545  027260  005037  001124                    CLR     $GDDAT          ;$GDDAT=S/B.
3546
3547  027264  012712  020000          L31:      MOV     #UPE,(R2)       ;SET UNIBUS PARITY ERROR SET.
3548  027270  000240                            NOP
3549  027272  000240                            NOP
3550
3551  027274  012712  000040                    MOV     #CLR,(R2)       ;CONTROLLER CLEAR.
3552
3553  027300  011237  001126                    MOV     (R2),$BDDAT     ;READ STATUS
3554  027304  042737  157777  001126            BIC     #^C<UPE>,$BDDAT ;SAVE ERROR.
3555
3556  027312  023737  001126  001124            CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS.
3557  027322  104057                            ERROR   57
3558
3559
```

```
3560
3561                                    .REM    %
3562                                    SET THE UNIT SELECT REGISTER TO DRIVE #7 ALL BITS SET. GENERATE A
3563                                    CONTROLLER CLEAR AND VERIFY THAT THE UNIT SELECT REGISTER IS CLEARED.
3564                                    %
3565  027342  012777  000040  165404            MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3566
3567  027350  012706  001000                    MOV     #STACK,SP       ;SET STACK POINTER.
3568
3569  027354  013702  014754                    MOV     RHCS2,R2        ;R2=CS2 ADDRESS.
3570  027360  010237  045530                    MOV     R2,REGADR
3571  027364  005037  001124                    CLR     $GDDAT          ;$GDDAT=S/B.
3572
3573  027370  012712  000007    L34:            MOV     #US4!US2!US1,(R2)  ;LOAD RHCS2 DRIVE SELECT
3574  027374  000240                           NOP
3575  027376  000240                           NOP
3576
3577  027400  012712  000040                    MOV     #CLR,(R2)       ;GENERATE CLR.
3578
3579  027404  011237  001126                    MOV     (R2),$BDDAT     ;READ RHCS2
3580  027410  042737  177770  001126            BIC     #^C<US4!US2!US1>,$BDDAT;SAVE DRIVE SELECT BITS.
3581
3582  027416  023737  001126  001124            CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS.
3583  027426  104060                           ERROR   60
3584
3585
```

```
3586
3587                                        .REM    %
3588                                        SET THE UPE FLOP UNIBUS PARITY ERROR. VERIFY THAT THE SETTING OF (UPE)
3589                                        ALSO SETS THE TRANSFER ERROR (TRE) FLOP
3590                                                %
3591
3592  027460  012777  000040  165266                MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3593
3594  027466  012706  001000                        MOV     #STACK,SP       ;SET STACK POINTER.
3595
3596  027472  013702  014754                        MOV     RHCS2,R2        ;R2=RHCS2 ADDRESS.
3597  027476  012737  040000  001124                MOV     #TRE,$GDDAT     ;$GDDAT=S/B.
3598
3599  027504  012712  020000          L35:          MOV     #UPE,(R2)       ;SET UNIBUS PARITY ERROR.
3600
3601  027510  013702  014756                        MOV     RHCS1,R2        ;R2=RHCS1 ADDRESS.
3602  027514  010237  045530                        MOV     R2,REGADR
3603
3604  027520  011237  001126                        MOV     (R2),$BDDAT     ;READ REGISTER
3605  027524  042737  137777  001126                BIC     #^C<TRE>,$BDDAT ;SAVE TRANSFER ERROR.
3606
3607  027532  023737  001126  001124                CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS
3608  027542  104061                                ERROR   61
3609
3610
```

```
3611
3612                                        .REM    %
3613                                        SET THE NED BIT NON EXISTANT DRIVE VERIFY THAT THIS SETS THE (TRE) BIT.
3614                                        %
3615  027562  012777  000040  165164              MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3616
3617  027570  012706  001000                    MOV     #STACK,SP       ;SET STACK POINTER.
3618
3619  027574  013702  014754                    MOV     RHCS2,R2        ;R2=CS2 ADDRESS.
3620  027600  012737  040000  001124            MOV     #TRE,$GDDAT     ;$GDDAT=S/B.
3621
3622  027606  013700  015136                    MOV     @#TOTALAT,R0    ;GET ALL AVAILABLE DRIVES DATA
3623  027612  005001                            CLR     R1
3624  027614  006000                    S36:    ROR     R0
3625  027616  103420                            BCS     N36
3626  027620  010112                    L36:    MOV     R1,(R2)           ;SET NED.
3627
3628  027622  013702  014756                    MOV     RHCS1,R2        ;R2=RHCS1 ADDRESS.
3629  027626  010237  045530                    MOV     R2,REGADR
3630
3631  027632  011237  001126                    MOV     (R2),$BDDAT     ;READ REGISTER.
3632  027636  042737  137777  001126            BIC     #^C<TRE>,$BDDAT ;SAVE TRE.
3633
3634  027644  023737  001126  001124            CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS.
3635  027654  104062                            ERROR   62
3636
3637  027660  005201                    N36:    INC     R1
3638  027662  020127  000010                    CMP     R1,#8.
3639  027666  001352                            BNE     S36
3640
3641
```

```
3642
3643                                        .REM    %
3644                                        SET THE PORT SELECT FLOP VERIFY THAT WE CAN READ IT BACK.
3645                                        %
3646  027706  012777  000040  165040               MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3647
3648  027714  012706  001000                       MOV     #STACK,SP       ;SET STACK POINTER.
3649
3650  027720  013702  014756                       MOV     RHCS1,R2        ;R2=RHCS1 ADDRESS.
3651  027724  010237  045530                       MOV     R2,REGADR
3652  027730  005037  001124                       CLR     $GDDAT          ;$GDDAT=S/B.
3653
3654  027734  012712  002000               L40:    MOV     #PSEL,(R2)      ;SET P SELECT.
3655
3656  027740  012777  000040  165006               MOV     #CLR,@RHCS2     ;DO A CONTROLLER CLEAR.
3657
3658  027746  011237  001126                       MOV     (R2),$BDDAT     ;READ BACK P SELECT BIT.
3659  027752  042737  175777  001126               BIC     #^C<PSEL>,$BDDAT;SAVE ONLY PORT SELECT.
3660
3661  027760  023737  001126  001124               CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS.
3662  027770  104064                               ERROR   64
3663
```

```
                                        .REM    %
                                        VERIFY THAT CONTROLLER CLEAR WILL CLEAR THE COMMAND REGISTER.
                                        %

030010  012777  000040  164736           MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER

030016  012706  001000                   MOV     #STACK,SP       ;SET STACK POINTER.

030022  013702  014756                   MOV     RHCS1,R2        ;R2=RHCS1 ADDRESS.
030026  010237  045530                   MOV     R2,REGADR
030032  005037  001124                   CLR     $GDDAT          ;$GDDAT=S/B.

030036  012712  000011           L41:    MOV     #11,(R2)        ;ISSUE A DRIVE CLR AND GO.

030042  012777  000040  164704           MOV     #CLR,@RHCS2     ;CONTROLLER CLEAR.

030050  011237  001126                   MOV     (R2),$BDDAT     ;READ COMMAND REGISTER.
030054  042737  177770  001126           BIC     #^C<7>,$BDDAT   ;SAVE ONLY THE COMMAND BITS.

030062  023737  001126  001124           CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS.
030072  104065                           ERROR   65
```

```
                                .REM    %
                                THE COMMAND REGISTER IS SUPPOSED TO BE SELF CLEARING THAT IS WHEN THE
                                RH11 IS SELECTED WHEATHER OR NOT WE ADDRESS IT DIRECTLY THE LOGIC
                                SHOULD CLEAR THE COMMAND REGISTER.
                                THIS IS TESTED BY LOADING A COMMAND INTO IT READING ANOTHER REGISTER
                                THAN RETURNING TO CHECK THE COMMAND REGISTER TO DETERMINE IF IT CLEARED.
                                %
030112  012777  000040  164634        MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER

030120  012706  001000                MOV     #STACK,SP       ;SET STACK POINTER.

030124  013702  014756                MOV     RHCS1,R2        ;R2=RHCS1 ADDRESS.
030130  010237  045530                MOV     R2,REGADR
030134  005037  001124                CLR     $GDDAT          ;$GDDAT=S/B.

030140  012737  000340  177776        MOV     #340,PS         ;SET PRIORITY TO #7.

030146  012712  000011        L42:    MOV     #11,(R2)        ;ISSUE A DRIVE CLR AND GO

030152  011237  001176                MOV     (R2),$TMP0      ;DO A RESELECT OF THE REGISTER.

030156  011237  001126                MOV     (R2),$BDDAT     ;READ THE COMMAND REGISTER.
030162  042737  177770  001126        BIC     #^C<7>,$BDDAT   ;SAVE ONLY THE COMMAND BITS.

030170  023737  001126  001124        CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS.
030200  104066                        ERROR   66
```

```
3714
3715                                        .REM    %
3716                                        HERE WE TEST TO SEE IF SETTING (IE) AND (RDY) WILL CAUSE AN INTERRUPT.
3717                                        %
3718  030220  012777  000040  164526        MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3719
3720  030226  013702  014756                MOV     RHCS1,R2        ;R2=RPCS1 ADDRESS.
3721  030232  010237  045530                MOV     R2,REGADR
3722  030236  005037  001126                CLR     $BDDAT          ;$BDDAT=WAS.
3723  030242  005037  001124                CLR     $GDDAT          ;$GDDAT=S/B.
3724
3725  030246  017737  164472  001176        MOV     @RPVEC,$TMP0    ;SAVE RH11 INTERRUPT VECTOR.
3726  030254  012777  030310  164462        MOV     #043,@RPVEC     ;SET RH11 INTERRUPT VECTOR 043
3727
3728  030262  012706  001000        L43:    MOV     #STACK,SP       ;SET STACK POINTER.
3729  030266  005037  177776                CLR     PS
3730
3731  030272  052712  000300                BIS     #IE!RDY,(R2)    ;THIS WILL CAUSE AN INTERRUPT.
3732  030276  000240                        NOP
3733  030300  000240                        NOP
3734
3735  030302  011237  001124                MOV     (R2),$GDDAT
3736  030306  104067                        ERROR   67
3737
3738  030310  013777  001176  164426  043:  MOV     $TMP0,@RPVEC    ;RESTORE RH11 INTERRUPT VECTOR.
```

```
3739
3740                                        .REM    %
3741                                        THE (IE) BIT IS SET CAUSING AN INTERRUPT. AT THE COMPLETION OF THE INTERRUPT
3742                                        WE CHECK TO SEE IF THE (IE) BIT HAS CLEARED.
3743                                        %
3744  030334  012777  000040  164412        MOV     #CLR,@RHCS2         ;CLEAR RH11 CONTROLLER
3745
3746  030342  013702  014756                MOV     RHCS1,R2            ;R2=RPCS1 ADDRESS.
3747  030346  010237  045530                MOV     R2,REGADR
3748  030352  005037  001124                CLR     $GDDAT              ;$GDDAT=S/B.
3749
3750  030356  017737  164362  001176        MOV     @RPVEC,$TMP0        ;SAVE RH11 INTERRUPT VECTOR.
3751  030364  012777  030414  164352        MOV     #044,@RPVEC         ;SET VECTOR TO 044.
3752
3753  030372  012706  001000        L44:    MOV     #STACK,SP           ;SET STACK POINTER.
3754  030376  005037  177776                CLR     PS
3755
3756  030402  052712  000100                BIS     #IE,(R2)            ;THIS WILL CAUSE AN INTERRUPT
3757  030406  000240                        NOP
3758  030410  000240                        NOP
3759
3760  030412  000411                        BR      E44                 ;NO INTERRUPT.
3761
3762  030414  011237  001126        044:    MOV     (R2),$BDDAT         ;READ RHCS1.
3763  030420  042737  177677  001126        BIC     #^C<IE>,$BDDAT      ;SAVE ONLY THE "IE" BIT.
3764
3765  030426  023737  001126  001124        CMP     $BDDAT,$GDDAT       ;COMPARE RESULTS.
3766  030434  001401                        BEQ     R44                 ;OK!
3767
3768  030436  104070                E44:    ERROR   70
3769
3770  030440  013777  001176  164276  R44:  MOV     $TMP0,@RPVEC        ;RESTORE RH11 INTERRUPT VECTOR.
3771
```

```
3772
3773                                    .REM    %
3774                                    HERE WE VERIFY THAT WRITING INTO THE "AS" REGISTER OWILL NOT CAUSE
3775                                    AN (NED) ERROR.
3776                                    %
3777  030464  012777  000040  164262            MOV     #CLR,@RHCS2     ;CLEAR RH11 CONTROLLER
3778
3779  030472  012706  001000                    MOV     #STACK,SP       ;SET STACK POINTER.
3780
3781  030476  013702  014754                    MOV     RHCS2,R2        ;R2=RH11 CS2 ADDRESS.
3782  030502  012737  000007  001124            MOV     #US4!US2!US1,$GDDAT;$GDDAT=S/B.
3783
3784  030510  013705  014756                    MOV     RHCS1,R5        ;ADDRESS OF A DEVICE REGISTER.
3785
3786  030514  012712  000007          L45:      MOV     #US4!US2!US1,(R2);LOAD A NON EXISTANT DEVICE.
3787
3788  030520  013702  014774                    MOV     RHAS,R2         ;R2="AS" ADDRESS.
3789
3790  030524  012712  000377                    MOV     #377,(R2)       ;ATTEMPT TO LOAD "AS".
3791
3792  030530  005012                            CLR     (R2)            ;CLEAR  "AS".
3793
3794  030532  013702  014754                    MOV     RHCS2,R2        ;R2=RH11 CS2 ADDRESS.
3795  030536  010237  045530                    MOV     R2,REGADR
3796
3797  030542  011237  001126                    MOV     (R2),$BDDAT     ;$BDDAT=WAS.
3798  030546  042737  167770  001126            BIC     #^C<NED!US4!US2!US1>,$BDDAT;SAVE NED AND DEVICE SELECTED.
3799
3800  030554  023737  001126  001124            CMP     $BDDAT,$GDDAT   ;COMPARE RESULTS.
3801  030564  104071                            ERROR   71
3802
```

```
                        .SBTTL  EXTENDED MEMORY, DRIVE TIMING & SECTOR SELECT TESTS

030624  012706  001000            MOV     #STACK,SP       ;RESET STACK

030650  004037  045702            JSR     R0,@#CLAREA     ;CLEAR SIMULATED DISK
030654  054566                    .WORD   DISK            ;FROM
030656  055612                    .WORD   TOLGAP+16       ;TO
030660  000000                    .WORD   0               ;DATA

                        ;THESE ARE SETUP FOR DISKLESS USE ONLY

030662  012737  010000  052650    MOV     #FMT22,@#CYL;CYLINDER 0
                                                        ;16 BITS PER WORD
030670  005037  052652            CLR     @#SECOTR        ;SECTOR 0 TRACK 0
030674  005037  052654            CLR     @#KEY1          ;KEY1 0
030700  005037  052656            CLR     @#KEY2          ;KEY2 0
030704  012737  000400  052716    MOV     #256.,@#NOWORD  ;NO OF DATA WORDS
030712  012737  000001  052660    MOV     #1,@#X          ;WRITE DATA
030720  004537  047172            JSR     R5,@#CRC        ;GO TO CALCULATE CRC
030724  052650                    CYL
030726  054550                    WCRC

                        ;THESE ARE REGULAR SETUPS

030730  004737  045764            JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
030734  012777  177400  164006    MOV     #-256.,@RHWC    ;256 DATA WORDS
030742  013777  001144  164002    MOV     @#$TKS,@RHBA    ;STARTING ADDRESS OF WRITE BUFFER
030750  017737  150170  015140    MOV     @$TKS,@#TMPILL  ;TEMPORARY STORAGE OF DATA
030756  005077  164000            CLR     @RHDST          ;SECTOR 0 TRACK 0
030762  012777  010000  163776    MOV     #FMT22,@RHOF    ;16 BITS PER WORD FORMAT
030770  005077  163774            CLR     @RHCA           ;CYLINDER 0
031006  013746  015170            MOV     @#WRIDAT,-(SP)  ;WRITE DATA=60
031012  052716  001400            BIS     #A16!A17,(SP)   ;SET HIGH ORDER UNIBUS BITS
031016  012611                    MOV     (SP)+,@RI       ;FILL RHCS1
031020  052777  000010  163726    BIS     #BAI,@RHCS2     ;SET BUS ADDRESS INHIBIT
031026  005037  015124            CLR     @#ERFLG$        ;CLEAR ERROR FLAG
031032  004737  052510            JSR     PC,@#COMHD      ;WRITE DATA
```

```
3849
3850                                    ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
3851                                    ;FROM THE "COMHD" ROUTINE IT MEANS SECTOR GAP, SYNC BYTE
3852                                    ;HEADER, HEADER CRC, HEADER GAP AND SYNC BYTE HAVE GONE BY
3853                                    ;AND SYNCS WERE CORRECTLY DETECTED.
3854
3855                                    ;DATA IS TO BE CHECKED.
3856
3857  031036  004737  045464                      JSR     PC,@#PUTREG     ;SAVE REGISTERS
3858  031042  005737  015124                      TST     @#ERFLG$        ;HAVE ANY ERRORS OCCURED?
3859  031050  013700  015140                      MOV     @#TMPILL,R0     ;GOOD DATA
3860  031054  012701  054566                      MOV     #DISK,R1        ;DATA WRITTEN INTO "DISK"
3861  031060  012702  000400                      MOV     #256.,R2        ;COUNTER
3862  031064  012737  000401  052770  1$:         MOV     #257.,@#ERWORD  ;FOR ERROR WORD
3863  031072  020021                              CMP     R0,(R1)+        ;COMPARE GOOD DATA WITH DATA ON DISK
3864  031074  001425                              BEQ     3$              ;BRANCH IF GOOD
3865  031076  013737  015140  001124              MOV     @#TMPILL,@#$GDDAT;GOOD DATA
3866  031104  014137  001126                      MOV     -(R1),@#$BDDAT  ;BAD DATA
3867  031110  160237  052770                      SUB     R2,@#ERWORD     ;ERROR WORD NO
3868  031114  005737  015124                      TST     @#ERFLG$        ;ANY ERRORS ALREADY THERE?
3869  031120  001002                              BNE     2$              ;BRANCH IF YES
3870  031122  104037                              ERROR   37              ;ERROR ON WRITE DATA COMMAND
3871                                                                      ;SEE NEXT ERROR COMMENTS
3872  031124  000401                              BR      64$             ;BRANCH TO AVOID PRINTING NEXT ERROR
3873  031126  104040                      2$:     ERROR   40              ;WORD NO GIVES WORD IN ERROR
3874                                                                      ;ERROR OCCURED WHILE WRITING
3875                                                                      ;WITH A16 A17 OF RHCS1 SET
3876  031130  005721                      64$:    TST     (R1)+           ;UNDO -(R1) FOR BAD DATA
3877  031132  017746  150002                      MOV     @SWR,-(SP)      ;GET SWITCH SETTING
3878  031136  042716  177177                      BIC     #177177,(SP)    ;KEEP ONLY SWITCH 7 AND 8
3879  031142  022726  000200                      CMP     #SW07,(SP)+     ;IS 7 SET AND 8 RESET
3880  031150  005302                      3$:     DEC     R2              ;IF NOT COUNT 256 WORDS
3881  031152  001344                              BNE     1$              ;BRANCH IF 256 NOT DONE
```

```
3882
3883  031156  012706  001000                  MOV     #STACK,SP       ;RESET STACK
3884
3885                                          ;*THESE ARE TO SETUP FOR DISKLESS USE
3886
3887  031170  012737  010000  052650          MOV     #FMT22,@#CYL    ;16 BITS PER WORD
3888                                                                  ;CYLINDER 0
3889  031176  005037  052652                  CLR     @#SECOTR        ;SECTOR 0
3890                                                                  ;TRACK 0
3891  031202  005037  052654                  CLR     @#KEY1          ;KEY1 = 0
3892  031206  005037  052656                  CLR     @#KEY2          ;KEY2 = 0
3893  031212  005037  052660                  CLR     @#X             ;THIS IS A READ COMMAND
3894  031216  004537  047172                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
3895  031222  052650                          CYL
3896  031224  054550                          WCRC
3897
3898                                          ;* THESE        ARE REGULAR SETUPS
3899
3900  031226  004737  045764                  JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
3901  031232  012777  177374  163510          MOV     #-260.,@RHWC    ;256 DATA WORDS, 4 HEADER WORDS
3902  031240  012777  016264  163504          MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF BUFFER
3903  031246  005077  163510                  CLR     @RHDST          ;TRACK = 0
3904                                                                  ;SECTOR = 0
3905  031252  012777  014000  163506          MOV     #FMT22!ECI,@RHOF ;16 BITS PER WORD
3906                                                                  ;ECC CORRECTION INHIBITED
3907  031260  005077  163504                  CLR     @RHCA           ;CYLINDER = 0
3908  031276  013711  015176                  MOV     @#REFOR,@R1     ;READ HEADER AND DATA = 72
3909
3910                                          ;*READ & SAVE REGISTERS FOR COMPARISON AFTER SIMULATED 'DTE'
3911
3912  031302  004037  046456                  JSR     R0,@#SAVER      ;READ IN SEQUENCE
3913  031306  014750                          RHWC                    ;FROM HARDWARE REGISTER
3914  031310  015024                          WC                      ;INTO CORE AT LOCATION
3915  031312  000023                          19.                     ;NUMBER OF REGISTERS TO READ
3916
3917
3918                                          ;*NOW 'GO' WILL BE GIVEN, EVERYTHING WILL BE TREATED
3919                                          ;*NORMALLY FOR THE HEADER. BUT WHEN IT IS TIME TO READ
3920                                          ;*DATA, ONLY SECTOR CLOCKS WILL BE GIVEN. NO DIAGNOSTIC DATA
3921                                          ;*CLOCKS WILL BE GIVEN. THIS SHOULD BRING SECTOR PULSE HIGH
3922                                          ;*WITHOUT PUTTING "READ" DOWN HENCE 'DTE' WILL COME UP.
3923
3924  031314  012737  177777  015144          MOV     #-1,@#TESDTE    ;SET DTE TEST
3925  031322  012737  177777  015124          MOV     #-1,@#ERFLG$    ;THIS WILL BRING THE READ HEADER
3926                                                                  ;AND DATA PROCESS OUT AFTER THE
3927                                                                  ;HEADER HAS BEEN CORRECTLY READ
3928
3929  031330  004737  052510                  JSR     PC,@#COMHD      ;ISSUE 'GO', SEARCH FOR THE SECTOR
3930                                                                  ;AND READ THE HEADER
3931
3932  031334  017737  163416  001176  SETCK1: MOV     @RHCS1,@#$TMPO  ;READ CS1 TO CHECK FOR ANY READ ERRORS
3933  031342  032737  100000  001176          BIT     #SC,@#$TMPO     ;TEST FOR "SPECIAL CONDITION" - 'SC'
3934  031350  001405                          BEQ     8$              ;CONTINUE WITH TEST IF 'SC' = 0 (NO ERRORS)
3935  031352  004737  045464                  JSR     PC,@#PUTREG     ;READ & SAVE ALL REGISTERS AGAIN IF AN
3936                                                                  ;UNDEFINED DATA TRANSFER ERROR OCCURRED
3937  031356  104040                          ERROR   40              ;READ/WRITE HEADER & DATA ERROR DURING
```

```
3938
3939  031364                          8$:     ;NOW THE HEADER HAS BEEN READ OK
3940                                          ;*NOW 560 SECTOR CLOCKS WILL BE GIVEN
3941                                          ;*GAP 11 BYTES, SYNC 1 BYTE, DATA 512, ECC 4 BYTES
3942                                          ;*GAP 2 BYTES, TOLERANCE 28 BYTES, EXTRA 2
3943
3944                                          ;*THESE 560 SECTOR CLOCKS ARE DIVIDED INTO TWO GROUPS
3945                                          ;*24 SECTOR CLOCKS WITH NORMAL DIAGNOSTIC DATA CLOCKS,
3946                                          ;*AND 536 SECTOR CLOCKS WITHOUT ANY DIAGNOSTIC DATA CLOCKS.
3947
3948
3949                                          ;*THIS GIVES 24 SECTOR CLOCKS WITH DIAGNOSTIC DATA CLOCKS
3950                                          ;*WHICH EQUALS 3 BYTES OF DATA
3951
3952  031364  012701  000030                  MOV     #24.,R1         ;LOAD COUNTER
3953  031370  013700  014776                  MOV     @#RHMR,R0       ;GET RHMR ADDRESS
3954  031374  012710  000001                  MOV     #DMD,@R0        ;SET DIAGNOSTIC MODE
3955  031400  052710  000012          1$:     BIS     #MSTCK!MCLK,@R0 ;SET SECTOR CLOCK AND DATA CLOCK
3956  031404  042710  000012                  BIC     #MSTCK!MCLK,@R0 ;CLEAR SECTOR CLOCK AND DATA CLOCK
3957  031410  012702  000007                  MOV     #7,R2           ;LOAD COUNTER FOR DIAGNOSTIC CLOCKS
3958  031414  052710  000002          4$:     BIS     #MCLK,@R0       ;SET CLOCK
3959  031420  042710  000002                  BIC     #MCLK,@R0       ;CLEAR CLOCK
3960  031424  005302                          DEC     R2              ;COUNT TO 7
3961  031426  001372                          BNE     4$              ;BRANCH IF 7 NOT DONE
3962  031430  005301                          DEC     R1              ;COUNT TO 24
3963  031432  001362                          BNE     1$              ;BRANCH IF 24 NOT DONE
3964
3965                                          ;*THIS GIVES 536 SECTOR CLOCKS WITHOUT DIAGNOSTIC DATA CLOCKS
3966
3967  031434  012701  001030                  MOV     #536.,R1        ;LOAD SECTOR CLOCK COUNTER
3968  031440  052710  000010          5$:     BIS     #MSTCK,@R0      ;SET SECTOR CLOCK
3969  031444  042710  000010                  BIC     #MSTCK,@R0      ;CLEAR SECTOR CLOCK
3970  031450  005301                          DEC     R1              ;COUNT
3971  031452  001372                          BNE     5$              ;BRANCH IF 536 NOT DONE
3972
3973
3974                                          ;*NOW 'DTE' SHOULD BE SET
3975                                          ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
3976
3977  031454  012737  177400  015024          MOV     #-256.,@#WC     ;SAVED RHWC
3978  031462  012737  016274  015026          MOV     #REINTO+<4.*2>,@#BA ;SAVED RHBA
3979  031470  052737  140000  015032          BIS     #SC!TRE,@#CS1   ;SAVED RHCS1
3980  031476  052737  010000  015034          BIS     #DTE,@#ER1      ;SAVED RHER1
3981  031504  012737  000401  015052          MOV     #401,@#MR       ;SAVED RHMR
3982  031512  052737  140000  015054          BIS     #ATA!ERR,@#DS1  ;SAVED RHDS1
3983  031520  012737  000100  015066          MOV     #100,@#LA       ;SAVED RHLA
3984  031526  012737  000001  015036          MOV     #1,@#DST        ;SAVED RHDST
3985  031534  013737  015134  015050          MOV     @#ATTENT,@#AS   ;SAVED RHAS
3986
3987
3988                                          ;*NOW READ & SAVE REGISTERS AGAIN SO THAT COMPARISONS
3989                                          ;*CAN BE DONE (USE THE 'WRFROM' SAVE BUFFER THIS TIME)
3990
3991  031542  004037  046456                  JSR     R0,@#SAVER      ;READ IN SEQUENCE
3992  031546  014750                          RHWC                    ;FROM HARDWARE REGISTER
3993  031550  015220                          WRFROM                  ;INTO CORE BUFFER
```

```
 3994  031552  000023                          19.                         ;NUMBER OF REGISTERS TO READ
 3995
 3996                                          ;*FOR RHAS UPPER BYTE
 3997  031554  113737  015051  015245          MOVB    @#AS+1,@#WRFROM+25 ;UPPER RHAS
 3998
 3999                                          ;*COMPARE THE HEADER READ
 4000
 4001  031562  004037  046660                  JSR     R0,@#COMPAR         ;COMPARE
 4002  031566  052650                          CYL                         ;GOOD BUFFER
 4003  031570  016264                          REINTO                      ;TEST BUFFER
 4004  031572  000004                          4.                          ;NUMBER
 4005  031574  031602                          6$                          ;RETURN FOR ERROR
 4006  031576  031602                          6$                          ;SAME
 4007  031600  031606                          7$                          ;RETURN FOR GOOD COMPARISON
 4008  031602  104010              6$:         ERROR   10                  ;HEADER READ IN DURING THIS TEST IS
 4009                                                                      ;IN ERROR
 4010
 4011  031604  000207                          RTS     PC                  ;RETURN
 4012
 4013  031606                      7$:                                     ;GOOD COMPARISON, CONTINUE
 4014
 4015
 4016                                          ;*COMPARE REGISTERS BEFORE COMMAND WITH REGISTERS AFTER COMMAND
 4017
 4018  031606  004037  046660                  JSR     R0,@#COMPAR         ;COMPARE
 4019  031612  015024                          WC                          ;INITIAL SNAPSHOT BUFFER (CHANGED)
 4020  031614  015220                          WRFROM                      ;TEST SNAPSHOT BUFFER
 4021  031616  000022                          18.                         ;NUMBER OF REGISTERS
 4022  031620  031626                          2$                          ;RETURN FOR ERROR
 4023  031622  031626                          2$                          ;SAME
 4024  031624  031646                          3$                          ;RETURN FOR GOOD COMPARISON
 4025
 4026  031626  013705  052770          2$:     MOV     @#ERWORD,R5         ;GETTING READY TO INDEX
 4027  031632  060505                          ADD     R5,R5               ;DOUBLE ERROR WORD
 4028  031634  016537  014746  045530          MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER
 4029  031642  104001                          ERROR   1                   ;IMPROPER REGISTER
 4030                                                                      ;CHANGE AFTER FORCING
 4031                                                                      ;'DTE' ERROR
 4032  031644  000207                          RTS     PC                  ;RETURN
 4033
 4034  031646                      3$:                                     ;GOOD - REGISTERS OK, GO ON TO NEXT TEST
 4035
 4036
```

```
                                        ;RESET STACK
031650  012706  001000          MOV     #STACK,SP
                                ;*THESE ARE TO SETUP FOR DISKLESS USE

031662  012737  010000  052650  MOV     #FMT22,@#CYL    ;16 BITS PER WORD
                                ;CYLINDER 0
031670  005037  052652          CLR     @#SECOTR        ;SECTOR 0
                                ;TRACK 0
031674  005037  052654          CLR     @#KEY1          ;KEY1 = 0
031700  005037  052656          CLR     @#KEY2          ;KEY2 = 0
031704  012737  177777  052660  MOV     #-1,@#X         ;THIS IS FOR WRITE DATA COMMAND
031712  004537  047172          JSR     R5,@#CRC        ;GO TO CALCULATE CRC
031716  052650                  CYL
031720  054550                  WCRC

                                ;* THESE        ARE REGULAR SETUPS & CHECKS

031722  004737  045764          JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
031726  012777  177400  163014  MOV     #-256.,@RHWC    ;256 DATA WORDS
031734  012777  015220  163010  MOV     #WRFROM,@RHBA   ;STARTING ADDRESS OF BUFFER
031742  005077  163014          CLR     @RHDST          ;TRACK = 0
                                ;SECTOR = 0
031746  012777  010000  163012  MOV     #FMT22,@RHOF    ;16 BITS PER WORD
                                ;ECC CORRECTION INHIBITED
031754  005077  163010          CLR     @RHCA           ;CYLINDER = 0
031772  013711  015170          MOV     @#WRIDAT,@R1    ;WRITE DATA = 60

                                ;*READ & SAVE REGISTERS FOR COMPARISON AFTER SIMULATED 'DTE'

031776  004037  046456          JSR     R0,@#SAVER      ;READ REGISTERS IN SEQUENCE
032002  014750                  RHWC                    ;FROM HARDWARE REGISTER
032004  015024                  WC                      ;INTO CORE BUFFER LOCATION
032006  000023                  19.                     ;NUMBER OF REGISTERS TO READ

                                ;*NOW 'GO' WILL BE GIVEN, EVERYTHING WILL BE TREATED
                                ;*NORMALLY FOR THE HEADER.  WHEN IT IS TIME TO WRITE
                                ;*DATA, ONLY SECTOR CLOCKS WILL BE GIVEN, NO DIAGNOSTIC DATA
                                ;*CLOCKS WILL BE GIVEN. THIS SHOULD BRING SECTOR PULSE HIGH
                                ;*WITHOUT PUTTING READ DOWN, HENCE 'DTE' WILL COME UP.

032010  012737  177777  015144  MOV     #-1,@#TESDTE    ;SET DTE TEST
032016  012737  177777  015124  MOV     #-1,@#ERFLG$    ;THIS WILL BRING THE READ HEADER
                                ;AND DATA PROCESS OUT AFTER THE
                                ;HEADER HAS BEEN CORRECTLY READ

032024  004737  052510          JSR     PC,@#COMHD      ;ISSUE 'GO', SEARCH FOR SECTOR,
                                ;READ HEADER AND DATA.

032030  017737  162722  001176  SETCK2: MOV     @RHCS1,@#$TMPO  ;READ CS1 TO CHECK FOR READ ERRORS
032036  032737  100000  001176          BIT     #SC,@#$TMPO     ;TEST FOR "SPECIAL CONDITION" - 'SC'
032044  001405                  BEQ     6$              ;CONTINUE TESTING IF NO ERROR
032046  004737  045464          JSR     PC,@#PUTREG     ;READ & SAVE REGISTERS AGAIN IF UNDE-
                                ;FINED ERROR HAS OCCURRED
032052  104040                  ERROR   40              ;THERE WAS A READ/WRITE HEADER ERROR
                                ;DURING 'DTE' TEST SETUP
```

```
4093  032060                        6$:   ;NOW THE HEADER HAS BEEN READ,
4094                                      ;*560 SECTOR CLOCKS WILL BE GIVEN -
4095                                      ;*GAP 11 BYTES, SYNC 1 BYTE, DATA 512, ECC 4 BYTES
4096                                      ;*GAP 2 BYTES, TOLERANCE 28 BYTES, EXTRA 2
4097
4098                                      ;*THESE 560 SECTOR CLOCKS ARE DIVIDED INTO TWO GROUPS
4099                                      ;*24 SECTOR CLOCKS WITH NORMAL DIAGNOSTIC DATA CLOCKS
4100                                      ;*AND 536 SECTOR CLOCKS WITHOUT ANY DIAGNOSTIC DATA CLOCKS
4101
4102
4103                                      ;*THIS GIVES 24 SECTOR CLOCKS WITH DIAGNOSTIC DATA CLOCKS
4104
4105  032060  012701  000030              MOV     #24.,R1         ;LOAD SECTOR CLOCK COUNTER
4106  032064  013700  014776              MOV     @#RHMR,R0       ;GET RHMR ADDRESS
4107  032070  012710  000001              MOV     #DMD,@R0        ;SET DIAGNOSTIC MODE
4108  032074  052710  000012        1$:   BIS     #MSTCK!MCLK,@R0 ;SET SECTOR CLOCK AND DATA CLOCK
4109  032100  042710  000012              BIC     #MSTCK!MCLK,@R0 ;CLEAR SECTOR CLOCK AND DATA CLOCK
4110  032104  012702  000007              MOV     #7,R2           ;LOAD COUNTER FOR DIAGNOSTIC DATA CLOCKS
4111  032110  052710  000002        4$:   BIS     #MCLK,@R0       ;SET CLOCK (DATA)
4112  032114  042710  000002              BIC     #MCLK,@R0       ;CLEAR CLOCK (DATA)
4113  032120  005302                      DEC     R2              ;COUNT
4114  032122  001372                      BNE     4$              ;BRANCH IF 7 NOT DONE
4115  032124  005301                      DEC     R1              ;COUNT
4116  032126  001362                      BNE     1$              ;BRANCH IF 24 NOT DONE
4117
4118                                      ;*THIS GIVES 536 SECTOR CLOCKS WITHOUT DIAGNOSTIC DATA CLOCKS
4119
4120  032130  012701  001030              MOV     #536.,R1        ;LOAD SECTOR CLOCK COUNTER
4121  032134  052710  000010        5$:   BIS     #MSTCK,@R0      ;SET SECTOR CLOCK
4122  032140  042710  000010              BIC     #MSTCK,@R0      ;CLEAR SECTOR CLOCK
4123  032144  005301                      DEC     R1              ;COUNT
4124  032146  001372                      BNE     5$              ;BRANCH IF 536 NOT DONE
4125
4126
4127
4128                                      ;*ECC PATTERN REGISTER IS NOT CHECKED
4129  032150  017737  162634  015064      MOV     @RHEC2,@#EC2    ;RHEC2 IS NOT CHECKED
4130
4131
4132                                      ;*NOW 'DTE' SHOULD BE SET, CHANGE SAVED REGISTERS TO EXPECTED VALUES
4133
4134  032156  005737  017334              TST     @#RH70          ;CHECK FOR RH70 CONTROLLER
4135  032162  001412                      BEQ     7$              ;SKIP RH70 CODE AND DO RH11 IF NOT
4136
4137  032164  012737  177416  015024      MOV     #-242.,@#WC     ;SAVED RHWC
4138  032172  012737  015254  015026      MOV     #WRFROM+<14.*2>,@#BA ;SAVED RHBA
4139  032200  052737  000300  015030      BIS     #IR!OR,@#CS2    ;SAVED RHCS2
4140  032206  000414                      BR      8$              ;SKIP NEXT RH11 CODE
4141
4142  032210  012737  177511  015024  7$: MOV     #-183.,@#WC     ;SAVED RHWC
4143  032216  012737  015442  015026      MOV     #WRFROM+<73.*2>,@#BA ;SAVED RHBA
4144  032224  042737  000100  015030      BIC     #IR,@#CS2       ;SAVED RHCS2
4145  032232  052737  000200  015030      BIS     #OR,@#CS2       ;SAVED RHCS2
4146
4147  032240  052737  140000  015032  8$: BIS     #SC!TRE,@#CS1   ;SAVED RHCS1
4148  032246  052737  010000  015034      BIS     #DTE,@#ER1      ;SAVED RHER1
```

```
4149  032254  012737  000201  015052          MOV     #DENVL!DMD,@#MR ;SAVED RHMR
4150  032262  052737  140000  015054          BIS     #ATA!ERR,@#DS1 ;SAVED RHDS1
4151  032270  012737  000100  015066          MOV     #100,@#LA       ;SAVED RHLA
4152  032276  012737  000001  015036          MOV     #1,@#DST        ;SAVED RHDST
4153  032304  013737  015134  015050          MOV     @#ATTENT,@#AS   ;SAVED RHAS
4154
4155                                          ;*NOW READ & SAVE REGISTERS AGAIN SO THAT COMPARISONS
4156                                          ;CAN BE DONE (USING 'WRFROM' BUFFER THIS TIME)
4157
4158  032312  004037  046456                  JSR     R0,@#SAVER      ;READ IN SEQUENCE
4159  032316  014750                          RHWC                    ;FROM HARDWARE REGISTER
4160  032320  016264                          REINTO                  ;INTO CORE BUFFER LOCATION
4161  032322  000023                          19.                     ;NUMBER OF REGISTERS TO READ
4162
4163                                          ;*FOR RHAS UPPER BYTE
4164  032324  113737  015051  016311          MOVB    @#AS+1,@#REINTO+25 ;UPPER RHAS
4165
4166
4167                                          ;*COMPARE CHANGED REGISTER SNAPSHOT BEFORE COMMAND WITH
4168                                          ;*SNAPSHOT AFTER COMMAND
4169
4170  032332  004037  046660                  JSR     R0,@#COMPAR     ;COMPARE
4171  032336  015024                          WC                      ;CHANGED INITIAL SNAPSHOT BUFFER
4172  032340  016264                          REINTO                  ;SNAPSHORT BUFFER AFTER COMMAND
4173  032342  000022                          18.                     ;NUMBER OF REGISTERS TO COMPARE
4174  032344  032352                          2$                      ;RETURN FOR ERROR
4175  032346  032352                          2$                      ;SAME
4176  032350  032372                          3$                      ;RETURN FOR GOOD COMPARISON
4177
4178  032352  013705  052770          2$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
4179  032356  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
4180  032360  016537  014746  045530          MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER
4181  032366  104001                          ERROR   1               ;IMPROPER REGISTER
4182                                                                  ;CHANGE AFTER FORCING
4183                                                                  ;'DTE' ERROR
4184  032370  000207                          RTS     PC              ;RETURN
4185
4186  032372                          3$:     ;GOOD, REGISTERS OK - GO ON TO NEXT TEST
4187
4188
```

```
4189
4190  032374  012706  001000                  MOV     #STACK,SP       ;RESET STACK
4191
4192                                          ;*THESE ARE TO SET UP FOR DISKLESS USE ONLY
4193
4194  032406  012737  010000  056006          MOV     #FMT22,@#WCYL   ;FORMAT 22=16 BITWORDS AND
4195                                                                  ;CYLINDER 0
4196  032414  005037  056010                  CLR     @#WSECTR        ;TRACK=0, SECTOR=0
4197  032420  005037  056012                  CLR     @#WKEY1         ;KEY1=0
4198  032424  005037  056014                  CLR     @#WKEY2         ;KEY2=0
4199  032430  012737  000400  056046          MOV     #256.,@#FNWORD  ;256 DATAWORDS
4200  032436  004537  047172                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
4201  032442  056006                          WCYL
4202  032444  056016                          GCRC
4203
4204                                          ;* THESE        ARE REGULAR SETUPS & CHECKS
4205
4206  032446  004737  045764                  JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
4207  032452  012777  177374  162270          MOV     #-260.,@RHWC    ;256 DATA WORDS & 4 HEADER WORDS
4208  032460  012777  015220  162264          MOV     #WRFROM,@RHBA   ;STARTING ADDRESS OF BUFFER
4209  032466  005077  162270                  CLR     @RHDST          ;TRACK = 0
4210                                                                  ;SECTOR = 0
4211  032472  012777  014000  162266          MOV     #FMT22!ECI,@RHOF ;16 BITS PER WORD
4212                                                                  ;ECC CORRECTION INHIBITED
4213  032500  005077  162264                  CLR     @RHCA           ;CYLINDER = 0
4214  032516  013711  015172                  MOV     @#WRIFOR,@R1    ;WRITE HEADER AND DATA = 62
4215
4216                                          ;*READ & SAVE REGISTERS FOR COMPARISON AFTER SIMULATED 'DTE'
4217
4218  032522  004037  046456                  JSR     R0,@#SAVER      ;READ IN SEQUENCE
4219  032526  014750                          RHWC                    ;FROM HARDWARE REGISTER
4220  032530  015024                          WC                      ;INTO CORE BUFFER LOCATION
4221  032532  000023                          19.                     ;NUMBER OF REGISTERS TO READ
```

```
4222
4223                                        ;*NOW 'GO' WILL BE GIVEN. EVERYTHING WILL BE TREATED
4224                                        ;*NORMALLY TILL HEADER IS TO BE GIVEN, THEN ONLY
4225                                        ;*SECTOR CLOCKS WILL BE GIVEN. NO DIAGNOSTIC DATA
4226                                        ;*CLOCKS WILL BE GIVEN, THIS SHOULD BRING SECTOR PULSE HIGH
4227                                        ;*WITHOUT PUTTING "READ" DOWN, HENCE 'DTE' WILL COME UP.
4228
4229  032534  012737  177777  015144        MOV     #-1,@#TESDTE    ;SET DTE TEST
4230  032542  012737  177777  015124        MOV     #-1,@#ERFLG$    ;THIS WILL BRING THE READ HEADER
4231                                                                ;AND DATA PROCESS OUT AFTER THE
4232                                                                ;HEADER HAS BEEN CORRECTLY READ
4233
4234  032550  004737  055632                JSR     PC,@#COMWHD     ;ISSUE 'GO', SEARCH FOR SECTOR,
4235                                                                ;WRITE HEADER AND DATA.
4236
4237  032554  017737  162176  001176  SETCK3: MOV   @RHCS1,@#$TMPO  ;READ CS1 TO CHECK FOR ERRORS DURING WRITE
4238  032562  032737  100000  001176        BIT     #SC,@#$TMPO     ;TEST FOR "SPECIAL CONDITION" - 'SC'
4239  032570  001405                        BEQ     4$              ;CONTINUE TEST IF NO ERROR ('SC' = 0)
4240  032572  004737  045464                JSR     PC,@#PUTREG     ;READ & SAVE REGISTERS AGAIN IF ERROR
4241  032576  104040                        ERROR   40              ;THERE WAS A READ/WRITE HEADER ERROR
4242                                                                ;DURING 'DTE' TEST SETUP
4243
4244  032604                        4$:     ;NOW SECTOR HAS BEEN FOUND OK
4245
4246                                        ;*609 SECTOR CLOCKS WILL BE GIVEN,
4247                                        ;*39 BYTES FOR SECTOR GAP,
4248                                        ;*1 BYTE FOR HEADER SYNC,
4249                                        ;*8 BYTES FOR HEADER,
4250                                        ;*GAP 11 BYTES, SYNC 1 BYTE, DATA 512, ECC 4 BYTES
4251                                        ;*GAP 2 BYTES, TOLERANCE 28 BYTES, EXTRA 3
4252
4253
4254                                        ;*THIS GIVES 609 SECTOR CLOCKS WITHOUT DIAGNOSTIC DATA CLOCKS
4255
4256  032604  012701  001141        5$:     MOV     #609.,R1        ;LOAD SECTOR CLOCK COUNTER
4257  032610  052710  000010                BIS     #MSTCK,@R0      ;SET SECTOR CLOCK
4258  032614  042710  000010                BIC     #MSTCK,@R0      ;CLEAR SECTOR CLOCK
4259  032620  005301                        DEC     R1              ;COUNT
4260  032622  001372                        BNE     5$              ;BRANCH IF 536 NOT DONE
4261
4262                                        ;*NOW 'DTE' SHOULD BE SET, CHANGE SAVED REGISTERS
4263                                        ;TO EXPECTED VALUES
4264
4265  032624  005737  017334                TST     @#RH70          ;CHECK FOR RH70 CONTROLLER
4266  032630  001407                        BEQ     6$              ;SKIP RH70 CODE AND DO RH11 IF NOT
4267
4268  032632  012737  177404  015024        MOV     #-252.,@#WC     ;SAVED RHWC
4269  032640  012737  015240  015026        MOV     #WRFROM+<8.*2>,@#BA ;SAVED RHBA
4270  032646  000406                        BR      7$              ;SKIP NEXT RH11 CODE
4271
4272  032650  012737  177477  015024  6$:   MOV     #-193.,@#WC     ;SAVED RHWC
4273  032656  012737  015426  015026        MOV     #WRFROM+<67.*2>,@#BA ;SAVED RHBA
4274
4275  032664  052737  140000  015032  7$:   BIS     #SC!TRE,@#CS1   ;SAVED RHCS1
4276  032672  042737  000100  015030        BIC     #IR,@#CS2       ;SAVED RHCS2
4277  032700  052737  000200  015030        BIS     #OR,@#CS2       ;SAVED RHCS2
```

```
4278  032706  052737  010000  015034        BIS     #DTE,@#ER1      ;SAVED RHER1
4279  032714  012737  000401  015052        MOV     #401,@#MR       ;SAVED RHMR
4280  032722  052737  140000  015054        BIS     #ATA!ERR,@#DS1  ;SAVED RHDS1
4281  032730  012737  000100  015066        MOV     #100,@#LA       ;SAVED RHLA
4282  032736  012737  000001  015036        MOV     #1,@#DST        ;SAVED RHDST
4283  032744  013737  015134  015050        MOV     @#ATTENT,@#AS   ;SAVED RHAS
4284
4285                                        ;*NOW READ & SAVE REGISTERS AGAIN SO THAT COMPARISONS CAN BE DONE
4286
4287  032752  004037  046456                JSR     R0,@#SAVER      ;READ IN SEQUENCE
4288  032756  014750                        RHWC                    ;FROM HARDWARE REGISTER
4289  032760  016264                        REINTO                  ;INTO CORE BUFFER LOCATION
4290  032762  000023                        19.                     ;NUMBER OF REGISTERS TO READ
4291
4292                                        ;*FOR RHAS UPPER BYTE
4293  032764  113737  015051  016311        MOVB    @#AS+1,@#REINTO+25 ;UPPER RHAS
4294
4295                                        ;*COMPARE CHANGED REGISTER SNAPSHOT BEFORE COMMAND
4296                                        ;*WITH REGISTER SNAPSHOT AFTER COMMAND
4297
4298  032772  004037  046660                JSR     R0,@#COMPAR     ;COMPARE
4299  032776  015024                        WC                      ;CHANGED REGISTER SNAPSHOT
4300  033000  016264                        REINTO                  ;SNAPSHOT AFTER COMMAND
4301  033002  000022                        18.                     ;NUMBER OF REGISTERS TO COMPARE
4302  033004  033012                        2$                      ;RETURN FOR ERROR
4303  033006  033012                        2$                      ;SAME
4304  033010  033032                        3$                      ;RETURN FOR GOOD COMPARISON
4305
4306  033012  013705  052770        2$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
4307  033016  060505                        ADD     R5,R5           ;DOUBLE ERROR WORD
4308  033020  016537  014746  045530        MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER
4309  033026  104001                        ERROR   1               ;IMPROPER REGISTER
4310                                                                ;CHANGE AFTER FORCING
4311                                                                ;'DTE' ERROR
4312  033030  000207                        RTS     PC              ;RETURN
4313
4314  033032                        3$:                             ;GOOD, REGISTERS COMPARE OK
4315                                                                ;GO ON TO THE NEXT TEST
4316
```

```
 4317  033034  012706  001000                  MOV     #STACK,SP       ;RESET STACK
 4318  033046  004737  045764                  JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS & CLEAR
 4319                                                                  ;THE DRIVE
 4320  033052  012737  000026  015146          MOV     #22.,@#TAGDTE   ;22 SECTORS
 4321                                                                  ;THIS TEST REPEATS
 4322                                                                  ;ITSELF 22 TIMES
 4323
 4324                                          ;*THE FOLLOWING INITIALIZES FOR SECTOR 0
 4325
 4326  033060  005037  033164                  CLR     @#SS3+2         ;HEADER (SECTOR)
 4327  033064  012737  000025  033170          MOV     #21.,@#SS4+2    ;HEADER (KEY1)
 4328  033072  012737  000025  033174          MOV     #21.,@#SS5+2    ;HEADER (KEY2)
 4329  033100  005037  033222                  CLR     @#SS7+2         ;DATA (SECTOR)
 4330  033104  005037  033304                  CLR     @#SS10+2        ;DATA
 4331  033110  005037  033334                  CLR     @#SS12+2        ;SECTOR (SIMULATED DISK)
 4332  033114  012737  000025  033342          MOV     #21.,@#SS13+2   ;KEY1 (SIMULATED DISK)
 4333  033122  012737  000025  033350          MOV     #21.,@#SS14+2   ;KEY2 (SIMULATED DISK)
 4334  033130  005037  033410                  CLR     @#SS15+2        ;SECTOR (RHDST)
 4335
 4336                                          ;*CLEAR SIMULATED DISK AREA
 4337  033134                          SS1:
 4338  033134  012700  054470          1$:     MOV     #SECGAP,R0      ;POINTER
 4339  033140  012701  000460                  MOV     #304.,R1        ;COUNTER
 4340  033144  005020                  2$:     CLR     (R0)+           ;CLEAR SIMULATED DISK AREA
 4341  033146  005301                          DEC     R1              ;COUNT
 4342  033150  001375                          BNE     2$
 4343
 4344
 4345                                          ;*SETUP WRITE FROM BUFFER
 4346  033152  012700  015220                  MOV     #WRFROM,R0
 4347
 4348                                          ;*HEADER
 4349  033156  012720  010000                  MOV     #FMT22,(R0)+    ;FORMAT 16 BITS PER WORD
 4350                                                                  ;CYLINDER 0
 4351  033162  012720  000000          SS3:    MOV     #0,(R0)+        ;SECTOR TO VARY
 4352  033166  012720  000025          SS4:    MOV     #21.,(R0)+      ;KEY1 TO VARY
 4353  033172  012720  000025          SS5:    MOV     #21.,(R0)+      ;KEY2 TO VARY
 4354
 4355                                          ;*DATA IN WRITE FROM BUFFER ALTHOUGH THIS IS DATA AND NOT
 4356                                          ;*HEADER, THE SECTOR WITH SYNC BYTES WILL BE GIVEN AS DATA.
 4357
 4358                                          ;*DATA IS - 19 WORDS OF ZEROS - SYNC WORDS, 4 HEADER WORDS
 4359                                          ;*1 CRC WORD, 5 WORDS OF ZEROS, 1 SYNC WORD, 100 ZEROS
 4360                                          ;*(DATA), 1 SYNC WORD, 70 SECTOR NUMBER TO VARY
 4361
 4362  033176  012705  000023                  MOV     #19.,R5         ;COUNTER
 4363  033202  005020          6$:     CLR     (R0)+           ;19 ZEROS
 4364  033204  005305                  DEC     R5              ;COUNT
 4365  033206  001375                  BNE     6$              ;19 DONE?
 4366  033210  013720  052752          MOV     @#RSYNC,(R0)+   ;SYNC = 14400
 4367  033214  012720  010000          MOV     #FMT22,(R0)+    ;CYLINDER 0
 4368  033220  012720  000000          SS7:    MOV     #0,(R0)+        ;SECTOR TO VARY
 4369  033224  005020                  CLR     (R0)+
 4370  033226  005020                  CLR     (R0)+
 4371  033230  004537  047172                  JSR     R5,@#CRC        ;CALCULATE CRC FOR ABOVE 4 WORDS
 4372  033234  015270                  WRFROM+50                       ;4 WORDS START FROM HERE
```

```
4373  033236  015300                          WRFROM+60               ;PUT CRC HERE
4374
4375  033240  005720                          TST     (R0)+           ;INCREMENT R0
4376
4377  033242  012705  000005                  MOV     #5.,R5
4378  033246  005020                  8$:     CLR     (R0)+           ;5 WORDS OF ZEROS
4379  033250  005305                          DEC     R5              ;COUNT
4380  033252  001375                          BNE     8$              ;BRANCH IF 5 NOT DONE
4381
4382  033254  013720  052752                  MOV     @#RSYNC,(R0)+   ;SYNC = 14400
4383
4384  033260  012705  000144                  MOV     #100.,R5
4385  033264  005020                  9$:     CLR     (R0)+           ;100 WORDS OF ZEROS
4386  033266  005305                          DEC     R5
4387  033270  001375                          BNE     9$
4388
4389  033272  013720  052752                  MOV     @#RSYNC,(R0)+   ;SYNC = 14400
4390  033276  012705  000106                  MOV     #70.,R5
4391  033302  012720  000000          SS10:   MOV     #0,(R0)+        ;SECTOR TO VARY
4392  033306  005305                          DEC     R5
4393  033310  001374                          BNE     SS10
4394
4395                                          ;*CLEAR REST OF 256 WORDS THAT IS 54 WORDS OF ZEROS
4396
4397  033312  012705  000066                  MOV     #54.,R5
4398  033316  005020                  11$:    CLR     (R0)+
4399  033320  005305                          DEC     R5
4400  033322  001375                          BNE     11$
4401
4402                                          ;*THESE ARE TO BE SET UP FOR DISKLESS USE ONLY
4403
4404  033324  012737  010000  056006          MOV     #FMT22,@#WCYL   ;FORMAT = 16 BIT WORDS
4405                                                                  ;CYLINDER = 0
4406  033332  012737  000000  056010  SS12:   MOV     #0,@#WSECTR     ;SECTOR TO VARY
4407  033340  012737  000025  056012  SS13:   MOV     #21.,@#WKEY1    ;KEY1 TO VARY
4408  033346  012737  000025  056014  SS14:   MOV     #21.,@#WKEY2    ;KEY2 TO VARY
4409  033354  012737  000312  056046          MOV     #202.,@#FNWORD  ;202 DATA WORDS
4410  033362  004537  047172                  JSR     R5,@#CRC        ;CALCULATE CRC
4411  033366  056006                          WCYL                    ;FIRST WORD
4412  033370  056016                          GCRC                    ;PUT HERE
4413
4414                                          ;*THESE ARE REGULAR SETUPS
4415
4416  033372  012777  177400  161350          MOV     #-256.,@RHWC    ;202 DATA, 4 HEADER
4417  033400  012777  015220  161344          MOV     #WRFROM,@RHBA   ;FILL BUS ADDRESS
4418  033406  012777  000000  161346  SS15:   MOV     #0,@RHDST       ;SECTOR TO VARY
4419  033414  013777  015172  161334          MOV     @#WRIFOR,@RHCS1 ;GET READY TO DO
4420                                                                  ;WRITE HEADER AND DATA
4421                                                                  ;WITH 62 FUNCTION CODE IN RHCS1
4422  033422  012777  010000  161336          MOV     #FMT22,@RHOF    ;16 BITS PER WORD FORMAT
4423  033430  005077  161334                  CLR     @RHCA           ;CYLINDER = 0
4424
4425  033434  005037  015124                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
4426
4427
4428  033452  004737  055632                  JSR     PC,@#COMWHD     ;ISSUE 'GO', COUNT SECTOR CLOCKS.
```

```
                                                ;WRITE HEADER AND DATA
033456  005737  015124                TST     @#ERFLG$        ;HAVE ANY ERRORS OCCURRED ?


                                        ;*NOW COMPARE "DISK" BUFFER WITH "REINTO" BUFFER
033476  004037  046660                JSR     R0,@#COMPAR     ;CHECK
033502  015230                        WRFROM+8.               ;GOOD BUFFER
033504  054566                        DISK                    ;TEST BUFFER
033506  000400                        256.                    ;NUMBER OF WORDS
033510  033516                        16$                     ;RETURN POINT FOR ERROR  HEADER
033512  033522                        17$                     ;RETURN POINT FOR ERROR DATA
033514  033526                        18$                     ;RETURN FOR GOOD COMPARISON
033516  104007          16$:          ERROR   7
033520  000207                        RTS     PC
033522  104010          17$:          ERROR   10
033524  000207                        RTS     PC


                                        ;*THE FOLLOWING INCREMENTS ARE TO CHANGE THE ABOVE SET UP
                                        ;*TO WRITE ON THE NEXT SECTOR
033526  005237  033164        18$:    INC     @#SS3+2         ;HEADER (SECTOR)
033532  005337  033170                DEC     @#SS4+2         ;HEADER (KEY1)
033536  005337  033174                DEC     @#SS5+2         ;HEADER (KEY2)
033542  005237  033222                INC     @#SS7+2         ;DATA (SECTOR)
033546  005237  033304                INC     @#SS10+2        ;DATA
033552  005237  033334                INC     @#SS12+2        ;SECTOR (SIMULATED DISK)
033556  005337  033342                DEC     @#SS13+2        ;KEY1 (SIMULATED DISK)
033562  005337  033350                DEC     @#SS14+2        ;KEY2 (SIMULATED DISK)
033566  005237  033410                INC     @#SS15+2        ;SECTOR (RHDST)

033572  005337  015146        SS2:    DEC     @#TAGDTE        ;COUNT DOWN FOR 22 SECTORS
033576  001001                        BNE     1$              ;BRANCH IF 22 SECTORS NOT DONE
033602  000137  033134        1$:     JMP     @#SS1           ;GO BACK TO NEXT SECTOR
```

```
                                        .SBTTL  DATA TRANSFER TESTS USING ECC


033610  012706  001000                  MOV     #STACK,SP       ;RESET STACK

033622  012700  054470                  MOV     #SECGAP,R0      ;POINTER
033626  012701  000402                  MOV     #258.,R1        ;COUNTER
033632  012720  177777          1$:     MOV     #-1,(R0)+       ;FILL SIMULATER DISK WITH ONES
033636  005301                          DEC     R1
033640  001374                          BNE     1$
033642  004737  045764                  JSR     PC,@#CLDISK     ;THIS IS USED TO SET GENERAL REGISTERS

                                        ;*THESE ARE FOR ECC TEST ONLY

033646  012737  177777  015142          MOV     #-1,@#TSECC     ;THIS IS AN ECC TEST
033654  005037  050472                  CLR     @#POSITI        ;CLEAR ERROR POSITION COUNTER
033660  013737  050466  050470          MOV     @#NCODE,@#NCOUNT  ;TEMPORARY N-CODE COUNTER
033666  013737  050474  050502          MOV     @#HARDER,@#HADTMP  ;TEMPORARY HARD ERROR COUNTER
033674  005037  050460                  CLR     @#GECC1         ;ECC LOW ORDER TO BE GENERATED
033700  005037  050462                  CLR     @#GECC2         ;ECC HIGH ORDER TO BE GENERATED
033704  005037  050476                  CLR     @#DATENV        ;CLEAR DATA ENVELOPE CLOCK COUNT
033710  005037  050500                  CLR     @#ZCODE         ;CLEAR LEADING ZEROS CLOCK COUNT


                                        ;*THESE ARE TO BE SETUP FOR DISKLESS USE ONLY

033714  012737  010000  056006          MOV     #FMT22,@#WCYL   ;FORMAT22=16BIT WORDS AND
                                                                ;CYLINDER 0
033722  012737  000001  056010          MOV     #1,@#WSECTR     ;TRACK=0, SECTOR=1
033730  005037  056012                  CLR     @#WKEY1         ;KEY1=0
033734  005037  056014                  CLR     @#WKEY2         ;KEY2=0
033740  012737  000400  056046          MOV     #256.,@#FNWORD  ;256 DATA WORDS
033746  004537  047172                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
033752  056006                          WCYL
033754  056016                          GCRC

                                        ;*THESE ARE REGULAR SETUPS

033756  012777  177374  160764          MOV     #-260.,@RHWC    ;256 DATA WORDS 4 HEADER WORDS
033764  012700  015220                  MOV     #WRFROM,R0      ;THESE TWO INSTRUCTIONS GETS
033770  010077  160756                  MOV     R0,@RHBA        ;ADDR. OF WRFROM INTO R0 AND
                                                                ;BUS ADDRESS REGISTER
033774  012720  010000                  MOV     #FMT22,(R0)+    ;FORMAT=16 BIT WORDS
                                                                ;CYLINDER=0
034000  012720  000001          2$:     MOV     #1,(R0)+        ;TRACK=0, SECTOR=1, KEYS=0
034004  005020                          CLR     (R0)+           ;KEY1=0
034006  005020                          CLR     (R0)+           ;KEY2=0
034010  012705  000400                  MOV     #256.,R5        ;COUNTER
034014  012720  000000          3$:     MOV     #0,(R0)+        ;MOVE ALL ZEROS FOR DATA
034020  005305                          DEC     R5
034022  001374                          BNE     3$              ;BRANCH IF DATA NOT COMPLETE
034024  012777  000001  160730          MOV     #1,@RHDST       ;TRACK=0 SECTOR=1
```

```
034044  013711  015172                  MOV     @#WRIFOR,@R1     ;GET READY FOR WRITE HEADER AND
                                                                 ;DATA WITH 62 IN RHCS1
034050  005037  015124                  CLR     @#ERFLGS         ;CLEAR ERROR FLAG
034054  012777  010000  160704          MOV     #FMT22,@RHOF     ;FORMAT BIT=1 (16 BIT WORDS)
034062  005077  160702                  CLR     @RHCA            ;CYLINDER =0
034066  004737  055632                  JSR     PC,@#COMWHD      ;WRITE HEADER AND DATA

                                        ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
                                        ;*FROM THE "COMWHD" ROUTINE THAT MEANS ALL HEADER ON DISK
                                        ;*IS GOOD IE. ONLY DATA IS TO BE CHECKED TO SEE IF THEY ARE
                                        ;*ALL ZEROS AND WRITE DATA GAP AND TOLERANCE GAP TO SEE IF
                                        ;*THEY ARE ALL ZEROS

034072  005737  015124                  TST     @#ERFLGS          ;HAS ANY ERRORS OCCURED?
                                                                  ;IF WRITE ERROR OCCURS ECC IS NOT CHECKED


                                        ;*COMPARE SOFTWARE GENERATED ECC WITH THAT GENERATED BY HARDWARE

034100  023737  050460  055566          CMP     @#GECC1,@#WECC1;COMPARE SOFTWARE ECC WITH HARDWARE ECC
034106  001402                          BEQ     6$               ;BRANCH IF GOOD
034110  104031                          ERROR   31               ;LOW ORDER ECC IN ERROR
034112  000405                          BR      7$               ;BRANCH TO CONTINUE
034114  023737  050462  055570  6$:     CMP     @#GECC2,@#WECC2;COMPARE SOFTWARE ECC WITH HARDWARE ECC
034122  001401                          BEQ     7$               ;BRANCH IF GOOD
034124  104031                          ERROR   31               ;HIGH ORDER ECC IN ERROR


034126                          7$:




                                        ;*FILL "REINTO" BUFFER WITH EXPECTED DATA

034140  004037  045702                  JSR     R0,@#CLAREA      ;FILL REINTO BUFFER
034144  016264                                  REINTO           ;FROM
034146  017262                                  REINTO+<255.*2>  ;TO
034150  000000                          .WORD   0                ;DATA

034152  013737  050460  017264          MOV     @#GECC1,@#REINTO+<256.*2>;FILL ECC1
034160  013737  050462  017266          MOV     @#GECC2,@#REINTO+<257.*2>;FILL ECC2
034166  004037  045702                  JSR     R0,@#CLAREA      ;FILL REST
034172  017270                          REINTO+<258.*2>          ;FROM
034174  017324                          REINTO+<272.*2>          ;TO
034176  000000                          0                        ;DATA


034200  005037  015124                  CLR     @#ERFLGS         ;CLEAR ERROR FLAG


                                        ;*NOW COMPARE "DISK" BUFFER WITH "REINTO"

034204  004037  046660                  JSR     R0,@#COMPAR      ;CHECK
034210  016264                          REINTO                   ;GOOD BUFFER
034212  054566                          DISK                     ;TEST BUFFER
```

```
4577  034214  000402                        258.                    ;NUMBER OF WORDS CHECKED
4578  034216  034224                        4$                      ;RETURN POINT FOR ERROR HEADER
4579  034220  034230                        5$                      ;RETURN POINT FOR ERROR DATA
4580  034224  104007              4$:       ERROR   7               ;READ ERROR 10 NEXT
4581  034226  000207                        RTS     PC              ;RETURN TO COMPARE
4582  034230  104010              5$:       ERROR   10              ;WORD NOS 1 TO 256 ARE
4583                                                                ;DATA WORDS
4584                                                                ;WORD NOS 257 AND 258
4585                                                                ;ARE ECC WHICH ARE CHECKED
4586                                                                ;WORD NOS 259
4587                                                                ;IS DATA GAP
4588                                                                ;WORD NOS 260 TO 273
4589                                                                ;ARE TOLERANCE GAP
4590  034232  000207                        RTS     PC              ;RETURN TO COMPARE
4591
4592
4593
4594
4595
```

```
4596
4597  034236  012706  001000                  MOV     #STACK,SP       ;RESET STACK
4598
4599
4600
4601                                  ;       SETUP FOR WHAT IS TO BE READ
4602                                  ;       HEADER CRC IS RESTORED FROM A SUBROUTINE
4603
4604  034250  012746  000000                  MOV     #0,-(SP)        ;DATA TO BE READ
4605  034254  012705  000400                  MOV     #256.,R5        ;COUNTER
4606  034260  012700  054566                  MOV     #DISK,R0        ;START OF SIMULATED DISK DATA
4607  034264  011620                  1$:     MOV     (SP),(R0)+      ;MOVE IN DATA ON TO SIMULATED DISK
4608  034266  005305                          DEC     R5              ;COUNT
4609  034270  001375                          BNE     1$              ;BRANCH IF 256 NOT COMPLETE
4610  034272  005726                          TST     (SP)+           ;UNDO -(SP)
4611  034274  022020                          CMP     (R0)+,(R0)+     ;JUMP OVER THE TWO ECC WORDS
4612  034276  012705  000017                  MOV     #15.,R5         ;1 DATA GAP
4613                                                                  ;14 TOLERANCE GAP
4614  034302  005020                  2$:     CLR     (R0)+           ;CLEAR DATA GAP, AND
4615  034304  005305                          DEC     R5              ;TOLERANCE GAP
4616  034306  001375                          BNE     2$              ;BRANCH IF NOT COMPLETE
4617
4618
4619  034310  004737  051254                  JSR     PC,@#FILLEC     ;INSERT THE TWO ECC WORDS ON THE DISK
4620                                                                  ;IN THE CORRECT PLACE
4621
4622                                  ;*THESE ARE FOR ECC TEST ONLY
4623
4624  034314  012737  177777  015142          MOV     #-1,@#TSECC     ;THIS IS AN ECC TEST
4625  034322  005037  050472                  CLR     @#POSITI        ;CLEAR ERROR POSITION COUNTER
4626  034326  013737  050466  050470          MOV     @#NCODE,@#NCOUNT  ;TEMPORARY N-CODE COUNTER
4627  034334  013737  050474  050502          MOV     @#HARDER,@#HADTMP  ;TEMPORARY HARD ERROR COUNTER
4628  034342  005037  050460                  CLR     @#GECC1         ;ECC LOW ORDER TO BE GENERATED
4629  034346  005037  050462                  CLR     @#GECC2         ;ECC HIGH ORDER TO BE GENERATED
4630  034352  005037  050476                  CLR     @#DATENV        ;CLEAR DATA ENVELOPE CLOCK COUNT
4631  034356  005037  050500                  CLR     @#ZCODE         ;CLEAR LEADING ZEROS CLOCK COUNT
4632
4633
4634                                  ;*THESE ARE TO SETUP FOR DISKLESS USE ONLY
4635
4636  034362  012737  010000  052650          MOV     #FMT22,@#CYL    ;16 BITS PER WORD
4637                                                                  ;CYLINDER 0, FORMAT 16 BITS
4638  034370  112737  000000  052653          MOVB    #0,@#SECOTR+1   ;TRACK 0
4639  034376  112737  000000  052652          MOVB    #0,@#SECOTR     ;SECTOR 0
4640  034404  012737  000000  052654          MOV     #0,@#KEY1       ;KEY1=0
4641  034412  012737  000000  052656          MOV     #0,@#KEY2       ;KEY2=0
4642  034420  012737  000400  052730          MOV     #256.,  @#DAWORD  ;NO. OF DATA WORDS
4643  034426  005037  052660                  CLR     @#X             ;THIS IS A READ COMMAND
4644  034432  004537  047172                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
4645  034436  052650                          CYL
4646  034440  054550                          WCRC
4647
4648
4649
4650
4651                                  ;*THESE ARE REGULAR SETUPS
```

```
4652
4653  034442  004737  045764                  JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
4654  034446  012777  177374  160274          MOV     #-256.,-4.,@RHWC ;256. DATA 4 HEADER WORDS
4655  034454  012777  016264  160270          MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
4656  034462  112746  000000                  MOVB    #0,-(SP)        ;IN LOWER BYTE GET SECTOR
4657  034466  112766  000000  000001          MOVB    #0,1(SP)        ;GET TRACK IN HIGHER BYTE
4658  034474  012677  160262                  MOV     (SP)+,  @RHDST  ;TRACK/SECTOR IN RHDST
4659  034500  012777  010000  160260          MOV     #FMT22,@RHOF    ;16 BITS PER WORD
4660                                                                  ;ECC CORRECTION NOT INHIBIT
4661                                                                  ;BECAUSE ECC IS NOT GOING
4662                                                                  ;TO BE CHECKED
4663  034506  005077  160256                  CLR     @RHCA           ;CYLINDER 0
4664
4665
4666  034524  013711  015176                  MOV     @#REFOR,@R1     ;READ HEADER AND DATA=72
4667  034530  005037  015124                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
4668  034534  004737  052510                  JSR     PC,@#COMHD      ;READ HEADER AND DATA
4669                                                                  ;IF THERE ARE READ ERRORS THEN
4670                                                                  ;ECC WILL NOT BE CHECKED
4671
4672
4673                                          ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
4674                                          ;*FROM THE "COMHD" ROUTINE THAT MEANS SECTOR GAP,
4675                                          ;*FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
4676                                          ;*SYNC BYTE HAVE GONE BY AND SYNCS WERE CORRECTLY
4677                                          ;*DETECTED
4678                                          ;*HEADER AND DATA ARE TO BE CHECKED.
4679                                          ;*IN CHECKING READ DATA THE WRITE FROM BUFFER
4680                                          ;*"WRFROM" IS FILLED WITH EXPECTED DATA AND
4681                                          ;*COMPARISONS ARE MADE
4682
4683  034540  005737  015124                  TST     @#ERFLG$        ;ANY ERRORS ALREADY THERE
4684  034546  004737  045464                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
4685  034552  005737  015034                  TST     @#ER1           ;NO ERRORS SHOULD BE SET
4686  034556  001401                          BEQ     6$              ;BRANCH IF NO ERRORS SET
4687  034560  104032                          ERROR   32              ;32 BIT ECC REGISTER SHOULD BE ZERO
4688                                                                  ;ONLY 11 OF THE 32 BITS CAN BE SEEN
4689                                                                  ;IN THE PATERN REGISTER
4690                                                                  ;DCK SHOULD BE SET IN RHER1
4691  034562  013746  050460          6$:     MOV     @#GECC1,-(SP)   ;GET PATTERN REGISTER
4692  034566  042716  174000                  BIC     #174000,(SP)    ;KEEP ONLY 11 BITS
4693  034572  022637  015064                  CMP     (SP)+,@#EC2     ;COMPARE PATTERN REGISTER
4694  034576  001401                          BEQ     7$              ;BRANCH IF GOOD
4695  034600  104032                          ERROR   32              ;11 BITS OF THE 32 BIT ECC REGISTER INCORRECT
4696
4697
4698
4699
4700                                          ;*ADD 16 MAINTENANCE CLOCKS TO
4701                                          ;*BRING EBL DOWN
4702
4703  034602  012700  000020          7$:     MOV     #16.,R0         ;COUNTER
4704  034606  052777  000002  160162  8$:     BIS     #MCLK,@RHMR     ;SET CLOCK
4705  034614  042777  000002  160154          BIC     #MCLK,@RHMR     ;CLEAR CLOCK
4706  034622  005300                          DEC     R0              ;COUNT
4707  034624  001370                          BNE     8$              ;BRANCH IF 16 CLOCKS NOT DONE
```

```
 4708  034640  012700  015220                MOV     #WRFROM,R0      ;GETTING READY TO FILL EXPECTED DATA
 4709  034644  012720  010000                MOV     #0!FMT22,(R0)+  ;CYLINDER 0
 4710  034650  112746  000000                MOVB    #0,-(SP)        ;IN LOWER BYTE GET SECTOR
 4711  034654  112766  000000  000001        MOVB    #0,1(SP)        ;GET TRACK IN HIGHER BYTE
 4712  034662  012620                        MOV     (SP)+,(R0)+     ;GET TRACK/SECTOR IN BUFFER
 4713  034664  012720  000000                MOV     #0,(R0)+        ;KEY1 IN BUFFER
 4714  034670  012720  000000                MOV     #0,(R0)+        ;KEY2 IN BUFFER
 4715  034674  012701  000400                MOV     #256.,R1        ;DATA WORD COUNTER
 4716  034700  012702  000000                MOV     #0,R2           ;DATA
 4717  034704  010220              3$:       MOV     R2,(R0)+        ;DATA INTO BUFFER
 4718  034706  005301                        DEC     R1              ;COUNT
 4719  034710  001375                        BNE     3$              ;BRANCH IF 256 NOT DONE
 4720
 4721
 4722  034712  005037  015124                CLR     @#ERFLG$        ;CLEAR ERROR FLAG
 4723  034716  004737  045464                JSR     PC,@#PUTREG     ;SAVE REGISTERS
 4724
 4725
 4726
 4727                                        ;*NOW READ DATA BUFFER WILL BE CHECKED
 4728
 4729  034722  004037  046660                JSR     R0,@#COMPAR     ;CHECK
 4730  034726  015220                        WRFROM                  ;GOOD BUFFER
 4731  034730  016264                        REINTO                  ;TEST BUFFER
 4732  034732  000404                        4+256.                  ;NUMBER OF WORDS CHECKED
 4733  034734  034742                        4$                      ;RETURN POINT FOR ERROR HEADER
 4734  034736  034746                        5$                      ;RETURN POINT FOR ERROR DATA
 4735  034742  104004              4$:       ERROR   4               ;READ NEXT ERROR
 4736  034744  000207                        RTS     PC              ;RETURN TO "COMPAR"
 4737  034746  104005              5$:       ERROR   5               ;WORD NOS 1 TO 4 ARE
 4738                                                                ;HEADER WORDS
 4739                                                                ;5 TO 260 ARE DATA WORDS
 4740  034750  000207                        RTS     PC              ;RETURN TO "COMPAR"
 4741
 4742
 4743
 4744
```

```
4745
4746  034754  012706  001000                  MOV     #STACK,SP       ;RESET STACK
4747
4748
4749
4750                                          ;*SETUP FOR WHAT IS TO BE READ
4751                                          ;*HEADER CRC IS RESTORED FROM A SUBROUTINE
4752
4753  034766  012746  000000                  MOV     #0,-(SP)        ;DATA TO BE READ
4754  034772  012705  000400                  MOV     #256.,R5        ;COUNTER
4755  034776  012700  054566                  MOV     #DISK,R0        ;START OF SIMULATED DISK DATA
4756  035002  011620                  1$:     MOV     (SP),(R0)+      ;MOVE IN DATA ON TO SIMULATED DISK
4757  035004  005305                          DEC     R5              ;COUNT
4758  035006  001375                          BNE     1$              ;BRANCH IF 256 NOT COMPLETE
4759  035010  005726                          TST     (SP)+           ;UNDO -(SP)
4760  035012  022020                          CMP     (R0)+,(R0)+     ;JUMP OVER THE TWO ECC WORDS
4761  035014  012705  000017                  MOV     #15.,   R5      ;1 DATA GAP
4762                                                                  ;14 TOLERANCE GAP
4763  035020  005020                  2$:     CLR     (R0)+           ;CLEAR DATA GAP, AND
4764  035022  005305                          DEC     R5              ;TOLERANCE GAP
4765  035024  001375                          BNE     2$              ;BRANCH IF NOT COMPLETE
4766
4767
4768  035026  004737  051254                  JSR     PC,@#FILLEC     ;INSERT ECC IN PROPER PLACE ON DISK
4769
4770
4771
4772                                          ;*THESE ARE FOR ECC TEST ONLY
4773
4774  035032  012737  177777  015142          MOV     #-1,@#TSECC     ;THIS IS AN ECC TEST
4775  035040  005037  050472                  CLR     @#POSITI        ;CLEAR ERROR POSITION COUNTER
4776  035044  013737  050466  050470          MOV     @#NCODE,@#NCOUNT  ;TEMPORARY N-CODE COUNTER
4777  035052  013737  050474  050502          MOV     @#HARDER,@#HADTMP  ;TEMPORARY HARD ERROR COUNTER
4778  035060  005037  050460                  CLR     @#GECC1         ;ECC LOW ORDER TO BE GENERATED
4779  035064  005037  050462                  CLR     @#GECC2         ;ECC HIGH ORDER TO BE GENERATED
4780  035070  005037  050476                  CLR     @#DATENV        ;CLEAR DATA ENVELOPE CLOCK COUNT
4781  035074  005037  050500                  CLR     @#ZCODE         ;CLEAR LEADING ZEROS CLOCK COUNT
4782
4783
4784                                          ;*THESE ARE TO SETUP FOR DISKLESS USE ONLY
4785
4786  035100  012737  010000  052650          MOV     #FMT22,@#CYL    ;16 BITS PER WORD
4787                                                                  ;CYLINDER 0, FORMAT 16 BITS
4788  035106  112737  000000  052653          MOVB    #0,@#SECOTR+1   ;TRACK 0
4789  035114  112737  000000  052652          MOVB    #0,@#SECOTR     ;SECTOR 0
4790  035122  012737  000000  052654          MOV     #0,@#KEY1       ;KEY1=0
4791  035130  012737  000000  052656          MOV     #0,@#KEY2       ;KEY2=0
4792  035136  012737  000400  052730          MOV     #256.,  @#DAWORD ;NO. OF DATA WORDS
4793  035144  005037  052660                  CLR     @#X             ;THIS IS A READ COMMAND
4794  035150  004537  047172                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
4795  035154  052650                          CYL
4796  035156  054550                          WCRC
4797
4798
4799                                          ;*THIS IS TO INSERT ERROR
4800                                          ;*THE DISK DATA IS IN LOCATION STARTING FROM 'DISK'
```

```
                                        ;*THE POSITION OF THE ERROR CAN BE CHANGED BY CHANGING
                                        ;*THIS MOVE
                                        ;
035160  012737  100000  054570         MOV     #100000,@#DISK+2 ;FORCE ERROR ON BIT NUMBER 32
                                                                 ;SO ERROR POSITION REGISTER WILL SHOW
                                                                 ;22
035166  012737  000026  035342         MOV     #22.,@#8$        ;INSERT POSITION REG.
                                        ;
                                        ;*THESE ARE REGULAR SETUPS
                                        ;
035174  004737  045764                 JSR     PC,@#CLDISK      ;SETUP GENERAL REGISTERS
035200  012777  177374  157542         MOV     #-256.-4.,@RHWC  ;256. DATA 4 HEADER WORDS
035206  012777  016264  157536         MOV     #REINTO,@RHBA    ;STARTING ADDRESS OF READ BUFFER
035214  112746  000000                 MOVB    #0,-(SP)         ;IN LOWER BYTE GET SECTOR
035220  112766  000000  000001         MOVB    #0,1(SP)         ;GET TRACK IN HIGHER BYTE
035226  012677  157530                 MOV     (SP)+,  @RHDST   ;TRACK/SECTOR IN RHDST
035232  012777  010000  157526         MOV     #FMT22,@RHOF     ;16 BITS PER WORD
                                                                 ;ECC CORRECTION NOT INHIBIT
                                                                 ;BECAUSE ECC IS NOT GOING
                                                                 ;TO BE CHECKED
035240  005077  157524                 CLR     @RHCA            ;CYLINDER 0
                                        ;
035256  013711  015176                 MOV     @#REFOR,@R1      ;READ HEADER AND DATA=72
035262  005037  015124                 CLR     @#ERFLG$         ;CLEAR ERROR FLAG
035266  004737  052510                 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
                                        ;
035272  005737  015124                 TST     @#ERFLG$         ;ANY ERRORS ALREADY THERE
035300  004737  045464                 JSR     PC,@#PUTREG      ;SAVE REGISTERS
035304  022737  100000  015034         CMP     #DCK,@#ER1       ;ONLY DATA CHECK ERROR SHOULD BE SET
035312  001401                         BEQ     6$               ;BRANCH IF YES
035314  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
035316  013746  050460         6$:     MOV     @#GECC1,-(SP)    ;GET PATTERN REGISTER
035322  042716  174000                 BIC     #174000,(SP)     ;KEEP ONLY 11 BITS
035326  022637  015064                 CMP     (SP)+,@#EC2      ;COMPARE PATTERN REGISTER
035332  001401                         BEQ     7$               ;BRANCH IF GOOD
035334  104032                         ERROR   32               ;11 BITS OF THE 32 BIT ECC REGISTER INCORRECT
```

```
4857  035336  004037  051102          7$:     JSR     R0,@#ECORR      ;GO TO ECC CORRECTION PROCESS
4858  035342  000026                  8$:     22.                     ;EXPECTED POSITION REG. WHEN CORRECTION
4859                                                                  ;IS COMPLETE
4860
4861
4862
4863  035356  012700  015220                  MOV     #WRFROM,R0      ;GETTING READY TO FILL EXPECTED DATA
4864  035362  012720  010000                  MOV     #0!FMT22,(R0)+  ;CYLINDER 0
4865  035366  112746  000000                  MOVB    #0,-(SP)        ;IN LOWER BYTE GET SECTOR
4866  035372  112766  000000  000001          MOVB    #0,1(SP)        ;GET TRACK IN HIGHER BYTE
4867  035400  012620                          MOV     (SP)+,  (R0)+   ;GET TRACK/SECTOR IN BUFFER
4868  035402  012720  000000                  MOV     #0,(R0)+        ;KEY1 IN BUFFER
4869  035406  012720  000000                  MOV     #0,(R0)+        ;KEY2 IN BUFFER
4870  035412  012701  000400                  MOV     #256.,  R1      ;DATA WORD COUNTER
4871  035416  012702  000000                  MOV     #0,R2           ;DATA
4872  035422  010220                  3$:     MOV     R2,(R0)+        ;DATA INTO BUFFER
4873  035424  005301                          DEC     R1              ;COUNT
4874  035426  001375                          BNE     3$              ;BRANCH IF 256 NOT DONE
4875
4876                                          ;*ONLY GOOD DATA HAS BEEN PUT IN 'WRFROM'
4877                                          ;*NOW THE INSERTED ERROR WILL BE PUT IN
4878  035430  012737  100000  015232          MOV     #100000,@#WRFROM+<5*2>  ;INSERTED ERROR
4879
4880
4881
4882  035436  005037  015124                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
4883  035442  004737  045464                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
4884
4885
4886                                          ;*NOW READ DATA BUFFER WILL BE CHECKED
4887
4888  035446  004037  046660                  JSR     R0,@#COMPAR     ;CHECK
4889  035452  015220                          WRFROM                  ;GOOD BUFFER
4890  035454  016264                          REINTO                  ;TEST BUFFER
4891  035456  000404                          4+256.                  ;NUMBER OF WORDS CHECKED
4892  035460  035466                          4$                      ;RETURN POINT FOR ERROR HEADER
4893  035462  035472                          5$                      ;RETURN POINT FOR ERROR DATA
4894  035466  104004                  4$:     ERROR   4               ;READ NEXT ERROR
4895  035470  000207                          RTS     PC              ;RETURN TO "COMPAR"
4896  035472  104005                  5$:     ERROR   5               ;WORD NOS 1 TO 4 ARE
4897                                                                  ;HEADER WORDS
4898                                                                  ;5 TO 260 ARE DATA WORDS
4899  035474  000207                          RTS     PC              ;RETURN TO "COMPAR"
4900
```

```
4901
4902  035500  012706  001000                  MOV     #STACK,SP       ;RESET STACK
4903
4904
4905
4906                                  ;       SETUP FOR WHAT IS TO BE READ
4907                                  ;       HEADER CRC IS RESTORED FROM A SUBROUTINE
4908
4909  035512  012746  000000                  MOV     #0,-(SP)        ;DATA TO BE READ
4910  035516  012705  000400                  MOV     #256.,R5        ;COUNTER
4911  035522  012700  054566                  MOV     #DISK,R0        ;START OF SIMULATED DISK DATA
4912  035526  011620                  1$:     MOV     (SP),(R0)+      ;MOVE IN DATA ON TO SIMULATED DISK
4913  035530  005305                          DEC     R5              ;COUNT
4914  035532  001375                          BNE     1$              ;BRANCH IF 256 NOT COMPLETE
4915  035534  005726                          TST     (SP)+           ;UNDO -(SP)
4916  035536  022020                          CMP     (R0)+,(R0)+     ;JUMP OVER THE TWO ECC WORDS
4917  035540  012705  000017                  MOV     #15.,R5         ;1 DATA GAP
4918                                                                  ;14 TOLERANCE GAP
4919  035544  005020                  2$:     CLR     (R0)+           ;CLEAR DATA GAP, AND
4920  035546  005305                          DEC     R5              ;TOLERANCE GAP
4921  035550  001375                          BNE     2$              ;BRANCH IF NOT COMPLETE
4922
4923
4924  035552  004737  051254                  JSR     PC,@#FILLEC     ;INSERT THE TWO ECC WORDS ON THE DISK
4925                                                                  ;IN THE CORRECT PLACE
4926
4927                                          ;*THESE ARE FOR ECC TEST ONLY
4928
4929  035556  012737  177777  015142          MOV     #-1,@#TSECC     ;THIS IS AN ECC TEST
4930  035564  005037  050472                  CLR     @#POSITI        ;CLEAR ERROR POSITION COUNTER
4931  035570  013737  050466  050470          MOV     @#NCODE,@#NCOUNT  ;TEMPORARY N-CODE COUNTER
4932  035576  013737  050474  050502          MOV     @#HARDER,@#HADTMP  ;TEMPORARY HARD ERROR COUNTER
4933  035604  005037  050460                  CLR     @#GECC1         ;ECC LOW ORDER TO BE GENERATED
4934  035610  005037  050462                  CLR     @#GECC2         ;ECC HIGH ORDER TO BE GENERATED
4935  035614  005037  050476                  CLR     @#DATENV        ;CLEAR DATA ENVELOPE CLOCK COUNT
4936  035620  005037  050500                  CLR     @#ZCODE         ;CLEAR LEADING ZEROS CLOCK COUNT
4937
4938
4939                                          ;*THESE ARE TO SETUP FOR DISKLESS USE ONLY
4940
4941  035624  012737  010000  052650          MOV     #FMT22,@#CYL    ;16 BITS PER WORD
4942                                                                  ;CYLINDER 0, FORMAT 16 BITS
4943  035632  112737  000000  052653          MOVB    #0,@#SECOTR+1   ;TRACK 0
4944  035640  112737  000000  052652          MOVB    #0,@#SECOTR     ;SECTOR 0
4945  035646  012737  000000  052654          MOV     #0,@#KEY1       ;KEY1=0
4946  035654  012737  000000  052656          MOV     #0,@#KEY2       ;KEY2=0
4947  035662  012737  000400  052730          MOV     #256.,@#DAWORD  ;NO. OF DATA WORDS
4948  035670  005037  052660                  CLR     @#X             ;THIS IS A READ COMMAND
4949  035674  004537  047172                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
4950  035700  052650                          CYL
4951  035702  054550                          WCRC
4952
4953
4954                                          ;*THIS IS TO INSERT ERROR
4955                                          ;*THE DISK DATA IS IN LOCATION STARTING FROM 'DISK'
4956                                          ;*THE POSITION OF THE ERROR CAN BE CHANGED BY CHANGING
```

```
4957                                    ;*THIS MOVE
4958  035704  012737  177760  054570    MOV     #177760,@#DISK+2  ;FORCE ERROR ON BIT NUMBER 21 THRU 32
4959                                                              ;SO ERROR POSITION REGISTER WILL SHOW
4960                                                              ;22
4961  035712  012737  010040  036066    MOV     #4128.,@#B$       ;INSERT POSITION REG.
4962
4963
4964                                    ;*THESE ARE REGULAR SETUPS
4965
4966  035720  004737  045764            JSR     PC,@#CLDISK       ;SETUP GENERAL REGISTERS
4967  035724  012777  177374  157016    MOV     #-256.-4.,@RHWC   ;256. DATA 4 HEADER WORDS
4968  035732  012777  016264  157012    MOV     #REINTO,@RHBA     ;STARTING ADDRESS OF READ BUFFER
4969  035740  112746  000000            MOVB    #0,-(SP)          ;IN LOWER BYTE GET SECTOR
4970  035744  112766  000000  000001    MOVB    #0,1(SP)          ;GET TRACK IN HIGHER BYTE
4971  035752  012677  157004            MOV     (SP)+,  @RHDST    ;TRACK/SECTOR IN RHDST
4972  035756  012777  010000  157002    MOV     #FMT22,@RHOF      ;16 BITS PER WORD
4973                                                              ;ECC CORRECTION NOT INHIBIT
4974                                                              ;BECAUSE ECC IS NOT GOING
4975                                                              ;TO BE CHECKED
4976  035764  005077  157000            CLR     @RHCA             ;CYLINDER 0
4977  036002  013711  015176            MOV     @#REFOR,@R1       ;READ HEADER AND DATA=72
4978  036006  005037  015124            CLR     @#ERFLG$          ;CLEAR ERROR FLAG
4979  036012  004737  052510            JSR     PC,     @#COMHD   ;READ HEADER AND DATA
4980                                                              ;IF THERE ARE READ ERRORS THEN
4981                                                              ;ECC WILL NOT BE CHECKED
4982
```

```
                                        ;*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

036016  005737  015124                TST     @#ERFLG$        ;ANY ERRORS ALREADY THERE
036024  004737  045464                JSR     PC,@#PUTREG     ;SAVE REGISTERS
036030  022737  100000  015034        CMP     #DCK,@#ER1      ;ONLY DATA CHECK ERROR SHOULD BE SET
036036  001401                        BEQ     6$              ;BRANCH IF YES
036040  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
036042  013746  050460        6$:     MOV     @#GECC1,-(SP)   ;GET PATTERN REGISTER
036046  042716  174000                BIC     #174000,(SP)    ;KEEP ONLY 11 BITS
036052  022637  015064                CMP     (SP)+,@#EC2     ;COMPARE PATTERN REGISTER
036056  001401                        BEQ     7$              ;BRANCH IF GOOD
036060  104032                        ERROR   32              ;11 BITS OF THE 32 BIT ECC REGISTER INCORRECT

036062  004037  051102        7$:     JSR     R0,@#ECORR      ;GO TO ECC CORRECTION PROCESS
036066  000000        8$:     .WORD                           ;EXPECTED POSITION REG. WHEN CORRECTION
                                                              ;IS COMPLETE


036070  004737  045464                JSR     PC,@#PUTREG     ;SAVE REGISTERS
036074  022737  100100  015034        CMP     #DCK!ECH,@#ER1  ;WITH ERRORS INSERTED IN BIT POSITION 21
                                                              ;THRU 32 HARD ERROR BIT SHOULD SET
036102  001401                        BEQ     9$              ;BRANCH IF GOOD
036104  104036                        ERROR   36              ;WITH ERROR INSERTED IN BIT POSITION 21 THRU
                                                              ;32 ECH SHOULD SET


036106                        9$:
036120  012700  015220                MOV     #WRFROM,R0      ;GETTING READY TO FILL EXPECTED DATA
036124  012720  010000                MOV     #0!FMT22,(R0)+  ;CYLINDER 0
036130  112746  000000                MOVB    #0,-(SP)        ;IN LOWER BYTE GET SECTOR
036134  112766  000000  000001        MOVB    #0,1(SP)        ;GET TRACK IN HIGHER BYTE
036142  012620                        MOV     (SP)+,(R0)+     ;GET TRACK/SECTOR IN BUFFER
036144  012720  000000                MOV     #0,(R0)+        ;KEY1 IN BUFFER
036150  012720  000000                MOV     #0,(R0)+        ;KEY2 IN BUFFER
036154  012701  000400                MOV     #256.,R1        ;DATA WORD COUNTER
036160  012702  000000                MOV     #0,R2           ;DATA
036164  010220                3$:     MOV     R2,(R0)+        ;DATA INTO BUFFER
036166  005301                        DEC     R1              ;COUNT
036170  001375                        BNE     3$              ;BRANCH IF 256 NOT DONE
```

```
                                                ;ONLY GOOD DATA HAS BEEN PUT IN 'WRFROM'
                                                ;*NOW THE INSERTED ERROR WILL BE PUT IN
                                                
036172  012737  177760  015232                 MOV     #177760,@#WRFROM+(5*2)  ;INSERTED ERROR
                                                
                                                
036200  005037  015124                         CLR     @#ERFLG$                ;CLEAR ERROR FLAG
036204  004737  045464                         JSR     PC,@#PUTREG     ;SAVE REGISTERS
                                                
                                                ;*NOW READ DATA BUFFER WILL BE CHECKED
                                                
036210  004037  046660                         JSR     R0,@#COMPAR     ;CHECK
036214  015220                                 WRFROM                  ;GOOD BUFFER
036216  016264                                 REINTO                  ;TEST BUFFER
036220  000404                                 4+256.                  ;NUMBER OF WORDS CHECKED
036222  036230                                 4$                      ;RETURN POINT FOR ERROR HEADER
036224  036234                                 5$                      ;RETURN POINT FOR ERROR DATA
036230  104004                         4$:     ERROR   4               ;READ NEXT ERROR
036232  000207                                 RTS     PC              ;RETURN TO "COMPAR"
036234  104005                         5$:     ERROR   5               ;WORD NOS 1 TO 4 ARE
                                                                       ;HEADER WORDS
                                                                       ;5 TO 260 ARE DATA WORDS
036236  000207                                 RTS     PC              ;RETURN TO "COMPAR"
```

```
5069
5070  036242  012706  001000                  MOV     #STACK,SP       ;RESET STACK
5071
5072  036254  012700  054470                  MOV     #SECGAP,R0      ;POINTER
5073  036260  012701  000460                  MOV     #304.,R1        ;COUNTER
5074  036264  005020                  1$:     CLR     (R0)+           ;CLEAR SIMULATED DISK AREA
5075  036266  005301                          DEC     R1
5076  036270  001375                          BNE     1$
5077  036272  004737  045764                  JSR     PC,CLDISK       ;THIS IS USED TO SET GENERAL REGISTERS
5078
5079                                          ;*THESE ARE FOR ECC TEST ONLY
5080
5081  036276  012737  177777  015142          MOV     #-1,@#TSECC     ;THIS IS AN ECC TEST
5082  036304  005037  050472                  CLR     @#POSITI        ;CLEAR ERROR POSITION COUNTER
5083  036310  013737  050466  050470          MOV     @#NCODE,@#NCOUNT  ;TEMPORARY N-CODE COUNTER
5084  036316  013737  050474  050502          MOV     @#HARDER,@#HADTMP  ;TEMPORARY HARD ERROR COUNTER
5085  036324  005037  050460                  CLR     @#GECC1         ;ECC LOW ORDER TO BE GENERATED
5086  036330  005037  050462                  CLR     @#GECC2         ;ECC HIGH ORDER TO BE GENERATED
5087  036334  005037  050476                  CLR     @#DATENV        ;CLEAR DATA ENVELOPE CLOCK COUNT
5088  036340  005037  050500                  CLR     @#ZCODE         ;CLEAR LEADING ZEROS CLOCK COUNT
5089
5090
5091
5092
5093                                          ;*THESE ARE TO BE SETUP FOR DISKLESS USE ONLY
5094
5095  036344  012737  010000  056006          MOV     #FMT22,@#WCYL   ;FORMAT22=16BIT WORDS AND
5096                                                                  ;CYLINDER 0
5097  036352  012737  000001  056010          MOV     #1,@#WSECTR     ;TRACK=0, SECTOR=1
5098  036360  005037  056012                  CLR     @#WKEY1         ;KEY1=0
5099  036364  005037  056014                  CLR     @#WKEY2         ;KEY2=0
5100  036370  012737  000400  056046          MOV     #256.,@#FNWORD  ;256 DATA WORDS
5101  036376  004537  047172                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
5102  036402  056006                          WCYL
5103  036404  056016                          GCRC
5104
5105                                          ;*THESE ARE REGULAR SETUPS
5106
5107  036406  012777  177374  156334          MOV     #-260.,@RHWC    ;256 DATA WORDS 4 HEADER WORDS
5108  036414  012700  015220                  MOV     #WRFROM,R0      ;THESE TWO INSTRUCTIONS GETS
5109  036420  010077  156326                  MOV     R0,@RHBA        ;ADDR. OF WRFROM INTO R0 AND
5110                                                                  ;BUS ADDRESS REGISTER
5111  036424  012720  010000                  MOV     #FMT22,(R0)+    ;FORMAT=16 BIT WORDS
5112                                                                  ;CYLINDER=0
5113  036430  012720  000001          2$:     MOV     #1,(R0)+        ;TRACK=0, SECTOR=1, KEYS=0
5114  036434  005020                          CLR     (R0)+           ;KEY1=0
5115  036436  005020                          CLR     (R0)+           ;KEY2=0
5116  036440  012705  000400                  MOV     #256.,R5        ;COUNTER
5117  036444  012720  177777          3$:     MOV     #-1,(R0)+       ;MOVE ALL ONES FOR DATA
5118  036450  005305                          DEC     R5
5119  036452  001374                          BNE     3$              ;BRANCH IF DATA NOT COMPLETE
5120  036454  012777  000001  156300          MOV     #1,@RHDST       ;TRACK=0 SECTOR=1
5121
5122
5123  036474  013711  015172                  MOV     @#WRIFOR,@R1    ;GET READY FOR WRITE HEADER AND
5124                                                                  ;DATA WITH 62 IN RHCS1
```

```
5125  036500  005037  015124                    CLR     @#ERFLG$          ;CLEAR ERROR FLAG
5126  036504  012777  010000  156254            MOV     #FMT22,@RHOF      ;FORMAT BIT=1 (16 BIT WORDS)
5127  036512  005077  156252                    CLR     @RHCA             ;CYLINDER =0
5128  036516  004737  055632                    JSR     PC,@#COMWHD       ;WRITE HEADER AND DATA
5129
5130                                            ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
5131                                            ;*FROM THE "COMWHD" ROUTINE THAT MEANS ALL HEADER ON DISK
5132                                            ;*IS GOOD IE. ONLY DATA IS TO BE CHECKED TO SEE IF THEY ARE
5133                                            ;*ALL ONES AND WRITE DATA GAP AND TOLERANCE GAP TO SEE IF
5134                                            ;*THEY ARE ALL ZEROS
5135
5136  036522  005737  015124                    TST     @#ERFLG$          ;HAS ANY ERRORS OCCURED?
5137                                                                      ;IF WRITE ERROR OCCURS ECC IS NOT CHECKED
5138
5139
5140                                            ;*COMPARE SOFTWARE GENERATED ECC WITH THAT GENERATED BY HARDWARE
5141
5142  036530  023737  050460  055566            CMP     @#GECC1,@#WECC1;COMPARE SOFTWARE ECC WITH HARDWARE ECC
5143  036536  001402                            BEQ     6$                ;BRANCH IF GOOD
5144  036540  104031                            ERROR   31                ;LOW ORDER ECC IN ERROR
5145  036542  000405                            BR      7$                ;BRANCH TO CONTINUE
5146  036544  023737  050462  055570  6$:       CMP     @#GECC2,@#WECC2;COMPARE SOFTWARE ECC WITH HARDWARE ECC
5147  036552  001401                            BEQ     7$                ;BRANCH IF GOOD
5148  036554  104031                            ERROR   31                ;HIGH ORDER ECC IN ERROR
5149
5150
5151  036556                          7$:
5152
5153
5154
5155
5156                                            ;*FILL "REINTO" BUFFER WITH EXPECTED DATA
5157
5158  036570  004037  045702                    JSR     R0,@#CLAREA       ;FILL REINTO BUFFER
5159  036574  016264                                    REINTO            ;FROM
5160  036576  017262                                    REINTO+<255.*2>   ;TO
5161  036600  177777                            .WORD   -1                ;DATA
5162
5163  036602  013737  050460  017264            MOV     @#GECC1,@#REINTO+<256.*2>;FILL ECC1
5164  036610  013737  050462  017266            MOV     @#GECC2,@#REINTO+<257.*2>;FILL ECC2
5165  036616  004037  045702                    JSR     R0,@#CLAREA       ;FILL REST
5166  036622  017270                            REINTO+<258.*2>           ;FROM
5167  036624  017324                            REINTO+<272.*2>           ;TO
5168  036626  000000                            0                         ;DATA
5169
5170
5171  036630  005037  015124                    CLR     @#ERFLG$          ;CLEAR ERROR FLAG
5172
5173
5174                                            ;*NOW COMPARE "DISK" BUFFER WITH "REINTO"
5175
5176  036634  004037  046660                    JSR     R0,@#COMPAR       ;CHECK
5177  036640  016264                            REINTO                    ;GOOD BUFFER
5178  036642  054566                            DISK                      ;TEST BUFFER
5179  036644  000402                            258.                      ;NUMBER OF WORDS CHECKED
5180  036646  036654                            4$                        ;RETURN POINT FOR ERROR HEADER
```

```
036650  036660                  5$                              ;RETURN POINT FOR ERROR DATA
036654  104007          4$:     ERROR   7                       ;READ ERROR 10 NEXT
036656  000207                  RTS     PC                      ;RETURN TO COMPARE
036660  104010          5$:     ERROR   10                      ;WORD NOS 1 TO 256 ARE
                                                                ;DATA WORDS
                                                                ;WORD NOS 257 AND 258
                                                                ;ARE ECC WHICH ARE CHECKED
                                                                ;WORD NOS 259
                                                                ;IS DATA GAP
                                                                ;WORD NOS 260 TO 273
                                                                ;ARE TOLERANCE GAP
036662  000207                  RTS     PC                      ;RETURN TO COMPARE
```

```
5198
5199  036666  012706  001000                  MOV     #STACK,SP       ;RESET STACK
5200
5201
5202
5203                                  ;       SETUP FOR WHAT IS TO BE READ
5204                                  ;       HEADER CRC IS RESTORED FROM A SUBROUTINE
5205
5206  036700  012746  177777                  MOV     #-1,-(SP)       ;DATA TO BE READ
5207  036704  012705  000400                  MOV     #256.,R5        ;COUNTER
5208  036710  012700  054566                  MOV     #DISK,R0        ;START OF SIMULATED DISK DATA
5209  036714  011620                  1$:     MOV     (SP),(R0)+      ;MOVE IN DATA ON TO SIMULATED DISK
5210  036716  005305                          DEC     R5              ;COUNT
5211  036720  001375                          BNE     1$              ;BRANCH IF 256 NOT COMPLETE
5212  036722  005726                          TST     (SP)+           ;UNDO -(SP)
5213  036724  022020                          CMP     (R0)+,(R0)+     ;JUMP OVER THE TWO ECC WORDS
5214  036726  012705  000017                  MOV     #15.,R5         ;1 DATA GAP
5215                                                                  ;14 TOLERANCE GAP
5216  036732  005020                  2$:     CLR     (R0)+           ;CLEAR DATA GAP, AND
5217  036734  005305                          DEC     R5              ;TOLERANCE GAP
5218  036736  001375                          BNE     2$              ;BRANCH IF NOT COMPLETE
5219
5220
5221  036740  004737  051254                  JSR     PC,@#FILLEC     ;INSERT THE TWO ECC WORDS ON THE DISK
5222                                                                  ;IN THE CORRECT PLACE
5223
5224                                          ;*THESE ARE FOR ECC TEST ONLY
5225
5226  036744  012737  177777  015142          MOV     #-1,@#TSECC     ;THIS IS AN ECC TEST
5227  036752  005037  050472                  CLR     @#POSITI        ;CLEAR ERROR POSITION COUNTER
5228  036756  013737  050466  050470          MOV     @#NCODE,@#NCOUNT  ;TEMPORARY N-CODE COUNTER
5229  036764  013737  050474  050502          MOV     @#HARDER,@#HADTMP  ;TEMPORARY HARD ERROR COUNTER
5230  036772  005037  050460                  CLR     @#GECC1         ;ECC LOW ORDER TO BE GENERATED
5231  036776  005037  050462                  CLR     @#GECC2         ;ECC HIGH ORDER TO BE GENERATED
5232  037002  005037  050476                  CLR     @#DATENV        ;CLEAR DATA ENVELOPE CLOCK COUNT
5233  037006  005037  050500                  CLR     @#ZCODE         ;CLEAR LEADING ZEROS CLOCK COUNT
5234
5235
5236                                          ;*THESE ARE TO SETUP FOR DISKLESS USE ONLY
5237
5238  037012  012737  010000  052650          MOV     #FMT22,@#CYL    ;16 BITS PER WORD
5239                                                                  ;CYLINDER 0, FORMAT 16 BITS
5240  037020  112737  000000  052653          MOVB    #0,@#SECOTR+1   ;TRACK 0
5241  037026  112737  000000  052652          MOVB    #0,@#SECOTR     ;SECTOR 0
5242  037034  012737  000000  052654          MOV     #0,@#KEY1       ;KEY1=0
5243  037042  012737  000000  052656          MOV     #0,@#KEY2       ;KEY2=0
5244  037050  012737  000400  052730          MOV     #256.,@#DAWORD  ;NO. OF DATA WORDS
5245  037056  005037  052660                  CLR     @#X             ;THIS IS A READ COMMAND
5246  037062  004537  047172                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
5247  037066  052650                          CYL
5248  037070  054550                          WCRC
5249
5250
5251
5252
5253                                          ;*THESE ARE REGULAR SETUPS
```

```
5254
5255  037072  004737  045764                  JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
5256  037076  012777  177374  155644          MOV     #-256.-4,@RHWC  ;256. DATA 4 HEADER WORDS
5257  037104  012777  016264  155640          MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
5258  037112  112746  000000                  MOVB    #0,-(SP)        ;IN LOWER BYTE GET SECTOR
5259  037116  112766  000000  000001          MOVB    #0,1(SP)        ;GET TRACK IN HIGHER BYTE
5260  037124  012677  155632                  MOV     (SP)+,@RHDST    ;TRACK/SECTOR IN RHDST
5261  037130  012777  010000  155630          MOV     #FMT22,@RHOF    ;16 BITS PER WORD
5262                                                                  ;ECC CORRECTION NOT INHIBIT
5263                                                                  ;BECAUSE ECC IS NOT GOING
5264                                                                  ;TO BE CHECKED
5265  037136  005077  155626                  CLR     @RHCA           ;CYLINDER 0
5266
5267
5268  037154  013711  015176                  MOV     @#REFOR,@R1     ;READ HEADER AND DATA=72
5269  037160  005037  015124                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
5270  037164  004737  052510                  JSR     PC,@#COMHD      ;READ HEADER AND DATA
5271                                                                  ;IF THERE ARE READ ERRORS THEN
5272                                                                  ;ECC WILL NOT BE CHECKED
5273
5274
5275                                          ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
5276                                          ;*FROM THE "COMHD" ROUTINE THAT MEANS SECTOR GAP,
5277                                          ;*FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
5278                                          ;*SYNC BYTE HAVE GONE BY AND SYNCS WERE CORRECTLY
5279                                          ;*DETECTED
5280                                          ;*HEADER AND DATA ARE TO BE CHECKED.
5281                                          ;*IN CHECKING READ DATA THE WRITE FROM BUFFER
5282                                          ;*"WRFROM" IS FILLED WITH EXPECTED DATA AND
5283                                          ;*COMPARISONS ARE MADE
5284
5285  037170  005737  015124                  TST     @#ERFLG$        ;ANY ERRORS ALREADY THERE
5286  037176  004737  045464                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
5287  037202  005737  015034                  TST     @#ER1           ;NO ERRORS SHOULD BE SET
5288  037206  001401                          BEQ     6$              ;BRANCH IF NO ERRORS SET
5289  037210  104032                          ERROR   32              ;32 BIT ECC REGISTER SHOULD BE ZERO
5290                                                                  ;ONLY 11 OF THE 32 BITS CAN BE SEEN
5291                                                                  ;IN THE PATERN REGISTER
5292                                                                  ;DCK SHOULD BE SET IN RHER1
5293  037212  013746  050460          6$:     MOV     @#GECC1,-(SP)   ;GET PATTERN REGISTER
5294  037216  042716  174000                  BIC     #174000,(SP)    ;KEEP ONLY 11 BITS
5295  037222  022637  015064                  CMP     (SP)+,@#EC2     ;COMPARE PATTERN REGISTER
5296  037226  001401                          BEQ     7$              ;BRANCH IF GOOD
5297  037230  104032                          ERROR   32              ;11 BITS OF THE 32 BIT ECC REGISTER INCORRECT
5298
5299
5300
5301
5302                                          ;*ADD 16 MAINTENANCE CLOCKS TO
5303                                          ;*BRING EBL DOWN
5304
5305  037232  012700  000020          7$:     MOV     #16.,R0         ;COUNTER
5306  037236  052777  000002  155532  8$:     BIS     #MCLK,@RHMR     ;SET CLOCK
5307  037244  042777  000002  155524          BIC     #MCLK,@RHMR     ;CLEAR CLOCK
5308  037252  005300                          DEC     R0              ;COUNT
5309  037254  001370                          BNE     8$              ;BRANCH IF 16 CLOCKS NOT DONE
```

```
5310  037270  012700  015220                MOV     #WRFROM,R0      ;GETTING READY TO FILL EXPECTED DATA
5311  037274  012720  010000                MOV     #0!FMT22,(R0)+  ;CYLINDER 0
5312  037300  112746  000000                MOVB    #0,     -(SP)   ;IN LOWER BYTE GET SECTOR
5313  037304  112766  000000  000001        MOVB    #0,     1(SP)   ;GET TRACK IN HIGHER BYTE
5314  037312  012620                        MOV     (SP)+,  (R0)+   ;GET TRACK/SECTOR IN BUFFER
5315  037314  012720  000000                MOV     #0,     (R0)+   ;KEY1 IN BUFFER
5316  037320  012720  000000                MOV     #0,     (R0)+   ;KEY2 IN BUFFER
5317  037324  012701  000400                MOV     #256.,  R1      ;DATA WORD COUNTER
5318  037330  012702  177777                MOV     #-1,    R2      ;DATA
5319
5320  037334  010220              3$:       MOV     R2,     (R0)+   ;DATA INTO BUFFER
5321  037336  005301                        DEC     R1              ;COUNT
5322  037340  001375                        BNE     3$              ;BRANCH IF 256 NOT DONE
5323  037342  005037  015124                CLR     @#ERFLG$        ;CLEAR ERROR FLAG
5324  037346  004737  045464                JSR     PC,@#PUTREG     ;SAVE REGISTERS
5325
5326                                        ;NOW READ DATA BUFFER WILL BE CHECKED
5327
5328  037352  004037  046660                JSR     R0,@#COMPAR     ;CHECK
5329  037356  015220                        WRFROM                  ;GOOD BUFFER
5330  037360  016264                        REINTO                  ;TEST BUFFER
5331  037362  000404                        4+256.                  ;NUMBER OF WORDS CHECKED
5332  037364  037372                        4$                      ;RETURN POINT FOR ERROR HEADER
5333  037366  037376                        5$                      ;RETURN POINT FOR ERROR DATA
5334  037372  104004              4$:       ERROR   4               ;READ NEXT ERROR
5335  037374  000207                        RTS     PC              ;RETURN TO "COMPAR"
5336  037376  104005              5$:       ERROR   5               ;WORD NOS 1 TO 4 ARE
5337                                                                ;HEADER WORDS
5338                                                                ;5 TO 260 ARE DATA WORDS
5339  037400  000207                        RTS     PC              ;RETURN TO "COMPAR"
5340
5341
5342
5343
```

```
5344
5345  037404  012706  001000                  MOV     #STACK,SP       ;RESET STACK
5346
5347
5348
5349                                          ;*SETUP FOR WHAT IS TO BE READ
5350                                          ;*HEADER CRC IS RESTORED FROM A SUBROUTINE
5351
5352  037416  012746  177777                  MOV     #-1,-(SP)       ;DATA TO BE READ
5353  037422  012705  000400                  MOV     #256.,R5        ;COUNTER
5354  037426  012700  054566                  MOV     #DISK,R0        ;START OF SIMULATED DISK DATA
5355  037432  011620                  1$:     MOV     (SP),(R0)+      ;MOVE IN DATA ON TO SIMULATED DISK
5356  037434  005305                          DEC     R5              ;COUNT
5357  037436  001375                          BNE     1$              ;BRANCH IF 256 NOT COMPLETE
5358  037440  005726                          TST     (SP)+           ;UNDO -(SP)
5359  037442  022020                          CMP     (R0)+,(R0)+     ;JUMP OVER THE TWO ECC WORDS
5360  037444  012705  000017                  MOV     #15.,   R5      ;1 DATA GAP
5361                                                                  ;14 TOLERANCE GAP
5362  037450  005020                  2$:     CLR     (R0)+           ;CLEAR DATA GAP, AND
5363  037452  005305                          DEC     R5              ;TOLERANCE GAP
5364  037454  001375                          BNE     2$              ;BRANCH IF NOT COMPLETE
5365
5366
5367  037456  004737  051254                  JSR     PC,@#FILLEC     ;INSERT ECC IN PROPER PLACE ON DISK
5368
5369
5370
5371                                          ;*THESE ARE FOR ECC TEST ONLY
5372
5373  037462  012737  177777  015142          MOV     #-1,@#TSECC     ;THIS IS AN ECC TEST
5374  037470  005037  050472                  CLR     @#POSITI        ;CLEAR ERROR POSITION COUNTER
5375  037474  013737  050466  050470          MOV     @#NCODE,@#NCOUNT  ;TEMPORARY N-CODE COUNTER
5376  037502  013737  050474  050502          MOV     @#HARDER,@#HADTMP  ;TEMPORARY HARD ERROR COUNTER
5377  037510  005037  050460                  CLR     @#GECC1         ;ECC LOW ORDER TO BE GENERATED
5378  037514  005037  050462                  CLR     @#GECC2         ;ECC HIGH ORDER TO BE GENERATED
5379  037520  005037  050476                  CLR     @#DATENV        ;CLEAR DATA ENVELOPE CLOCK COUNT
5380  037524  005037  050500                  CLR     @#ZCODE         ;CLEAR LEADING ZEROS CLOCK COUNT
5381
5382
5383                                          ;*THESE ARE TO SETUP FOR DISKLESS USE ONLY
5384
5385  037530  012737  010000  052650          MOV     #FMT22,@#CYL    ;16 BITS PER WORD
5386                                                                  ;CYLINDER 0, FORMAT 16 BITS
5387  037536  112737  000000  052653          MOVB    #0,@#SECOTR+1   ;TRACK 0
5388  037544  112737  000000  052652          MOVB    #0,@#SECOTR     ;SECTOR 0
5389  037552  012737  000000  052654          MOV     #0,@#KEY1       ;KEY1=0
5390  037560  012737  000000  052656          MOV     #0,@#KEY2       ;KEY2=0
5391  037566  012737  000400  052730          MOV     #256.,@#DAWORD  ;NO. OF DATA WORDS
5392  037574  005037  052660                  CLR     @#X             ;THIS IS A READ COMMAND
5393  037600  004537  047172                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
5394  037604  052650                          CYL
5395  037606  054550                          WCRC
5396
5397
5398                                          ;*THIS IS TO INSERT ERROR
5399                                          ;*THE DISK DATA IS IN LOCATION STARTING FROM 'DISK'
```

```
5400                                        ;*THE POSITION OF THE ERROR CAN BE CHANGED BY CHANGING
5401                                        ;*THIS MOVE
5402
5403  037610  012737  077777  054570        MOV     #77777,@#DISK+2 ;FORCE ERROR ON BIT NUMBER 32
5404                                                                ;SO ERROR POSITION REGISTER WILL SHOW
5405                                                                ;22
5406  037616  012737  000026  037772        MOV     #22.,@#8$       ;INSERT POSITION REG.
5407
5408
5409                                        ;*THESE ARE REGULAR SETUPS
5410
5411  037624  004737  045764                JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
5412  037630  012777  177374  155112        MOV     #-256.,-4.,@RHWC ;256. DATA 4 HEADER WORDS
5413  037636  012777  016264  155106        MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
5414  037644  112746  000000                MOVB    #0,-(SP)        ;IN LOWER BYTE GET SECTOR
5415  037650  112766  000000  000001        MOVB    #0,1(SP)        ;GET TRACK IN HIGHER BYTE
5416  037656  012677  155100                MOV     (SP)+,@RHDST    ;TRACK/SECTOR IN RHDST
5417  037662  012777  010000  155076        MOV     #FMT22,@RHOF    ;16 BITS PER WORD
5418                                                                ;ECC CORRECTION NOT INHIBIT
5419                                                                ;BECAUSE ECC IS NOT GOING
5420                                                                ;TO BE CHECKED
5421  037670  005077  155074                CLR     @RHCA           ;CYLINDER 0
5422  037706  013711  015176                MOV     @#REFOR,@R1     ;READ HEADER AND DATA=72
5423  037712  005037  015124                CLR     @#ERFLG$        ;CLEAR ERROR FLAG
5424  037716  004737  052510                JSR     PC,@#COMHD      ;READ HEADER AND DATA
5425                                                                ;IF THERE ARE READ ERRORS THEN
5426                                                                ;ECC WILL NOT BE CHECKED
5427
5428
5429                                        ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
5430                                        ;*FROM THE "COMHD" ROUTINE THAT MEANS SECTOR GAP,
5431                                        ;*FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
5432                                        ;*SYNC BYTE HAVE GONE BY AND SYNCS WERE CORRECTLY
5433                                        ;*DETECTED
5434                                        ;*HEADER AND DATA ARE TO BE CHECKED.
5435                                        ;*IN CHECKING READ DATA THE WRITE FROM BUFFER
5436                                        ;*"WRFROM" IS FILLED WITH EXPECTED DATA AND
5437                                        ;*COMPARISONS ARE MADE
5438
5439  037722  005737  015124                TST     @#ERFLG$        ;ANY ERRORS ALREADY THERE
5440  037730  004737  045464                JSR     PC,@#PUTREG     ;SAVE REGISTERS
5441  037734  022737  100000  015034        CMP     #DCK,@#ER1      ;ONLY DATA CHECK ERROR SHOULD BE SET
5442  037742  001401                        BEQ     6$              ;BRANCH IF YES
5443  037744  104032                        ERROR   32              ;32 BIT ECC REGISTER SHOULD BE NON
5444                                                                ;ZERO
5445                                                                ;ONLY 11 OF THE 32 BITS CAN BE SEEN
5446                                                                ;IN THE PATERN REGISTER
5447                                                                ;DCK SHOULD BE SET IN RHER1
5448  037746  013746  050460        6$:     MOV     @#GECC1,-(SP)   ;GET PATTERN REGISTER
5449  037752  042716  174000                BIC     #174000,(SP)    ;KEEP ONLY 11 BITS
5450  037756  022637  015064                CMP     (SP)+,@#EC2     ;COMPARE PATTERN REGISTER
5451  037762  001401                        BEQ     7$              ;BRANCH IF GOOD
5452  037764  104032                        ERROR   32              ;11 BITS OF THE 32 BIT ECC REGISTER INCORRECT
5453
5454  037766  004037  051102        7$:     JSR     R0,@#ECORR      ;GO TO ECC CORRECTION PROCESS
5455  037772  000026                8$:     22.                     ;EXPECTED POSITION REG. WHEN CORRECTION
```

```
5456                                                    ;IS COMPLETE
5457
5458
5459
5460  040006  012700  015220                  MOV     #WRFROM,R0      ;GETTING READY TO FILL EXPECTED DATA
5461  040012  012720  010000                  MOV     #0!FMT22,(R0)+  ;CYLINDER 0
5462  040016  112746  000000                  MOVB    #0,-(SP)        ;IN LOWER BYTE GET SECTOR
5463  040022  112766  000000  000001          MOVB    #0,1(SP)        ;GET TRACK IN HIGHER BYTE
5464  040030  012620                          MOV     (SP)+,(R0)+     ;GET TRACK/SECTOR IN BUFFER
5465  040032  012720  000000                  MOV     #0,(R0)+        ;KEY1 IN BUFFER
5466  040036  012720  000000                  MOV     #0,(R0)+        ;KEY2 IN BUFFER
5467  040042  012701  000400                  MOV     #256.,R1        ;DATA WORD COUNTER
5468  040046  012702  177777                  MOV     #-1,R2          ;DATA
5469  040052  010220                  3$:     MOV     R2,(R0)+        ;DATA INTO BUFFER
5470  040054  005301                          DEC     R1              ;COUNT
5471  040056  001375                          BNE     3$              ;BRANCH IF 256 NOT DONE
5472
5473                                          ;*ONLY GOOD DATA HAS BEEN PUT IN 'WRFROM'
5474                                          ;*NOW THE INSERTED ERROR WILL BE PUT IN
5475
5476  040060  012737  077777  015232          MOV     #77777,@#WRFROM+<5*2>   ;INSERTED ERROR
5477  040066  004737  045464                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
5478  040072  005037  015124                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
5479
5480
5481                                          ;*NOW READ DATA BUFFER WILL BE CHECKED
5482
5483  040076  004037  046660                  JSR     R0,@#COMPAR     ;CHECK
5484  040102  015220                          WRFROM                  ;GOOD BUFFER
5485  040104  016264                          REINTO                  ;TEST BUFFER
5486  040106  000404                          4+256.                  ;NUMBER OF WORDS CHECKED
5487  040110  040116                          4$                      ;RETURN POINT FOR ERROR HEADER
5488  040112  040122                          5$                      ;RETURN POINT FOR ERROR DATA
5489  040116  104004                  4$:     ERROR   4               ;READ NEXT ERROR
5490  040120  000207                          RTS     PC              ;RETURN TO "COMPAR"
5491  040122  104005                  5$:     ERROR   5               ;WORD NOS 1 TO 4 ARE
5492                                                                  ;HEADER WORDS
5493                                                                  ;5 TO 260 ARE DATA WORDS
5494  040124  000207                          RTS     PC              ;RETURN TO "COMPAR"
5495
```

```
5496
5497  040130  012706  001000                  MOV     #STACK,SP       ;RESET STACK
5498
5499
5500
5501                                  ;       SETUP FOR WHAT IS TO BE READ
5502                                  ;       HEADER CRC IS RESTORED FROM A SUBROUTINE
5503
5504  040142  012746  177777                  MOV     #-1,-(SP)       ;DATA TO BE READ
5505  040146  012705  000400                  MOV     #256.,R5        ;COUNTER
5506  040152  012700  054566                  MOV     #DISK,R0        ;START OF SIMULATED DISK DATA
5507  040156  011620                  1$:     MOV     (SP),(R0)+      ;MOVE IN DATA ON TO SIMULATED DISK
5508  040160  005305                          DEC     R5              ;COUNT
5509  040162  001375                          BNE     1$              ;BRANCH IF 256 NOT COMPLETE
5510  040164  005726                          TST     (SP)+           ;UNDO -(SP)
5511  040166  022020                          CMP     (R0)+,(R0)+     ;JUMP OVER THE TWO ECC WORDS
5512  040170  012705  000017                  MOV     #15.,R5         ;1 DATA GAP
5513                                                                  ;14 TOLERANCE GAP
5514  040174  005020                  2$:     CLR     (R0)+           ;CLEAR DATA GAP, AND
5515  040176  005305                          DEC     R5              ;TOLERANCE GAP
5516  040200  001375                          BNE     2$              ;BRANCH IF NOT COMPLETE
5517
5518
5519  040202  004737  051254                  JSR     PC,@#FILLEC     ;INSERT THE TWO ECC WORDS ON THE DISK
5520                                                                  ;IN THE CORRECT PLACE
5521
5522                                          ;*THESE ARE FOR ECC TEST ONLY
5523
5524  040206  012737  177777  015142          MOV     #-1,@#TSECC     ;THIS IS AN ECC TEST
5525  040214  005037  050472                  CLR     @#POSITI        ;CLEAR ERROR POSITION COUNTER
5526  040220  013737  050466  050470          MOV     @#NCODE,@#NCOUNT  ;TEMPORARY N-CODE COUNTER
5527  040226  013737  050474  050502          MOV     @#HARDER,@#HADTMP  ;TEMPORARY HARD ERROR COUNTER
5528  040234  005037  050460                  CLR     @#GECC1         ;ECC LOW ORDER TO BE GENERATED
5529  040240  005037  050462                  CLR     @#GECC2         ;ECC HIGH ORDER TO BE GENERATED
5530  040244  005037  050476                  CLR     @#DATENV        ;CLEAR DATA ENVELOPE CLOCK COUNT
5531  040250  005037  050500                  CLR     @#ZCODE         ;CLEAR LEADING ZEROS CLOCK COUNT
5532
5533
5534                                          ;*THESE ARE TO SETUP FOR DISKLESS USE ONLY
5535
5536  040254  012737  010000  052650          MOV     #FMT22,@#CYL    ;16 BITS PER WORD
5537                                                                  ;CYLINDER 0, FORMAT 16 BITS
5538  040262  112737  000000  052653          MOVB    #0,@#SECOTR+1   ;TRACK 0
5539  040270  112737  000000  052652          MOVB    #0,@#SECOTR     ;SECTOR 0
5540  040276  012737  000000  052654          MOV     #0,@#KEY1       ;KEY1=0
5541  040304  012737  000000  052656          MOV     #0,@#KEY2       ;KEY2=0
5542  040312  012737  000400  052730          MOV     #256.,@#DAWORD  ;NO. OF DATA WORDS
5543  040320  005037  052660                  CLR     @#X             ;THIS IS A READ COMMAND
5544  040324  004537  047172                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
5545  040330  052650                          CYL
5546  040332  054550                          WCRC
5547
5548
5549                                          ;*THIS IS TO INSERT ERROR
5550                                          ;*THE DISK DATA IS IN LOCATION STARTING FROM 'DISK'
5551                                          ;*THE POSITION OF THE ERROR CAN BE CHANGED BY CHANGING
```

```
5552                                                    ;*THIS MOVE
5553
5554   040334  012737  077757  054570          MOV     #77757,@#DISK+2 ;FORCE ERROR ON BIT NUMBER 32 AND 21
5555                                                    ;SO ERROR POSITION REGISTER WILL SHOW
5556                                                    ;22
5557   040342  012737  010040  040516          MOV     #4128.,@#8$     ;INSERT POSITION REG.
5558
5559
5560                                                    ;*THESE ARE REGULAR SETUPS
5561
5562   040350  004737  045764                  JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
5563   040354  012777  177374  154366          MOV     #-256.-4.,@RHWC ;256. DATA 4 HEADER WORDS
5564   040362  012777  016264  154362          MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
5565   040370  112746  000000                  MOVB    #0,-(SP)        ;IN LOWER BYTE GET SECTOR
5566   040374  112766  000000  000001          MOVB    #0,1(SP)        ;GET TRACK IN HIGHER BYTE
5567   040402  012677  154354                  MOV     (SP)+,@RHDST    ;TRACK/SECTOR IN RHDST
5568   040406  012777  010000  154352          MOV     #FMT22,@RHOF    ;16 BITS PER WORD
5569                                                    ;ECC CORRECTION NOT INHIBIT
5570                                                    ;BECAUSE ECC IS NOT GOING
5571                                                    ;TO BE CHECKED
5572   040414  005077  154350                  CLR     @RHCA           ;CYLINDER 0
5573
5574
5575   040432  013711  015176                  MOV     @#REFOR,@R1     ;READ HEADER AND DATA=72
5576   040436  005037  015124                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
5577   040442  004737  052510                  JSR     PC,@#COMHD      ;READ HEADER AND DATA
5578                                                    ;IF THERE ARE READ ERRORS THEN
5579                                                    ;ECC WILL NOT BE CHECKED
5580
5581
5582                                           ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
5583                                           ;*FROM THE "COMHD" ROUTINE THAT MEANS SECTOR GAP,
5584                                           ;*FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
5585                                           ;*SYNC BYTE HAVE GONE BY AND SYNCS WERE CORRECTLY
5586                                           ;*DETECTED
5587                                           ;*HEADER AND DATA ARE TO BE CHECKED.
5588                                           ;*IN CHECKING READ DATA THE WRITE FROM BUFFER
5589                                           ;*"WRFROM" IS FILLED WITH EXPECTED DATA AND
5590                                           ;*COMPARISONS ARE MADE
5591
5592   040446  005737  015124                  TST     @#ERFLG$        ;ANY ERRORS ALREADY THERE
5593   040454  004737  045464                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
5594   040460  022737  100000  015034          CMP     #DCK,@#ER1      ;ONLY DATA CHECK ERROR SHOULD BE SET
5595   040466  001401                          BEQ     6$              ;BRANCH IF YES
5596   040470  104032                          ERROR   32              ;32 BIT ECC REGISTER SHOULD BE NON
5597                                                    ;ZERO
5598                                                    ;ONLY 11 OF THE 32 BITS CAN BE SEEN
5599                                                    ;IN THE PATERN REGISTER
5600                                                    ;DCK SHOULD BE SET IN RHER1
5601   040472  013746  050460          6$:     MOV     @#GECC1,-(SP)   ;GET PATTERN REGISTER
5602   040476  042716  174000                  BIC     #174000,(SP)    ;KEEP ONLY 11 BITS
5603   040502  022637  015064                  CMP     (SP)+,@#EC2     ;COMPARE PATTERN REGISTER
5604   040506  001401                          BEQ     7$              ;BRANCH IF GOOD
5605   040510  104032                          ERROR   32              ;11 BITS OF THE 32 BIT ECC REGISTER INCORRECT
5606
5607   040512  004037  051102          7$:     JSR     R0,@#ECORR      ;GO TO ECC CORRECTION PROCESS
```

```
5608  040516  000000                  8$:     .WORD                   ;EXPECTED POSITION REG. WHEN CORRECTION
5609                                                                    ;IS COMPLETE
5610
5611  040520  004737  045464                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
5612
5613
5614  040524  022737  100100  015034          CMP     #DCK!ECH,@#ER1  ;WITH ERRORS INSERTED IN BIT POSITION 21
5615                                                                    ;AND 32 HARD ERROR BIT SHOULD SET
5616  040532  001401                          BEQ     9$              ;BRANCH IF GOOD
5617  040534  104036                          ERROR   36              ;WITH ERROR INSERTED IN BIT POSITION 21 THRU
5618                                                                    ;32 HCE SHOULD SET
5619
5620
5621
5622
5623  040536                          9$:
5624  040550  012700  015220                  MOV     #WRFROM,R0      ;GETTING READY TO FILL EXPECTED DATA
5625  040554  012720  010000                  MOV     #0!FMT22,(R0)+  ;CYLINDER 0
5626  040560  112746  000000                  MOVB    #0,-(SP)        ;IN LOWER BYTE GET SECTOR
5627  040564  112766  000000  000001          MOVB    #0,1(SP)        ;GET TRACK IN HIGHER BYTE
5628  040572  012620                          MOV     (SP)+,(R0)+     ;GET TRACK/SECTOR IN BUFFER
5629  040574  012720  000000                  MOV     #0,(R0)+        ;KEY1 IN BUFFER
5630  040600  012720  000000                  MOV     #0,(R0)+        ;KEY2 IN BUFFER
5631  040604  012701  000400                  MOV     #256.,R1        ;DATA WORD COUNTER
5632  040610  012702  177777                  MOV     #-1,R2          ;DATA
5633  040614  010220                  3$:     MOV     R2,(R0)+        ;DATA INTO BUFFER
5634  040616  005301                          DEC     R1              ;COUNT
5635  040620  001375                          BNE     3$              ;BRANCH IF 256 NOT DONE
5636
5637                                          ;*ONLY GOOD DATA HAS BEEN PUT IN 'WRFROM'
5638                                          ;*NOW THE INSERTED ERROR WILL BE PUT IN
5639
5640  040622  012737  077757  015232          MOV     #77757,@#WRFROM+<5*2>   ;INSERTED ERROR
5641  040630  004737  045464                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
5642  040634  005037  015124                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
5643
5644
5645                                          ;*NOW READ DATA BUFFER WILL BE CHECKED
5646
5647  040640  004037  046660                  JSR     R0,@#COMPAR     ;CHECK
5648  040644  015220                          WRFROM                  ;GOOD BUFFER
5649  040646  016264                          REINTO                  ;TEST BUFFER
5650  040650  000404                          4+256.                  ;NUMBER OF WORDS CHECKED
5651  040652  040660                          4$                      ;RETURN POINT FOR ERROR HEADER
5652  040654  040664                          5$                      ;RETURN POINT FOR ERROR DATA
5653  040660  104004                  4$:     ERROR   4               ;READ NEXT ERROR
5654  040662  000207                          RTS     PC              ;RETURN TO "COMPAR"
5655  040664  104005                  5$:     ERROR   5               ;WORD NOS 1 TO 4 ARE
5656                                                                    ;HEADER WORDS
5657                                                                    ;5 TO 260 ARE DATA WORDS
5658  040666  000207                          RTS     PC              ;RETURN TO "COMPAR"
```

```
040672  012706  001000                        MOV     #STACK,SP          ;RESET STACK

040704  012700  054470                        MOV     #SECGAP,R0         ;POINTER
040710  012701  000460                        MOV     #304.,R1           ;COUNTER
040714  005020                        1$:     CLR     (R0)+              ;CLEAR SIMULATED DISK AREA
040716  005301                                DEC     R1
040720  001375                                BNE     1$
040722  004737  045764                        JSR     PC,CLDISK          ;THIS IS USED TO SET GENERAL REGISTERS

                                      ;*THESE ARE FOR ECC TEST ONLY

040726  012737  177777  015142                MOV     #-1,@#TSECC        ;THIS IS AN ECC TEST
040734  005037  050472                        CLR     @#POSITI           ;CLEAR ERROR POSITION COUNTER
040740  013737  050466  050470                MOV     @#NCODE,@#NCOUNT   ;TEMPORARY N-CODE COUNTER
040746  013737  050474  050502                MOV     @#HARDER,@#HADTMP  ;TEMPORARY HARD ERROR COUNTER
040754  005037  050460                        CLR     @#GECC1            ;ECC LOW ORDER TO BE GENERATED
040760  005037  050462                        CLR     @#GECC2            ;ECC HIGH ORDER TO BE GENERATED
040764  005037  050476                        CLR     @#DATENV           ;CLEAR DATA ENVELOPE CLOCK COUNT
040770  005037  050500                        CLR     @#ZCODE            ;CLEAR LEADING ZEROS CLOCK COUNT




                                      ;*THESE ARE TO BE SETUP FOR DISKLESS USE ONLY

040774  012737  010000  056006                MOV     #FMT22,@#WCYL      ;FORMAT22=16BIT WORDS AND
                                                                         ;CYLINDER 0
041002  012737  000001  056010                MOV     #1,@#WSECTR        ;TRACK=0, SECTOR=1
041010  005037  056012                        CLR     @#WKEY1            ;KEY1=0
041014  005037  056014                        CLR     @#WKEY2            ;KEY2=0
041020  012737  000400  056046                MOV     #256.,@#FNWORD     ;256 DATA WORDS
041026  004537  047172                        JSR     R5,@#CRC           ;GO TO CALCULATE CRC
041032  056006                                WCYL
041034  056016                                GCRC

                                      ;*THESE ARE REGULAR SETUPS

041036  012777  177374  153704                MOV     #-260.,@RHWC       ;256 DATA WORDS 4 HEADER WORDS
041044  012700  015220                        MOV     #WRFROM,R0         ;THESE TWO INSTRUCTIONS GETS
041050  010077  153676                        MOV     R0,@RHBA           ;ADDR. OF WRFROM INTO R0 AND
                                                                         ;BUS ADDRESS REGISTER
041054  012720  010000                        MOV     #FMT22,(R0)+       ;FORMAT=16 BIT WORDS
                                                                         ;CYLINDER=0
041060  012720  000001                2$:     MOV     #1,(R0)+           ;TRACK=0, SECTOR=1, KEYS=0
041064  005020                                CLR     (R0)+              ;KEY1=0
041066  005020                                CLR     (R0)+              ;KEY2=0
041070  012705  000400                        MOV     #256.,R5           ;COUNTER
041074  012720  052525                3$:     MOV     #52525,(R0)+       ;MOVE ALL 52525 FOR DATA
041100  005305                                DEC     R5
041102  001374                                BNE     3$                 ;BRANCH IF DATA NOT COMPLETE
041104  012777  000001  153650                MOV     #1,@RHDST          ;TRACK=0 SECTOR=1


041124  013711  015172                        MOV     @#WRIFOR,@R1       ;GET READY FOR WRITE HEADER AND
                                                                         ;DATA WITH 62 IN RHCS1
```

```
5715  041130  005037  015124            CLR     @#ERFLG$          ;CLEAR ERROR FLAG
5716  041134  012777  010000  153624    MOV     #FMT22,@RHOF      ;FORMAT BIT=1 (16 BIT WORDS)
5717  041142  005077  153622            CLR     @RHCA             ;CYLINDER =0
5718  041146  004737  055632            JSR     PC,@#COMWHD       ;WRITE HEADER AND DATA
5719
5720                                    ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
5721                                    ;*FROM THE "COMWHD" ROUTINE THAT MEANS ALL HEADER ON DISK
5722                                    ;*IS GOOD IE. ONLY DATA IS TO BE CHECKED TO SEE IF THEY ARE
5723                                    ;*ALL 52525 AND WRITE DATA GAP AND TOLERANCE GAP TO SEE IF
5724                                    ;*THEY ARE ALL ZEROS
5725
5726  041152  005737  015124            TST     @#ERFLG$          ;HAS ANY ERRORS OCCURED?
5727                                                              ;IF WRITE ERROR OCCURS ECC IS NOT CHECKED
5728
5729
5730                          ;COMPARE SOFTWARE GENERATED ECC WITH THAT GENERATED BY HARDWARE
5731  041160  023737  050460  055566        CMP     @#GECC1,@#WECC1;COMPARE SOFTWARE ECC WITH HARDWARE ECC
5732  041166  001402                        BEQ     6$              ;BRANCH IF GOOD
5733  041170  104031                        ERROR   31              ;LOW ORDER ECC IN ERROR
5734  041172  000405                        BR      7$              ;BRANCH TO CONTINUE
5735  041174  023737  050462  055570  6$:   CMP     @#GECC2,@#WECC2 ;COMPARE SOFTWARE ECC WITH HARDWARE ECC
5736  041202  001401                        BEQ     7$              ;BRANCH IF GOOD
5737  041204  104031                        ERROR   31              ;HIGH ORDER ECC IN ERROR
5738
5739
5740  041206                          7$:
5741
5742
5743
5744
5745                                    ;*FILL "REINTO" BUFFER WITH EXPECTED DATA
5746
5747  041220  004037  045702            JSR     R0,@#CLAREA       ;FILL REINTO BUFFER
5748  041224  016264                            REINTO            ;FROM
5749  041226  017262                            REINTO+<255.*2>   ;TO
5750  041230  052525                    .WORD   52525             ;DATA
5751
5752  041232  013737  050460  017264    MOV     @#GECC1,@#REINTO+<256.*2>;FILL ECC1
5753  041240  013737  050462  017266    MOV     @#GECC2,@#REINTO+<257.*2>;FILL ECC2
5754  041246  004037  045702            JSR     R0,@#CLAREA       ;FILL REST
5755  041252  017270                    REINTO+<258.*2>           ;FROM
5756  041254  017324                    REINTO+<272.*2>           ;TO
5757  041256  000000                    0                         ;DATA
5758
5759
5760  041260  005037  015124            CLR     @#ERFLG$          ;CLEAR ERROR FLAG
5761
5762
5763                                    ;*NOW COMPARE "DISK" BUFFER WITH "REINTO"
5764
5765  041264  004037  046660            JSR     R0,@#COMPAR       ;CHECK
5766  041270  016264                    REINTO                    ;GOOD BUFFER
5767  041272  054566                    DISK                      ;TEST BUFFER
5768  041274  000402                    258.                      ;NUMBER OF WORDS CHECKED
5769  041276  041304                    4$                        ;RETURN POINT FOR ERROR HEADER
5770  041300  041310                    5$                        ;RETURN POINT FOR ERROR DATA
```

```
5771   041304  104007          4$:     ERROR   7       ;READ ERROR 10 NEXT
5772   041306  000207                  RTS     PC      ;RETURN TO COMPARE
5773   041310  104010          5$:     ERROR   10      ;WORD NOS 1 TO 256 ARE
5774                                                    ;DATA WORDS
5775                                                    ;WORD NOS 257 AND 258
5776                                                    ;ARE ECC WHICH ARE CHECKED
5777                                                    ;WORD NOS 259
5778                                                    ;IS DATA GAP
5779                                                    ;WORD NOS 260 TO 273
5780                                                    ;ARE TOLERANCE GAP
5781   041312  000207                  RTS     PC      ;RETURN TO COMPARE
```

```
                                        ;*SETUP FOR WHAT IS TO BE READ
                                        ;*HEADER CRC IS RESTORED FROM A SUBROUTINE

041330  012746  052525                  MOV     #52525,-(SP)    ;DATA TO BE READ
041334  012705  000400                  MOV     #256.,R5        ;COUNTER
041340  012700  054566                  MOV     #DISK,R0        ;START OF SIMULATED DISK DATA
041344  011620          1$:             MOV     (SP),(R0)+      ;MOVE IN DATA ON TO SIMULATED DISK
041346  005305                          DEC     R5              ;COUNT
041350  001375                          BNE     1$              ;BRANCH IF 256 NOT COMPLETE
041352  005726                          TST     (SP)+           ;UNDO -(SP)
041354  022020                          CMP     (R0)+,(R0)+     ;JUMP OVER THE TWO ECC WORDS
041356  012705  000017                  MOV     #15.,R5         ;1 DATA GAP
                                                                ;14 TOLERANCE GAP
041362  005020          2$:             CLR     (R0)+           ;CLEAR DATA GAP, AND
041364  005305                          DEC     R5              ;TOLERANCE GAP
041366  001375                          BNE     2$              ;BRANCH IF NOT COMPLETE

041370  004737  051254                  JSR     PC,@#FILLEC     ;INSERT THE TWO ECC WORDS ON THE DISK
                                                                ;IN THE CORRECT PLACE

                                        ;*THESE ARE FOR ECC TEST ONLY

041374  012737  177777  015142          MOV     #-1,@#TSECC     ;THIS IS AN ECC TEST
041402  005037  050472                  CLR     @#POSITI        ;CLEAR ERROR POSITION COUNTER
041406  013737  050466  050470          MOV     @#NCODE,@#NCOUNT  ;TEMPORARY N-CODE COUNTER
041414  013737  050474  050502          MOV     @#HARDER,@#HADTMP  ;TEMPORARY HARD ERROR COUNTER
041422  005037  050460                  CLR     @#GECC1         ;ECC LOW ORDER TO BE GENERATED
041426  005037  050462                  CLR     @#GECC2         ;ECC HIGH ORDER TO BE GENERATED
041432  005037  050476                  CLR     @#DATENV        ;CLEAR DATA ENVELOPE CLOCK COUNT
041436  005037  050500                  CLR     @#ZCODE         ;CLEAR LEADING ZEROS CLOCK COUNT


                                        ;*THESE ARE TO SETUP FOR DISKLESS USE ONLY

041442  012737  010000  052650          MOV     #FMT22,@#CYL    ;16 BITS PER WORD
                                                                ;CYLINDER 0, FORMAT 16 BITS
041450  112737  000000  052653          MOVB    #0,@#SECOTR+1   ;TRACK 0
041456  112737  000000  052652          MOVB    #0,@#SECOTR     ;SECTOR 0
041464  012737  000000  052654          MOV     #0,@#KEY1       ;KEY1=0
041472  012737  000000  052656          MOV     #0,@#KEY2       ;KEY2=0
041500  012737  000400  052730          MOV     #256.,@#DAWORD  ;NO. OF DATA WORDS
041506  005037  052660                  CLR     @#X             ;THIS IS A READ COMMAND
041512  004537  047172                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
041516  052650                          CYL
041520  054550                          WCRC


                                        ;*THESE ARE REGULAR SETUPS

041522  004737  045764                  JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
```

```
5838  041526  012777  177374  153214          MOV     #-256.-4.,@RHWC ;256. DATA 4 HEADER WORDS
5839  041534  012777  016264  153210          MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
5840  041542  112746  000000                  MOVB    #0,-(SP)        ;IN LOWER BYTE GET SECTOR
5841  041546  112766  000000  000001          MOVB    #0,1(SP)        ;GET TRACK IN HIGHER BYTE
5842  041554  012677  153202                  MOV     (SP)+,@RHDST    ;TRACK/SECTOR IN RHDST
5843  041560  012777  010000  153200          MOV     #FMT22,@RHOF ;16 BITS PER WORD
5844                                                                  ;ECC CORRECTION NOT INHIBIT
5845                                                                  ;BECAUSE ECC IS NOT GOING
5846                                                                  ;TO BE CHECKED
5847  041566  005077  153176                  CLR     @RHCA           ;CYLINDER 0
5848
5849
5850  041604  013711  015176                  MOV     @#REFOR,@R1     ;READ HEADER AND DATA=72
5851  041610  005037  015124                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
5852  041614  004737  052510                  JSR     PC,@#COMHD      ;READ HEADER AND DATA
5853                                                                  ;IF THERE ARE READ ERRORS THEN
5854                                                                  ;ECC WILL NOT BE CHECKED
5855
5856
5857                                          ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
5858                                          ;*FROM THE "COMHD" ROUTINE THAT MEANS SECTOR GAP,
5859                                          ;*FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
5860                                          ;*SYNC BYTE HAVE GONE BY AND SYNCS WERE CORRECTLY
5861                                          ;*DETECTED
5862                                          ;*HEADER AND DATA ARE TO BE CHECKED.
5863                                          ;*IN CHECKING READ DATA THE WRITE FROM BUFFER
5864                                          ;*"WRFROM" IS FILLED WITH EXPECTED DATA AND
5865                                          ;*COMPARISONS ARE MADE
5866
5867  041620  005737  015124                  TST     @#ERFLG$        ;ANY ERRORS ALREADY THERE
5868  041626  004737  045464                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
5869  041632  005737  015034                  TST     @#ER1           ;NO ERRORS SHOULD BE SET
5870  041636  001401                          BEQ     6$              ;BRANCH IF NO ERRORS SET
5871  041640  104032                          ERROR   32              ;32 BIT ECC REGISTER SHOULD BE ZERO
5872                                                                  ;ONLY 11 OF THE 32 BITS CAN BE SEEN
5873                                                                  ;IN THE PATERN REGISTER
5874                                                                  ;DCK SHOULD BE SET IN RHER1
5875  041642  013746  050460          6$:     MOV     @#GECC1,-(SP)   ;GET PATTERN REGISTER
5876  041646  042716  174000                  BIC     #174000,(SP)    ;KEEP ONLY 11 BITS
5877  041652  022637  015064                  CMP     (SP)+,@#EC2     ;COMPARE PATTERN REGISTER
5878  041656  001401                          BEQ     7$              ;BRANCH IF GOOD
5879  041660  104032                          ERROR   32              ;11 BITS OF THE 32 BIT ECC REGISTER INCORRECT
5880
5881
5882
5883
5884                                          ;*ADD 16 MAINTENANCE CLOCKS TO
5885                                          ;*BRING EBL DOWN
5886
5887  041662  012700  000020          7$:     MOV     #16.,R0         ;COUNTER
5888  041666  052777  000002  153102  8$:     BIS     #MCLK,@RHMR     ;SET CLOCK
5889  041674  042777  000002  153074          BIC     #MCLK,@RHMR     ;CLEAR CLOCK
5890  041702  005300                          DEC     R0              ;COUNT
5891  041704  001370                          BNE     8$              ;BRANCH IF 16 CLOCKS NOT DONE
5892  041720  012700  015220                  MOV     #WRFROM,R0      ;GETTING READY TO FILL EXPECTED DATA
5893  041724  012720  010000                  MOV     #0!FMT22,(R0)+  ;CYLINDER 0
```

```
 5894  041730  112746  000000            MOVB    #0,-(SP)        ;IN LOWER BYTE GET SECTOR
 5895  041734  112766  000000  000001    MOVB    #0,1(SP)        ;GET TRACK IN HIGHER BYTE
 5896  041742  012620                    MOV     (SP)+,(R0)+     ;GET TRACK/SECTOR IN BUFFER
 5897  041744  012720  000000            MOV     #0,(R0)+        ;KEY1 IN BUFFER
 5898  041750  012720  000000            MOV     #0,(R0)+        ;KEY2 IN BUFFER
 5899  041754  012701  000400            MOV     #256.,R1        ;DATA WORD COUNTER
 5900  041760  012702  052525            MOV     #52525,R2       ;DATA
 5901  041764  010220              3$:   MOV     R2,(R0)+        ;DATA INTO BUFFER
 5902  041766  005301                    DEC     R1              ;COUNT
 5903  041770  001375                    BNE     3$              ;BRANCH IF 256 NOT DONE
 5904  041772  005037  015124            CLR     @#ERFLG$        ;CLEAR ERROR FLAG
 5905  041776  004737  045464            JSR     PC,@#PUTREG     ;SAVE REGISTERS
 5906
 5907                                    ;*NOW READ DATA BUFFER WILL BE CHECKED
 5908
 5909  042002  004037  046660            JSR     R0,@#COMPAR     ;CHECK
 5910  042006  015220                    WRFROM                  ;GOOD BUFFER
 5911  042010  016264                    REINTO                  ;TEST BUFFER
 5912  042012  000404                    4+256.                  ;NUMBER OF WORDS CHECKED
 5913  042014  042022                    4$                      ;RETURN POINT FOR ERROR HEADER
 5914  042016  042026                    5$                      ;RETURN POINT FOR ERROR DATA
 5915  042022  104004              4$:   ERROR   4               ;READ NEXT ERROR
 5916  042024  000207                    RTS     PC              ;RETURN TO "COMPAR"
 5917  042026  104005              5$:   ERROR   5               ;WORD NOS 1 TO 4 ARE
 5918                                                            ;HEADER WORDS
 5919                                                            ;5 TO 260 ARE DATA WORDS
 5920  042030  000207                    RTS     PC              ;RETURN TO "COMPAR"
```

```
5921
5922  042034  012706  001000                  MOV     #STACK,SP       ;RESET STACK
5923
5924                                          ;*SETUP FOR WHAT IS TO BE READ
5925                                          ;*HEADER CRC IS RESTORED FROM A SUBROUTINE
5926
5927  042046  012746  052525                  MOV     #52525, -(SP)   ;DATA TO BE READ
5928  042052  012705  000400                  MOV     #256.,R5        ;COUNTER
5929  042056  012700  054566                  MOV     #DISK,R0        ;START OF SIMULATED DISK DATA
5930  042062  011620          1$:             MOV     (SP),(R0)+      ;MOVE IN DATA ON TO SIMULATED DISK
5931  042064  005305                          DEC     R5              ;COUNT
5932  042066  001375                          BNE     1$              ;BRANCH IF 256 NOT COMPLETE
5933  042070  005726                          TST     (SP)+           ;UNDO -(SP)
5934  042072  022020                          CMP     (R0)+,(R0)+     ;JUMP OVER THE TWO ECC WORDS
5935  042074  012705  000017                  MOV     #15.,   R5      ;1 DATA GAP
5936                                                                  ;14 TOLERANCE GAP
5937  042100  005020          2$:             CLR     (R0)+           ;CLEAR DATA GAP, AND
5938  042102  005305                          DEC     R5              ;TOLERANCE GAP
5939  042104  001375                          BNE     2$              ;BRANCH IF NOT COMPLETE
5940
5941
5942  042106  004737  051254                  JSR     PC,@#FILLEC     ;INSERT ECC IN PROPER PLACE ON DISK
5943
5944
5945
5946                                          ;*THESE ARE FOR ECC TEST ONLY
5947
5948  042112  012737  177777  015142          MOV     #-1,@#TSECC     ;THIS IS AN ECC TEST
5949  042120  005037  050472                  CLR     @#POSITI        ;CLEAR ERROR POSITION COUNTER
5950  042124  013737  050466  050470          MOV     @#NCODE,@#NCOUNT  ;TEMPORARY N-CODE COUNTER
5951  042132  013737  050474  050502          MOV     @#HARDER,@#HADTMP  ;TEMPORARY HARD ERROR COUNTER
5952  042140  005037  050460                  CLR     @#GECC1         ;ECC LOW ORDER TO BE GENERATED
5953  042144  005037  050462                  CLR     @#GECC2         ;ECC HIGH ORDER TO BE GENERATED
5954  042150  005037  050476                  CLR     @#DATENV        ;CLEAR DATA ENVELOPE CLOCK COUNT
5955  042154  005037  050500                  CLR     @#ZCODE         ;CLEAR LEADING ZEROS CLOCK COUNT
5956
5957
5958                                          ;*THESE ARE TO SETUP FOR DISKLESS USE ONLY
5959
5960  042160  012737  010000  052650          MOV     #FMT22,@#CYL    ;16 BITS PER WORD
5961                                                                  ;CYLINDER 0, FORMAT 16 BITS
5962  042166  112737  000000  052653          MOVB    #0,@#SECOTR+1   ;TRACK 0
5963  042174  112737  000000  052652          MOVB    #0,@#SECOTR     ;SECTOR 0
5964  042202  012737  000000  052654          MOV     #0,@#KEY1       ;KEY1=0
5965  042210  012737  000000  052656          MOV     #0,@#KEY2       ;KEY2=0
5966  042216  012737  000400  052730          MOV     #256.,@#DAWORD  ;NO. OF DATA WORDS
5967  042224  005037  052660                  CLR     @#X             ;THIS IS A READ COMMAND
5968  042230  004537  047172                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
5969  042234  052650                          CYL
5970  042236  054550                          WCRC
5971
5972
5973                                          ;*THIS IS TO INSERT ERROR
5974                                          ;*THE DISK DATA IS IN LOCATION STARTING FROM 'DISK'
5975                                          ;*THE POSITION OF THE ERROR CAN BE CHANGED BY CHANGING
5976                                          ;*THIS MOVE
```

```
                                        ;*THIS CHANGES THE LAST BIT OF THE ECC
                                        MOV     @#WECC2,-(SP)   ;GET LAST ECC
                                        COM     (SP)            ;INVERT ALL BITS OF WECC2
                                        BIC     #^C100000,(SP)  ;KEEP BIT 16
                                        BIC     #100000,@#WECC2 ;CLEAR BIT 16 IN ECC
                                        BIS     (SP)+,@#WECC2   ;THIS WILL SET BIT 16 IF IT WAS 0
                                                                ;OR WILL SET NOTHING IF IT WAS A 1

                                        MOV     #4118.,@#8$     ;INSERT POSITION REG.

                                        ;*THESE ARE REGULAR SETUPS

                                        JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
                                        MOV     #-256.,-4.,@RHWC ;256. DATA 4 HEADER WORDS
                                        MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
                                        MOVB    #0,     -(SP)   ;IN LOWER BYTE GET SECTOR
                                        MOVB    #0,     1(SP)   ;GET TRACK IN HIGHER BYTE
                                        MOV     (SP)+,  @RHDST  ;TRACK/SECTOR IN RHDST
                                        MOV     #FMT22,@RHOF ;16 BITS PER WORD
                                                                ;ECC CORRECTION NOT INHIBIT
                                                                ;BECAUSE ECC IS NOT GOING
                                                                ;TO BE CHECKED
                                        CLR     @RHCA           ;CYLINDER 0

                                        MOV     @#REFOR,@R1     ;READ HEADER AND DATA=72
                                        CLR     @#ERFLG$        ;CLEAR ERROR FLAG
                                        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

                                        TST     @#ERFLG$        ;ANY ERRORS ALREADY THERE
                                        JSR     PC,@#PUTREG     ;SAVE REGISTERS
                                        CMP     #DCK,@#ER1      ;ONLY DATA CHECK ERROR SHOULD BE SET
                                        BEQ     6$              ;BRANCH IF YES
                                        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
                                6$:     MOV     @#GECC1,-(SP)   ;GET PATTERN REGISTER
                                        BIC     #174000,(SP)    ;KEEP ONLY 11 BITS
```

```
6033  042424  022637  015064          CMP     (SP)+,@#EC2     ;COMPARE PATTERN REGISTER
6034  042430  001401                  BEQ     7$              ;BRANCH IF GOOD
6035  042432  104032                  ERROR   32              ;11 BITS OF THE 32 BIT ECC REGISTER INCORRECT
6036
6037  042434  004037  051102    7$:   JSR     R0,@#ECORR      ;GO TO ECC CORRECTION PROCESS
6038  042440  010026              8$:   4118.                   ;EXPECTED POSITION REG. WHEN CORRECTION
6039                                                            ;IS COMPLETE
6040
6041
6042
6043  042454  012700  015220          MOV     #WRFROM,R0      ;GETTING READY TO FILL EXPECTED DATA
6044  042460  012720  010000          MOV     #0!FMT22,(R0)+  ;CYLINDER 0
6045  042464  112746  000000          MOVB    #0,-(SP)        ;IN LOWER BYTE GET SECTOR
6046  042470  112766  000000  000001  MOVB    #0,1(SP)        ;GET TRACK IN HIGHER BYTE
6047  042476  012620                  MOV     (SP)+,(R0)+     ;GET TRACK/SECTOR IN BUFFER
6048  042500  012720  000000          MOV     #0,(R0)+        ;KEY1 IN BUFFER
6049  042504  012720  000000          MOV     #0,(R0)+        ;KEY2 IN BUFFER
6050  042510  012701  000400          MOV     #256.,R1        ;DATA WORD COUNTER
6051  042514  012702  052525          MOV     #52525,R2       ;DATA
6052  042520  010220          3$:     MOV     R2,(R0)+        ;DATA INTO BUFFER
6053  042522  005301                  DEC     R1              ;COUNT
6054  042524  001375                  BNE     3$              ;BRANCH IF 256 NOT DONE
6055
6056                                  ;*ONLY GOOD DATA HAS BEEN PUT IN 'WRFROM'
6057                                  ;*NOW THE INSERTED ERROR WILL BE PUT IN
6058                                  ;*BUT INSERTED ERROR IS IN ECC SO DATA IS NOT WRONG
6059
6060
6061  042526  004737  045464          JSR     PC,@#PUTREG     ;SAVE REGISTERS
6062  042532  005037  015124          CLR     @#ERFLG$        ;CLEAR ERROR FLAG
6063
6064
6065                                  ;*NOW READ DATA BUFFER WILL BE CHECKED
6066
6067  042536  004037  046660          JSR     R0,@#COMPAR     ;CHECK
6068  042542  015220                  WRFROM                  ;GOOD BUFFER
6069  042544  016264                  REINTO                  ;TEST BUFFER
6070  042546  000404                  4+256.                  ;NUMBER OF WORDS CHECKED
6071  042550  042556                  4$                      ;RETURN POINT FOR ERROR HEADER
6072  042552  042562                  5$                      ;RETURN POINT FOR ERROR DATA
6073  042556  104004          4$:     ERROR   4               ;READ NEXT ERROR
6074  042560  000207                  RTS     PC              ;RETURN TO "COMPAR"
6075  042562  104005          5$:     ERROR   5               ;WORD NOS 1 TO 4 ARE
6076                                                          ;HEADER WORDS
6077                                                          ;5 TO 260 ARE DATA WORDS
6078  042564  000207                  RTS     PC              ;RETURN TO "COMPAR"
6079
```

```
6080
6081  042570  012706  001000                  MOV     #STACK,SP       ;RESET STACK
6082
6083
6084                                          ;*SETUP FOR WHAT IS TO BE READ
6085                                          ;*HEADER CRC IS RESTORED FROM A SUBROUTINE
6086
6087  042602  012746  052525                  MOV     #52525,-(SP)    ;DATA TO BE READ
6088  042606  012705  000400                  MOV     #256.,R5        ;COUNTER
6089  042612  012700  054566                  MOV     #DISK,R0        ;START OF SIMULATED DISK DATA
6090  042616  011620                  1$:     MOV     (SP),(R0)+      ;MOVE IN DATA ON TO SIMULATED DISK
6091  042620  005305                          DEC     R5              ;COUNT
6092  042622  001375                          BNE     1$              ;BRANCH IF 256 NOT COMPLETE
6093  042624  005726                          TST     (SP)+           ;UNDO -(SP)
6094  042626  022020                          CMP     (R0)+,(R0)+     ;JUMP OVER THE TWO ECC WORDS
6095  042630  012705  000017                  MOV     #15.,   R5      ;1 DATA GAP
6096                                                                  ;14 TOLERANCE GAP
6097  042634  005020                  2$:     CLR     (R0)+           ;CLEAR DATA GAP, AND
6098  042636  005305                          DEC     R5              ;TOLERANCE GAP
6099  042640  001375                          BNE     2$              ;BRANCH IF NOT COMPLETE
6100
6101
6102  042642  004737  051254                  JSR     PC,@#FILLEC     ;INSERT THE TWO ECC WORDS ON THE DISK
6103                                                                  ;IN THE CORRECT PLACE
6104
6105                                          ;*THESE ARE FOR ECC TEST ONLY
6106
6107  042646  012737  177777  015142          MOV     #-1,@#TSECC     ;THIS IS AN ECC TEST
6108  042654  005037  050472                  CLR     @#POSITI        ;CLEAR ERROR POSITION COUNTER
6109  042660  013737  050466  050470          MOV     @#NCODE,@#NCOUNT  ;TEMPORARY N-CODE COUNTER
6110  042666  013737  050474  050502          MOV     @#HARDER,@#HADTMP  ;TEMPORARY HARD ERROR COUNTER
6111  042674  005037  050460                  CLR     @#GECC1         ;ECC LOW ORDER TO BE GENERATED
6112  042700  005037  050462                  CLR     @#GECC2         ;ECC HIGH ORDER TO BE GENERATED
6113  042704  005037  050476                  CLR     @#DATENV        ;CLEAR DATA ENVELOPE CLOCK COUNT
6114  042710  005037  050500                  CLR     @#ZCODE         ;CLEAR LEADING ZEROS CLOCK COUNT
6115
6116
6117                                          ;*THESE ARE TO SETUP FOR DISKLESS USE ONLY
6118
6119  042714  012737  010000  052650          MOV     #FMT22,@#CYL    ;16 BITS PER WORD
6120                                                                  ;CYLINDER 0, FORMAT 16 BITS
6121  042722  112737  000000  052653          MOVB    #0,@#SECOTR+1   ;TRACK 0
6122  042730  112737  000000  052652          MOVB    #0,@#SECOTR     ;SECTOR 0
6123  042736  012737  000000  052654          MOV     #0,@#KEY1       ;KEY1=0
6124  042744  012737  000000  052656          MOV     #0,@#KEY2       ;KEY2=0
6125  042752  012737  000400  052730          MOV     #256.,@#DAWORD  ;NO. OF DATA WORDS
6126  042760  005037  052660                  CLR     @#X             ;THIS IS A READ COMMAND
6127  042764  004537  047172                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
6128  042770  052650                          CYL
6129  042772  054550                          WCRC
6130
6131
6132                                          ;*THIS IS TO INSERT ERROR
6133                                          ;*THE DISK DATA IS IN LOCATION STARTING FROM 'DISK'
6134                                          ;*THE POSITION OF THE ERROR CAN BE CHANGED BY CHANGING
6135                                          ;*THIS MOVE
```

```
6136
6137  042774  012737  152652  054632          MOV     #152652,@#DISK+44;INSERT ERROR IN POSITION 296 THRU 304
6138                                                                    ;IN WORD NUMBER 19
6139  043002  012737  052532  054634          MOV     #52532,@#DISK+46;INSERT ERROR IN POSITION 305 THRU 308
6140                                                                    ;IN WORD NUMBER 20
6141  043010  012737  010040  043164          MOV     #4128.,@#8$     ;INSERT POSITION REG.
6142
6143
6144                                          ;*THESE ARE REGULAR SETUPS
6145
6146  043016  004737  045764                  JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
6147  043022  012777  177374  151720          MOV     #-256.-4.,@RHWC ;256. DATA 4 HEADER WORDS
6148  043030  012777  016264  151714          MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
6149  043036  112746  000000                  MOVB    #0,-(SP)        ;IN LOWER BYTE GET SECTOR
6150  043042  112766  000000  000001          MOVB    #0,1(SP)        ;GET TRACK IN HIGHER BYTE
6151  043050  012677  151706                  MOV     (SP)+, @RHDST   ;TRACK/SECTOR IN RHDST
6152  043054  012777  010000  151704          MOV     #FMT22,@RHOF    ;16 BITS PER WORD
6153                                                                  ;ECC CORRECTION NOT INHIBIT
6154                                                                  ;BECAUSE ECC IS NOT GOING
6155                                                                  ;TO BE CHECKED
6156  043062  005077  151702                  CLR     @RHCA           ;CYLINDER 0
6157  043100  013711  015176                  MOV     @#REFOR,@R1     ;READ HEADER AND DATA=72
6158  043104  005037  015124                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
6159  043110  004737  052510                  JSR     PC,@#COMHD      ;READ HEADER AND DATA
6160                                                                  ;IF THERE ARE READ ERRORS THEN
6161                                                                  ;ECC WILL NOT BE CHECKED
6162
6163
6164                                          ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
6165                                          ;*FROM THE "COMHD" ROUTINE THAT MEANS SECTOR GAP,
6166                                          ;*FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
6167                                          ;*SYNC BYTE HAVE GONE BY AND SYNCS WERE CORRECTLY
6168                                          ;*DETECTED
6169                                          ;*HEADER AND DATA ARE TO BE CHECKED.
6170                                          ;*IN CHECKING READ DATA THE WRITE FROM BUFFER
6171                                          ;*"WRFROM" IS FILLED WITH EXPECTED DATA AND
6172                                          ;*COMPARISONS ARE MADE
6173
6174  043114  005737  015124                  TST     @#ERFLG$        ;ANY ERRORS ALREADY THERE
6175  043122  004737  045464                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
6176  043126  022737  100000  015034          CMP     #DCK,@#ER1      ;ONLY DATA CHECK ERROR SHOULD BE SET
6177  043134  001401                          BEQ     6$              ;BRANCH IF YES
6178  043136  104032                          ERROR   32              ;32 BIT ECC REGISTER SHOULD BE NON
6179                                                                  ;ZERO
6180                                                                  ;ONLY 11 OF THE 32 BITS CAN BE SEEN
6181                                                                  ;IN THE PATERN REGISTER
6182                                                                  ;DCK SHOULD BE SET IN RHER1
6183  043140  013746  050460          6$:     MOV     @#GECC1,-(SP)   ;GET PATTERN REGISTER
6184  043144  042716  174000                  BIC     #174000,(SP)    ;KEEP ONLY 11 BITS
6185  043150  022637  015064                  CMP     (SP)+,@#EC2     ;COMPARE PATTERN REGISTER
6186  043154  001401                          BEQ     7$              ;BRANCH IF GOOD
6187  043156  104032                          ERROR   32              ;11 BITS OF THE 32 BIT ECC REGISTER INCORRECT
6188
6189  043160  004037  051102          7$:     JSR     R0,@#ECORR      ;GO TO ECC CORRECTION PROCESS
6190  043164  000000                  8$:     .WORD                   ;EXPECTED POSITION REG. WHEN CORRECTION
6191                                                                  ;IS COMPLETE
```

```
043166  004737  045464                 JSR     PC,@#PUTREG       ;SAVE REGISTERS
043172  022737  100100  015034         CMP     #DCK!ECH,@#ER1    ;WITH ERRORS INSERTED IN BIT POSITION 21
                                                                 ;THRU 32 HARD ERROR BIT SHOULD SET
043200  001401                         BEQ     9$                ;BRANCH IF GOOD
043202  104036                         ERROR   36                ;WITH ERROR INSERTED IN BIT POSITION 21 THRU
                                                                 ;32 HCE SHOULD SET



043204                         9$:
043216  012700  015220                 MOV     #WRFROM,R0        ;GETTING READY TO FILL EXPECTED DATA
043222  012720  010000                 MOV     #0!FMT22,(R0)+    ;CYLINDER 0
043226  112746  000000                 MOVB    #0,-(SP)          ;IN LOWER BYTE GET SECTOR
043232  112766  000000  000001         MOVB    #0,1(SP)          ;GET TRACK IN HIGHER BYTE
043240  012620                         MOV     (SP)+,(R0)+       ;GET TRACK/SECTOR IN BUFFER
043242  012720  000000                 MOV     #0,(R0)+          ;KEY1 IN BUFFER
043246  012720  000000                 MOV     #0,(R0)+          ;KEY2 IN BUFFER
043252  012701  000400                 MOV     #256.,R1          ;DATA WORD COUNTER
043256  012702  052525                 MOV     #52525,R2         ;DATA
043262  010220                 3$:     MOV     R2,(R0)+          ;DATA INTO BUFFER
043264  005301                         DEC     R1                ;COUNT
043266  001375                         BNE     3$                ;BRANCH IF 256 NOT DONE

                                       ;*ONLY GOOD DATA HAS BEEN PUT IN 'WRFROM'
                                       ;*NOW THE INSERTED ERROR WILL BE PUT IN

043270  012737  152652  015274         MOV     #152652,@#WRFROM+54;INSERT ERROR IN POSITION 296 THRU 304
                                                                 ;IN WORD NUMBER 19 IN DATA
043276  012737  052532  015276         MOV     #52532,@#WRFROM+56;INSERT ERROR IN POSITION 305 THRU 308
                                                                 ;IN WORD NUMBER 20 IN DATA

043304  004737  045464                 JSR     PC,@#PUTREG       ;SAVE REGISTERS
043310  005037  015124                 CLR     @#ERFLG$          ;CLEAR ERROR FLAG


                                       ;*NOW READ DATA BUFFER WILL BE CHECKED

043314  004037  046660                 JSR     R0,@#COMPAR       ;CHECK
043320  015220                         WRFROM                    ;GOOD BUFFER
043322  016264                         REINTO                    ;TEST BUFFER
043324  000404                         4+256.                    ;NUMBER OF WORDS CHECKED
043326  043334                         4$                        ;RETURN POINT FOR ERROR HEADER
043330  043340                         5$                        ;RETURN POINT FOR ERROR DATA
043334  104004                 4$:     ERROR   4                 ;READ NEXT ERROR
043336  000207                         RTS     PC                ;RETURN TO "COMPAR"
043340  104005                 5$:     ERROR   5                 ;WORD NOS 1 TO 4 ARE
                                                                 ;HEADER WORDS
                                                                 ;5 TO 260 ARE DATA WORDS
043342  000207                         RTS     PC                ;RETURN TO "COMPAR"
```

```
6245
6246  043346  012706  001000                MOV     #STACK,SP       ;RESET STACK
6247
6248
6249
6250                                        ;*SETUP FOR WHAT IS TO BE READ
6251                                        ;*HEADER CRC IS RESTORED FROM A SUBROUTINE
6252
6253  043360  012746  052525                MOV     #52525,-(SP)    ;DATA TO BE READ
6254  043364  012705  000400                MOV     #256.,R5        ;COUNTER
6255  043370  012700  054566                MOV     #DISK,R0        ;START OF SIMULATED DISK DATA
6256  043374  011620                1$:     MOV     (SP),(R0)+      ;MOVE IN DATA ON TO SIMULATED DISK
6257  043376  005305                DEC     R5                      ;COUNT
6258  043400  001375                BNE     1$                      ;BRANCH IF 256 NOT COMPLETE
6259  043402  005726                TST     (SP)+                   ;UNDO -(SP)
6260  043404  022020                CMP     (R0)+,(R0)+             ;JUMP OVER THE TWO ECC WORDS
6261  043406  012705  000017                MOV     #15.,R5         ;1 DATA GAP
6262                                                                ;14 TOLERANCE GAP
6263  043412  005020                2$:     CLR     (R0)+           ;CLEAR DATA GAP, AND
6264  043414  005305                DEC     R5                      ;TOLERANCE GAP
6265  043416  001375                BNE     2$                      ;BRANCH IF NOT COMPLETE
6266
6267
6268  043420  004737  051254                JSR     PC,@#FILLEC     ;INSERT THE TWO ECC WORDS ON THE DISK
6269                                                                ;IN THE CORRECT PLACE
6270
6271                                        ;*THESE ARE FOR ECC TEST ONLY
6272
6273  043424  012737  177777  015142        MOV     #-1,@#TSECC     ;THIS IS AN ECC TEST
6274  043432  005037  050472                CLR     @#POSITI        ;CLEAR ERROR POSITION COUNTER
6275  043436  013737  050466  050470        MOV     @#NCODE,@#NCOUNT  ;TEMPORARY N-CODE COUNTER
6276  043444  013737  050474  050502        MOV     @#HARDER,@#HADTMP  ;TEMPORARY HARD ERROR COUNTER
6277  043452  005037  050460                CLR     @#GECC1         ;ECC LOW ORDER TO BE GENERATED
6278  043456  005037  050462                CLR     @#GECC2         ;ECC HIGH ORDER TO BE GENERATED
6279  043462  005037  050476                CLR     @#DATENV        ;CLEAR DATA ENVELOPE CLOCK COUNT
6280  043466  005037  050500                CLR     @#ZCODE         ;CLEAR LEADING ZEROS CLOCK COUNT
6281
6282
6283                                        ;*THESE ARE TO SETUP FOR DISKLESS USE ONLY
6284
6285  043472  012737  010000  052650        MOV     #FMT22,@#CYL    ;16 BITS PER WORD
6286                                                                ;CYLINDER 0, FORMAT 16 BITS
6287  043500  112737  000000  052653        MOVB    #0,@#SECOTR+1   ;TRACK 0
6288  043506  112737  000000  052652        MOVB    #0,@#SECOTR     ;SECTOR 0
6289  043514  012737  000000  052654        MOV     #0,@#KEY1       ;KEY1=0
6290  043522  012737  000000  052656        MOV     #0,@#KEY2       ;KEY2=0
6291  043530  012737  000400  052730        MOV     #256.,@#DAWORD  ;NO. OF DATA WORDS
6292  043536  005037  052660                CLR     @#X             ;THIS IS A READ COMMAND
6293  043542  004537  047172                JSR     R5,@#CRC        ;GO TO CALCULATE CRC
6294  043546  052650                        CYL
6295  043550  054550                        WCRC
6296
6297
6298                                        ;*THIS IS TO INSERT ERROR
6299                                        ;*THE DISK DATA IS IN LOCATION STARTING FROM 'DISK'
6300                                        ;*THE POSITION OF THE ERROR CAN BE CHANGED BY CHANGING
```

```
 6301                                                ;*THIS MOVE
 6302
 6303  043552  012737  152525  054570               MOV     #152525,@#DISK+2  ;FORCE ERROR ON BIT NUMBER 32
 6304  043560  012737  152525  055564               MOV     #152525,@#DISK+(255.*2);FORCE ERROR IN BIT 4096
 6305  043566  012737  010040  043742               MOV     #4128.,@#8$       ;INSERT POSITION REG.
 6306
 6307
 6308                                                ;*THESE ARE REGULAR SETUPS
 6309
 6310  043574  004737  045764                       JSR     PC,@#CLDISK       ;SETUP GENERAL REGISTERS
 6311  043600  012777  177374  151142               MOV     #-256.-4.,@RHWC   ;256. DATA 4 HEADER WORDS
 6312  043606  012777  016264  151136               MOV     #REINTO,@RHBA     ;STARTING ADDRESS OF READ BUFFER
 6313  043614  112746  000000                       MOVB    #0,-(SP)          ;IN LOWER BYTE GET SECTOR
 6314  043620  112766  000000  000001               MOVB    #0,1(SP)          ;GET TRACK IN HIGHER BYTE
 6315  043626  012677  151130                       MOV     (SP)+,@RHDST      ;TRACK/SECTOR IN RHDST
 6316  043632  012777  010000  151126               MOV     #FMT22,@RHOF      ;16 BITS PER WORD
 6317                                                                        ;ECC CORRECTION NOT INHIBIT
 6318                                                                        ;BECAUSE ECC IS NOT GOING
 6319                                                                        ;TO BE CHECKED
 6320  043640  005077  151124                       CLR     @RHCA             ;CYLINDER 0
 6321
 6322
 6323  043656  013711  015176                       MOV     @#REFOR,@R1       ;READ HEADER AND DATA=72
 6324  043662  005037  015124                       CLR     @#ERFLG$          ;CLEAR ERROR FLAG
 6325  043666  004737  052510                       JSR     PC,@#COMHD        ;READ HEADER AND DATA
 6326                                                                        ;IF THERE ARE READ ERRORS THEN
 6327                                                                        ;ECC WILL NOT BE CHECKED
 6328
 6329
 6330                                                ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
 6331                                                ;*FROM THE "COMHD" ROUTINE THAT MEANS SECTOR GAP,
 6332                                                ;*FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
 6333                                                ;*SYNC BYTE HAVE GONE BY AND SYNCS WERE CORRECTLY
 6334                                                ;*DETECTED
 6335                                                ;*HEADER AND DATA ARE TO BE CHECKED.
 6336                                                ;*IN CHECKING READ DATA THE WRITE FROM BUFFER
 6337                                                ;*"WRFROM" IS FILLED WITH EXPECTED DATA AND
 6338                                                ;*COMPARISONS ARE MADE
 6339
 6340  043672  005737  015124                       TST     @#ERFLG$          ;ANY ERRORS ALREADY THERE
 6341  043700  004737  045464                       JSR     PC,@#PUTREG       ;SAVE REGISTERS
 6342  043704  022737  100000  015034               CMP     #DCK,@#ER1        ;ONLY DATA CHECK ERROR SHOULD BE SET
 6343  043712  001401                               BEQ     6$                ;BRANCH IF YES
 6344  043714  104032                               ERROR   32                ;32 BIT ECC REGISTER SHOULD BE NON
 6345                                                                        ;ZERO
 6346                                                                        ;ONLY 11 OF THE 32 BITS CAN BE SEEN
 6347                                                                        ;IN THE PATERN REGISTER
 6348                                                                        ;DCK SHOULD BE SET IN RHER1
 6349  043716  013746  050460          6$:          MOV     @#GECC1,-(SP)     ;GET PATTERN REGISTER
 6350  043722  042716  174000                       BIC     #174000,(SP)      ;KEEP ONLY 11 BITS
 6351  043726  022637  015064                       CMP     (SP)+,@#EC2       ;COMPARE PATTERN REGISTER
 6352  043732  001401                               BEQ     7$                ;BRANCH IF GOOD
 6353  043734  104032                               ERROR   32                ;11 BITS OF THE 32 BIT ECC REGISTER INCORRECT
 6354
 6355  043736  004037  051102          7$:          JSR     R0,@#ECORR        ;GO TO ECC CORRECTION PROCESS
 6356  043742  000000                   8$:          .WORD                     ;EXPECTED POSITION REG. WHEN CORRECTION
```

```
                                                                       ;IS COMPLETE



043744  004737  045464                      JSR     PC,@#PUTREG        ;SAVE REGISTERS
043750  022737  100100  015034              CMP     #DCK!ECH,@#ER1     ;WITH ERRORS INSERTED IN BIT POSITION 32
                                                                       ;AND 4096 HARD ERROR BIT SHOULD SET
043756  001401                              BEQ     9$                 ;BRANCH IF GOOD
043760  104036                              ERROR   36                 ;WITH ERROR INSERTED IN BIT POSITION 21 THRU
                                                                       ;32 HCE SHOULD SET




043762                              9$:
043774  012700  015220                      MOV     #WRFROM,R0         ;GETTING READY TO FILL EXPECTED DATA
044000  012720  010000                      MOV     #0!FMT22,(R0)+     ;CYLINDER 0
044004  112746  000000                      MOVB    #0,-(SP)           ;IN LOWER BYTE GET SECTOR
044010  112766  000000  000001              MOVB    #0,1(SP)           ;GET TRACK IN HIGHER BYTE
044016  012620                              MOV     (SP)+,(R0)+        ;GET TRACK/SECTOR IN BUFFER
044020  012720  000000                      MOV     #0,(R0)+           ;KEY1 IN BUFFER
044024  012720  000000                      MOV     #0,(R0)+           ;KEY2 IN BUFFER
044030  012701  000400                      MOV     #256.,R1           ;DATA WORD COUNTER
044034  012702  052525                      MOV     #52525,R2          ;DATA
044040  010220                      3$:     MOV     R2,(R0)+           ;DATA INTO BUFFER
044042  005301                              DEC     R1                 ;COUNT
044044  001375                              BNE     3$                 ;BRANCH IF 256 NOT DONE

                                    ;*ONLY GOOD DATA HAS BEEN PUT IN 'WRFROM'
                                    ;*NOW THE INSERTED ERROR WILL BE PUT IN

044046  012737  152525  015232              MOV     #152525,@#WRFROM+<5*2>    ;INSERTED ERROR IN BIT 32
044054  012737  152525  016226              MOV     #152525,@#WRFROM+<259.*2>   ;INSERT ERROR IN BIT 4096

044062  005037  015124                      CLR     @#ERFLGS           ;CLEAR ERROR FLAG
044066  004737  045464                      JSR     PC,@#PUTREG        ;SAVE REGISTERS

                                    ;*NOW READ DATA BUFFER WILL BE CHECKED

044072  004037  046660                      JSR     R0,@#COMPAR        ;CHECK
044076  015220                              WRFROM                     ;GOOD BUFFER (CHANGED)
044100  016264                              REINTO                     ;TEST BUFFER
044102  000404                              4+256.                     ;NUMBER OF WORDS CHECKED
044104  044112                              4$                         ;RETURN POINT FOR ERROR HEADER
044106  044116                              5$                         ;RETURN POINT FOR ERROR DATA
044112  104004                      4$:     ERROR   4                  ;READ NEXT ERROR
044114  000207                              RTS     PC                 ;RETURN TO "COMPAR"
044116  104005                      5$:     ERROR   5                  ;WORD NOS 1 TO 4 ARE
                                                                       ;HEADER WORDS
                                                                       ;5 TO 260 ARE DATA WORDS
044120  000207                              RTS     PC                 ;RETURN TO "COMPAR"
```

```
                                        .SBTTL  CURSORY INTERRUPT LOGIC TESTS


                                        MOV     #STACK,SP       ;RESET STACK

                                        JSR     PC,@#CLDISK     ;CLEAR DISK
                                        MOV     @#RPVEC,R0      ;GET VECTOR ADDRESS
                                        MOV     #RPTRP1,(R0)+   ;SET INTERRUPT VECTOR
                                        MOV     #340,(R0)       ;SET SERVICE ROUTINE PRIORITY
                                        MOV     #200,PS         ;SET PROCESSOR PRIORITY
                                        MOV     #RDY!IE,@R1     ;RDY, IE IN RHSC1 SHOULD CAUSE INTERRUPT
                                        MOV     @#TIMCNT,@#$TMP1;COUNTER
                                1$:     DEC     @#$TMP1         ;WAIT FOR INTERRUPT
                                        BNE     1$              ;BRANCH IF NOT ZERO
                                                                ;BEFORE THIS IS ZERO INTERRUPT SHOULD
                                                                ;OCCUR
                                        JSR     PC,@#PUTREG     ;SAVE REGISTERS
                                        ERROR   21              ;INTERRUPT DID NOT OCCUR


                                RPTRP1: CMP     (SP)+,(SP)+     ;RESTORE STACK
                                        JSR     PC,@#PUTREG     ;SAVE REGISTERS
                                        CMP     #DVA!RDY,@#CS1  ;'IE' SHOULD BE LOW
                                        ERROR   21              ;INTERRUPT OCCURRED BUT
                                                                ;'IE' FAILED TO RESET
```

```
6435
6436
6437
6438  044244  012706  001000                  MOV     #STACK,SP       ;RESET STACK
6439
6440  044250  004737  045764                  JSR     PC,@#CLDISK     ;CLEAR DISK
6441  044254  013700  014744                  MOV     @#RPVEC,R0      ;GET VECTOR ADDRESS
6442  044260  012720  044320                  MOV     #RPTRP2,(R0)+   ;SET INTERRUPT VECTOR
6443  044264  012710  000340                  MOV     #340,(R0)       ;SET SERVICE ROUTINE PRIORITY
6444  044270  012737  000240  177776          MOV     #240,PS         ;SET PROCESSOR PRIORITY
6445  044276  012711  000300                  MOV     #RDY!IE,@R1     ;RDY, IE IN RHSC1 SHOULD CAUSE INTERRUPT
6446  044302  013737  046334  001200          MOV     @#TIMCNT,@#$TMP1;COUNTER
6447  044310  005337  001200          1$:     DEC     @#$TMP1         ;WAIT FOR INTERRUPT
6448  044314  001375                  BNE     1$              ;BRANCH IF NOT ZERO
6449                                                          ;BEFORE THIS IS ZERO INTERRUPT SHOULD
6450                                                          ;OCCUR
6451
6452  044320  022626          RPTRP2: CMP     (SP)+,(SP)+     ;RESTORE STACK
6453  044322  004737  045464          JSR     PC,@#PUTREG     ;SAVE REGISTERS
6454  044326  104021                  ERROR   21              ;INTERRUPT OCCURRED WITH
6455                                                          ;PROCESSOR STATUS SAME
6456                                                          ;AS DISK
6457
6458
6459
6460
```

```
 6461  044340  004737  045764                    JSR     PC,@#CLDISK
 6462
 6463  044344  012737  000000  177776            MOV     #0,PS           ;REINSTATE PS TO 0
 6464  044432  013746  015112                    MOV     @#UNIT,-(SP)    ;GET READY TO TYPE UNIT NUMBER
 6465  044436  104405                            TYPDS
 6466  044452  013746  001112                    MOV     @#$ERTTL,-(SP)  ;GET READY TO TYPE NUMBER OF ERRORS
 6467  044456  104405                            TYPDS
 6468  044460  005037  001112                    CLR     @#$ERTTL        ;CLEAR TOTAL NUMBER OF ERRORS
 6469  044464  005037  001102                    CLR     @#$TSTNM        ;CLEAR TEST NUMBER
 6470  044470  005737  015120                    TST     @#SELECT        ;STARTING FROM 200 ?
 6471  044474  001413                            BEQ     3$              ;TEST NEXT DRIVE IF SO
 6472
 6473  044476  005237  001100                    INC     @#$PASS         ;INCREASE PASS COUNT
 6474  044502  104401  044665                    TYPE    ,$ENDMG         ;TYPE END PASS #
 6475  044506  013746  001100                    MOV     @#$PASS,-(SP)
 6476  044512  104405                            TYPDS
 6477  044514  104401  044662                    TYPE    ,$ENULL
 6478  044520  000137  022636                    JMP     @#TSTS          ;CONTINUE TESTING THIS DRIVE ---------->
 6479
 6480  044524  005337  015114          3$:       DEC     @#NOUNITS       ;NO. OF UNITS PRESENT DECREMENTED
 6481  044530  001413                            BEQ     $EOP            ;BRANCH IF ALL DRIVES COMPLETE
 6482  044532  013700  015112                    MOV     @#UNIT,R0       ;UNIT UNDER TEST
 6483  044536  012701  015072                    MOV     #UNITS,R1       ;TABLE
 6484  044542  022100                    1$:     CMP     (R1)+,R0        ;IS THIS UNIT JUST TESTED
 6485  044544  001401                            BEQ     2$              ;BRANCH IF YES
 6486  044546  000775                            BR      1$              ;BRANCH IF NO
 6487  044550  011137  015112          2$:       MOV     (R1),@#UNIT     ;THIS IS NEXT UNIT
 6488  044554  000137  022636                    JMP     @#TSTS          ;TEST NEXT DRIVE ---------->
```

```
                                        .SBTTL
                                        .SBTTL  ***SUBROUTINES***
                                        .SBTTL

                                ;*HERE IS A DETAILED EXPLAINATION OF HOW THE LOOP ON ERROR WORKS.
                                ;*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.

044702  000000          TESTAD: 0                       ;FIRST ADDRESS OF TEST

044704                  OPERSEL:
044704  005037  177776          CLR     PS              ;MAKE PROCESSOR STATUS ZERO
044760  013746  017330          MOV     @#TSTNM,-(SP)   ;GET READY TO TYPE TEST
044764  104402                  TYPOC                   ;NUMBER
045024  013746  001110          MOV     @#$LPERR,-(SP)  ;GET READY TO TYPE LOOP BACK PC
045030  104402                  TYPOC
045032  104401  001223          TYPE    ,$CRLF
045242  104412                  RDOCT
045244  062716  000002          ADD     #2,(SP)         ;GET LPADR
045250  012637  001106          MOV     (SP)+,@#$LPADR
045430  104412                  RDOCT
045432  012637  001110          MOV     (SP)+,@#$LPERR  ;GET LPERR
045436  013746  001106          MOV     @#$LPADR,-(SP)
                                ;THIS CLEARS UP GARBAGE
045442  005037  052764          CLR     @#NOSYNC                ;CLEAR FLAG FOR HEADER ERROR COMMANDS
045446  005037  015142          CLR     @#TSECC                 ;CLEAR FLAG FOR ECC TEST
                                                                ;WHEN =177777 IT IS AN ECC TEST
                                                                ;WHEN =0 IT IS NOT AN ECC TEST

045452  005037  050464          CLR     @#TSECCG                ;EVEN IN AN ECC TEST EVERY CLOCK
                                                                ;IS NOT TO GENERATE ECC
```

```
 6545
 6546
 6547  045456  005037  015144                CLR     @#TESDTE                ;IF =177777 GENERATE ECC
 6548  045462  000002                        RTI                             ;IF =0 DO NOT GENERATE ECC
                                                                               ;DRIVE TIMING ERROR TEST
```

```
                                        .SBTTL  SAVE REGISTERS ROUTINE

                                        ;THIS SAVES THE CONTENTS OF ALL HARDWARE REGISTERS
                                        ;IN MEMORY LOCATIONS TAGED FROM "WC" TO "EC2"
                                        ;
                                        ;
                                        ;THIS IS DONE SO THAT COMPARES ARE DONE WITH SAVED LOCATIONS
                                        ;AND NOT THE REGISTERS THEMSELVES.   THIS WILL MAKE
                                        ;ERROR PRINTOUTS FOR GOOD AND BAD DATA ALWAYS DIFFRENT

045464                          PUTREG:
045472  012700  014750                  MOV     #RHWC,R0        ;STARTING ADDRESS OF REG
045476  012701  015024                  MOV     #WC,R1          ;STARTING ADDRESS OF SAVE LOCATIONS
045502  012702  000023                  MOV     #RHCC-RHWC+2/2,R2 ;NUMBER OF REG. INTO R2
045506  013021                  10$:    MOV     @(R0)+,(R1)+    ;SAVE HARDWARE REG.
045510  005302                          DEC     R2
045512  001375                          BNE     10$
045522  000207                          RTS     PC
```

```
                                        .SBTTL  FLOAT 1 AND 0

                                ;*FLOAT A ONE AND A ZERO THRU A DESIGNATED REGISTER
                                ;*ABSOLUTE ADDRESS OF REG. UNDER TEST IS IN R4

045524  000000          MASK:   0                       ;BITS UNDER TEST
045526  000000          LERR:   0                       ;ERROR HLT ADDRESS
045530  000000          REGADR: 0

045532  012537  045524  BITST:  MOV     (R5)+,  MASK    ;FETCH DATA MASK
045536  012504                  MOV     (R5)+,  R4      ;GET ADDRESS OF REG. UNDER TEST
045540  010437  045530          MOV     R4,     REGADR
045544  010537  045526          MOV     R5,     LERR    ;GET ERROR RETURN ADDR.
045550  062705  000004          ADD     #4,     R5      ;MODIFY RETURN ADDR. TO JUMP OVER RTS
045554  012703  000001          MOV     #1,     R3      ;INITIALIZE DATA PATTERN
045560  004737  045602  BLT1:   JSR     PC,     BLT2    ;OUTPUT FLOATING ZERO
045564  004737  045602          JSR     PC,     BLT2    ;OUTPUT FLOATING ONE
045570  000241                  CLC
045572  006103                  ROL     R3              ;SHIFT PATTERN
045574  005703                  TST     R3
045576  001370                  BNE     BLT1            ;BRANCH IF NOT COMPLETE
045600  000205                  RTS     R5              ;RETURN TO TEST
045602  005103          BLT2:   COM     R3              ;COMPLEMENT PATTERN
045604  012737  045612  045744  MOV     #BLT3,  @#LAD   ;SET SCOPE LOOP
045612  010337  001124  BLT3:   MOV     R3,@#$GDDAT     ;STORE GOOD DATA
045616  005137  045524          COM     @#MASK          ;AND MASK WITH PATTERN
045622  043737  045524  001124  BIC     @#MASK, @#$GDDAT;CLEAR THE REST
045630  005137  045524          COM     @#MASK          ;RESTORE MASK
045634  013714  001124          MOV     @#$GDDAT,(R4)   ;OUTPUT TO REGISTER
045640  011437  001126          MOV     (R4),@#$BDDAT   ;INPUT FROM REGISTER
045644  005137  045524          COM     @#MASK
045650  043737  045524  001126  BIC     @#MASK,@#$BDDAT ;AND MASK OUT RECEIVED DATA
045656  005137  045524          COM     @#MASK          ;RESTORE MASK
045662  023737  001124  001126  CMP     @#$GDDAT,@#$BDDAT;IS DATA CORRECT
045670  001403                  BEQ     1$              ;BRANCH IF GOOD
045672  004777  177630          JSR     PC,     @LERR   ;GO TO REPORT ERROR
045676  104413                  SCOP1                   ;LOCAL SCOPE LOOP
045700  000207          1$:     RTS     PC
```

```
                                .SBTTL  CLEAR MEMORY ROUTINE


                        ;*      THIS CLEARS ANY BLOCK OF MEMORY
                        ;*      FILLING IT WITH ANY DATA
                        ;*
                        ;*      CALL
                        ;*      JSR     R0,CLAREA
                        ;*      X                       ;STARTING ADDRESS OF BLOCK
                        ;*      Y
                        ;*      Z                       ;DATA TO BE FILLED
                        ;*R1 WILL HAVE STARTING ADDRESS OF BLOCK TO BE FILLED
                        ;*R2 AFTER SUBTRACTION WILL HAVE TWICE NUMBER OF LOCATIONS
                        ;*R3 WILL HAVE DATA TO BE FILLED
                        ;*TO AVOID DIVIDE ROUTINE TWO DECREMENT R2 WILL BE USED

045702                  CLAREA:
045710  012001                  MOV     (R0)+,R1        ;FROM
045712  012002                  MOV     (R0)+,R2        ;TO
045714  012003                  MOV     (R0)+,R3        ;DATA
045716  160102                  SUB     R1,R2           ;NO. OF LOCATIONS MINUS TWO
045720  062702  000002          ADD     #2,R2           ;GET TWICE NO OF LOCATIONS
045724  010321          1$:     MOV     R3,(R1)+        ;MOVE IN DATA
045726  005302                  DEC     R2
045730  005302                  DEC     R2
045732  001374                  BNE     1$              ;BRANCH IF NOT COMPLETE
045742  000200                  RTS     R0              ;RETURN
```

```
                                        .SBTTL  LOCAL TRAPS
045744  000000                  LAD:    0

045746  032777  001000  133164  T.SCOP: BIT     #SW09,  @SWR
045754  001402                          BEQ     1$
045756  013716  045744                  MOV     @#LAD,  (SP)
045762  000002                  1$:     RTI

                                ;*EXAMPLE OF THE USE OF THE ABOVE
                                ;*THIS WILL LOOP BETWEEN X: AND SCOP1 PROVIDED THERE IS NO "NEWTST"
                                ;*MOV   #X,     @#LAD
                                ;*X:    ---     ---
                                ;*      ---     ---
                                ;*      ---     ---
                                ;*      SCOP1

                                        .SBTTL  CLEAR DISK ROUTINE

045764  013701  014756          CLDISK: MOV     @#RHCS1,R1      ;R1 WILL BE CONTROL AND STATUS1
045770  013702  014754                  MOV     @#RHCS2,R2      ;R2 WILL BE CONTROL AND STATUS2
045774  013703  015000                  MOV     @#RHDS1,R3      ;R3 WILL BE DISK STATUS REGISTER1
046000  013704  014760                  MOV     @#RHER1,R4      ;R4 WILL BE ERROR REGISTER #1

046004  012712  000040                  MOV     #CLR,@R2        ;CLEAR ALL REG.
046010  013712  015112                  MOV     @#UNIT,@R2      ;REINSTATE UNIT NO.
046014  005011                          CLR     @R1             ;CLEAR FUNCTION BITS
046016  000207                          RTS     PC
```

```
                                        .SBTTL  CHECK DISK STATUS ROUTINES

                                ;*THIS CHECKS THAT DEVICE AVAILABLE (DVA) AND READY (RDY) IN RHCS1 = 1
                                ;*AND CHECKS THAT DEVICE PRESENT (DPR), DEVICE READY (DRY) IN RHDS1 = 1
                                ;*IT ALSO CHECKS THAT ALL OTHER BITS IN THESE REGISTERS = 0

046020  011637  015132        CHECKT: MOV     (SP),@#PCJSR    ;SAVE PC OF JSR+4
046024  162737  000004  015132        SUB     #4,@#PCJSR      ;GET PC OF JSR
046032  004737  045464                JSR     PC,@#PUTREG     ;SAVE REGISTERS
046036  022737  004200  015032        CMP     #DVA!RDY,@#CS1  ;RHCS1 SHOULD HAVE DEVICE AVAILABLE
                                                              ;AND BE READY
046044  001423                        BEQ     3$              ;BRANCH IF GOOD TO RHDS1 CHECK

046046  032737  004000  015032        BIT     #DVA,@#CS1      ;BAD SO TEST DEVICE AVAILABLE
046054  001004                        BNE     1$              ;TEST READY IF DVA THERE
046056  010137  001122                MOV     R1,@#$BDADR     ;ADDRESS OF BAD REGISTER (RHCS1)
046062  104026                        ERROR   26              ;RHCS1 DID NOT HAVE DEVICE
                                                              ;AVAILABLE AT START OF TEST
046064  000413                        BR      3$              ;BRANCH TO RHDS1 CHECK

046066  032737  000200  015032  1$:   BIT     #RDY,@#CS1      ;TEST READY
046074  001003                        BNE     2$              ;IF RDY THERE BRANCH
046076  010137  001122                MOV     R1,@#$BDADR     ;ADDRESS OF BAD REGISTER (RHCS1)
046102  104026                        ERROR   26              ;RHCS1 DID NOT HAVE READY
                                                              ;AT THE START OF TEST
046104  000403                2$:     BR      3$              ;BRANCH TO NEXT COMPARE
046106  010137  001122                MOV     R1,@#$BDADR     ;ADDRESS OF BAD REGISTER (RHCS1)
046112  104026                        ERROR   26              ;RHCS1 HAD SOME BITS OTHER
                                                              ;THAN DVA AND RDY SET
                                                              ;ALL OTHER BITS SHOULD BE 0
                                                              ;AT START OF TEST

046114  013746  015054        3$:     MOV     @#DS1,-(SP)     ;GET RHDS1
046120  042716  001100                BIC     #VV!PROG,(SP)   ;CLEAR VV AND PROGRAMABLE BIT
046124  022726  000600                CMP     #DPR!DRY,(SP)+  ;RHDS1 SHOULD HAVE THESE SET
046130  001424                        BEQ     8$              ;RETURN TO TEST IF GOOD

046132  032737  000400  015054  4$:   BIT     #DPR,@#DS1      ;BAD SO TEST DRIVE PRESENT
046140  001004                        BNE     5$              ;CHECK DRY IF GOOD
046142  010337  001122                MOV     R3,@#$BDADR     ;ADDRESS OF BAD REGISTER (RHDS1)
046146  104026                        ERROR   26              ;RHDS1 DOES NOT HAVE DPR
046150  000413                        BR      7$              ;BRANCH OUT
046152  032737  000200  015054  5$:   BIT     #DRY,@#DS1      ;TEST DRIVE READY
046160  001004                        BNE     6$              ;IF DPR WAS THERE SO BRANCH
046162  010337  001122                MOV     R3,@#$BDADR     ;ADDRESS OF BAD REGISTER (RHDS1)
046166  104026                        ERROR   26              ;RHDS1 DOES NOT HAVE DRY
046170  000403                        BR      7$              ;BRANCH OUT
046172  010337  001122          6$:   MOV     R3,@#$BDADR     ;ADDRESS OF BAD REGISTER (RHDS1)
046176  104026                        ERROR   26              ;RHDS1 HAS SOME BITS OTHER
                                                              ;THAN MOL, DRY, DPR, SET
                                                              ;ALL OTHER BIT$ SHOULD BE 0
046200  000207                7$:     RTS     PC              ;RETURN TO TEST AND HALT - FATAL ERROR

046202  062716  000006        8$:     ADD     #6,(SP)         ;ADJUST STACK PTR TO GET OVER HALT IN TEST
```

```
                                        RTS     PC              ;RETURN TO TEST AND CONTINUE TESTING

                                ;*THIS CHECKS THAT DEVICE AVAILABLE (DVA) AND READY (RDY) IN RHCS1 = 1
                                ;*AND CHECKS THAT DEVICE PRESENT (DPR), DEVICE READY (DRY) IN RHDS1 = 1
                                ;*IT DOES NOT CHECK THAT OTHER BITS IN THESE REGISTERS = 0

046210  011637  015132          CHECKE: MOV     (SP),@#PCJSR    ;SAVE PC OF JSR+4
046214  162737  000004  015132          SUB     #4,@#PCJSR      ;GET PC OF JSR
046222  004737  045464                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
046226  032737  000200  015032          BIT     #RDY,@#CS1      ;RHCS1 SHOULD HAVE DEVICE AVAILABLE
                                                                ;AND BE READY
046234  001004                          BNE     1$              ;BRANCH IF GOOD
046236  010137  001122                  MOV     R1,@#$BDADR     ;FAILING REGISTER
046242  104026                          ERROR   26              ;RHCS1 IS IN ERROR
                                                                ;DOES NOT HAVE DVA, RDY
046244  000427                          BR      4$              ;BRANCH

046246  032737  004000  015032  1$:     BIT     #DVA,@#CS1      ;RHCS1 SHOULD HAVE DEVICE AVAILABLE
                                                                ;AND BE READY
046254  001004                          BNE     2$              ;BRANCH IF GOOD
046256  010137  001122                  MOV     R1,@#$BDADR     ;FAILING REGISTER
046262  104026                          ERROR   26              ;RHCS1 IS IN ERROR
                                                                ;DOES NOT HAVE DVA, RDY
046264  000417                          BR      4$              ;BRANCH OUT
046266  032737  000200  015054  2$:     BIT     #DRY,@#DS1      ;RHDS1 SHOULD HAVE DPR,DRY
046274  001004                          BNE     3$              ;BRANCH IF THERE
046276  010337  001122                  MOV     R3,@#$BDADR     ;FAILING REGISTER RHDS1
046302  104026                          ERROR   26              ;RHDS1 DOES NOT HAVE DPR,DRY
046304  000407                          BR      4$              ;BRANCH OUT
046306  032737  000400  015054  3$:     BIT     #DPR,@#DS1      ;RHDS1 SHOULD HAVE DPR,DRY
046314  001004                          BNE     5$              ;BRANCH IF THERE
046316  010337  001122                  MOV     R3,@#$BDADR     ;FAILING REGISTER RHDS1
046322  104026                          ERROR   26              ;RHDS1 DOES NOT HAVE DPR,DRY
046324  000207                  4$:     RTS     PC              ;RETURN TO TEST AND HALT - FATAL ERROR

046326  062716  000006          5$:     ADD     #6,(SP)         ;ADJUST STACK TO GET OVER HALT IN TEST
046332  000207                          RTS     PC              ;RETURN TO TEST AND CONTINUE TESTING

                                        .SBTTL  WAIT LOOP
                                ;*      WAIT LOOP
                                ;*      ONE  LOOP OR ONE COUNT = 5.15 MICROSEC WITH BIPOLAR MEMORY (MIN)
                                ;*      ONE LOOP OR ONE COUNT = 11.86 MICROSEC WITH CORE (MIN)
                                ;*      WITH CORE ERROR IS INDICATED AFTER ABOUT 650 MILLISEC (MIN)

046334  177777                  TIMCNT: 177777                  ;WAITING COUNT

046336  010046                  WAIT.T: MOV     R0,-(SP)        ;SAVE R0
046340  016600  000002                  MOV     2(SP),R0        ;GET ADRESS OF REG. ADDRESS
046344  010037  001204                  MOV     R0,@#$TMP3      ;WAT PC+2 IN $TMP3
046350  162737  000002  001204          SUB     #2,@#$TMP3      ;WAT PC FOR TYPEOUT
046356  012037  001176                  MOV     (R0)+,@#$TMP0   ;WAIT REGISTER ADDRESS
046362  012037  001200                  MOV     (R0)+,@#$TMP1   ;WAIT ON BIT
046366  010066  000002                  MOV     R0,2(SP)        ;RESTORE RETURN ON STACK
046372  012600                          MOV     (SP)+,R0        ;RESTORE R0
046374  013737  046334  001202          MOV     @#TIMCNT,@#$TMP2;TEMPORARY COUNT
```

```
6776
6777  046402  033777  001200  132566  1$:     BIT     @#$TMP1,@$TMP0  ;IS REQUIRED BIT THERE?
6778  046410  001021                          BNE     2$              ;BRANCH IF YES
6779  046412  005337  001202                  DEC     @#$TMP2         ;COUNT
6780  046416  001371                          BNE     1$              ;BRANCH IF NOT TIME UP
6781  046420  013737  046334  001202          MOV     @#TIMCNT,@#$TMP2;TEMPORARY COUNT
6782  046426  033777  001200  132542  3$:     BIT     @#$TMP1,@$TMP0  ;IS REQUIRED BIT THERE?
6783  046434  001007                          BNE     2$              ;BRANCH IF YES
6784  046436  005337  001202                  DEC     @#$TMP2         ;COUNT
6785  046442  001371                          BNE     3$              ;BRANCH IF NOT TIME UP
6786  046444  017737  132526  001126          MOV     @$TMP0,@#$BDDAT ;REGISTER CONTENTS
6787  046452  104016                          ERROR   16              ;WAITED ON BIT FAILED TO SET
6788  046454  000002                  2$:     RTI
6789
6790
6791                                  ;*      CALL FOR THE ABOVE WAITLOOP IS
6792                                  ;*
6793                                  ;*      MOV     @A,@#X$         ;A CONTAINS REGISTER ADDRESS
6794                                  ;*      -       -       -       ;HENCE X$ WILL HAVE ABSOLUTE REG. ADR.
6795                                  ;*      -       -       -
6796                                  ;*      -       -       -
6797                                  ;*      WAT
6798                                  ;*X$:   0                       ;ABSOLUTE REG. ADDRESS UNDER WAIT
6799                                  ;*      .WORD   0               ;BIT WAITED FOR
6800                                  ;*                              ;CONTINUE
```

```
                                        .SBTTL  SAVE ROUTINE

                                        ;THIS IS A SUBROUTINE TO READ & SAVE REGISTERS
                                        ;IN THE REGISTER TABLE TO ANY LOCATION
                                        ;THE CALL IS
                                        ;JSR     R0,@#SAVER
                                        ;FROM
                                        ;TO
                                        ;NUMBER OF WORDS SAVED

046456                          SAVER:
046464  012001                          MOV     (R0)+,R1        ;FROM
046466  012002                          MOV     (R0)+,R2        ;TO
046470  012003                          MOV     (R0)+,R3        ;NUMBER
046472  013122                  1$:     MOV     @(R1)+,(R2)+    ;SAVE REGISTER CONTENTS
046474  005303                          DEC     R3              ;COUNT
046476  001375                          BNE     1$              ;BRANCH IF NOT DONE
046506  000200                          RTS     R0


                                        .SBTTL  WRITE CHECK ROUTINE

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

                                        ;THESE ARE TO SET UP FOR DISKLESS USE ONLY

046510  012737  010000  052650  WRCHHD: MOV     #FMT22,@#CYL    ;CYLINDER 0 FORMAT 16 BIT WORDS
046516  112737  000001  052653          MOVB    #1,@#SECOTR+1   ;TRACK=1
046524  112737  000001  052652          MOVB    #1,@#SECOTR     ;SECTOR=1
046532  005037  052654                  CLR     @#KEY1          ;KEY1=0
046536  005037  052656                  CLR     @#KEY2          ;KEY2=0
046542  012737  000044  052730          MOV     #36.,DAWORD     ;NO OF DATA WORDS
046550  005037  052660                  CLR     @#X             ;THIS IS A READ OPERATION
046554  004537  047172                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
046560  052650                          CYL
046562  054550                          WCRC

                                        ;THESE ARE REGULAR SETUPS

046564  004737  045764                  JSR     PC,@#CLDISK     ;SET UP GENERAL REGISTERS
                                                                ;AND CLEAR DISK REGISTERS
046570  012777  177730  146152          MOV     #-40.,@RHWC     ;36 DATA WORDS 4 HEADER WORDS
046576  012777  016264  146146          MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
046604  112746  000001                  MOVB    #1,-(SP)        ;SECTOR=1
046610  112766  000001  000001          MOVB    #1,1(SP)        ;TRACK=1 IN UPPER BYTE
046616  012677  146140                  MOV     (SP)+,@RHDST    ;TRACK=1, SECTOR=1 IN RHDST
046622  012777  014000  146136          MOV     #FMT22!ECI,@RHOF ;16 BIT WORDS
                                                                ;ECC CORRECTION INHIBIT BECAUSE
                                                                ;ECC LOGIC IS NOT CHECKED YET
046630  005077  146134                  CLR     @RHCA           ;CYLINDER=0
046646  013711  015166                  MOV     @#WRCHDT,@R1    ;WRITE CHECK HEADER AND DATA=52
                                                                ;INTO RHCS1
```

```
 6857  046652  004737  052510                JSR     PC,@#COMHD      ;WRITE CHECK HEADER AND DATA
 6858                                                                  ;SAME AS READ HEADER AND DATA
 6859
 6860  046656  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

046660                  COMPAR:
046672  012001                  MOV     (R0)+,R1        ;ADDRESS OF GOOD DATA BUFFER
046674  012002                  MOV     (R0)+,R2        ;ADDRESS OF TEST DATA BUFFER
046676  012003                  MOV     (R0)+,R3        ;NO OF WORDS TO BE COMPARED
046700  012037  001176          MOV     (R0)+,$TMP0     ;RETURN ON ERROR TO PRINT HEADER
046704  012037  001200          MOV     (R0)+,$TMP1     ;RETURN ON ERROR TO PRINT DATA
046710  011000                  MOV     (R0),R0         ;RETURN ON NO ERROR
046712  010304                  MOV     R3,R4           ;NO OF WORDS TO BE COMPARED
046714  005204                  INC     R4
046716  010437  052770  1$:     MOV     R4,@#ERWORD     ;FOR ERROR WORD NO
046722  022122                  CMP     (R1)+,(R2)+     ;COMPARE GOOD WITH TEST DATA
046724  001426                  BEQ     3$              ;BRANCH IF GOOD

046726  014137  001124          MOV     -(R1),@#$GDDAT  ;GOOD DATA
046732  014237  001126          MOV     -(R2),@#$BDDAT  ;BAD DATA
046736  160337  052770          SUB     R3,@#ERWORD     ;ERROR WORD NO.
046742  005737  015124          TST     @#ERFLG$        ;ANY ERRORS ALREAY THERE
046746  001003                  BNE     2$              ;BRANCH IF YES
046750  004777  132222          JSR     PC,@$TMP0       ;RETURN TO PRINT HEADER
046754  000402                  BR      5$              ;BRANCH TO AVOID PRINTING NEXT ERROR
046756  004777  132216  2$:     JSR     PC,@$TMP1       ;RETURN TO PRINT DATA
046762  022122          5$:     CMP     (R1)+,(R2)+     ;UNDO -(R1) AND -(R2) FOR ERRORS
046764  017746  132150          MOV     @SWR,-(SP)      ;GET SWITCH SETTING
046770  042716  177177          BIC     #^C600,(SP)     ;KEEP ONLY SWITCH 7 AND 8
046774  022726  000200          CMP     #SW07,(SP)+     ;IS 7 SET AND 8 RESET
047000  001402                  BEQ     4$              ;BRANCH OUT IF YES
047002  005303          3$:     DEC     R3              ;COUNT
047004  001344                  BNE     1$              ;BRANCH IF ALL NOT DEVICE
047006                  4$:
047020  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

047022  012737  010000  052650  WRCHDA: MOV     #FMT22,@#CYL    ;CYLINDER 0 FORMAT 16 BIT WORDS
047030  112737  000001  052653          MOVB    #1,@#SECOTR+1   ;TRACK=1
047036  112737  000001  052652          MOVB    #1,@#SECOTR     ;SECTOR=1
047044  005037  052654                  CLR     @#KEY1          ;KEY1=0
047050  005037  052656                  CLR     @#KEY2          ;KEY2=0
047054  012737  000040  052730          MOV     #32.,@#DAWORD   ;NO OF DATA WORDS
047062  005037  052660                  CLR     @#X             ;THIS IS A READ OPERATION

047066  004537  047172                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
047072  052650                          CYL
047074  054550                          WCRC

                                        ;THESE ARE REGULAR SETUPS

047076  004737  045764                  JSR     PC,@#CLDISK     ;SET UP GENERAL REGISTERS
                                                                ;AND CLEAR DISK REGISTERS

047102  012777  177740  145640          MOV     #-32.,@RHWC     ;36 DATA WORDS 4 HEADER WORDS
047110  012777  016264  145634          MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
047116  112746  000001                  MOVB    #1,-(SP)        ;SECTOR=1
047122  112766  000001  000001          MOVB    #1,1(SP)        ;TRACK=1 IN UPPER BYTE
047130  012677  145626                  MOV     (SP)+,@RHDST    ;TRACK=1, SECTOR=1 IN RHDST
047134  012777  014000  145624          MOV     #FMT22!ECI,@RHOF        ;16 BIT WORDS
                                                                ;ECC CORRECTION INHIBIT BECAUSE
                                                                ;ECC LOGIC IS NOT CHECKED YET
047142  005077  145622                  CLR     @RHCA           ;CYLINDER=0
047160  013711  015164                  MOV     @#WRCHEK,@R1    ;WRITE CHECK DATA=50 INTO RHCS1
047164  004737  052510                  JSR     PC,@#COMHD      ;WRITE CHECK HEADER AND DATA
                                                                ;SAME AS READ HEADER AND DATA

047170  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

047172                          CRC:
047174  012500                  MOV     (R5)+,R0        ;GET POINTER TO CYL NO.
047206  005001                  CLR     R1              ;CLEAR WORKING LOCATION
047210  005037  001210          CLR     @#$TMP5
047214  012737  000004  001176  MOV     #4,@#$TMP0      ;WORD COUNT
047222  012037  001204  16$:    MOV     (R0)+,@#$TMP3   ;TEMPORARY WORD STORAGE
047226  012737  000020  001202  MOV     #16.,$TMP2      ;BIT COUNT
047234  013737  001204  001206  MOV     @#$TMP3,@#$TMP4 ;TEMPORARY WORD STORAGE
047242  006037  001204  15$:    ROR     @#$TMP3         ;GET LSB INTO "C"
047246  006037  001210          ROR     @#$TMP5         ;GET ABOVE "C" INTO $TMP5
047252  032701  000001          BIT     #BIT0,R1        ;IS POSITION 15 HIGH
047256  001403                  BEQ     1$              ;BRANCH IF POSITION 16 LOW
047260  012703  100000          MOV     #BIT15,R3       ;GET POSITION 16
047264  000401                  BR      2$
047266  005003          1$:     CLR     R3              ;GET POSITION 16
047270  063703  001210  2$:     ADD     @#$TMP5,R3      ;XOR POSITION 16 WITH D
                                                        ;TO GIVE B
047274  032701  040000          BIT     #BIT14,R1       ;IS POSITION 2 HIGH
047300  001403                  BEQ     3$              ;BRANCH IF POSITION 2 LOW
047302  012702  100000          MOV     #BIT15,R2       ;GET POSITION 2
047306  000401                  BR      4$
047310  005002          3$:     CLR     R2              ;GET POSITION 2
047312  060302          4$:     ADD     R3,R2           ;XOR B WITH POSITION 2
                                                        ;TO GIVE C
047314  032701  000002          BIT     #BIT1,R1        ;IS POSITION 15 HIGH
047320  001403                  BEQ     5$              ;BRANCH IF POSITION 15 LOW
047322  012704  100000          MOV     #BIT15,R4       ;GET POSITION 15
047326  000401                  BR      6$
```

```
 6998  047330  005004                  5$:     CLR     R4              ;GET POSITION 15
 6999  047332  060304                  6$:     ADD     R3,R4           ;XOR POSITION 15 WITH B
 7000                                                                  ;TO GIVE E
 7001  047334  006037  001206                  ROR     @#$TMP4         ;GET LSB INTO "C"
 7002  047340  006001                          ROR     R1              ;GET ABOVE C INTO R1
 7003  047342  005703                          TST     R3              ;TEST B
 7004  047344  100403                          BMI     7$              ;BRANCH IF B=1
 7005  047346  042701  100000                  BIC     #BIT15,R1       ;SET B IN POSITION 1
 7006  047352  000402                          BR      10$
 7007  047354  052701  100000            7$:   BIS     #BIT15,R1       ;SET B IN POSITION 1
 7008  047360  005702                  10$:    TST     R2              ;TEST C
 7009  047362  100403                          BMI     11$             ;BRANCH IF C=1
 7010  047364  042701  020000                  BIC     #BIT13,R1       ;GET C IN POSITION 3
 7011  047370  000402                          BR      12$
 7012  047372  052701  020000            11$:  BIS     #BIT13,R1       ;GET C IN POSITION 3
 7013  047376  005704                  12$:    TST     R4              ;TEST E
 7014  047400  100403                          BMI     13$             ;BRANCH IF E=1
 7015  047402  042701  000001                  BIC     #BIT0,R1        ;GET E IN POSITION 16
 7016  047406  000402                          BR      14$
 7017  047410  052701  000001            13$:  BIS     #BIT0,R1        ;GET E IN POSITION 16
 7018  047414  005337  001202            14$:  DEC     @#$TMP2         ;BIT COUNTER
 7019  047420  001310                          BNE     15$             ;BRANCH IF 16 NOT DONE
 7020  047422  005337  001176                  DEC     @#$TMP0         ;WORD COUNTER
 7021  047426  001275                          BNE     16$             ;BRANCH IF 4 NOT DONE
 7022  047430  010135                          MOV     R1,@(R5)+       ;PUT CRC WHERE DESIRED
 7023  047444  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

047446                          SETDSK:
047454  012700  177400                  MOV     #177400,R0      ;DATA IN THE DISK
047460  012701  000400                  MOV     #256.,R1        ;COUNTER
047464  012702  054566                  MOV     #DISK,R2        ;START OF SIMULATOR DISK
047470  010022                  1$:     MOV     R0,(R2)+        ;MOVE IN DATA
047472  005301                  DEC     R1                      ;COUNT FOR 256
047474  001375                  BNE     1$                      ;BRANCH IF 256 NOT COMPLETE
047476  012701  000021                  MOV     #17.,R1         ;2 ECC WORDS, 1 DATA GAP
                                                                ;14 TOLERANCE GAP
047502  005022                  2$:     CLR     (R2)+           ;CLEAR ECC,DATA GAP AND
                                                                ;TOLERANCE GAP
047504  005301                  DEC     R1                      ;COUNT
047506  001375                  BNE     2$                      ;BRANCH IF NOT COMPLETE

                                        ;NOW SET UP FOR DISKLESS USE

047510  012737  010000  052650          MOV     #FMT22,@#CYL    ;CYLINDER 0 (16 BIT WORDS)
047516  112737  000001  052653          MOVB    #1,@#SECOTR+1   ;TRACK=1
047524  112737  000001  052652          MOVB    #1,@#SECOTR     ;SECTOR=1
047532  012737  000001  052654          MOV     #1,@#KEY1       ;KEY1=1
047540  012737  000001  052656          MOV     #1,@#KEY2       ;KEY2=1
047546  013737  000400  052730          MOV     256.,@#DAWORD   ;NO. OF DATA WORDS
047554  004537  047172                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
047560  052650                          CYL                     ;FIRST CRC WORD
047562  054550                          WCRC                    ;PUT CALCULATED CRC
047572  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

047574  010037  015132        HCCRCE: MOV     R0,@#PCJSR      ;SAVE PC OF JSR+4
047600  162737  000004  015132        SUB     #4,@#PCJSR      ;GET PC OF JSR
047606  004737  045764                JSR     PC,@#CLDISK     ;INIT AND SETUP GENERAL REG.
047624  011037  001210                MOV     (R0),@#$TMPS    ;SAVE COMMAND
047630  012011                        MOV     (R0)+,@R1       ;COMMAND
047632  012077  145132                MOV     (R0)+,@RHCA     ;CYLINDER
047636  112046                        MOVB    (R0)+,-(SP)     ;SECTOR
047640  105720                        TSTB    (R0)+           ;UP DATE R0
047642  112066  000001                MOVB    (R0)+,1(SP)     ;TRACK
047646  105720                        TSTB    (R0)+           ;UPDATE R0
047650  012677  145106                MOV     (SP)+,@RHDST    ;TRACK SECTOR
047654  012077  145070                MOV     (R0)+,@RHWC     ;NO. OF DATA WORDS +4 HEADER
                                                                ;IF A READ HEADER AND DATA
047660  012077  145066                MOV     (R0)+,@RHBA     ;STARTING ADDRESS OF BUFFER
047664  012037  052660                MOV     (R0)+,@#X       ;X=0 READ HEADER AND DATA
                                                                ;X=1 WRITE DATA
047670  012777  014000  145070        MOV     #FMT22!ECI,@RHOF ;16 BITS PER WORD
                                                                ;ECC CORRECTION INHIBIT
047676  005037  015124                CLR     @#ERFLG$        ;CLEAR ERROR FLAG
047702  004737  052510                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.

047706  004737  045464                JSR     PC,@#PUTREG     ;SAVE REGISTERS
047712  005737  015124                TST     @#ERFLG$        ;ANY ERRORS ALREADY THERE
047716  001034                        BNE     10$             ;BRANCH IF YES
047720  005737  052660                TST     @#X             ;IS THIS A READ
047724  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
```

```
7119
7120  047726  004037  046660          JSR     R0,@#COMPAR     ;CHECK
7121  047732  015220                  WRFROM                  ;GOOD DATA
7122  047734  016264                  REINTO                  ;TEST BUFFER
7123  047736  000400                  256.                    ;4 HEADER 252 DATA
7124  047740  047746                  1$                      ;RETURN POINT FOR ERROR HEADER
7125  047742  047752                  2$                      ;RETURN POINT FOR ERROR DATA
7126  047744  050010                  10$                     ;RETURN FOR GOOD COMPARISON
7127  047746  104004          1$:     ERROR   4               ;READ NEXT ERROR 5
7128  047750  000207                  RTS     PC              ;RETURN TO COMPARISON SUBROUTINE
7129  047752  104005          2$:     ERROR   5               ;WORD NO 1 THRU 4 ARE
7130                                                          ;HEADER WORDS AND HENCE
7131                                                          ;SHOULD BE READ AS WRITTEN ON
7132                                                          ;DISK, WORD NOS. 5 ONWARDS
7133                                                          ;SHOULD NOT BE READ AND HENCE
7134                                                          ;READ INTO BUFFER
7135                                                          ;SHOULD BE UNCHANGED
7136  047754  000207                  RTS     PC              ;RETURN TO COMPARISON
7137
7138  047756  000414                  BR      10$             ;JUMP OUT
7139
7140                                  ;NOW THE DISK WILL BE CHECKED
7141                                  ;NO DATA SHOULD BE WRITTEN
7142                                  ;REINTO BUFFER HAS BEEN FILLED WITH EXPECTED DATA
7143                                  ;DISK HAS BEEN FILLED WITH 177400
7144                                  ;WRFROM HAS BEEN FILLED WITH 125252
7145
7146  047760  004037  046660  3$:     JSR     R0,@#COMPAR     ;CHECK
7147  047764  016264                  REINTO                  ;GOOD DATA BUFFER
7148  047766  054566                  DISK                    ;TEST BUFFER
7149  047770  000400                  256.
7150  047772  050000                  4$                      ;RETURN POINT FOR ERROR HEADER
7151  047774  050004                  5$                      ;RETURN POINT FOR ERROR DATA
7152  047776  050010                  10$                     ;RETURN POINT FOR GOOD COMPARISON
7153  050000  104004          4$:     ERROR   4               ;READ NEXT ERROR 5
7154  050002  000207                  RTS     PC              ;RETURN TO COMPARISON SUBROUTINE
7155  050004  104005          5$:     ERROR   5               ;WORD NO ARE ALL DATA
7156                                                          ;WORDS THE SHOULD NOT
7157                                                          ;HAVE BEEN CHANGED BY THE
7158                                                          ;WRITE COMMAND
7159  050006  000207                  RTS     PC              ;RETURN TO COMPARISON SUBROUTINE
7160  050010  005720          10$:    TST     (R0)+           ;IS THIS A HCRC ON HCE CHECK?
7161  050012  001442                  BEQ     6$              ;BRANCH IF HCRC
7162  050014  022737  000072  001210  CMP     #72,@#$TMP5     ;IS THIS A READ COMMAND
7163  050022  001417                  BEQ     11$             ;BRANCH IF YES
7164  050024  017737  144730  001126  MOV     @RHER1,@#$BDDAT ;TEST DATA
7165  050032  022737  000200  001126  CMP     #HCE,@#$BDDAT   ;ONLY HEADER COMPARE BIT?
7166                                                          ;SHOULD BE SET
7167  050040  001470                  BEQ     7$              ;BRANCH IF GOOD
7168  050042  013737  014760  045530  MOV     @#RHER1,@#REGADR ;REGISTER ADDRESS RHER1
7169  050050  012737  000200  001124  MOV     #HCE,@#$GDDAT   ;GOOD DATA
7170  050056  104027                  ERROR   27              ;AFTER AN ERROR ON THE
7171                                                          ;HEADER ONLY HCE SHOULD
7172  050060  000460                  BR      7$              ;BE SET
7173  050062                  11$:
7174  050062  017737  144672  001126  MOV     @RHER1,@#$BDDAT ;TEST DATA
```

```
7175  050070  022737  100200  001126          CMP     #DCK!HCE,@#$BDDAT       ;ONLY HEADER COMPARE BIT?
7176                                                                          ;SHOULD BE SET
7177                                                                          ;DCK IS SET BECAUSE ECC IS NOT READ
7178  050076  001451                          BEQ     7$                      ;BRANCH IF GOOD
7179  050100  013737  014760  045530          MOV     @#RHER1,@#REGADR        ;REGISTER ADDRESS RHER1
7180  050106  012737  100200  001124          MOV     #DCK!HCE,@#$GDDAT       ;GOOD DATA
7181  050114  104027                          ERROR   27                      ;AFTER AN ERROR ON THE
7182                                                                          ;HEADER ONLY HCE SHOULD
7183  050116  000441                          BR      7$                      ;BE SET
7184  050120  022737  000072  001210  6$:     CMP     #72,@#$TMP5             ;IS THIS A READ COMMAND?
7185  050126  001417                          BEQ     12$                     ;BRANCH IF A READ
7186  050130  017737  144624  001126          MOV     @RHER1,@#$BDDAT         ;TEST DATA
7187  050136  022737  000400  001126          CMP     #HCRC,@#$BDDAT          ;ONLY CRC ERROR SHOULD BE THERE
7188  050144  001426                          BEQ     7$
7189  050146  013737  014760  045530          MOV     @#RHER1,@#REGADR        ;REG. ADDR = RHER1
7190  050154  012737  000400  001124          MOV     #HCRC,@#$GDDAT          ;GOOD DATA
7191  050162  104027                          ERROR   27                      ;AFTER A CRC ERROR ONLY CRC
7192                                                                          ;SHOULD BE SET
7193  050164  000416                          BR      7$                      ;BRANCH OUT
7194  050166  017737  144566  001126  12$:    MOV     @RHER1,@#$BDDAT         ;TEST DATA
7195
7196  050174  022737  100400  001126          CMP     #DCK!HCRC,@#$BDDAT      ;HCRC AND DCK SHOULD BE SET
7197                                                                          ;DCK IS SET BECAUSE ECC IS NOT READ
7198  050202  001407                          BEQ     7$                      ;BRANCH IF GOOD
7199  050204  012737  100400  001124          MOV     #DCK!HCRC,@#$GDDAT      ;GOOD DATA
7200  050212  013737  014760  045530          MOV     @#RHER1,@#REGADR        ;FAILING REGISTER RHER1
7201  050220  104027                          ERROR   27                      ;AFTER A CRC ERROR ON A READ
7202                                                                          ;DCK AND HCRC SHOULD BE SET
7203                                                                          ;DCK IS SET BECAUSE ECC IS NOT READ
7204  050222  000200                  7$:     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

050224                          MIDDLE:
050230  013777  015172  144520          MOV     @#WRIFOR,@RHCS1 ;WRITE HEADER AND DATA=62
                                                                ;IN RHCS1
050236  012777  177766  144504          MOV     #-10.,@RHWC     ;10 WORDS
050244  012777  015220  144500          MOV     #WRFROM,@RHBA   ;BUS ADDRESS=WRFROM
050252  012777  000010  144502          MOV     #10,@RHDST      ;DESIRED TRACK=0 SECTOR=10
050260  052777  000010  144466          BIS     #BAI,@RHCS2     ;BUS ADDRESS INCREMENT INHIBIT
050266  012777  010000  144472          MOV     #FMT22,@RHOF    ;FORMAT 16 BIT WORDS
050274  005077  144470                  CLR     @RHCA           ;CYLINDER=0
050300  012737  000001  050326          MOV     #1,@#MID        ;SECTOR IS SET TO 1 SO THAT
                                                                ;WE CAN GET OUT AT THE
                                                                ;MIDDLE OF AN OPERATION
                                                                ;LOOKING FOR SECTOR 10
050306  012777  000001  144462          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
050314  052777  000001  144434          BIS     #GO,@RHCS1              ;GO TO RHCS1 WITH 62
050322  004137  056716                  JSR     R1,@#SEARCH
050326  000000                  MID:    .WORD   0               ;SECTOR
050334  000207                          RTS     PC
```

```
                                        .SBTTL  JAM CURRENT CYLINDER ROUTINE

                                ;*THIS SUBROUTINE WILL CHANGE THE CURRENT CYLINDER REGISTER
                                ;*THIS IS DONE BY GIVING A SEEK COMMAND THEN AN INIT
                                ;*WHICH WILL LOAD THE CURRENT CYLINDER WITH THE DESIRED CYLINDER VALUE
                                ;*
                                ;*CALL IS:
                                ;*      JSR     R0,@#MAKECYL
                                ;*      XC                      ;DESIRED VALUE OF CURRENT CYLINDER

050336                          MAKECYL:
050340  010037  015132                  MOV     R0,@#PCJSR      ;PC OF JSR+4
050344  162737  000004  015132          SUB     #4,@#PCJSR      ;SAVE PC OF JSR
050352  012005                          MOV     (R0)+,R5        ;GETTING READY TO FILL DESIRED CYLINDER
050354  010577  144410                  MOV     R5,@RHCA        ;FILL DESIRED CYLINDER REGISTER
050360  005077  144376                  CLR     @RHDST          ;MAKE SURE DESIRED SECTOR TRACK IS NOT ILLEGAL
050364  013777  015200  144364          MOV     @#SEECOM,@RHCS1 ;FILL SEEK COMMAND
050372  012777  000001  144376          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
050400  052777  000001  144350          BIS     #GO,@RHCS1      ;GO TO SEEK
050406  000240                          NOP                     ;ALLOW TIME FOR SEEK TO HANG UP
050410  000240                          NOP                     ;ALLOW TIME FOR SEEK TO HANG UP
050412  000240                          NOP                     ;ALLOW TIME FOR SEEK TO HANG UP
050414  000240                          NOP                     ;ALLOW TIME FOR SEEK TO HANG UP
050416  004737  045764                  JSR     PC,@#CLDISK     ;GIVE INIT
050422  017737  144366  001126          MOV     @RHCC,@#$BDDAT  ;TEST DATA
050430  020537  001126                  CMP     R5,@#$BDDAT     ;COMPARE CURRENT CYLINDER
050434  001406                          BEQ     1$              ;BRANCH IF GOOD
050436  010537  001124                  MOV     R5,@#$GDDAT     ;GOOD VALUE OF RHCC
050442  013737  015014  045530          MOV     @#RHCC,@#REGADR ;FAILING REGISTER ADDRESS
050450  104030                          ERROR   30              ;CURRENT CYLINDER DOES NOT MATCH DESIRED CYLINDER
                                                                ;REGISTER AFTER A SEEK AND AN INIT
050452                          1$:
050454  000200                          RTS     R0
```

```
                        .SBTTL  ECC GENERATION AND COMPARISON ROUTINE

                        ;*THIS SUBROUTINE GENERATES AND TESTS ECC
                        ;*CALL    JSR PC,ECTEST


        100000          PIE1    =100000
        040000          PIE2    =40000
        020000          PIE3    =20000
        010000          PIE4    =10000
        004000          PIE5    =4000
        002000          PIE6    =2000
        001000          PIE7    =1000
        000400          PIE8    =400
        000200          PIE9    =200
        000100          PIE10   =100
        000040          PIE11   =40
        000020          PIE12   =20
        000010          PIE13   =10
        000004          PIE14   =4
        000002          PIE15   =2
        000001          PIE16   =1
        100000          PIE17   =100000
        040000          PIE18   =40000
        020000          PIE19   =20000
        010000          PIE20   =10000
        004000          PIE21   =4000
        002000          PIE22   =2000
        001000          PIE23   =1000
        000400          PIE24   =400
        000200          PIE25   =200
        000100          PIE26   =100
        000040          PIE27   =40
        000020          PIE28   =20
        000010          PIE29   =10
        000004          PIE30   =4
        000002          PIE31   =2
        000001          PIE32   =1

050456  000000          ECDATA: 0               ;DATA BIT FOR ECC
                                                ;IF ALL ONES THEN CURRENT BIT IS A ONE
                                                ;IF ZERO THEN CURRENT BIT IS A ZERO

050460  000000          GECC1:  0               ;LOW ORDER ECC WORD TO BE GENERATED HERE
                                                ;=R1

050462  000000          GECC2:  0               ;HIGH ORDER ECC WORD TO BE GENERATED HERE
                                                ;=R2

050464  000000          TSECCG: 0               ;IF =177777 GENERATE AND TEST ECC FOR THIS BIT
                                                ;IF =0 DO NOT GENERATE AND TEST ECC FOR THIS BIT

050466  113713          NCODE:  38859.          ;N-CODE WORD
```

```
7323  050470  000000                NCOUNT: 0               ;TEMPORARY N CODE
7324  050472  000000                POSITI: 0               ;POSITION REGISTER
7325  050474  010041                HARDER: 4129.           ;HARD ERROR COUNT
7326                                                        ;TRUE COUNT IS 4128 BUT AS COMPARES ARE
7327                                                        ;DONE ONE STAGE LATER SO 4129
7328  050476  000000                DATENV: 0               ;DATA ENVELOPE FOR TYPE OUT
7329                                                        ;MAX FOR WRITE IS 4096
7330                                                        ;MAX FOR READ IS 4128
7331  050500  000000                ZCODE:  0               ;LEADING ZEROS ENVELOPE FOR TYPE OUT
7332                                                        ;THIS IS SHUT OFF WHEN POSITION COUNTER
7333                                                        ;IN ENABLED
7334                                                        ;MAX COUNT IS 38859
7335
7336
7337
7338  050502  000000                HADTMP: 0               ;TEMPORARY HARD ERROR COUNT
7339  050504  000000                P3:     0
7340  050506  000000                P12:    0
7341  050510  000000                P22:    0
7342  050512  000000                P24:    0
7343
7344
7345
7346
7347
7348  050514                        ECTEST:
7349  050530  013701  050460                MOV     @#GECC1,R1              ;ECC1 WORD
7350  050534  013702  050462                MOV     @#GECC2,R2              ;ECC2 WORD
7351  050540  005737  050456                TST     @#ECDATA                ;IS CURRENT BIT A ONE
7352  050544  001406                        BEQ     2$                      ;BRANCH IF CURRENT DATA D=0
7353
7354                                        ;IF CARRY IS NOT ZERO THEN D=1
7355                                        ;INVERT X32 TO GIVE R0
7356
7357  050546  010103                1$:     MOV     R1,R3
7358  050550  052703  177776              BIS     #^CPIE32,R3
7359  050554  005103                        COM     R3
7360  050556  010300                        MOV     R3,R0
7361  050560  000404                        BR      3$
7362
7363                                        ;IF CARRY IS ZERO THEN D=0
7364                                        ;X32 BECOMES R0
7365  050562  010103                2$:     MOV     R1,R3
7366  050564  042703  177776              BIC     #^CPIE32,R3
7367  050570  010300                        MOV     R3,R0
7368
7369  050572  000241                3$:     CLC
7370  050574  006000                        ROR     R0
7371  050576  006000                        ROR     R0
7372  050600  005700                        TST     R0
7373  050602  001462                        BEQ     10$                     ;BRANCH IF R0=0
7374                                        ;INVERT X2
7375
7376  050604  010203                        MOV     R2,R3
7377  050606  052703  137777              BIS     #^CPIE2,R3
7378  050612  005103                        COM     R3
```

```
7379  050614  010337  050504          MOV     R3,@#P3
7380  050620  006237  050504          ASR     @#P3
7381
7382                                  ;INVERT X11
7383
7384
7385  050624  010203                  MOV     R2,R3
7386  050626  052703  177737          BIS     #^CPIE11,R3
7387  050632  005103                  COM     R3
7388  050634  010337  050506          MOV     R3,@#P12
7389  050640  006237  050506          ASR     @#P12
7390
7391                                  ;INVERT X21
7392
7393  050644  010103                  MOV     R1,R3
7394  050646  052703  173777          BIS     #^CPIE21,R3
7395  050652  005103                  COM     R3
7396  050654  010337  050510          MOV     R3,@#P22
7397  050660  006237  050510          ASR     @#P22
7398
7399                                  ;INVERT X23
7400
7401  050664  010103                  MOV     R1,R3
7402  050666  052703  176777          BIS     #^CPIE23,R3
7403  050672  005103                  COM     R3
7404  050674  010337  050512          MOV     R3,@#P24
7405  050700  006237  050512          ASR     @#P24
7406
7407                                  ;NOW THAT R0 FOR POSITION 1
7408                                  ;         P3 FOR POSITION 3
7409                                  ;        P12 FOR POSITION 12
7410                                  ;        P22 FOR POSITION 22
7411                                  ;        P24 FOR POSITION 24
7412                                  ;ARE KNOWN THE ROTATE WILL BE DONE AND
7413                                  ;THESE BITS JAMED IN
7414
7415  050704  006002                  ROR     R2
7416  050706  006001                  ROR     R1
7417  050710  053700  050504          BIS     @#P3,R0
7418  050714  053700  050506          BIS     @#P12,R0
7419  050720  042702  120020          BIC     #PIE1!PIE3!PIE12,R2
7420  050724  050002                  BIS     R0,R2
7421
7422  050726  005000                  CLR     R0
7423  050730  053700  050510          BIS     @#P22,R0
7424  050734  053700  050512          BIS     @#P24,R0
7425  050740  042701  002400          BIC     #PIE22!PIE24,R1
7426  050744  050001                  BIS     R0,R1
7427  050746  000404                  BR      12$
7428
7429                                  ;THE PROGRAM COMES HERE IF R0=0
7430                                  ;SO AFTER ROTATE R0 GETS PUT INTO POSITION 1
7431
7432  050750  006002          10$:    ROR     R2
7433  050752  006001                  ROR     R1
7434  050754  042702  100000          BIC     #PIE1,R2
```

```
7435  050760  010137  050460          12$:    MOV     R1,@#GECC1      ;SAVE ECC1
7436  050764  010237  050462                  MOV     R2,@#GECC2      ;SAVE ECC2
7437  050770  005737  050464                  TST     @#↑SECCG        ;IS HARDWARE TO BE CHECKED
7438                                                                  ;IF =1777777 TEST HARDWARE
7439                                                                  ;IF = 0 DO NOT TEST HARDWARE
7440  050774  001432                          BEQ     14$             ;BRANCH IF HARDWARE NOT TO BE CHECKED
7441
7442
7443                                  ;*CHECK HARDWARE
7444  050776  032777  000400  130134          BIT     #SW8,@SWR       ;IS SWITCH 8 SET
7445  051004  001005                          BNE     15$             ;BRANCH IF SW8 IS SET
7446  051006  032777  000100  130124          BIT     #SW6,@SWR       ;IS SWITCH 6 SET
7447  051014  001401                          BEQ     15$             ;BRANCH IF SW6 IS NOT SET
7448  051016  000421                          BR      14$             ;IF SWITCH 8 IS NOT SET AND
7449                                                                  ;SWITCH 6 IS SET THEN
7450                                                                  ;DO NOT DO COMPARES
7451  051020  010146          15$:    MOV     R1,-(SP)                ;GOOD PATTERN REGISTER
7452  051022  042716  174000          BIC     #174000,(SP)            ;GET ONLY PATTERN BITS
7453  051026  022677  143756          CMP     (SP)+,@RHEC2            ;COMPARE PATTERN REGISTER
7454  051032  001404                  BEQ     13$                     ;BRANCH IF GOOD
7455                                  ;TO SAVE TIME
7456  051034  004737  045464          JSR     PC,@#PUTREG             ;SAVE REGISTERS
7457  051040  104035                  ERROR   35                      ;PATTERN REGISTER IN 11 BITS IN ERROR
7458  051042  000407                  BR      14$                     ;BRANCH OUT
7459  051044  023777  050472  143734  13$:    CMP     @#POSITI,@RHEC1 ;COMPARE POISTION REGISTER
7460  051052  001403                  BEQ     14$                     ;BRANCH IF GOOD
7461                                  ;TO SAVE TIME
7462  051054  004737  045464          JSR     PC,@#PUTREG             ;SAVE REGISTERS
7463  051060  104035                  ERROR   35                      ;POSITION REGISTER IN ERROR
7464                                                                  ;"DATA ENVLOP" GIVES NUMBER OF CLOCK
7465                                                                  ;PULSES FROM BEGINING OF COMMAND
7466                                                                  ;THAT IS THE CLOCKS IN THE R/W DATA FIELD ENVELOPE
7467                                                                  ;
7468                                                                  ;IN A WRITE THERE ARE 10000 OCTAL CLOCKS
7469                                                                  ;IN A READ THERE ARE 10040 OCTAL CLOCKS
7470                                                                  ;
7471                                                                  ;
7472                                                                  ;"N-CODE ZEROS" GIVE THE NUMBER OF CLOCKS
7473                                                                  ; GIVEN FOR THE LEADING ZEROS FIELD
7474                                                                  ;MAX COUNT IS 113713 OCTAL
7475                                                                  ;
7476                                                                  ;"GOOD POSITION" GIVES NUMBER OF CLOCKS
7477                                                                  ;GIVEN AFTER LEADING ZEROS WHICH IS FOR THE DATA
7478                                                                  ;FIELD
7479                                                                  ;MAX COUNT IS 10040 OR 10041 OCTAL
7480
7481
7482  051062                  14$:
7483  051076  000207                  RTS     PC
7484                                  .SBTTL  ECC GENERATION CONTROL ROUTINE
7485
7486                          ;*THIS SUBROUTINE WILL CONTROL THE ECC GENERATION ROUTINE
7487                          ;*FOR ERROR CORRECTION PROCESS
7488                          ;*CALL  JSR,    PC,@#ECORR
7489                          ;*      XP      ;EXPECTED POSITION REGISTER WHEN CORRECTION IS COMPLETE
7490
```

```
                                ERPOS:  0                       ;POSITION REG. WHEN CORRECTION IS COMPLETE

                                ECORR:  MOV     R0,@#PCJSR      ;SAVE PC OF JSR + 4
                                        SUB     #4,@#PCJSR      ;SAVE PC OF JSR
                                        MOV     (R0)+,@#ERPOS   ;GET POSITION REG. WHEN CORRECTION IS COMPLETE
                                        MOV     @#RHMR,R1       ;MAINTENANCE REGISTER
                                        MOV     #DMD,@R1        ;SET DIAGNOSTIC MODE BIT
                                        CLR     @#ECDATA        ;ECC DATA IS ZERO

                                1$:     TST     @#POSITI        ;IS  SOFTWARE POSITION NON ZERO
                                        BNE     2$              ;BRANCH IF N-CODE S COMPLETE
                                        DEC     @#NCOUNT        ;DECREMENT N-CODE
                                        BNE     6$              ;BRANCH IF N-CODE IS NOT COMPLETE
                                        BR      2$              ;BRANCH  AS N-CODE IS COMPLETE
                                6$:     INC     @#ZCODE         ;INCREMENT CLOCKS GIVEN FOR LEADING ZEROS
                                        BR      3$              ;BRANCH AS N-CODE IS NOT COMPLETE
                                                                ;GO TO GIVE CLOCK AND TEST ECC
                                2$:     INC     @#POSITI        ;INCREMENT SOFTWARE POSITION
                                        CMP     @#ERPOS,@#POSITI;HAVE ENOUGH CLOCKS BEEN GIVEN TO DETECT ERROR
                                        BHIS    3$              ;BRANCH IF MORE CLOCKS TO BE GIVEN
                                        CMP     @#HADTMP,@#POSITI;HAVE ENOUGH CLOCKS BEEN GIVEN FOR HARD ERROR
                                                                ;THAT IS HAVE 4128 MORE CLOCKS BEEN GIVEN
                                        BEQ     5$              ;BRANCH IF YES
                                        BIT     #ZER,@R1        ;CHECK ZERO DETECT BIT IN RHMR
                                        BNE     4$              ;BRANCH IS ZER SET
                                        ;TO SAVE TIME
                                        JSR     PC,@#PUTREG     ;SAVE REGISTERS
                                        ERROR   34              ;ZERO DETECT BIT NOT HIGH
                                                                ;WHEN 21 BITS IN ECC 32 BIT REGISTER IS  0

                                3$:     BIS     #MCLK,@R1       ;SET CLOCK
                                        BIC     #MCLK,@R1       ;CLEAR CLOCK
                                        JSR     PC,@#ECTEST     ;GO TO GENERATE AND TEST ECC
                                        BR      1$              ;CONTINUE

                                ;THIS EXTRA CLOCK IS TO BRING ECH HIGH
                                ;AFTER THIS CLOCK POSITION REGISTER MAY BE 10040 OR 10041 OCTAL

                                5$:     BIS     #MCLK,@R1       ;SET CLOCK
                                        BIC     #MCLK,@R1       ;CLEAR CLOCK

                                4$:
                                        RTS     R0
```

```
                                        .SBTTL  SOFTWARE DISK DATA ECC GEN. ROUTINE

                                ;*THIS SUBROUTINE GENERATES THE ECC FOR WHAT IS ON DISK AND INSERTS THEM
                                ;*ON LOCATIONS "DISK+1000" AND "DISK+1002"

051254                          FILLEC:
051270  005037  050472                  CLR     @#POSITI        ;CLEAR POSITION
051274  005037  050460                  CLR     @#GECC1         ;CLEAR GECC1
051300  005037  050462                  CLR     @#GECC2         ;CLEAR
051304  012701  054566                  MOV     #DISK,R1        ;POINTER TO DATA FOR ECC GENERATION
051310  012702  000400                  MOV     #256.,R2        ;COUNTER FOR NUMBER OF DATA WORDS
051314  012703  000020          9$:     MOV     #16.,R3         ;COUNTER FOR NUMBER OF BITS PER WORD
051320  012104                  10$:    MOV     (R1)+,R4        ;DATA IN R4
051322  006004                          ROR     R4              ;GET ONE DATA BIT IN CARRY
051324  103004                          BCC     11$             ;BRANCH IF DATA BIT IS ZERO
051326  012737  177777  050456          MOV     #-1,@#ECDATA    ;ECC DATA BIT IS A ONE
051334  000402                          BR      12$             ;BRANCH TO GENERATE ECC
051336  005037  050456          11$:    CLR     @#ECDATA        ;ECC DATA BIT IS A ZERO
051342  004737  050514          12$:    JSR     PC,@#ECTEST     ;GO TO GENERATE ECC
051346  005303                          DEC     R3              ;DECREMENT BIT COUNT
051350  001364                          BNE     10$             ;BRANCH IF 16 BITS NOT DONE
051352  005302                          DEC     R2              ;DECREMENT WORD COUNT
051354  001357                          BNE     9$              ;BRANCH IF 256 WORDS NOT DONE
051356  013737  050460  055566          MOV     @#GECC1,@#DISK+<256.*2> ;INSERT ECC1 ON DISK
051364  013737  050462  055570          MOV     @#GECC2,@#DISK+<257.*2> ;INSERT ECC2 ON DISK
051406  000207                          RTS     PC
```

```
7567
7568                                    .SBTTL  RH BASE ADDRESS CHANGE ROUTINE
7569
7570                            ;*      THIS ROUTINE WILL ALLOW THE CHANGE OF THE BASE
7571                            ;*      ADDRESS FROM 176700 TO ANY TYPED VALUE
7572
7573  051410                    BASECH:
7574  051470  013746  014756           MOV     @#RHCS1,-(SP)   ;GET READY TO TYPE OLD BASE
7575  051474  104402                   TYPOC
7576  051556  004737  060346           JSR     PC,@#$TKINT     ;INITIALIZE THE TTY KEYBOARD
7577  051562  104412                   RDOCT
7578  051564  012700  014746           MOV     #RHDB,R0        ;GET STARTING ADDRESS OF REGISTERS
7579  051570  012701  000026           MOV     #22.,R1         ;NUMBER OF REGISTERS
7580  051574  012737  052374  000004   MOV     #ADTIMO,@#4     ;SET UP TO TEST THIS ADDRESS
7581  051602  021637  014756           CMP     @SP,@#RHCS1     ;NEW CSR?
7582  051606  001431                   BEQ     1$              ;NO, THE OLD ONE WAS RETYPED
7583  051610  005776  000000           TST     @0(SP)          ;ACCESS THE NEW ADDRESS
7584  051614  163716  014756           SUB     @#RHCS1,@SP     ;GET THE ADDRESS OFFSET
7585  051620  061620                2$: ADD     @SP,(R0)+       ;AND PLUG IT IN
7586  051622  005301                   DEC     R1              ;ONE LESS ADDRESS TO CHANGE
7587  051624  001375                   BNE     2$              ;BUT DO SOME MORE
7588  051672  013746  014744        1$: MOV     @#RPVEC,-(SP)   ;GET READY TO TYPE OLD VECTOR ADDRESS
7589  051676  104402                   TYPOC
7590  052004  104412                   RDOCT
7591  052006  012637  014744           MOV     (SP)+,@#RPVEC   ;SETUP VECTOR ADDRESS
7592  052054  013746  014756           MOV     @#RHCS1,-(SP)
7593  052060  104402                   TYPOC
7594  052124  013746  014744           MOV     @#RPVEC,-(SP)
7595  052130  104402                   TYPOC
7596  052350  012746  000200           MOV     #RA,-(SP)
7597  052354  104402                   TYPOC
7598  052370  000137  017356           JMP     @#BEGIN         ;ALL DONE, NOW START OVER!
7599  052374  022626                ADTIMO: CMP     (SP)+,(SP)+
7600  052454  000137  051410           JMP     @#BASECH
7601
7602
7603
7604                            ;*THIS IS A LITTLE ROUTINE THAT TESTS NED BIT 11 IN RHCS2
7605                            ;*THIS LOOPS HERE FOR EVER
7606                            ;*TO BE USED ONLY IF DRIVES PRESENT LOOKING AT NED DOES NOT AGREE
7607                            ;*WITH WHAT IS REALY THERE
7608  052460  000000            ERUNIT: 0                       ;UNIT UNDER MANUAL TEST
7609  052462  004737  045764   ERSTART:JSR     PC,@#CLDISK     ;SET GENERAL REG.
7610  052466  013712  052460           MOV     @#ERUNIT,@R2    ;SELECT UNIT
7611  052472  005714            1$:    TST     @R4             ;TEST RHER1
7612  052474  032712  010000           BIT     #NED,@R2        ;TEST NED
7613  052500  001401                   BEQ     2$              ;BRANCH IF GOOD
7614  052502  000773                   BR      1$              ;NED NOT SET
7615  052504  000772            2$:    BR      1$              ;NED SET
```

```
                        .SBTTL  DISK SIMULATION
;*IN A WRITE HEADER AND DATA COMMAND FILL THE FOLLOWING
;*WCLY=WITH CYLINDER TO BE ON DISK
;*WSECTR=WITH SECTOR AND TRACK TO BE ON DISK
;*WKEY1= WITH KEY1 TO BE ON DISK
;*WKEY2= WITH KEY2 TO BE ON DISK
;*FNWORD= NO OF DATA WORDS TO BE WRITTEN ON DISK
;*THE COMMAND THEN IS   JSR PC,COMWHD
;*
;*
;*
;*IN A WRITE DATA COMMAND FILL THE FOLLOWING
;*CYL=WITH CYLINDER TO BE FOUND ON DISK
;*SECOTR= WITH SECTOR AND TRACK TO BE FOUND ON DISK
;*KEY1= WITH KEY1 TO BE FOUND ON DISK
;*KEY2= WITH KEY2 TO BE FOUND ON DISK
;*X= 1  MUST BE ONE
;*NOWORD= WITH NUMBER OF DATA WORDS TO BE WRITTEN
;*THE COMMAND THEN IS JSR PC,COMHD
;*
;*
;*
;*
;*
;*IN A READ HEADER AND DATA COMMAND FILL THE FOLLOWING
;*CYL= WITH CYLINDER TO BE FOUND ON DISK
;*SECOTR= WITH SECTOR AND TRACK TO BE FOUND ON DISK
;*KEY1= WITH KEY1 TO BE FOUND ON DISK
;*KEY2=WITH KEY2 TO BE FOUND ON DISK
;*DAWORD= WITH NUMBER OF WORDS TO BE FOUND ON DISK
;*;X=0 MUST BE ZERO
;*THE COMMAND THEN IS JSR PC,COMHD
;*
;*
;*
;*
;*
;*
;*
;*IN A READ DATA COMMAND FILL THE FOLLOWING
;*CYL= WITH CYLINDER TO BE FOUND ON DISK
;*SECOTR= WITH SECTOR AND TRACK TO BE FOUND ON DISK
;*KEY1= WITH KEY1 TO BE FOUND ON DISK
;*KEY2=WITH KEY2 TO BE FOUND ON DISK
;*DAWORD= WITH NUMBER OF WORDS TO BE FOUND ON DISK
;*;X=0 MUST BE ZERO
;*THE COMMAND THEN IS JSR PC,COMHD
;*
```

```
                                        ;*WRITE DATA COMMAND
                                        ;*OR READ COMMAND, I.E DATA ONLY, OR HEADER AND DATA



                                        ;*THIS SUBROUTINE IS THE FIRST IN A SERIES OF NESTED SUBROUTINES

                                        ;*IT ISSUES DIAGNOSTIC MODE, AND EXTRA DIAGNOSTIC INDEX, AND THE
                                        ;*'GO' BIT

                                        ;*IT THEN CALLS THREE OTHER SUBROUTINES, WHICH IN TURN CALL
                                        ;*OTHER SUBROUTINES.  THE SUBROUTINES CALLED HERE ARE:

                                        ;*      SEARCH          ;ISSUES SECTOR CLOCKS TO SET SECTOR FOUND FLOP
                                        ;*      RDHEAD          ;READS THE SECTOR HEADER
                                        ;*      WRDATA          ;WRITES THE SECTOR DATA (WRITE OPERATION)
                                        ;*      REDATA          ;READS THE SECTOR DATA (READ OPERATION)


052506  000000                  RUNCTR: .WORD   0
052510  011637  015132          COMHD:  MOV     (SP),@#PCJSR    ;SAVE PC OF JSR + 4
052514  162737  000004  015132          SUB     #4,@#PCJSR      ;SAVE PC OF JSR

052536  012777  000001  142232          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
052544  052777  000004  142224          BIS     #MINX,@RHMR     ;SET DIAGNOSTIC INDEX
052552  042777  000004  142216          BIC     #MINX,@RHMR     ;CLEAR DIAGNOSTIC INDEX
052560  052777  000001  142170          BIS     #GO,@RHCS1      ;ISSUE 'GO' BIT & STALL 'TILL 'RUN'
                                                                ;FUNCTION CODE WAS ISSUED BY THE TEST
052566  012737  000113  052506  RUNWAT: MOV     #75.,@#RUNCTR   ;LOAD STALL COUNT = APPROX. 450US FOR 11/50 CPU
052574  005337  052506          1$:     DEC     @#RUNCTR        ;COUNT DOWN ONE
052600  001375                          BNE     1$              ;CONTINUE UNTIL = 0

052602  013746  052652                  MOV     SECOTR, -(SP)   ;GET DESIRED SECTOR/TRACK
052606  042716  177740                  BIC     #177740,(SP)    ;MAKE ONLY SECTOR
052612  012637  052622                  MOV     (SP)+,@#TRK     ;SAVE SECTOR
052616  004137  056716                  JSR     R1,@#SEARCH     ;ISSUE SECTOR CLOCKS <---------------->

052622  000000                  TRK:    .WORD   0
052624  012701  000240                  MOV     #+NOP,R1        ;GOING TO MOVE NOPS
052630  010137  052662                  MOV     R1,@#SSYN       ;NOP INTO SSYN
052634  010137  052664                  MOV     R1,@#HEDGAP     ;NOP INTO HEDGAP
052640  010137  052666                  MOV     R1,@#HEDSYN     ;NOP INTO HEDSYN
052644  004137  052772                  JSR     R1,@#RDHEAD     ;READ THE HEADER <------------->

052650  000000                  CYL:    .WORD   0               ;CYLINDER ADDRESS
```

```
7725  052652  000000                SECOTR: .WORD   0               ;SECTOR/TRACK ADDRESS
7726  052654  000000                KEY1:   .WORD   0               ;KEY1 WORD
7727  052656  000000                KEY2:   .WORD   0               ;KEY2 WORD
7728  052660  000000                X:      .WORD   0               ;X=1 WRITE COMMAND
7729                                                                ;X=0 READ COMMAND
7730
7731
7732                                      ;DUMMY ERROR CALL LOCATIONS FOR THE READ HEADER OPERATION
7733
7734  052662  000240                SSYN:   NOP                     ;IF "ERROR 2" INSERTED BY RDHEAD
7735                                                                ;SUBROUTINE, THEN THE FIRST SYNC
7736                                                                ;IS NOT DETECTED.  NO BAD DATA
7737                                                                ;IS GIVEN BECAUSE SYNC=144000
7738                                                                ;CANNOT BE READ.  WORD NUMBER
7739                                                                ;IS "1" BECAUSE THIS IS THE FIRST
7740                                                                ;WORD TESTED.
7741
7742  052664  000240                HEDGAP: NOP                     ;IF "ERROR 3" INSERTED BY
7743                                                                ;RDHEAD SUBROUTINE, THEN THE
7744                                                                ;HEADER GAP 0'S WERE NOT
7745                                                                ;WRITTEN RIGHT.
7746
7747                                                                ;IF "WORD NO" CONTAINS, SAY
7748                                                                ;3(8), THEN IT IS THE THIRD
7749                                                                ;WORD OF A 5 WORD HEADER
7750                                                                ;GAP THAT IS WRONG.
7751
7752                                                                ;"BAD DATA" CONTAINS WHAT IS
7753                                                                ;GOING ON THE DISK.
7754
7755  052666  000240                HEDSYN: NOP
7756                                                                ;IF "ERROR 3" INSERTED BY RDHEAD
7757                                                                ;SUBROUTINE, THEN THE HEADER SYNC
7758                                                                ;GENERATED BY DCL IS WRONG,
7759                                                                ;OR THE LAST BYTE
7760                                                                ;OF THE HEADER GAP 0'S IS WRONG.
7761
7762                                                                ;IN EITHER CASE WORD NO=6,
7763                                                                ;RIGHT BYTE IS HEADER 0,
7764                                                                ;LEFT BYTE IS SYNC.
7765
7766                                                                ;"BAD DATA" HAS WHAT IS GOING
7767                                                                ;ON DISK.
7768
7769  052670  005737  015124        TST     @#ERFLG$                ;WERE ANY ERRORS DETECTED ?
7770  052674  001017                BNE     OUT                     ;IF YES, EXIT ---------------------›
7771
7772  052676  005737  052660        TST     @#X                     ;IS IT A DATA WRITE OPERATION ?
7773  052702  001410                BEQ     DAREAD                  ;NO...THEN DO A DATA READ
7774  052704  005737  052764        TST     @#NOSYNC                ;IS THIS FORCED HEADER ERROR COMMAND ?
7775                                                                ;IF YES  NOSYNC=-1 THEN WRITE OR READ
7776                                                                ;IS SHUT OFF, SO BRANCH OUT
7777                                                                ;IF  NOSYNC=0 THEN CONTINUE
7778  052710  001011                BNE     OUT                     ;EXIT IF SET ----------------------›
7779
7780  052712  004137  054236        JSR     R1,@#WRDATA             ;WRITE DATA ‹----------------------›
```

```
 7781  052716  000000                NOWORD: .WORD   0               ;NO OF WORDS TO BE WRITTEN
 7782  052720  000000                Y:      .WORD   0               ;
 7783  052722  000404                        BR      OUT             ;EXIT ------------------------------>
 7784
 7785  052724  004137  057172        DAREAD: JSR     R1,@#REDATA     ;READ DATA <------------------------>
 7786  052730  000000                DAWORD: .WORD   0               ;NO OF WORDS TO BE READ
 7787  052732  000000                        .WORD   0
 7788
 7789  052734                        OUT:
 7790  052750  000207                        RTS     PC              ;EXIT
 7791
```

```
                                        ;*THE DISK SECTOR IS DEVIDED AS FOLLOWS
                                        ;*19 WORDS OF 0, ONE WORD 144000
                                        ;*THESE MAKE 39 BYTES FOR SECTOR GAP AND ONE SYNC. BYTE

7800   052752  014400                  RSYNC:  14400
7801   052754  000000                  RCYL:   0
7802   052756  000000                  RSETR:  0
7803   052760  000000                  RKEY1:  0
7804   052762  000000                  RKEY2:  0

                                        ;*5 WORDS OF 0'S, ONE WORD 144000
                                        ;*THESE MAKE 11 BYTES FOR HEADER GAP AND ONE SYNC. BYTE
                                        ;*THESE ARE DCL GENERATED


                                        ;*THERE ARE 256 WORDS OF DATA
                                        ;*THERE ARE 2 WORDS FOR ECC GENERATED BY DCL
                                        ;*15 WORDS OF 0'S FOR DATA GAP AND TOLERANCE GAP
```

```
                                ;*READ DISK HEADER
                                
                                
                                
                                
                                
                                
052764  000000          NOSYNC: 0                       ;FORCED HEADER ERROR = -1
                                                        ;NORMAL = 0
052766  000000          TY:     0                       ;ERROR TYPE NO.
052770  000000          ERWORD: 0                       ;ERROR WORD NO.
                                
                                
                                
                                
052772  012137  052754  RDHEAD: MOV     (R1)+,@#RCYL    ;STORE CYLINDER ADDRESS
052776  012137  052756          MOV     (R1)+,@#RSETR   ;STORE SECTOR AND TRACK ADDRESS
053002  012137  052760          MOV     (R1)+,@#RKEY1   ;STORE KEY1
053006  012137  052762          MOV     (R1)+,@#RKEY2   ;STORE KEY2
053012  012137  053562          MOV     (R1)+,@#COMPA   ;STORE COMPARE OR NOT
                                
053020  013700  014776          MOV     @#RHMR,R0       ;R0 CONTAINS MAINTANENCE REG.
053024  012705  000002          MOV     #2,R5           ;R5 IS A COUNTER FOR WORDS
053030  012710  000001          MOV     #DMD,@R0        ;SET DIAG. MODE
053034  052710  000010          BIS     #MSTCK,@R0      ;SET SECTOR CLOCK FOR FIRST WORD
053040  052710  000002          BIS     #MCLK,@R0       ;SET MAINT.CLOCK FOR FIRST WORD
053044  042710  000012          BIC     #MSTCK!MCLK,@R0 ;RESET THEM
                                
053050  000404                  BR      2$              ;DON'T GIVE SECTOR CLOCK FIRST TIME
053052  012710  000013  1$:     MOV     #MSTCK!MCLK!DMD,@R0;SET SECTOR, CLOCK, DIAG. MODE, RESET INDEX
053056  042710  000012          BIC     #MSTCK!MCLK,@R0 ;RESET SECTOR & MAINT.CLOCK
                                
053062  012702  000007  2$:     MOV     #7,R2           ;LOAD BYTE COUNTER
053066  052710  000002  3$:     BIS     #MCLK,@R0       ;SET MAINT. CLOCK
053072  042710  000002          BIC     #MCLK,@R0       ;RESET IT
053076  005302                  DEC     R2              ;BYTE COUNTER
053100  001372                  BNE     3$              ;CONTINUE IF BYTE NOT COMPLETE
053102  005305                  DEC     R5              ;WORD COUNTER
053104  001362                  BNE     1$              ;CONTINUE IF WORD NOT COMPLETE
                                
053106  012702  000022          MOV     #18.,R2         ;LOAD NO OF WORDS OF ZEROS
053112  005037  053560  4$:     CLR     @#WORD          ;DO 0'S
053116  004737  053564          JSR     PC,@#READ       ;READ 0'S <------------------->
053122  005302                  DEC     R2              ;COUNT DOWN WORDS
053124  001372                  BNE     4$              ;CONTINUE
                                
053126  013737  052752  053560  MOV     @#RSYNC,@#WORD  ;SYNC. WORD
053134  004737  053564          JSR     PC,@#READ       ;READ IT <---------------------->
053140  032710  001000          BIT     #DTSY,  @R0     ;SYNC. BYTE DETECTED?
053144  001012                  BNE     5$              ;CONTINUE IF SYNC DETECTED
053146  012737  000001  052770  MOV     #1,@#ERWORD     ;ERROR WORD NO
053154  013737  052752  001124  MOV     @#RSYNC,@#$GDDAT;SYNC WORD
053162  012737  104002  052662  MOV     #104002,@#SSYN  ;INSERT "ERROR 2" IN SSYN
053170  000571                  BR      13$             ;BRANCH OUT ------------------->
                                
053172  013737  052754  053560  5$:     MOV     @#RCYL, @#WORD  ;SETUP CYLINDER
053200  004737  053564          JSR     PC,@#READ       ;READ IT <---------------------->
```

```
7878  053204  013737  052756  053560          MOV     @#RSETR,@#WORD  ;SETUP SECTOR/TRACK
7879  053212  004737  053564                  JSR     PC,@#READ       ;READ THEM <--------------------
7880  053216  013737  052760  053560          MOV     @#RKEY1,@#WORD  ;SETUP KEY1
7881  053224  004737  053564                  JSR     PC,@#READ       ;READ IT <----------------------
7882  053230  013737  052762  053560          MOV     @#RKEY2,@#WORD  ;SETUP KEY2
7883  053236  004737  053564                  JSR     PC,@#READ       ;READ IT <----------------------
7884  053242  013737  054550  053560          MOV     @#WCRC,@#WORD   ;SETUP CRC
7885  053250  004737  053564                  JSR     PC,@#READ       ;READ IT <----------------------
7886
7887  053254  005737  015144                  TST     @#TESDTE        ;IS THIS A DRIVE TIMING ERROR ?
7888  053260  001135                          BNE     13$             ;BRANCH OUT IF YES ------------>
7889  053262  005737  053562                  TST     @#COMPA         ;IS THIS A READ OR WRITE COMMAND ?
7890  053266  001472                          BEQ     11$             ;DO READ IF = 0
7891
7892                                  ;*CONTINUE WITH DIAGNOSTIC WRITE COMMAND
7893
7894  053270  012705  054552                  MOV     #HEGAP, R5      ;POINTER FOR HEADER GAP
7895  053274  012702  000005                  MOV     #5,R2           ;NO OF WORDS OF ZEROS
7896  053300  012737  000006  052770  6$:     MOV     #6,@#ERWORD     ;ERROR WORD NO SET
7897  053306  004737  054016                  JSR     PC,@#WRITE      ;FOR HEADER GAP
7898  053312  005737  054014                  TST     @#WWORD         ;TEST WRITTEN WORD
7899  053316  001413                          BEQ     7$              ;CONTINUE IF GOOD, THAT IS = 0
7900  053320  160237  052770                  SUB     R2,@#ERWORD     ;WORD NO IN ERROR
7901  053324  005037  001124                  CLR     @#$GDDAT        ;GOOD WORD SHOULD BE 0
7902  053330  013737  054014  001126          MOV     @#WWORD,$BDDAT  ;BAD DATA
7903  053336  012737  104003  052664          MOV     #104003,@#HEDGAP;"ERROR 2" GOES IN HEDGAP
7904  053344  000503                          BR      13$             ;BRANCH OUT ---------------------->
7905
7906  053346  013725  054014          7$:     MOV     @#WWORD,(R5)+   ;SAVE HEADER GAP
7907  053352  005302                          DEC     R2
7908  053354  001351                          BNE     6$
7909  053356  004737  054016                  JSR     PC,@#WRITE      ;WRITE HEADER (DATA) GAP SYNC
7910  053362  023737  052752  054014          CMP     @#RSYNC,@#WWORD
7911  053370  001426                          BEQ     10$
7912  053372  005737  052764                  TST     @#NOSYNC        ;IS THIS FORCED HEADER ERROR COMMAND ?
7913                                                                  ;IF YES  NOSYNC=-1 THEN WRITE OR READ
7914                                                                  ;IS SHUT OFF SO BRANCH OUT
7915                                                                  ;IF NO NOSYNC=0 THEN CONTINUE
7916  053376  001406                          BEQ     14$             ;PRINT IT IF TRUE ERROR
7917
7918  053400  005737  054014                  TST     @#WWORD         ;IS IT = 0 ?
7919  053404  001420                          BEQ     10$             ;CONTINUE IF GOOD
7920  053406  005037  001124                  CLR     @#$GDDAT        ;IT SHOULD BE ZERO
7921  053412  000403                          BR      15$             ;BRANCH TO TYPE ERROR
7922  053414  013737  052752  001124  14$:     MOV     @#RSYNC,@#$GDDAT;GOOD DATA
7923  053422  013737  054014  001126  15$:     MOV     @#WWORD,@#$BDDAT;BAD DATA
7924  053430  012737  000006  052770          MOV     #6,@#ERWORD
7925  053436  012737  104003  052666          MOV     #104003,@#HEDSYN
7926  053444  000443                          BR      13$             ;BRANCH OUT ---------------------->
7927
7928  053446  013725  054014          10$:    MOV     @#WWORD,(R5)+   ;SAVE DATA SYNC.
7929  053452  000440                          BR      13$             ;EXIT ---------------------------->
7930
7931                                  ;*READ COMMAND START FROM HERE
7932
7933  053454  012702  000005          11$:    MOV     #5,R2
```

```
7934   053460  005037  053560          12$:    CLR     WORD
7935   053464  004737  053564                  JSR     PC,@#READ       ;READ HEADER GAP <---------------
7936   053470  005302                          DEC     R2              ;ARE 5 HEADER GAP ZEROS COMPLETE ?
7937   053472  001372                          BNE     12$             ;IF NOT CONTINUE
7938   053474  013737  052752  053560          MOV     @#RSYNC,@#WORD  ;SYNC WORD
7939   053502  004737  053564                  JSR     PC,@#READ       ;READ HEADER (DATA) SYNC)
7940   053506  005737  052764                  TST     @#NOSYNC        ;FORCED SYNC ERROR ?
7941   053512  001404                          BEQ     16$             ;IF NOT ERROR COMMAND CONTINUE
7942   053514  032710  001000                  BIT     #DTSY,@RO       ;SYNC. DETECTED
7943   053520  001415                          BEQ     13$             ;IF ZERO BRANCH OUT ------------>
7944   053522  000403                          BR      17$             ;IF NOT ZERO BRANCH TO ERROR
7945
7946   053524  032710  001000          16$:    BIT     #DTSY,  @RO     ;SYNC. DETECTED ?
7947   053530  001011                          BNE     13$             ;EXIT IF YES ------------------->
7948   053532  012737  000006  052770  17$:    MOV     #6,@#ERWORD     ;ERROR WORD NO.
7949   053540  013737  052752  001124          MOV     @#RSYNC,@#$GDDAT;SYNC WORD
7950   053546  012737  104002  052666          MOV     #104002,@#HEDSYN;MOVE "ERROR 2"
7951   053554                          13$:
7952   053556  000201                          RTS     R1              ;EXIT ------------------------>
7953
7954
7955
7956
7957
7958
7959
7960
```

```
                                        ;*READ ONE WORD IN "WORD"

053560  000000                  WORD:   0
053562  000000                  COMPA:  0

053564                          READ:
053566  012705  000002                  MOV     #2,R5           ;WORD COUNTER
053572  012710  000001                  MOV     #DMD,@R0        ;SET DIAG. MODE
053576  006037  053560                  ROR     @#WORD          ;CHECKING IF THERE IS A ONE
053602  103002                          BCC     1$              ;IF NO ONE BRANCH
053604  052710  000020                  BIS     #MRD,@R0        ;SET BIT 4 IF DATA HAS ONE
053610  012702  000007          1$:     MOV     #7,R2           ;BYTE COUNTER
053614  052710  000012                  BIS     #MSTCK!MCLK,@R0 ;SET CLOCK,DATA IF ANY, SECTOR
053620  005737  050464                  TST     @#TSECCG        ;IS THIS BIT TO GENERATE AND TEST ECC ?
053624  001411                          BEQ     6$              ;BRANCH IF NO
053626  032710  000020                  BIT     #MRD,@R0        ;IS DATA BIT A ONE ?
053632  001404                          BEQ     5$              ;BRANCH IF DATA BIT IS 0
053634  012737  177777  050456          MOV     #-1,@#ECDATA    ;ECC DATA BIT IS A ONE
053642  000402                          BR      6$              ;BRANCH
053644  005037  050456          5$:     CLR     @#ECDATA        ;ECC DATA BIT IS A 0
053650  012746  000001          6$:     MOV     #DMD,-(SP)      ;KEEP ONLY DIAG. MODE
053654  006037  053560                  ROR     @#WORD          ;CHECKING IF THERE IS A ONE
053660  103002                          BCC     2$              ;IF NO ONE BRANCH
053662  012716  000021                  MOV     #MRD!DMD,(SP)   ;KEEP DATA AND DIAG. MODE
053666  012610                  2$:     MOV     (SP)+,@R0       ;PUT IN DATA,RESET CLOCK, SECTOR
053670  005737  050464                  TST     @#TSECCG        ;IS ECC TO BE GENERATED FOR THIS BIT
053674  001404                          BEQ     3$              ;BRANCH IF NO
053676  005237  050476                  INC     @#DATENV        ;NUMBER OF CLOCKS GIVEN FOR DATA ENVELOPE
053702  004737  050514                  JSR     PC,@#ECTEST     ;GO TO GENERATE AND TEST ECC
053706  052710  000002          3$:     BIS     #MCLK,@R0       ;SET CLOCK
053712  005737  050464                  TST     @#TSECCG        ;IS THIS BIT TO GENERATE ECC
053716  001411                          BEQ     8$              ;BRANCH IF NO
053720  032710  000020                  BIT     #MRD,@R0        ;IS DATA BIT A ONE
053724  001404                          BEQ     7$              ;BRANCH IF DATA BIT IS = 0
053726  012737  177777  050456          MOV     #-1,@#ECDATA    ;ECC DATA BIT IS A ONE
053734  000402                          BR      8$              ;BRANCH
053736  005037  050456          7$:     CLR     @#ECDATA        ;ECC DATA BIT IS = 0
053742  012746  000001          8$:     MOV     #DMD,-(SP)      ;KEEP DIAG. MODE
053746  006037  053560                  ROR     @#WORD          ;CHECKING IF THERE IS A ONE
053752  103002                          BCC     4$              ;BRANCH IF NO ONE
053754  012716  000021                  MOV     #MRD!DMD,(SP)   ;KEEP DIAG. MODE AND DATA
053760  012610                  4$:     MOV     (SP)+,@R0       ;SET DATA, DIAG. MODE, CLEAR CLOCK
053762  005737  050464                  TST     @#TSECCG        ;IS THIS BIT TO GENERATE ECC
053766  001404                          BEQ     9$              ;BRANCH IF NO
053770  005237  050476                  INC     @#DATENV        ;NUMBER OF CLOCKS FOR DATA ENVELOPE
053774  004737  050514                  JSR     PC,@#ECTEST     ;GO TO GENERATE AND TEST ECC

054000  005302                  9$:     DEC     R2              ;BYTE COUNTER
054002  001341                          BNE     3$              ;CONTINUE IF ONE BYTE NOT COMPLETE
054004  005305                          DEC     R5              ;WORD COUNTER
```

```
8017  054006  001300                    BNE     1$              ;CONTINUE IF ONE WORD NOT COMPLETE
8018  054012  000207                    RTS     PC              ;EXIT ------------------------------>
8019
8020
8021
8022
8023
8024
```

```
8025                                    ;*WRITE ONE WORD WHICH COMES BACK IN "WWORD"
8026
8027
8028
8029
8030
8031
8032  054014  000000                    WWORD:  0
8033
8034
8035
8036
8037  054016                            WRITE:
8038  054026  012705  000002                    MOV     #2,R5           ;WORD COUNTER
8039  054032  012710  000001                    MOV     #1,@R0          ;SET DIAG. MODE
8040  054036  012702  000007            1$:     MOV     #7,R2           ;BYTE COUNTER
8041  054042  012710  000013                    MOV     #MSTCK!MCLK!DMD,@R0;SET SECTOR & MAINT. CLOCKS
8042  054046  032710  000040                    BIT     #MWR,@R0        ;CHECK WRITEBIT IN MAINT. REG.
8043  054052  001406                            BEQ     2$              ;BRANCH IF ZERO
8044  054054  012737  177777  050456            MOV     #-1,@#ECDATA    ;ECC DATA BIT IS A ONE
8045  054062  000261                            SEC                     ;SET CARRY
8046  054064  006003                            ROR     R3              ;MOVE 1 FORWARD
8047  054066  000404                            BR      3$
8048  054070  005037  050456            2$:     CLR     @#ECDATA        ;ECC DATA BIT IS = 0
8049  054074  000241                            CLC                     ;CLEAR CARRY
8050  054076  006003                            ROR     R3              ;MOVE 0 FOR WWORD
8051  054100  012710  000001            3$:     MOV     #DMD,@R0        ;CLEAR SECTOR AND CLOCK
8052  054104  005737  050464                    TST     @#TSECCG        ;IS THIS BIT TO GENERATE ECC ?
8053  054110  001404                            BEQ     4$              ;BRANCH IF NO
8054  054112  005237  050476                    INC     @#DATENV        ;NUMBER OF CLOCKS FOR DATA ENVELOPE
8055  054116  004737  050514                    JSR     PC,@#ECTEST     ;GO TO GENERATE AND TEST ECC <-------->
8056
8057  054122  052710  000002            4$:     BIS     #MCLK,@R0       ;SET CLOCK
8058  054126  032710  000040                    BIT     #MWR,@R0        ;CHECK WRITE BIT IN MAINT. REG.
8059  054132  001406                            BEQ     5$              ;BRANCH IF ZERO
8060  054134  012737  177777  050456            MOV     #-1,@#ECDATA    ;ECC DATA BIT IS A ONE
8061  054142  000261                            SEC                     ;SET CARRY
8062  054144  006003                            ROR     R3              ;MOVE 1 FOR WWORD
8063  054146  000404                            BR      6$
8064  054150  005037  050456            5$:     CLR     @#ECDATA        ;ECC DATA BIT IS ZERO
8065  054154  000241                            CLC                     ;CLEAR CARRY
8066  054156  006003                            ROR     R3              ;MOVE 0 FOR WWORD
8067  054160  012710  000001            6$:     MOV     #DMD,@R0        ;CLEAR CLOCK
8068  054164  005737  050464                    TST     @#TSECCG        ;IS THIS BIT TO GENERATE ECC ?
8069  054170  001404                            BEQ     7$              ;BRANCH IF NO
8070  054172  005237  050476                    INC     @#DATENV        ;NUMBER OF CLOCKS FOR DATA ENVELOPE
8071  054176  004737  050514                    JSR     PC,@#ECTEST     ;GO TO GENERATE AND TEST ECC <-------->
8072
8073  054202  005302                    7$:     DEC     R2              ;COUNT FOR BYTE END
8074  054204  001346                            BNE     4$              ;IF NOT BYTE END CONTINUE
8075  054206  005305                            DEC     R5              ;COUNT FOR WORD END
8076  054210  001312                            BNE     1$              ;IF NOT WORD END CONTINUE
8077  054212  010337  054014                    MOV     R3,@#WWORD      ;STORE THE WORD
8078  054226  000207                            RTS     PC              ;EXIT ---------------------------------->
8079
8080
```

```
8081
8082
8083
8084
```

```
8085                                    ;*WRITE DATA HOUSEKEEPING ROUTINE
8086
8087
8088
8089
8090
8091
8092
8093  054230  000000          COUNTD: 0
8094  054232  000400          FORMAT: 256.
8095  054234  000000          ZWORDS: 0
8096  054236                  WRDATA:
8097  054236  011137  054230          MOV     (R1),@#COUNTD   ;STORE NO. OF WORDS TO BE WRITTEN
8098  054242  012102                  MOV     (R1)+,R2        ;SAME IN R2
8099  054244  012137  053562          MOV     (R1)+,@#COMPA   ;COMPARE OR NOT
8100  054262  012701  000016          MOV     #14.,R1         ;NO. OF TOLERANCE GAP WORDS
8101  054266  012703  055574          MOV     #TOLGAP,R3      ;START OF TOLERANCE GAP TABLE
8102  054272  012723  177777  1$:     MOV     #-1,(R3)+       ;MAKE IT 177777
8103  054276  005301                  DEC     R1              ;IS 14 COMPLETED ?
8104  054300  001374                  BNE     1$              ;IF NOT, CONTINUE
8105
8106  054302  013700  014776          MOV     @#RHMR,R0       ;R0 CONTAINS MAINTANENCE REG.
8107  054306  013746  054232          MOV     @#FORMAT,-(SP)
8108  054312  163716  054230          SUB     @#COUNTD,(SP)
8109  054316  011637  054234          MOV     (SP),@#ZWORDS   ;NO. OF ZERO WORDS TO BE WRITTEN
8110  054322  012604                  MOV     (SP)+,R4
8111  054324  005737  015142          TST     @#TSECC         ;IS THIS AN ECC TEST
8112  054330  001403                  BEQ     7$              ;BRANCH IF NO
8113  054332  012737  177777  050464  MOV     #-1,@#TSECCG    ;THESE BITS ARE TO GENERATE ECC
8114  054340  012703  054566  7$:     MOV     #DISK,R3        ;SIMULATED DISK AREA
8115  054344  004737  054016  2$:     JSR     PC,@#WRITE      ;WRITE ON SIMULATED DISK
8116  054350  013723  054014          MOV     @#WWORD,(R3)+   ;STORE ON SIMULATED DISK
8117  054354  005302                  DEC     R2
8118  054356  001372                  BNE     2$
8119  054360  005704                  TST     R4              ;ANY ZEROS TO BE WRITTEN ?
8120  054362  001406                  BEQ     4$              ;BRANCH IF NONE TO BE WRITTEN
8121
8122  054364  004737  054016  3$:     JSR     PC,@#WRITE      ;WRITE ZEROS ON SIMULATED DISK
8123  054370  013723  054014          MOV     @#WWORD,(R3)+   ;STORE THEM
8124  054374  005304                  DEC     R4
8125  054376  001372                  BNE     3$
8126  054400  005037  050464  4$:     CLR     @#TSECCG        ;NO MORE ECC TO BE GENERATED
8127  054404  012701  000002          MOV     #2,R1
8128  054410  004737  054016  5$:     JSR     PC,@#WRITE      ;WRITE ECC1 AND ECC2 ON SIMULATED DISK
8129  054414  013723  054014          MOV     @#WWORD,(R3)+   ;STORE IN WEEC1 AND WEEC2
8130  054420  005301                  DEC     R1
8131  054422  001372                  BNE     5$
8132  054424  004737  054016          JSR     PC,@#WRITE      ;WRITE DATA GAP
8133  054430  013723  054014          MOV     @#WWORD,(R3)+   ;STORE IT
8134  054434  012701  000016          MOV     #14.,R1
8135  054440  004737  054016  6$:     JSR     PC,@#WRITE      ;WRITE TOLERANCE GAP ZEROS
8136  054444  013723  054014          MOV     @#WWORD,(R3)+   ;STORE THEM
8137  054450  005301                  DEC     R1
8138  054452  001372                  BNE     6$
8139  054466  000201                  RTS     R1              ;EXIT ---------------------------->
8140
```

```
8141
8142
8143
8144
8145
8146
8147
```

```
                                   ;*THIS IS THE SIMULATED DISK
                                   ;*ONLY ONE SECTOR OF SPACE IS ALLOCATED








054470  000023                     SECGAP: .BLKW   19.            ;SECTOR GAP 38 BYTES OF 0
054536  000001                     WSSYNC: .BLKW   1              ;SECTOR GAP 1 BYTE OF 0 ONE SYNC BYTE
054540  000004                     HEADER: .BLKW   4              ;HEADER = CYL, SECTOR/TRACK, KEY1, KEY2
054550  000001                     WCRC:   .BLKW   1              ;CRC
054552  000005                     HEGAP:  .BLKW   5              ;HEADER GAP 10 BYTES OF 0
054564  000001                     HDWSYN: .BLKW   1              ;HEADER GAP 1 BYTE OF 0 ONE SYNC. BYTE
054566                             SILOTB:                        ;(ALSO USED IN SILO TEST AS SILO TABLE)
054566  000400                     DISK:   .BLKW   256.           ;DATA SPACE
055566  000001                     WECC1:  .BLKW   1              ;ECC1
055570  000001                     WECC2:  .BLKW   1              ;ECC2
055572  000001                     DTAGAP: .BLKW   1              ;DATA GAP 2 BYTES OF 0
055574  000016                     TOLGAP: .BLKW   14.            ;TOLERANCE GAP 28 BYTES OF 0
```

```
                                ;*WRITE HEADER AND DATA
                                ;*THIS SUBROUTINE IS THE FIRST IN A SERIES OF NESTED SUBROUTINES
                                ;*IT ISSUES DIAGNOSTIC MODE, AN EXTRA DIAGNOSTIC INDEX, AND THE
                                ;*'GO' BIT
                                ;*IT THEN CALLS THREE OTHER SUBROUTINES, WHICH IN TURN CALL OTHER
                                ;*SUBROUTINES.  THE THREE SUBROUTINES CALLED HERE ARE:
                                ;*      SEARCH          ;ISSUES SECTOR CLOCKS TO SET SECTOR FOUND FLOP
                                ;*      WRHEAD          ;WRITES THE SECTOR HEADER
                                ;*      WRDATA          ;WRITES THE ACTUAL SECTOR DATA
                                ;*ALL OF THE ABOVE MENTIONED "WRITING" IS ACTUALLY DONE INTO A CORE
                                ;*BUFFER AREA CALLED 'DISK' VIA THE MAINTENANCE REGISTER (RHMR)

055630  000000          RNCTR1: .WORD   0               ;'RUN' LINE STALL COUNTER

055632  011637  015132  COMWHD: MOV     (SP),@#PCJSR    ;SAVE PC OF JSR + 4
055636  162737  000004  015132  SUB     #4,@#PCJSR      ;SAVE PC OF JSR

055660  012777  000001  137110  MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
055666  052777  000004  137102  BIS     #MINX,@RHMR     ;SET DIAGNOSTIC INDEX
055674  042777  000004  137074  BIC     #MINX,@RHMR     ;CLEAR DIAGNOSTIC INDEX
055702  052777  000001  137046  BIS     #GO,@RHCS1      ;SET 'GO' BIT & STALL 'TILL 'RUN'
                                                        ;(FUNCTION CODE WAS SET UP BY THE TEST)
055710  012737  000113  055630  RNWAT1: MOV     #75.,@#RNCTR1   ;LOAD STALL COUNTER = APPROX 450US
                                                        ;FOR 11/50 CPU
055716  005337  055630          1$:     DEC     @#RNCTR1        ;COUNT DOWN 1 TIME
055722  001375                  BNE     1$              ;CONTINUE UNTIL = 0

055724  013746  056010          MOV     @#WSECTR,-(SP)  ;GET DESIRED SECTOR/TRACK
055730  042716  177740          BIC     #177740,(SP)    ;MAKE ONLY SECTOR
055734  012637  055744          MOV     (SP)+,@#WTRK    ;SAVE SECTOR
055740  004137  056716          2$:     JSR     R1,@#SEARCH     ;ISSUE SECTOR CLOCKS TO GET TO
                                                        ;DESIRED SECTOR <------------------->

055744  000000          WTRK:   .WORD   0               ;SECTOR NO.
055746  012701  000240          MOV     #+NOP,R1        ;GOING TO MOVE NOPS
055752  010137  056020          MOV     R1,@#SEGPER     ;NOP INTO SEGAP
055756  010137  056022          MOV     R1,@#FSYNER     ;NOP INTO FSYNER
055762  010137  056024          MOV     R1,@#ERHEAD     ;NOP INTO ERHEAD
055766  010137  056026          MOV     R1,@#ERCRC      ;NOPINTO ERCRC
055772  010137  056030          MOV     R1,@#ERHDGP     ;NOP INTO ERHDGAP
055776  010137  056032          MOV     R1,@#HDESYN     ;NOP INTO HDESYN
```

```
8232  056002  004137  056102                  JSR     R1,@#WRHEAD     ;WRITE THE HEADER <----------------->
8233
8234  056006  000000                  WCYL:   0                       ;CYLINDER
8235  056010  000000                  WSECTR: 0                       ;SECTOR AND TRACK
8236  056012  000000                  WKEY1:  0                       ;KEY1
8237  056014  000000                  WKEY2:  0                       ;KEY2
8238  056016  000000                  GCRC:   0                       ;GOOD CRC
8239
8240
8241                                  ;DUMMY ERROR CALL LOCATIONS FOR THE WRITE HEADER OPERATION
8242
8243  056020  000240                  SEGPER: NOP                     ;IF "ERROR 6" INSERTED BY
8244                                                                  ;WRHEAD SUBROUTINE, THEN
8245                                                                  ;SECTOR GAP GOING ON DISK
8246                                                                  ;IS NOT RIGHT.
8247
8248                                                                  ;WORD NO. CONTAINS WHICH
8249                                                                  ;WORD IS WRONG, THAT IS
8250                                                                  ;FIRST OF TENTH, OR WHAT EVER.
8251
8252                                                                  ;BAD WORD IS WHAT IS GOING ON DISK
8253
8254  056022  000240                  FSYNER: NOP                     ;IF "ERROR 6" INSERTED BY
8255                                                                  ;WRHEAD SUBROUTINE, THEN
8256                                                                  ;THE LAST 0 BYTE OF SECTOR
8257                                                                  ;GAP OF FIRST SYNC. BYTE
8258                                                                  ;AFTER SECTOR GAP IS IN
8259                                                                  ;ERROR.
8260
8261                                                                  ;WORD NO. CONTAINS 20
8262                                                                  ;RIGHT BYTE IS SECTOR GAP
8263                                                                  ;LEFT BYTE IS SYNC. BYTE.
8264
8265                                                                  ;BAD WORD IS WHAT IS GOING ON
8266                                                                  ;DISK
8267
8268  056024  000240                  ERHEAD: NOP
8269                                                                  ;IF "ERROR 6" INSERTED BY
8270                                                                  ;WRHEAD SUBROUTINE, THEN
8271                                                                  ;HEADER GOING ON DISK
8272                                                                  ;IS WRONG.
8273
8274                                                                  ;WORD NO 1 = CYLINDER NO
8275                                                                  ;WORD NO 2 = SECTOR/TRACK
8276                                                                  ;WORD NO 3 = KEY1
8277                                                                  ;WORD NO 4 = KEY2
8278
8279                                                                  ;BAD WORD IS WHAT IS GOING ON
8280                                                                  ;DISK
8281
8282  056026  000240                  ERCRC:  NOP
8283                                                                  ;IF "ERROR 6" INSERTED BY
8284                                                                  ;WRHEAD SUBROUTINE, THEN CRC WRITTEN
8285                                                                  ;ON DISK IS IN ERROR.
8286
8287                                                                  ;GOOD DATA IS WHAT SHOULD BE ON DISK
```

```
8288                                                    ;BAD DATA IS WHAT IS GOING ON DISK.
8289
8290                                                    ;WORD NO IS 5
8291
8292  056030  000240                ERHDGP: NOP
8293                                                    ;IF "ERROR 6" INSERTED BY
8294                                                    ;WRHEAD SUBROUTINE, THEN HEADER
8295                                                    ;GAP GOING ON DISK IS WRONG.
8296
8297                                                    ;WORD NO GIVES WHICH OF
8298                                                    ;THE HEADER GAP WORDS
8299                                                    ;ARE WRONG. FOR EXAMPLE:
8300
8301                                                    ;WORD NO 1 = FIRST HEADER
8302                                                    ;            GAP WORD
8303                                                    ;BAD WORD IS WHAT IS GOING ON DISK
8304
8305  056032  000240                HDESYN: NOP
8306                                                    ;IF "ERROR 6" INSERTED BY
8307                                                    ;WRHEAD SUBROUTINE, THEN LAST
8308                                                    ;HEADER GAP BYTE OR HEADER
8309                                                    ;SYNC BYTE GOING ON DISK IS WRONG.
8310
8311                                                    ;WORD NO = 5
8312
8313                                                    ;BAD DATA IS WHAT IS GOING
8314                                                    ;ON DISK, RIGHT BYTE IS HEADER
8315                                                    ;GAP 0'S BYTE, LEFT BYTE IS HEADER
8316                                                    ;GAP SYNC.
8317
8318
8319  056034  005737  015124                TST     @#ERFLG$        ;ARE ANY ERRORS DETECTED ?
8320  056040  001004                        BNE     FOUT            ;IF YES BRANCH
8321  056042  004137  054236                JSR     R1,@#WRDATA     ;WRITE THE DATA
8322
8323  056046  000000                FNWORD: .WORD   0               ;FORMAT COMMAND NO. OF DATA
8324  056050  000000                        .WORD   0
8325
8326  056052                        FOUT:
8327  056066  000207                        RTS     PC              ;EXIT
```

```
                                ;*WRITE HEADER
                                ;*R0 = MAINT.REG.
                                ;*R1 = SIMULATED DISK
                                ;*R2 = BYTE COUNT
                                ;*R3 = WRITE WORD
                                ;*R5 = WORD COUNT
056070  000000          SCYL:   0                       ;
056072  000000          SSECTR: 0                       ;
056074  000000          SKEY1:  0                       ;
056076  000000          SKEY2:  0                       ;
056100  000000          SCRC:   0                       ;

056102  012137  056070  WRHEAD: MOV     (R1)+,@#SCYL
056106  012137  056072          MOV     (R1)+,@#SSECTR
056112  012137  056074          MOV     (R1)+,@#SKEY1
056116  012137  056076          MOV     (R1)+,@#SKEY2
056122  012137  056100          MOV     (R1)+,@#SCRC

056130  012701  054470          MOV     #SECGAP,R1      ;SIMULATED DISK INDICATOR
056134  013700  014776          MOV     @#RHMR,R0       ;R0 NOW HAS MAINT. REG. ADDR.
056140  012710  000001          MOV     #DMD,@R0        ;SET DIAG. MODE
056144  012705  000002          MOV     #2,R5           ;WORD COUNTER
056150  052710  000010          BIS     #MSTCK,@R0      ;SET SECTOR FOR FIRST BYTE
056154  012710  000013  1$:     MOV     #MSTCK!MCLK!DMD,@R0;SET SECTOR, CLOCK, DIAG.
                                                        ;MODE, RESET INDEX
056160  032710  000040          BIT     #MWR,@R0        ;CHECK WRITE BIT IN MAINT. REG.
056164  001403                  BEQ     2$
056166  000261                  SEC                     ;SET CARRY
056170  006003                  ROR     R3              ;MOVE ONE FORWARD
056172  000402                  BR      3$              ;
056174  000241          2$:     CLC                     ;CLEAR CARRY
056176  006003                  ROR     R3              ;MOVE ZERO FORWARD
056200  012710  000001  3$:     MOV     #DMD,@R0        ;CLEAR CLOCK, SECTOR
056204  012702  000007          MOV     #7,R2           ;BYTE COUNTER
056210  052710  000002  4$:     BIS     #MCLK,@R0       ;SET BIT CLOCK
056214  032710  000040          BIT     #MWR,@R0        ;CHECK WRITE BIT IN MAINT.REG.
056220  001403                  BEQ     5$              ;BRANCH IF ZERO

056222  000261                  SEC                     ;SET CARRY
056224  006003                  ROR     R3              ;MOVE ONE FORWARD
056226  000402                  BR      6$
056230  000241          5$:     CLC
056232  006003                  ROR     R3
056234  012710  000001  6$:     MOV     #DMD,@R0
056240  005302                  DEC     R2
```

```
8384  056242  001362                        BNE     4$
8385  056244  005305                        DEC     R5
8386  056246  001342                        BNE     1$
8387  056250  010321                        MOV     R3,(R1)+
8388  056252  005703                        TST     R3
8389  056254  001414                        BEQ     7$
8390
8391  056256  012737  000001  052770        MOV     #1,@#ERWORD
8392  056264  005037  001124                CLR     @#$GDDAT
8393  056270  010337  001126                MOV     R3,@#$BDDAT
8394  056274  012737  104006  056020        MOV     #104006,@#SEGPER
8395  056302  000137  056710                JMP     @#17$           ;BRANCH OUT ---------------------->
8396
8397  056306  012702  000022        7$:     MOV     #18.,R2         ;COUNT NO. OF SECTOR GAP
8398  056312  012737  000024  052770  10$:  MOV     #20.,@#ERWORD   ;COUNT TO GIVE ERROR WORD
8399  056320  004737  054016                JSR     PC,@#WRITE      ;WRITE SECTOR GAP
8400  056324  013721  054014                MOV     @#WWORD,(R1)+   ;STORE SECTOR GAP WORD
8401  056330  001413                        BEQ     11$
8402  056332  160237  052770                SUB     R2,@#ERWORD     ;IF NOT GET ERROR WORD NO.
8403  056336  005037  001124                CLR     @#$GDDAT        ;GOOD WORD
8404  056342  013737  054014  001126        MOV     @#WWORD,@#$BDDAT;BAD WORD
8405  056350  012737  104006  056020        MOV     #104006,@#SEGPER;STORE "ERROR 6" IN SEGPER
8406  056356  000554                        BR      17$             ;BRANCH OUT
8407  056360  005302                11$:    DEC     R2              ;HAVE 18 WORDS OF ZEROS BEEN WRITTEN ?
8408  056362  001353                        BNE     10$             ;IF NOT CONTINUE
8409
8410                                        ;AT THIS POINT THE SECTOR FOUND FLOP SHOULD
8411                                        ;BE HIGH,  SO THAT THE HEADER SYNC BYTE CAN BE GIVEN.
8412
8413                                        ;HOWEVER, IN THE DRIVE TIMING ERROR TEST THE REST OF THE ROUTINE
8414                                        ;IS ABORTED
8415
8416  056364  005737  015144                TST     @#TESDTE        ;IS THIS A DRIVE TIMING ERROR TEST ?
8417  056370  001147                        BNE     17$             ;BRANCH OUT IF YES ------------------>
8418
8419  056372  004737  054016                JSR     PC,@#WRITE      ;WRITE ONE SECTOR GAP 0 BYTE
8420                                                                ;AND ONE SYNC. BYTE = 230
8421  056376  013711  054014                MOV     @#WWORD,(R1)    ;SAVE 0 BYTE AND SYNC BYTE
8422  056402  023721  052752                CMP     @#RSYNC,(R1)+   ;IF SYNC. BYTE RIGHT
8423  056406  001414                        BEQ     12$             ;IF YES CONTINUE OPERATION
8424  056410  012737  000024  052770        MOV     #20.,@#ERWORD   ;IF NOT GET READY FOR ERROR PRINT
8425
8426  056416  013737  052752  001124        MOV     @#RSYNC,@#$GDDAT;GOOD WORD
8427  056424  014137  001126                MOV     -(R1),@#$BDDAT  ;BAD WORD
8428  056430  012737  104006  056022        MOV     #104006,@#FSYNER;INSERT "ERROR 6" IN FSYNER
8429  056436  000524                        BR      17$             ;BRANCH OUT ---------------------->
8430
8431  056440  012702  000004        12$:    MOV     #4,R2           ;FOUR HEADER WORDS
8432  056444  012703  056070                MOV     #SCYL,R3        ;POINTER FOR HEADER TABLE
8433  056450  012737  000005  052770  13$:  MOV     #5,@#ERWORD     ;ERROR WORD NO SET
8434  056456  004737  054016                JSR     PC,@#WRITE      ;WRITE 4 HEADER WORDS
8435  056462  013711  054014                MOV     @#WWORD,(R1)    ;STORE WRITTEN WORD
8436  056466  022321                        CMP     (R3)+,(R1)+     ;IS IT CORRECT ?
8437  056470  001412                        BEQ     14$             ;IF GOOD CONTINUE OPERATION
8438                                                                ;IF NOT GET READY FOR ERROR PRINT
8439
```

```
8440  056472  160237  052770                  SUB     R2,@#ERWORD     ;WORD NO
8441  056476  014337  001124                  MOV     -(R3),@#$GDDAT  ;GOOD DATA
8442  056502  014137  001126                  MOV     -(R1),@#$BDDAT  ;BAD DATA
8443  056506  012737  104006  056024          MOV     #104006,@#ERHEAD;INSERT "ERROR 6"
8444  056514  000475                          BR      17$             ;BRANCH OUT ------------------------>
8445
8446  056516  005302                  14$:    DEC     R2              ;ARE 4 HEADER WORDS DONE?
8447  056520  001353                          BNE     13$             ;IF NOT CONTINE
8448  056522  004737  054016                  JSR     PC,@#WRITE      ;WRITE CRC
8449  056526  013711  054014                  MOV     @#WWORD,(R1)    ;STORE CRC
8450  056532  022137  056016                  CMP     (R1)+,@#GCRC    ;COMPARE GOOD CRC
8451  056536  001414                          BEQ     20$             ;IF GOOD CONTINUE OPERATION
8452                                                                  ;IF NOT GET READY FOR ERROR PRINT
8453
8454  056540  014137  001126                  MOV     -(R1),@#$BDDATA ;BAD CRC WRITTEN
8455  056544  013737  056016  001124          MOV     @#GCRC,@#$GDDAT ;GOOD CRC
8456  056552  012737  000005  052770          MOV     #5,@#ERWORD     ;ERROR WORD NO
8457  056560  012737  104006  056026          MOV     #104006,@#ERCRC ;INSERT ERROR 6
8458  056566  000450                          BR      17$             ;EXIT ------------------------------>
8459
8460  056570  012702  000005          20$:    MOV     #5,R2           ;NO OF HEADER GAP
8461  056574  012737  000006  052770  15$:    MOV     #6,@#ERWORD     ;ERROR WORD NO SET
8462  056602  004737  054016                  JSR     PC,@#WRITE      ;WRITE HEADER GAP
8463  056606  013721  054014                  MOV     @#WWORD,(R1)+   ;STORE
8464  056612  001412                          BEQ     16$             ;IF GOOD CONTINUE OPERATION
8465                                                                  ;IF NOT GET READY FOR PRINT
8466
8467  056614  160237  052770                  SUB     R2,@#ERWORD     ;ERROR WORD NO
8468  056620  005037  001124                  CLR     @#$GDDAT        ;GOOD DATA
8469  056624  014137  001126                  MOV     -(R1),@#$BDDAT  ;BAD DATA
8470  056630  012737  104006  056030          MOV     #104006,@#ERHDGP;STORE "ERROR 6"
8471  056636  000424                          BR      17$             ;BRANCH OUT ------------------------>
8472
8473  056640  005302                  16$:    DEC     R2              ;ARE 5 HEADER GAP ZEROS DONE ?
8474  056642  001354                          BNE     15$             ;IF NOT CONTINUE
8475  056644  004737  054016                  JSR     PC,@#WRITE
8476  056650  013711  054014                  MOV     @#WWORD,(R1)
8477  056654  023721  052752                  CMP     @#RSYNC,(R1)+
8478  056660  001413                          BEQ     17$             ;A-OK, EXIT ------------------------>
8479
8480  056662  012737  000005  052770          MOV     #5,@#ERWORD     ;IF NOT GET READY FOR ERROR PRINT
8481  056670  014137  001126                  MOV     -(R1),@#$BDDAT
8482  056674  013737  052752  001124          MOV     @#RSYNC,@#$GDDAT
8483  056702  012737  104006  056032          MOV     #104006,@#HDESYN
8484
8485  056710                          17$:
8486  056712  000201                          RTS     R1              ;EXIT ------------------------------>
8487
8488
8489
8490
8491
8492
8493
8494
```

```
8495                                    ;*SEARCH SECTOR
8496
8497
8498
8499
8500
8501
8502                                    ;*      R0=RHMR ADDRESS
8503                                    ;*      R1=PASSED ARGUMENT (SECTOR SEARCHED FOR)
8504                                    ;*      R2=CLOCK COUNT (PER BYTE)
8505                                    ;*      R3=SECTOR COUNTER FROM R1
8506                                    ;*      R5=BYTES PER WORD COUNT
8507
8508                                    ;*BEFORE INDEX IS GIVEN TWO SECTOR CLOCKS ARE GIVEN TO RESET
8509                                    ;*SECTOR PULSE IN CASE IT IS SET.
8510
8511                                    ;*AT THE BEGINNING OF EACH SECTOR, ONE SECTOR CLOCK HAS TO RISE
8512                                    ;*BEFORE MAINT. CLOCK, THEN EVERY EIGHT MAINT.CLOCKS, ONE SECTOR CLOCK
8513                                    ;*IS IDENTICAL WITH THE MAINT. CLOCK
8514                                    ;*THE SECTOR CLOCKS ARE NUMBERED AS FOLLOWS:
8515
8516                                    ;*THE SECTOR CLOCK UNDER INDEX - 0
8517                                    ;*THE NEXT - 1
8518                                    ;*THE NEXT - 2
8519                                    ;*ETC.
8520                                    ;*THEN THE LAST SECTOR CLOCK IN ONE SECTOR HAS NUMBER - 608
8521                                    ;*THE NEXT SECTOR WILL HAVE 608 SECTOR CLOCKS
8522                                    ;*THE NEXT SECTOR THEN HAS ANOTHER 608 SECTOR CLOCKS
8523                                    ;*AND SO ON
8524
8525
8526
8527  056714  000000                SECTR:  0                       ;SECTOR SEARCHED FOR
8528
8529  056716  012137  056714        SEARCH: MOV     (R1)+,@#SECTR ;SAVE SECTOR SEARCHED FOR
8530  056736  013700  014776                MOV     @#RHMR, R0      ;NOW R0 HAS MAINTENANCE REG. ADR.
8531  056742  013703  056714                MOV     @#SECTR,R3      ;LOAD SECTOR COUNTER
8532
8533  056746  012710  000001                MOV     #DMD,@R0        ;SET DIAGNOSTIC MODE
8534  056752  052710  000010                BIS     #MSTCK, @R0     ;SET SECTOR CLOCK
8535  056756  042710  000010                BIC     #MSTCK, @R0     ;CLEAR SECTOR CLOCK
8536  056762  052710  000010                BIS     #MSTCK, @R0     ;SET SECTOR CLOCK
8537  056766  042710  000010                BIC     #MSTCK, @R0     ;CLEAR SECTOR CLOCK
8538                                                                ;THE ABOVE TWO SECTOR CLOCKS ARE GIVEN FOR
8539                                                                ;RESETING SECTOR PULSE
8540                                                                ;IN CASE IT STARTS SET
8541  056772  052710  000014                BIS     #MINX!MSTCK,@R0 ;SET INDEX AND SECTOR CLOCK
8542  056776  042710  000014                BIC     #MINX!MSTCK,@R0 ;RESET INDEX AND SECTOR CLOCK
8543  057002  005703                        TST     R3              ;TEST FOR SECTOR 0
8544  057004  001461                        BEQ     7$              ;IF SECTOR 0 IS REQUIRED, EXIT ------>
8545
8546                                    ;NOW 304 WORDS WILL START (608 BYTES)
8547
8548                                    ;*FOR FIRST BYTE MAINT. SECTOR CLOCK WILL GO HIGH, THEN MAINT. CLOCK WILL GO HIGH
8549                                    ;*BOTH WILL COME DOWN TOGETHER, THEN SEVEN MAINT. CLOCKS WILL BE GIVEN
8550                                    ;*FOR SECOND BYTE, AND ALL OTHER BYTES TILL NEXT SECTOR, SECTOR CLOCK
```

```
8551                                    ;*WILL BE IDENTICAL WITH ONE MAINT. CLOCK
8552
8553
8554                                    ;ONE WORD ONLY
8555
8556  057006  012702  000010      1$:   MOV     #8.,    R2        ;BYTE COUNTER
8557  057012  012705  000002            MOV     #2,     R5        ;BYTES PER WORD
8558  057016  052710  000010            BIS     #MSTCK,@R0        ;SET SECTOR CLOCK
8559  057022  052710  000002            BIS     #MCLK,@R0         ;SET MAINT. CLOCK
8560  057026  000402                    BR      3$                ;BRANCH TO CLEAR SECTOR AND CLOCK
8561  057030  052710  000012      2$:   BIS     #MSTCK!MCLK,@R0   ;SET BOTH CLOCKS
8562  057034  042710  000012      3$:   BIC     #MSTCK!MCLK,@R0   ;CLEAR BOTH CLOCKS
8563  057040  052710  000002      8$:   BIS     #MCLK,@R0         ;SET MAINT. CLOCK
8564  057044  042710  000002            BIC     #MCLK,@R0         ;CLEAR MAINT.CLOCK
8565  057050  005302                    DEC     R2                ;BYTE COUNTER
8566  057052  001372                    BNE     8$                ;CONTINUE IF BYTE NOT COMPLETE
8567
8568  057054  012702  000007            MOV     #7,R2             ;SETUP FOR SECOND BYTE
8569  057060  005305                    DEC     R5                ;IS WORD COMPLETE?
8570  057062  001362                    BNE     2$                ;CONTINUE IF NOT COMPLETE
8571                                                              ;TO GIVE SECTOR CLOCK AND MAINT. CLOCK
8572
8573
8574                                    ;NOW 303 WORDS ARE LEFT AND ALL ARE IDENTICAL
8575
8576  057064  012701  000457            MOV     #303.,R1          ;WORDS PER SECTOR COUNTER
8577  057070  012705  000002      4$:   MOV     #2,R5             ;BYTES PER WORD COUNTER
8578  057074  012702  000007      5$:   MOV     #7,R2             ;BIT COUNTER (MAINT.CLOCK COUNTER)
8579  057100  052710  000012            BIS     #MSTCK!MCLK,@R0   ;SET SECTOR CLOCK AND MAINT. CLOCK
8580  057104  042710  000012            BIC     #MSTCK!MCLK,@R0   ;CLEAR CLOCKS
8581  057110  052710  000002      6$:   BIS     #MCLK,@R0         ;SET MAINT. CLOCK
8582  057114  042710  000002            BIC     #MCLK,@R0         ;RESET MAINT. CLOCK
8583  057120  005302                    DEC     R2                ;IS BYTE COMPLETE ?
8584  057122  001372                    BNE     6$                ;CONTINUE IF NOT COMPLETE
8585
8586  057124  005305                    DEC     R5                ;IS WORD COMPLETE ?
8587  057126  001362                    BNE     5$                ;CONTINUE IF NOT
8588
8589  057130  005301                    DEC     R1                ;IS SECTOR COMPLETE ?
8590  057132  001356                    BNE     4$                ;CONTINUE IF NOT
8591
8592  057134  052710  000010            BIS     #MSTCK,@R0        ;SET SECTOR CLOCK 1 MORE TIME (FOR 609)
8593  057140  042710  000010            BIC     #MSTCK,@R0        ;CLEAR SECTOR CLOCK
8594  057144  005303                    DEC     R3                ;IS REQUIRED NO OF SECTORS COMPLETE ?
8595  057146  001317                    BNE     1$                ;CONTINUE IF NOT
8596
8597  057150                      7$:
8598  057164  000201                    RTS     R1
8599
8600
8601                                    ;*READ ONE SECTOR OF DATA
8602
8603  057166  000000            RNO:    0                         ;NO. OF WORDS READ
8604  057170  000000            RCOM:   0                         ;EXTRA STORAGE
8605
8606
```

```
 8607
 8608  057172  012137  057166          REDATA: MOV     (R1)+,@#RNO     ;SAVE NO. OF WORDS ONLY FOR INFORMATION
 8609  057176  012137  057170                  MOV     (R1)+,@#RCOM    ;EXTRA WORD ONLY FOR INFORMATION
 8610  057204  005737  015142                  TST     @#TSECC         ;IS THIS AN ECC TEST
 8611  057210  001403                          BEQ     1$              ;BRANCH IF NO
 8612  057212  012737  177777  050464          MOV     #-1,@#TSECCG    ;THESE BITS ARE TO GENERATE ECC
 8613  057220  012702  000402          1$:     MOV     #258.,R2        ;256 WORDS PER SECTOR
 8614                                                                  ;PLUS 2 ECC WORDS
 8615  057224  012703  054566                  MOV     #DISK,R3        ;POINTE TO DISK SIMULATION
 8616  057230  012337  053560          2$:     MOV     (R3)+,@#WORD    ;READY TO READ CONTENTS
 8617  057234  004737  053564                  JSR     PC,@#READ               ;READ
 8618  057240  005302                          DEC     R2              ;IS 256 WORDS DONE?
 8619  057242  001372                          BNE     2$              ;IF NOT BRANCH
 8620  057244  005737  015142                  TST     @#TSECC         ;IS THIS AN ECC TEST
 8621  057250  001012                          BNE     4$              ;BRANCH OUT IF YES
 8622  057252  005037  050464                  CLR     @#TSECCG        ;NO MORE ECC BITS ARE TO BE GENERATED
 8623  057256  012702  000017                  MOV     #15.,R2         ;ONE DATA GAP, 14 TOLERANCE GAP
 8624  057262  012337  053560          3$:     MOV     (R3)+,@#WORD    ;READY TO READ CONTENTS OF WORD
 8625  057266  004737  053564                  JSR     PC,@#READ       ;READ
 8626  057272  005302                          DEC     R2              ;COUNT
 8627  057274  001372                          BNE     3$              ;BRANCH IF 14 NOT DONE
 8628  057276                          4$:
 8629  057300  000201                          RTS     R1              ;RETURN
```

```
8630  057302                          RPVECT:
8631  057356  104402                          TYPOC                   ;TYPE FROM PC
8632  057360  012777  057302  135356          MOV     #RPVECT,@RPVEC  ;RESTORE TRAP RP04 VECTOR
8633  057366  000000                          HALT                    ;CHANGE TO CONTINUE
8634
```

SEQ 0208

```
8635
8636                                    .SBTTL
8637                                    .SBTTL  ***SYSMAC LIBRARY ROUTINES***
8638                                    .SBTTL
8639
8640
```

```
8641                                    ;FROM THE TTY
```

```
 8642
 8643          000001                  .END
```

```
ACL    = 000040            2432#
ACU    = 100000            2385#
ADDMOD   000204             495#  2673
ADTIMO   052374            7580   7599#
AOE    = 001000            2301#
AS       015050            1885   1888   2510#  3985*  3997   4153*  4164   4283*  4293
ATA    = 100000            2289#  3982   4150   4280
ATABLE   017336            2610#  2851
ATTENT   015134            2545#  2813*  2851*  3985   4153   4283
AT0    = 000001            2323#
AT1    = 000002            2324#
AT2    = 000004            2325#
AT3    = 000010            2326#
AT4    = 000020            2327#
AT5    = 000040            2328#
AT6    = 000100            2329#
AT7    = 000200            2330#
A16    = 000400            2264#  3844
A17    = 001000            2265#  3844
BA       015026            1881   2499#  3978*  4138*  4143*  4269*  4273*
BAI    = 000010            2239#  3846   7220
BASECH   051410             495   7573#  7600
BEGIN    017356             493   2620#  7598   8642
BEGIN2   017346             497   2618#
BITST    045532            2696   6578#
BIT0   = 000001             487#  6980   7015   7017
BIT00  = 000001             487#
BIT01  = 000002             487#
BIT02  = 000004             487#
BIT03  = 000010             487#
BIT04  = 000020             487#
BIT05  = 000040             487#
BIT06  = 000100             487#
BIT07  = 000200             487#
BIT08  = 000400             487#  8641
BIT09  = 001000             487#  8641   8642
BIT1   = 000002             487#  6994
BIT10  = 002000             487#  8642
BIT11  = 004000             487#  8641
BIT12  = 010000             487#
BIT13  = 020000             487#  7010   7012   8642
BIT14  = 040000             487#  6987   8641
BIT15  = 100000             487#  6982   6989   6996   7005   7007
BIT2   = 000004             487#
BIT3   = 000010             487#
BIT4   = 000020             487#
BIT5   = 000040             487#
BIT6   = 000100             487#
BIT7   = 000200             487#
BIT8   = 000400             487#
BIT9   = 001000             487#
BLT1     045560            6584#  6589
BLT2     045602            6584   6585   6591#
BLT3     045612            6592   6593#
BPTVEC=  000014             487#
CA       015044            2508#
```

```
CC       015070  2518#
CHECKD=  104414  8642#
CHECKE   046210  4432  4551  4708  4863  5025  5152  5310  5460  5624  5741  5892  6043  6205
                 6372  6727#
CHECKT   046020  3843  3908  4062  4214  4427  4521  4665  4824  4977  5122  5267  5422  5574
                 5712  5849  6004  6157  6322  6672#  6855  6936  7083  8642
CKSWR =  104407  8641  8642#
CLAREA   045702  3815  4556  4563  5158  5165  5747  5754  6624#
CLDISK   045764  2812  2892  2904  2956  3806  3836  3900  4054  4206  4318  4476  4653  4812
                 4966  5077  5255  5411  5562  5667  5837  5992  6146  6310  6415  6440  6461
                 6654#  6844  6924  7082  7256  7609
CLR    = 000040  2241#  2969  3005  3022  3045  3065  3089  3110  3146  3184  3231  3268  3300
                 3353  3393  3445  3473  3493  3515  3539  3551  3565  3577  3592  3615  3646
                 3656  3668  3678  3694  3718  3744  3777  6659
COMHD    052510  3848  3929  4082  4668  4827  4979  5270  5424  5577  5852  6007  6159  6325
                 6857  6937  7099  7700#
COMPA    053562  7840#  7889  7967#  8099#
COMPAR   046660  2938  4001  4018  4170  4298  4434  4574  4729  4888  5051  5176  5328  5483
                 5647  5765  5909  6067  6232  6396  6872#  7120  7146
COMWHD   055632  4234  4428  4527  5128  5718  8204#
COUNTD   054230  8093#  8097#  8108
CPHALT   005123  1423#  3843  3908  4062  4214  4427  4432  4521  4551  4665  4708  4824  4863
                 4977  5025  5122  5152  5267  5310  5422  5460  5574  5624  5712  5741  5849
                 5892  6004  6043  6157  6205  6322  6372  6855  6936  7083
CR     = 000015  487#  8641
CRC      047172  3829  3894  4048  4200  4371  4410  4500  4644  4794  4949  5101  5246  5393
                 5544  5691  5828  5968  6127  6293  6838  6918  6970#  7058
CRLF   = 000200  487#  8641
CSF    = 000002  2372#
CSU    = 000010  2374#
CS1      015032  1870  1873  1881  1888  1895  1905  1908  1912  1928  2503#  3979*  4147*  4275*
                 6432  6675  6679  6686  6730  6738
CS2      015030  1870  1873  1881  1895  1928  2500#  4139*  4144*  4145*  4276*  4277*
CYL      052650  3822*  3830  3887*  3895  4002  4041*  4049  4636*  4645  4786*  4795  4941*  4950
                 5238*  5247  5385*  5394  5536*  5545  5820*  5829  5960*  5969  6119*  6128  6285*
                 6294  6831*  6839  6910*  6919  7052*  7059  7724#
DAREAD   052724  7773  7785#
DATENV   050476  1924  4486*  4630*  4780*  4935*  5087*  5232*  5379*  5530*  5677*  5814*  5954*  6113*
                 6279*  7328#  7994*  8011*  8054*  8070*
DAWORD   052730  4642*  4792*  4947*  5244*  5391*  5542*  5826*  5966*  6125*  6291*  6836*  6915*  7057*
                 7786#
DB       015022  1881  2497#
DCK    = 100000  2307#  4844  4996  5016  5441  5594  5614  6024  6176  6196  6342  6362  7175
                 7180  7196  7199
DCL    = 000100  2433#
DCLEAR   015156  2572#
DDISP  = 177570  487#  511  2625
DENVL  = 000200  2317#  4149
DE1    = 000040  2279#
DFF20  = 000002  2275#
DF1      012516  523  642  649  1935#
DF11     012534  631  1942#
DF14     012543  662  1945#
DF15     012553  671  1948#
DF16     012556  682  1949#
DF17     012563  695  1951#
```

```
DF2     012523          1937#
DF20    012572           708    1954#
DF21    012601           719    1958#
DF22    012606           736     750    1960#
DF24    012612           765     778    1962#
DF26    012621           791    1965#
DF27    012631           806    1968#
DF3     012527           611     618    1939#
DF30    012637           819    1970#
DF31    012645           574     582     949     957    1973#
DF32    012655           558     599    1976#
DF33    012666           541    1979#
DF34    012676           837    1982#
DF35    012705           856     928    1985#
DF36    012715           871     888    1988#
DF37    012723           911    1990#
DF40    012734           974    1993#
DF42    014730           980     987    1014    2213#
DF44    014733           994    1000    1007    1021    1028    1035    1042    1056    1063    1070    1077    1084    1091
                        1098    1105    1112    1119    1140    2214#
DF5   = 000001          2274#
DF54    014740          1049    1126    1133    2216#
DH1     005451           518     637    1464#
DH11    005701           623    1492#
DH14    006056           654    1511#
DH15    006255           668    1533#
DH16    006305           676    1538#
DH17    006443           687    1555#
DH2     005567           605    1478#
DH20    006620           700    1574#
DH21    006774           713    1592#
DH22    007110           730    1606#
DH24    007204           758     771    1617#
DH26    007362           782    1636#
DH27    007560           798    1658#
DH30    007716           812    1675#
DH31    010050           564     939    1691#
DH32    010246           547     588    1713#
DH33    010463           532    1737#
DH34    010662           827    1759#
DH35    011035           847     918    1777#
DH36    011232           862     879    1799#
DH37    011370           899    1816#
DH40    011607           966    1841#
DH42    014410           978     985    1012    2169#
DH44    014471           992     998    1005    1019    1026    1033    1040    1054    1061    1068    1075    1082    1089
                        1096    1103    1110    1117    1138    2179#
DH54    014603          1047    1124    1131    2193#
DIGB  = 000004          2276#
DISK    054566          1915    3816    3860    4436    4576    4606    4755    4804*   4911    4958*   5178    5208    5354
                        5403*   5506    5554*   5767    5791    5929    6089    6137*   6139*   6255    6303*   6304*   7039
                        7148    7550    7564*   7565*   8114    8165*   8615
DISPLA  001142           511#   2625*   8641*   8642*
DISPRE  000174           488#   2625
DLT   = 100000          2251#
DL64  = 000020          2278#
```

```
DMD    = 000001   2311#  2906  2956  3806  3954  4107  4149  7227  7250  7500  7703  7844  7850
                  7974   7987  7990  8004  8007  8041  8051  8067  8207  8359  8362  8371  8382
                  8533
DPR    = 000400   2282#  6700  6703  6750
DRY    = 000200   2281#  6700  6708  6745
DST      015036    1892  2505# 3984# 4152# 4282#
DSWR   = 177570     487#   511  2625
DS1      015054    1870  1873  1885  1888  1895  1905  1908  1912  1928  2512# 3982# 4150# 4280#
                  6698  6703  6708  6745  6750
DT       015056   2513#
DTAGAP   055572   8168#
DTE    = 010000   2304#  3980  4148  4278
DTSY   = 001000   2319#  7869  7942  7946
DT1      011764    522   641   648  1863#
DT11     012026    630  1870#
DT14     012046    661  1873#
DT15     012070    670  1876#
DT16     012100    681  1878#
DT17     012116    694  1881#
DT2      012000   1865#
DT20     012136    707  1885#
DT21     012156    718  1888#
DT22     012172    735   749  1890#
DT24     012204    764   777  1892#
DT26     012224    790  1895#
DT27     012246    805  1898#
DT3      012012    610   617  1867#
DT30     012264    818  1901#
DT31     012302    573   581   948   956  1905#
DT32     012324    557   598  1908#
DT33     012350    540  1912#
DT34     012372    835  1915#
DT35     012412    854   926  1918#
DT36     012434    869   886  1921#
DT37     012452    909  1924#
DT40     012476    973  1928#
DT42     014672    979   986  1013  2207#
DT44     014702    993   999  1006  1020  1027  1034  1041  1055  1062  1069  1076  1083  1090
                  1097  1104  1111  1118  1139  2209#
DT54     014716   1048  1125  1132  2211#
DVA    = 004000   2267#  6432  6675  6679  6738
ECDATA   050456   7309#  7351  7501# 7556# 7558# 7984# 7986# 8001# 8003# 8044# 8048# 8060# 8064#
ECH    = 000100   2298#  5016  5614  6196  6362
ECI    = 004000   2398#  3905  4211  6851  6932  7096
ECORR    051102   4857   5009  5454  5607  6037  6189  6355  7496#
ECTEST   050514   7348#  7529  7559  7995  8012  8055  8071
EC1      015062   1918   1921  1924  2515#
EC2      015064   1918   1921  1924  2516#  4129# 4693  4853  5005  5295  5450  5603  5877  6033
                  6185   6351
EMTVEC=  000030    487#  2625# 2658# 2686#
EM1      002136    517   796  1151#
EM11     002307    622   635   780  1170#
EM14     002342    653  1175#
EM15     002362    666  1178#
EM16     002432    675  1185#
EM17     002453    686  1188#
```

```
EM2     002221   530    545     562    1160#
EM20    002477   699   1192#
EM21    002520   712   1195#
EM22    002542   723   1198#
EM24    003143   755   1245#
EM25    003236   768   1255#
EM30    003276   811   1261#
EM31    003420   824   1275#
EM32    003543   841   1290#
EM33    004032   861   1323#
EM34    004104   874   1330#
EM35    004177   893   1340#
EM36    004476   916   1375#
EM37    004664   936   1395#
EM40    004752   963   1404#
EM42    012744   977   1998#
EM43    013027   984   2008#
EM44    013073   991   2016#
EM45    013130   997   2022#
EM46    013157  1004   2027#
EM47    013206  1011   2032#
EM50    013243  1018   2038#
EM51    013301  1025   2045#
EM52    013322  1032   2049#
EM53    013362  1039   2056#
EM54    013422  1046   2063#
EM55    013461  1053   2070#
EM56    013474  1060   2073#
EM57    013511  1067   2077#
EM6     002250   586    604    1164#
EM60    013540  1074   2082#
EM61    013627  1081   2093#
EM62    013677  1088   2101#
EM63    013754  1095   2110#
EM64    014032  1102   2119#
EM65    014066  1109   2125#
EM66    014150  1116   2135#
EM67    014222  1123   2143#
EM70    014307  1130   2153#
EM71    014361  1137   2162#
ERCRC   056026  8228*  8282#  8457*
ERFLGS  015124  2534#  3847#  3858   3868   3925*  4078#  4230#  4425#  4430   4524#  4535   4569#  4667#
                4683   4722#  4826#  4842   4882#  4978#  4994   5045#  5125#  5136   5171#  5269#  5285
                5323#  5423#  5439   5478#  5576#  5592   5642#  5715#  5726   5760#  5851#  5867   5904#
                6006#  6022   6062#  6158#  6174   6227#  6324#  6340   6391#  6888   7098#  7109   7769
                8319   8642#
ERHDGP  056030  8229#  8292#  8470#
ERHEAD  056024  8227#  8268#  8443#
ERPOS   051100  7492#  7498#  7514
ERR     =040000 2288#  3982   4150   4280
ERRVEC= 000004   487#  2625#  2660#  2667#  2688#  8641#
ERSTAR  052462  7609#
ERUNIT  052460  7608#  7610
ERWORD  052770  1865   1867   1905   1908   1912   2945   3862#  3867#  4026   4178   4306   6881#  6887#
                7831#  7871#  7896#  7900#  7924#  7948#  8391#  8398#  8402#  8424#  8433#  8440#  8456#
                8461#  8467#  8480#
```

```
ER1     015034  1870  1873  1885  1895  1905  1908  1912  1918  1928  2504#  3980*  4148*  4278*
                4685  4844  4996  5016  5287  5441  5594  5614  5869  6024   6176   6196   6342
                6362
ER2     015040  1885  2506#
ER3     015046  1885  2509#
EXT1  = 000001  2352#
EXT10 = 000010  2355#
EXT2  = 000002  2353#
EXT20 = 000020  2356#
EXT4  = 000004  2354#
EXT40 = 000040  2357#
E00     024176  2974  2987#
E11     025604  3220  3227#
E12     025732  3254  3260#
E44     030436  3760  3768#
FEN   = 000200  2378#
FER   = 000020  2296#
FILLEC  051254  4619  4768  4924  5221  5367  5519  5803  5942  6102  6268  7546#
FIRST   017332  2598#  2631  2641*
FMT22 = 010000  2399#  3822  3841  3887  3905  4041  4059  4194  4211  4349  4367  4404  4422
                4494  4510  4525  4636  4659  4709  4786  4818  4864  4941  4972  5026  5095
                5111  5126  5238  5261  5311  5385  5417  5461  5536  5568  5625  5685  5701
                5716  5820  5843  5893  5960  5998  6044  6119  6152  6206  6285  6316  6373
                6831  6851  6910  6932  7052  7096  7221
FNWORD  056046  4199*  4409*  4499*  5100*  5690*  8323#
FORMAT  054232  8094#  8107
FOUT    056052  8320  8326#
FSYNER  056022  8226*  8254#  8428*
FUTABL  015150  2568#
GCRC    056016  4202  4412  4502  5103  5693  8238#  8450  8455
GECC1   050460  1915  1918  1924  4484*  4541  4561  4628*  4691  4778*  4851  4933*  5003  5085*
                5142  5163  5230*  5293  5377*  5448  5528*  5601  5675*  5731  5752  5812*  5875
                5952*  6031  6111*  6183  6277*  6349  7313#  7349  7435*  7548*  7564
GECC2   050462  1915  1918  1924  4485*  4545  4562  4629*  4779*  4934*  5086*  5146  5164  5231*
                5378*  5529*  5676*  5735  5753  5813*  5953*  6112*  6278#  7316#  7350  7436*  7549*
                7565
GNS   = ****** U  488  2638  2639  2640  2649  2673  2683  2686  2723  2746  2787  2789  2791
                2852  2854  2881  2883  2885  2895  2898  3196  6464  6466  6526  6528  6531
                6534  7574  7576  7588  7590  7592  7594  7596  7598  7600  8631  8642
GO    = 000001  2261#  2917  7228  7251  7706  8210
GRV   = 000010  2277#
GTSWR = 104406  2643  8642#
HADTMP  050502  4483*  4627*  4777*  4932*  5084*  5229*  5376*  5527*  5674*  5811*  5951*  6110*  6276*
                7338#  7516
HARDER  050474  4483  4627  4777  4932  5084  5229  5376  5527  5674  5811  5951  6110  6276
                7325#
HCCRCE  047574  7080#
HCE   = 000200  2299#  7165  7169  7175  7180
HCI   = 002000  2397#
HCRC  = 000400  2300#  7187  7190  7196  7199
HDESYN  056032  8230*  8305#  8483*
HDWSYN  054564  8163#
HEADER  054540  8160#
HEDGAP  052664  7720*  7742#  7903*
HEDSYN  052666  7721*  7755#  7925*  7950*
HEGAP   054552  7894  8162#
```

```
HT     = 000011             487#  8641
IAE    = 002000            2302#
IE     = 000100            2262#  3457  3484  3731  3756  3763  6420  6445
ILF    = 000001            2292#
ILLEGL   015212            2586#
ILR    = 000002            2293#
IOTVEC= 000020              487#  2625*
IR     = 000100            2242#  4139  4144  4276
IXE    = 004000            2382#
I00      024152            2978#  2984
I02      024256            3015#
I03      024352            3034#
I04      024452            3056#
I05      024552            3078#
I06      024652            3101#
I07      024752            3123#
I14      026124            3310#
I15      026326            3365#
I16      026470            3407#
I24      027054            3500#
KEY1     052654            3825*  3891*  4045*  4640*  4790*  4945*  5242*  5389*  5540*  5824*  5964*  6123*  6289*
                           6834*  6913*  7055*  7726#
KEY2     052656            3826*  3892*  4046*  4641*  4791*  4946*  5243*  5390*  5541*  5825*  5965*  6124*  6290*
                           6835*  6914*  7056*  7727#
KIPAR0= 172340               509#
KIPAR1= 172342               509#
KIPAR2= 172344               509#
KIPAR3= 172346               509#
KIPAR4= 172350               509#
KIPAR5= 172352               509#
KIPAR6= 172354               509#
KIPAR7= 172356               509#
KIPDR0= 172300               509#
KIPDR1= 172302               509#
KIPDR2= 172304               509#
KIPDR3= 172306               509#
KIPDR4= 172310               509#
KIPDR5= 172312               509#
KIPDR6= 172314               509#
KIPDR7= 172316               509#
LA       015066            2517#  3983*  4151*  4281*
LAD      045744            6592*  6636#  6640
LERR     045526            6575#  6581*  6604
LF     = 000012             487#  8641
LST    = 002000            2284#
LST14A   026154            3306   3319#
LST15A   026356            3359   3377#
LST16A   026526            3399   3423#
L00      024146            2976#
L02      024254            3013#
L03      024346            3030#
L04      024444            3053#
L05      024542            3073#
L06      024644            3097#
L07      024742            3118#
L10      025076            3162#
```

```
L11      025542            3212#
L12      025670            3246#
L13      026002            3277#
L14      026120            3308#  3318
L15      026320            3361#  3374
L16      026460            3401#  3420
L21      026652            3457#
L22      026736            3479#
L30      027164            3523#
L31      027264            3547#
L34      027370            3573#
L35      027504            3599#
L36      027620            3626#
L40      027734            3654#
L41      030036            3676#
L42      030146            3704#
L43      030262            3728#
L44      030372            3753#
L45      030514            3786#
MAKECY   050336            2956   3806   7243#
MASK     045524            6574#  6578*  6594*  6595   6596*  6599*  6600   6601*
MCLK  =  000002            2312#  3955   3956   3958   3959   4108   4109   4111   4112   4704   4705   5306   5307
                           5888   5889   7527   7528   7535   7536   7846   7847   7850   7851   7854   7855   7979
                           7996   8041   8057   8362   8373   8559   8561   8562   8563   8564   8579   8580   8581
                           8582
MCPE  =  020000            2268#
MHS   =  001000            2380#
MID      050326            7223*  7230#
MIDDLE   050224            7214#
MINX  =  000004            2313#  7704   7705   8208   8209   8541   8542
MMVEC =  000250             509#
MOL   =  010000            2286#  2893   2896
MPE   =  000400            2244#
MR       015052            1921   2511#  3981*  4149*  4279*
MRD   =  000020            2315#  7977   7982   7990   7999   8007
MSE   =  000020            2375#
MSTCK =  000010            2314#  3955   3956   3968   3969   4108   4109   4121   4122   4257   4258   7845   7847
                           7850   7851   7979   8041   8361   8362   8534   8535   8536   8537   8541   8542   8558
                           8561   8562   8579   8580   8592   8593
MWR   =  000040            2316#  8042   8058   8364   8374
MXF   =  001000            2245#  3498   3500   3504
NCODE    050466            4482   4626   4776   4931   5083   5228   5375   5526   5673   5810   5950   6109   6275
                           7322#
NCOUNT   050470            4482*  4626*  4776*  4931*  5083*  5228*  5375*  5526*  5673*  5810*  5950*  6109*  6275*
                           7323#  7507*
NED   =  010000            2248#  3208   3217   3251   3288   3798   7612
NEM   =  004000            2247#
NHS   =  002000            2381#
NOPERA   015150            2569#
NOSYNC   052764            6538*  7774   7828#  7912   7940   8641*
NOUNIT   015114            2527#  2760*  2796*  2803   2835*  6480*
NOWORD   052716            3827*  7781#
NUNIT    015116            2529#  2803*  2804*
N11      025572            3205   3222#
N12      025720            3240   3256#
N14      026146            3314   3317#
```

```
N36      027660          3625    3637#
OCYL   = 100000          2435#
OF       015042          2507#
OFREV  = 000200          2396#
OFSETC   015202          2582#
OF100  = 000004          2391#
OF200  = 000010          2392#
OF25   = 000001          2389#
OF400  = 000020          2393#
OF50   = 000002          2390#
OF800  = 000040          2394#
OPERSE   044704          6524#   8641
OPI    = 020000          2305#
OR     = 000200          2243#   4139    4145    4277
OUT      052734          7770    7778    7783    7789#
000      024200          2985    2989#
010      025114          3160    3170#
021      026670          3455    3464#
043      030310          3726    3738#
044      030414          3751    3762#
PAR    = 000010          2295#
PAT    = 000020          2240#
PCJSR    015132          1895    1898    1901    1908    1912    1921    2543#   6672#   6673#   6727#   6728#   7080#   7081#
                         7244#   7245#   7496#   7497#   7700#   7701#   8204#   8205#
PGE    = 002000          2246#
PIE1   = 100000          7276#   7419    7434
PIE10  = 000100          7285#
PIE11  = 000040          7286#   7386
PIE12  = 000020          7287#   7419
PIE13  = 000010          7288#
PIE14  = 000004          7289#
PIE15  = 000002          7290#
PIE16  = 000001          7291#
PIE17  = 100000          7292#
PIE18  = 040000          7293#
PIE19  = 020000          7294#
PIE2   = 040000          7277#   7377
PIE20  = 010000          7295#
PIE21  = 004000          7296#   7394
PIE22  = 002000          7297#   7425
PIE23  = 001000          7298#   7402
PIE24  = 000400          7299#   7425
PIE25  = 000200          7300#
PIE26  = 000100          7301#
PIE27  = 000040          7302#
PIE28  = 000020          7303#
PIE29  = 000010          7304#
PIE3   = 020000          7278#   7419
PIE30  = 000004          7305#
PIE31  = 000002          7306#
PIE32  = 000001          7307#   7358    7366
PIE4   = 010000          7279#
PIE5   = 004000          7280#
PIE6   = 002000          7281#
PIE7   = 001000          7282#
PIE8   = 000400          7283#
```

```
PIE9   = 000200          7284#
PIP    = 020000          2287#
PIRQ   = 177772           487#
PIRQVE= 000240            487#
PKACK   015206          2584#  2908
PLU    = 020000          2384#
POSITI  050472          1918   1924   4481*  4625*  4775*  4930*  5082*  5227*  5374*  5525*  5672*  5809*  5949*
                        6108*  6274*  7324#  7459   7505   7513*  7514   7516   7547*
PRE    = 000020          2431#
PROG   = 001000          2283#  6699
PR0    = 000000           487#
PR1    = 000040           487#
PR2    = 000100           487#
PR3    = 000140           487#
PR4    = 000200           487#
PR5    = 000240           487#
PR6    = 000300           487#
PR7    = 000340           487#
PS     = 177776           487#  2628*  3702*  3729*  3754*  6419*  6444*  6463*  6525*
PSEL   = 002000          2266#  3480   3654   3659
PSU    = 000001          2428#
PSW    = 177776           487#
PUTREG  045464          3857   3935   4088   4240   4684   4723   4843   4883   4995   5015   5046   5286   5324
                        5440   5477   5593   5611   5641   5868   5905   6023   6061   6175   6195   6226   6341
                        6361   6392   6426   6431   6453   6560#  6674   6729   7108   7456   7462   7522
PWRVEC= 000024            487#  2625*  8642*
P12     050506          7340#  7388*  7389*  7418
P22     050510          7341#  7396*  7397*  7423
P24     050512          7342#  7404*  7405*  7424
P3      050504          7339#  7379*  7380*  7417
RA      000200           493#  7596
RCOM    057170          8604#  8609*
RCYL    052754          7801#  7836*  7876
RDCHR  = 104410          8641   8642#
RDHEAD  052772          7722   7836#
RDLIN  = 104411          8642#
RDOCT  = 104412          2649   6531   6534   7577   7590   8642#
RDY    = 000200          2263#  3731   6420   6432   6445   6675   6686   6730
READ    053564          7863   7868   7877   7879   7881   7883   7885   7935   7939   7972#  8617   8625
READAT  015174          2579#
READIN  015210          2585#
RECALI  015154          2571#
REDATA  057172          7785   8608#
REFOR   015176          2580#  3908   4666   4825   4977   5268   5422   5575   5850   6005   6157   6323
REGADR  045530          1863   1898   1901   2207   2209   2211   2713*  2947*  2973*  3010*  3027*  3050*  3070*
                        3094*  3115*  3155*  3200*  3236*  3285*  3305*  3358*  3398*  3450*  3476*  3497*  3520*
                        3544*  3570*  3602*  3629*  3651*  3673*  3699*  3721*  3747*  3795*  4028*  4180*  4308*
                        6576#  6580*  7168*  7179*  7189*  7200*  7261*
REGSAI  061616          2658   8642#
REINTO  016264          2594#  2913   2920*  2933   2939   3900   3978   4003   4160   4164*  4172   4289   4293*
                        4300   4557   4558   4561*  4562*  4564   4565   4575   4655   4731   4814   4890   4968
                        5053   5159   5160   5163*  5164*  5166   5167   5177   5257   5330   5413   5485   5564
                        5649   5748   5749   5752*  5753*  5755   5756   5766   5839   5911   5994   6069   6148
                        6234   6312   6398   6847   6928   7122   7147
RELEAS  015160          2573#
RESVEC= 000010            4C
```

```
RETCL   015204  2583#
RHAS    014774  2477#  2708   2723   3279   3788
RHBA    014752  2468#  3304   3357   3397   3838*  3902*  4056*  4208*  4417*  4508*  4655*  4814*  4968*
                5109*  5257*  5413*  5564*  5699*  5839*  5994*  6148*  6312*  6847*  6928*  7093*  7218*
RHBAE   015020  2490#  2681
RHCA    014770  2475#  3842*  3907*  4061*  4213*  4423*  4526*  4663*  4822*  4976*  5127*  5265*  5421*
                5572*  5717*  5847*  6002*  6156*  6320*  6854*  6935*  7085*  7222*  7247*
RHCC    015014  2485#  6563   7257   7261
RHCS1   014756  2470#  2908*  2917*  2971   3210   3244   3275   3449   3475   3601   3628   3650   3672
                3698   3720   3746   3784   3932   4085   4237   4419*  6654   7215*  7228*  7249*  7251*
                7574   7581   7584   7592   7706*  8210#
RHCS2   014754  2469#  2695   2724   2766*  2969*  3005*  3022*  3045*  3065*  3089*  3110*  3146*  3184*
                3199   3231*  3235   3268*  3272   3284   3300*  3353*  3393*  3445*  3473*  3493*  3496
                3515*  3519   3539*  3543   3565*  3569   3592*  3596   3615*  3619   3646*  3656*  3668*
                3678*  3694*  3718*  3744*  3777*  3781   3794   3846*  6655   7220*
RHCS3   015016  2489#  3150
RHDB    014746  2466#  2662   7578
RHDST   014762  2472#  3840*  3903*  4057*  4209*  4418*  4519*  4658*  4817*  4971*  5120*  5260*  5416*
                5567*  5710*  5842*  5997*  6151*  6315*  6850*  6931*  7090*  7219*  7248*
RHDS1   015000  2479#  6656
RHDT    015002  2480#  2767   2769   2773   2775   2778   2780   2789   2854   2859   2861   2864   2866
                2869   2871   2887   2889
RHEC1   015006  2482#  7459
RHEC2   015010  2483#  2983   3153   4129   7453
RHER1   014760  2471#  2727   6657   7164   7168   7174   7179   7186   7189   7194   7200
RHER2   014764  2473#
RHER3   014772  2476#
RHLA    015012  2484#
RHMR    014776  2478#  2906*  2956*  3806*  3953   4106   4704*  4705*  5306*  5307*  5888*  5889*  7227*
                7250*  7499   7703*  7704*  7705*  7842   8106   8207*  8208*  8209*  8358   8530
RHOF    014766  2474#  3841*  3905*  4059*  4211*  4422*  4525*  4659*  4818*  4972*  5126*  5261*  5417*
                5568*  5716*  5843*  5998*  6152*  6316*  6851*  6932*  7096*  7221*
RHSN    015004  2481#  2854   2886   2888
RHWC    014750  2467#  2912   2926   2947   3009   3026   3049   3069   3093   3114   3837*  3901*  3913
                3992   4028   4055*  4067   4159   4180   4207*  4219   4288   4308   4416*  4506*  4654*
                4813*  4967*  5107*  5256*  5412*  5563*  5697*  5838*  5993*  6147*  6311*  6561   6563
                6846*  6927*  7091*  7217*
RH70    017334  2601#  2646*  2682*  2694   3148   3493   3514   3591   3814   4134   4265
RH70CK  020720  2644#
RKEY1   052760  7803#  7838*  7880
RKEY2   052762  7804#  7839*  7882
RMR     =000004 2294#
RNCTR1  055630  8202#  8212*  8214*
RNO     057166  8603#  8608*
RNWAT1  055710  8212#
RPTRP1  044214  6417   6430#
RPTRP2  044320  6442   6452#
RPVEC   014744  2457#  2629*  3454   3455*  3464*  3725   3726*  3738*  3750   3751*  3770*  6416   6441
                7588   7591*  7594   8632*
RPVECT  057302  2629   8630#  8632
RSETR   052756  7802#  7837*  7878
RSYNC   052752  4366   4382   4389   7800#  7867   7872   7910   7922   7938   7949   8422   8426   8477
                8482
RTN     021356  2683   2686#
RUNCTR  052506  7699#  7708*  7709*
RUNWAT  052566  7708#
```

```
R11       025524            3207#
R15       026350            3369    3373#
R16       026520            3415    3419#
R44       030440            3766    3770#
SAVDT     015126            2536#   2889*
SAVER     046456            2911    2925    3912    3991    4066    4158    4218    4287    6812#
SAVSN     015130            2539#   2888*
SC      = 100000            2270#   3933    3979    4086    4147    4238    4275
SCOP1   = 104413            6605    8542#
SCRC      056100            8348#   8355*
SCYL      056070            8344#   8351*   8432
SC1     = 000100            2358#
SC10    = 001000            2361#
SC2     = 000200            2359#
SC20    = 002000            2362#
SC4     = 000400            2360#
SEARCH    056716            7229    7715    8220    8529#
SECGAP    054470            4338    4471    5072    5662    8158#   8357
SECOTR    052652            3824*   3889*   4043*   4638*   4639*   4788*   4789*   4943*   4944*   5240*   5241*   5387*   5388*
                            5538*   5539*   5822*   5823*   5962*   5963*   6121*   6122*   6287*   6288*   6832*   6833*   6911*
                            6912*   7053*   7054*   7712    7725#
SECTR     056714            8527#   8529*   8531
SEECOM    015200            2581#   7249
SEGPER    056020            8225*   8243#   8394*   8405*
SELECT    015120            2531#   2618*   2620*   2648    2807    2822    6470
SELTST    022622            2742    2807#
SERCH     015162            2574#
SETCK1    031334            3932#
SETCK2    032030            4085#
SETCK3    032554            4237#
SETDSK    047446            7036#
SILOTB    054566            8164#
SKEY1     056074            8346#   8353*
SKEY2     056076            8347#   8354*
SKI     = 040000            2434#
SN        015060            2514#
SND1      020662            2635    2641#
SR0     = 177572             509#
SR1     = 177574             509#
SR2     = 177576             509#
SR3     = 172516             509#
SSECTR    056072            8345#   8352*
SSYN      052662            7719*   7734#   7873*
SS1       033134            4337#   4461
SS10      033302            4330*   4391#   4393    4453*
SS12      033332            4331*   4406#   4454*
SS13      033340            4332*   4407#   4455*
SS14      033346            4333*   4408#   4456*
SS15      033406            4334*   4418#   4457*
SS2       033572            4459#
SS3       033162            4326*   4351#   4449*
SS4       033166            4327*   4352#   4450*
SS5       033172            4328*   4353#   4451*
SS7       033220            4329*   4368#   4452*
STACK   = 001000             487#   2625    2657    2693    2902    2956    2976    3007    3024    3047    3067    3091    3112
                            3162    3197    3233    3270    3302    3355    3395    3447    3479    3495    3517    3541    3567
```

```
                        3594    3617    3648    3670    3696    3728    3753    3779    3806    3813    3883    4038    4190
                        4317    4469    4597    4746    4902    5070    5199    5345    5497    5660    5783    5922    6081
                        6246    6413    6438
START   017362          2619    2623#
STARTA  017604          2628#
STKLMT= 177774           487#
SWR     001140           511#   2625*   2643    2644    2671    2721    2740    3877    6638    6894    7444    7446    8641#
                        8642#
SWREG   000176           488#   2625    2643    8641
SW0    = 000001          487#
SW00   = 000001          487#
SW01   = 000002          487#
SW02   = 000004          487#
SW03   = 000010          487#
SW04   = 000020          487#
SW05   = 000040          487#
SW06   = 000100          487#
SW07   = 000200          487#   3879    6896
SW08   = 000400          487#
SW09   = 001000          487#   6638
SW1    = 000002          487#
SW10   = 002000          487#
SW11   = 004000          487#
SW12   = 010000          487#   2644
SW13   = 020000          487#   2671    2721    2740
SW14   = 040000          487#
SW15   = 100000          487#
SW2    = 000004          487#
SW3    = 000010          487#
SW4    = 000020          487#
SW5    = 000040          487#
SW6    = 000100          487#   7446
SW7    = 000200          487#
SW8    = 000400          487#   7444
SW9    = 001000          487#
S11     025520          3204#   3225
S12     025654          3239#   3259
S36     027614          3624#   3639
TAGDTE  015146          2558#   4320*   4459*
TBITVE= 000014           487#
TDF    = 000040         2376#
TESDTE  015144          2554#   3924*   4077*   4229*   6547*   7887    8416    8641*
TESTAD  044702          6522#
TIMCNT  046334          6421    6446    6765#   6775    6781
TIMOT  = 000004         2960#   2973    2974*   2989*   3159    3160*   3170*
TKVEC  = 000060          487#   8641*
TMPILL  015140          2548#   3839*   3859    3865
TOLGAP  055574          3817    8101    8169#
TOTALA  015136          2546#   2733*   2735*   3186    3202    3237    3622
TPVEC  = 000064          487#
TRAPVE= 000034           487#   2625*
TRE    = 040000         2269#   3597    3605    3620    3632    3979    4147    4275
TRK     052622          7714*   7717#
TRK1   = 004000         2363#
TRK10  = 040000         2366#
TRK2   = 010000         2364#
```

```
TRK20 = 100000          2367#
TRK4  = 020000          2365#
TRP     021316          2680    2684#
TRTVEC= 000014           487#
TSECC   015142          2550#   4480#   4624#   4774#   4929#   5081#   5226#   5373#   5524#   5671#   5808#   5948#   6107#
                        6273#   6539#   8111    8610    8620    8641#
TSECCG  050464          6543#   7319#   7437    7980    7992    7997    8009    8052    8068    8113#   8126#   8612#   8622#
                        8641#
TSTNM   017330          1863    1865    1867    1870    1873    1876    1878    1881    1885    1888    1890    1892    1895
                        1898    1901    1905    1908    1912    1915    1918    1921    1924    1928    2207    2209    2211
                        2597#   2658#   2694#   2707#   2892#   2903#   2956#   2969#   3005#   3022#   3045#   3065#   3089#
                        3110#   3146#   3184#   3231#   3268#   3300#   3353#   3393#   3445#   3473#   3493#   3514#   3539#
                        3565#   3591#   3615#   3646#   3668#   3694#   3718#   3744#   3777#   3806#   3814#   3884#   4039#
                        4191#   4318#   4471#   4599#   4748#   4904#   5072#   5201#   5347#   5499#   5662#   5784#   5923#
                        6082#   6248#   6413#   6438#   6526
TST1    021034          2649    2657#   6497
TST10   024046          2956#
TST11   024102          2969#
TST12   024206          3005#
TST13   024300          3019    3022#
TST14   024376          3041    3045#
TST15   024474          3061    3065#
TST16   024576          3085    3089#
TST17   024674          3106    3110#
TST2    021372          2693#
TST20   024776          3130    3146#
TST21   025122          3184#
TST22   025606          3225    3231#
TST23   025734          3258    3268#
TST24   026054          3291    3300#
TST25   026254          3319    3353#
TST26   026414          3375    3393#
TST27   026566          3421    3445#
TST3    021450          2694    2707#
TST30   026676          3473#
TST31   026774          3487    3493#
TST32   027104          3493    3507    3514#
TST33   027220          3514    3531    3539#
TST34   027324          3557    3565#
TST35   027430          3583    3591#
TST36   027544          3591    3608    3615#
TST37   027670          3635    3636    3646#
TST4    021522          2712    2718#
TST40   027772          3662    3668#
TST41   030074          3684    3694#
TST42   030202          3712    3718#
TST43   030316          3744#
TST44   030446          3777#
TST45   030566          3801    3806#
TST46   030622          3812#
TST47   031154          3814    3859    3880    3883#
TST5    022636          2808    2811#   6478    6488
TST50   031646          3938    4038#
TST51   032372          4092    4190#
TST52   033032          4243    4317#
TST53   033606          4431    4461    4469#
```

```
TST54   034234          4537    4580    4597#
TST55   034752          4684    4735    4746#
TST56   035476          4843    4894    4902#
TST57   036240          4995    5057    5070#
TST6    023342          2892#
TST60   036664          5138    5182    5199#
TST61   037402          5286    5334    5345#
TST62   040126          5440    5489    5497#
TST63   040670          5593    5653    5660#
TST64   041314          5728    5771    5783#
TST65   042032          5868    5915    5922#
TST66   042566          6023    6073    6081#
TST67   043344          6175    6238    6246#
TST7    023704          2894    2902#
TST70   044122          6341    6402    6412#
TST71   044234          6429    6433    6437#
TST72   044330          6451    6461#
TTNO  = 000071          2658#   2694#   2707#   2892#   2903#   2956#   2969#   3005#   3022#   3045#   3065#   3089#   3110#
                        3146#   3184#   3231#   3268#   3300#   3353#   3393#   3445#   3473#   3493#   3514#   3539#   3565#
                        3591#   3615#   3646#   3668#   3694#   3718#   3744#   3777#   3806#   3814#   3884#   4039#   4191#
                        4318#   4471#   4599#   4748#   4904#   5072#   5201#   5347#   5499#   5662#   5784#   5923#   6082#
                        6248#   6413#   6438#
TUF   = 000100          2377#
TY      052766          7830#
TYPDS = 104405          2788    2795    2853    6465    6467    6476    6497    8642#
TYPE  = 104401          2638    2639    2640    2649    2673    2683    2686    2723    2746    2787    2789    2791    2793
                        2852    2854    2881    2883    2885    2895    2898    3196    3843    3908    4062    4214    4427
                        4432    4521    4551    4665    4708    4824    4863    4977    5025    5122    5152    5267    5310
                        5422    5460    5574    5624    5712    5741    5849    5892    6004    6043    6157    6205    6322
                        6372    6464    6466    6474    6477    6497    6526    6528    6530    6531    6534    6855    6936
                        7083    7574    7576    7588    7590    7592    7594    7596    7598    7600    8631    8641    8642#
TYPOC = 104402          2676    2790    2854    6527    6529    7575    7589    7593    7595    7597    8631    8641    8642#
TYPON = 104404          8642#
TYPOS = 104403          8642#
T.SCOP  045746          6638#   8642
UN      021440          2695*   2697#
UNIT    015112          2526#   2651*   2802*   2808*   2817    2834*   2836    2852    6464    6482    6487*   6660
UNITS   015072          2525#   2754    2758    2802    2829    6483
UNITSL  015122          2532#   2652*   2808
UNLOAD  015152          2570#
UNS   = 040000          2306#
UPE   = 020000          2249#   3521    3528    3547    3554    3599
US1   = 000001          2236#   3217    3251    3273    3277    3288    3573    3580    3782    3786    3798
US2   = 000002          2237#   3217    3251    3273    3277    3288    3573    3580    3782    3786    3798
US4   = 000004          2238#   3217    3251    3273    3277    3288    3573    3580    3782    3786    3798
UWR   = 000010          2430#
U11     025476          3187    3197#
VUF   = 000002          2429#
VU3D  = 010000          2383#
VV    = 000100          2280#   2920    6699
WAIT.T  046336          6767#   8642
WAT   = 104415          8642#
WC      015024          1881    2498#   3914    3977*   4019    4068    4137*   4142*   4171    4220    4268*   4272*   4299
                        6562
WCE   = 040000          2250#
WCF   = 000040          2297#
```

```
WCRC    054550          3831    3896    4050    4646    4796    4951    5248    5395    5546    5830    5970    6129    6295
                        6840    6920    7060    7884    8161#
WCU   = 000001          2371#
WCYL    056006          4194*   4201    4404*   4411    4494*   4501    5095*   5102    5685*   5692    8234#
WECC1   055566          1915    4541    5142    5731    8166#
WECC2   055570          1915    4545    5146    5735    5979    5982*   5983*   8167#
WKEY1   056012          4197*   4407*   4497*   5098*   5688*   8236#
WKEY2   056014          4198*   4408*   4498*   5099*   5689*   8237#
WLE   = 004000          2303#
WORD    053560          7862*   7867*   7876*   7878*   7880*   7882*   7884*   7934*   7938*   7966#   7975*   7988*   8005*
                        8616*   8624*
WRCHDA  047022          6910#
WRCHDT  015166          2576#   6855
WRCHEK  015164          2575#   6936
WRCHHD  046510          6831#
WRDATA  054236          7780    8096#   8321
WRFROM  015220          2593#   2927    2933*   2940    3993    3997*   4020    4056    4138    4143    4208    4269    4273
                        4346    4372    4373    4417    4435    4507    4708    4730    4863    4878*   4889    5025    5041*
                        5052    5108    5310    5329    5460    5476*   5484    5624    5640*   5648    5698    5892    5910
                        6043    6068    6205    6221*   6223*   6233    6372    6388*   6389*   6397    7121    7218
WRHEAD  056102          8232    8351#
WRIDAT  015170          2577#   3843    4062
WRIFOR  015172          2578#   4214    4419    4522    5123    5713    7215
WRITE   054016          7897    7909    8037#   8115    8122    8128    8132    8135    8399    8419    8434    8448    8462
                        8475
WRL   = 004000          2285#
WRU   = 000400          2379#
WSECTR  056010          4196*   4406*   4496*   5097*   5687*   8217    8235*
WSSYNC  054536          8159#
WSU   = 000004          2373#
WTRK    055744          8219*   8223#
WWORD   054014          7898    7902    7906    7910    7918    7923    7928    8032#   8077*   8116    8123    8129    8133
                        8136    8400    8404    8421    8435    8449    8463    8476
X       052660          3828*   3893*   4047*   4643*   4793*   4948*   5245*   5392*   5543*   5827*   5967*   6126*   6292*
                        6837*   6916*   7094*   7111    7728#   7772
XE2     022264          2738    2752#
X11     025600          3193    3224#
Y       052720          7782#
Y11     025172          3192    3195#
ZCODE   050500          1924    4487*   4631*   4781*   4936*   5088*   5233*   5380*   5531*   5678*   5815*   5955*   6114*
                        6280*   7331#   7510*
ZER   = 000400          2318#   7519
ZWORDS  054234          8095#   8109*
$AUTOB  001134           511#   2643*   8641
$BDADR  001122           511#   1870    1873    1895    1928    6681*   6688*   6692*   6705*   6710*   6713*   6733*   6741*
                        6747*   6752*
$BDDAT  001126           511#   1863    1867    1873    1878    1890    1892    1898    1901    1905    1908    2209    2710*
                        2711    3016*   3018    3038*   3040    3058*   3060    3082*   3084    3103*   3105    3127*   3129
                        3156*   3216*   3217*   3219    3250*   3251*   3253    3287*   3288*   3290    3311*   3313    3366*
                        3368    3411*   3414    3451*   3483*   3484*   3486    3502*   3504*   3506    3527*   3528*   3530
                        3553*   3554*   3556    3579*   3580*   3582    3604*   3605*   3607    3631*   3632*   3634    3658*
                        3659*   3661    3680*   3681*   3683    3708*   3709*   3711    3722*   3762*   3763*   3765    3797*
                        3798*   3800    3866*   6598*   6600*   6602    6786*   6886*   7164*   7165    7174*   7175    7186*
                        7187    7194*   7196    7257*   7258    7902*   7923*   8393*   8404*   8427*   8442*   8454*   8469*
                        8481#
$BELL   001216           511#   8641    8642
```

```
$CHARC  060322          8641#*
$CKSWR  060634          8641#   8642
$CMTAG  001100           511#   2625
$CM1   = 000006          511#
$CM2   = 000014          511#
$CM3   = 000006          511#
$CM4   = 000006          511#
$CNTLC  061405          8641#
$CNTLG  061417          8641#
$CNTLU  061412          8641#
$CRLF   001223           511#   2793    6530    8641    8642
$DBLK   060076          8641#
$DOAGN  044656          6497#
$DTBL   060066          8641#
$ENDAD  044646           489   6497#
$ENDCT  044614          6497#
$ENDMG  044665          6474   6497#
$ENULL  044662          6477   6497#
$EOP    044560          2678   2747    6481    6497#
$EOPCT  044606          6497#
$ERFLG  001103           511#   8641*   8642*
$ERMAX  001115           511#   2625*   8641*
$ERROR  061606          2625   2686    8642#
$ERRPC  001116           511#   1863    1865    1867    1870    1873    1876    1878    1881    1885    1888    1890    1892
                        1895   1898    1901    1905    1908    1912    1915    1918    1921    1924    1928    2207    2209
                        2211   8642*
$ERRTB  001226           511#   8642
$ERRTY  061754          8642#
$ERTTL  001112           511#   6466    6468*   8642*
$ESCAP  001214           511#   2625*   8641*   8642
$FILLC  001156           511#   8641
$FILLS  001155           511#   8641
$GDADR  001120           511#
$GDDAT  001124           511#   1863    1865    1867    1890    1898    1901    1905    1908    1912    2209    2211    2712*
                        3011*   3018    3028*   3040    3051*   3060    3071*   3084    3095*   3105    3116*   3129    3157*
                        3207*   3208*   3219    3242*   3253    3273*   3290    3308*   3310    3313    3363*   3365    3368
                        3403*   3407    3412*   3414    3452*   3460*   3477*   3486    3498*   3506    3521*   3523    3530
                        3545*   3556    3571*   3582    3597*   3607    3620*   3634    3652*   3661    3674*   3683    3700*
                        3711    3723*   3735*   3748*   3765    3782*   3800    3865*   6593*   6595*   6597    6602    6885*
                        7169*   7180*   7190*   7199*   7260*   7872*   7901*   7920*   7922*   7949*   8392*   8403*   8426*
                        8441*   8455*   8468*   8482*
$GET42  044636          6497#
$GTSWR  060724          8641#   8642
$HD    = 000000           467
$HIOCT  061604          8642#*
$ICNT   001104           511#   8641*
$ILLUP  062612          8642#
$INTAG  001135           511#   8641*
$ITEMB  001114           511#   8642*
$LF     001224           511#   8641    8642
$LPADR  001106           511#   2625*   2811*   6533*   6536    8641*
$LPERR  001110           511#   2625*   6528    6535*   8641*   8642
$MAIL  = ******  U      2625    2643    8641    8642
$MNEW   061435          8641#
$MSWR   061424          8641#
$MXCNT  057660          8641#
```

```
$NULL   001154            511#  8641
$NWTST= 000001           2657#  2693#  2707#  2718#  2811#  2892#  2902#  2956#  2969#  3005#  3022#  3045#  3065#
                         3089#  3110#  3146#  3184#  3231#  3268#  3300#  3353#  3393#  3445#  3473#  3493#  3514#
                         3539#  3565#  3591#  3615#  3646#  3668#  3694#  3718#  3744#  3777#  3806#  3812#  3883#
                         4038#  4190#  4317#  4469#  4597#  4746#  4902#  5070#  5199#  5345#  5497#  5660#  5783#
                         5922#  6081#  6246#  6412#  6437#  6461#
$OCNT   062352           8642#*
$OMODE  062354           8642#*
$OVER   057644           8641#
$PASS   001100            511#  3189   6473*  6475   6497*  8641
$POWER  062620           8642#
$PWRAD  062606           8642#
$PWRDN  062446           2625   8642#
$PWRMG  062602           8642#
$PWRUP  062520           8642#
$QUES   001222            511#  8641   8642
$RDCHR  061176           8641#  8642
$RDDEC= ****** U         8642
$RDLIN  061266           8641#  8642
$RDOCT  061446           8642#
$RDSZ = 000011           8641#
$REGAD  001160            511#
$REG0   001162            511#
$REG1   001164            511#
$REG2   001166            511#
$REG3   001170            511#
$REG4   001172            511#
$REG5   001174            511#
$RTNAD  044660           6497#
$R2A  = ****** U         8642
$SAVRE= ****** U         8642
$SAVR6  062616           8642#*
$SCOPE  057370           2625   8641#
$SETUP= 000117           2624#  2625   2643   6497   8641   8642
$SS1  = 000000           2626#
$STUP = 177777           2624#
$SVLAD  057616           8641#
$SVPC = 000200            489#
$SWR  = 167770            467#   485    511   2625   2657   2693   2707   2718   2811   2892   2902   2956   2969
                         3005   3022   3045   3065   3089   3110   3146   3184   3231   3268   3300   3353   3393
                         3445   3473   3493   3514   3539   3565   3591   3615   3646   3668   3694   3718   3744
                         3777   3806   3812   3883   4038   4190   4317   4469   4597   4746   4902   5070   5199
                         5345   5497   5660   5783   5922   6081   6246   6412   6437   6461   6497   8641   8642
$SWRMK= 000000            485   8641
$TIMES  001212            511#  2625*  2657*  2693*  2707*  2718*  2811*  2969*  3005*  3022*  3045*  3065*  3089*
                         3110*  3146*  3184*  3231*  3268*  3300*  3353*  3393*  3445*  3473*  3493*  3514*  3539*
                         3565*  3591*  3615*  3646*  3668*  3694*  3718*  3744*  3777*  6461*  6497*  8641*
$TKB    001146            511#  8641
$TKCNT  060326           8641#*
$TKINT  060346           2630   2719   7576   8641#
$TKQEN= 060345           8641#
$TKQIN  060330           8641#*
$TKQOU  060332           8641#*
$TKQSR  060334           8641#
$TKS    001144            511#  3838   3839   8641*
$TKSRV  060416           8641#
```

```
$TMP0   001176          511#  1878  2762*  2763*  2973*  2989   3159*  3170   3214*  3248*  3281*  3454*  3464
                       3706*  3725*  3738   3750*  3770   3932*  3933   4085*  4086   4237*  4238   6771*  6777
                       6782   6786   6876*  6890   6974*  7020*
$TMP1   001200          511#  1876   1878   1892   2669*  6421*  6422*  6446*  6447*  6772*  6777   6782   6877*
                       6892
$TMP2   001202          511#  1892   6775*  6779*  6781*  6784*  6976*  7018*
$TMP3   001204          511#  1878   1892   6769*  6770*  6975*  6977   6978*
$TMP4   001206          511#  6977*  7001*
$TMP5   001210          511#  6973*  6979*  6985   7083*  7162   7184
$TN   = 000073          467#  2649   2657#  2658   2693#  2694   2707#  2712   2718#  2808   2811#  2892#  2894
                       2902#  2903   2956#  2969#  3005#  3019   3022#  3041   3045#  3061   3065#  3085   3089#
                       3106   3110#  3130   3146#  3184#  3224   3231#  3258   3268#  3291   3300#  3319   3353#
                       3375   3393#  3421   3445#  3473#  3487   3493#  3507   3514#  3531   3539#  3557   3565#
                       3583   3591#  3608   3615#  3635   3636   3646#  3662   3668#  3684   3694#  3712   3718#
                       3744#  3777#  3801   3806#  3812#  3814   3859   3880   3883#  3884   3938   4038#  4039
                       4092   4190#  4191   4243   4317#  4318   4431   4461   4469#  4471   4537   4580   4597#
                       4599   4684   4735   4746#  4748   4843   4894   4902#  4904   4995   5057   5070#  5072
                       5138   5182   5199#  5201   5286   5334   5345#  5347   5440   5489   5497#  5499   5593
                       5653   5660#  5662   5728   5771   5783#  5784   5868   5915   5922#  5923   6023   6073
                       6081#  6082   6175   6238   6246#  6248   6341   6402   6412#  6413   6429   6433   6437#
                       6436   6451   6461#
$TPB    001152          511#  8641*
$TPFLG  001157          511#  8641
$TPS    001150          511#  8641
$TRAP   062356         2625   8642#
$TRAP2  062400         8642#
$TRP  = 000016         8642#
$TRPAD  062412         8642#
$TSTNM  001102          511#  6469*  6497*  8641*  8642
$TTYIN  061374         8641#
$TYPBN= ****** U       8642
$TYPDS  057662         8641#  8642
$TYPE   060106         8641#  8642
$TYPEC  060256         8641#
$TYPEX  060324         8641#
$TYPOC  062154         8642#
$TYPON  062170         8642#
$TYPOS  062130         8642#
$XTSTR  057422         8641#
$$GET4= 000000         6497#
$OFILL  062353         8642#*
$40CAT= ****** U       8641   8642
.     = 062630          488#   489#   492#   510#   511#  1997#  2205#  2525#  2592#  2593#  2594#  2625   2638#
                       2639#  2640#  2683#  2686#  2723#  2746#  2787#  2789#  2791#  2852#  2881#  2883#  2885#
                       2895#  2898#  3196#  6464#  6466#  6497   6531#  6534#  7588#  7596#  7600#  8158#  8159#
                       8160#  8161#  8162#  8163#  8165#  8166#  8167#  8168#  8169#  8631#  8641#  8642#
```

```
CHECKA   484#  4432  4550  4708  4863  5024  5151  5310  5460  5623  5740  5892  6043  6204  6371
CHECKB   484#  3843  3908  4062  4214  4427  4521  4665  4824  4977  5122  5267  5422  5574  5712
         5849  6004  6157  6322  6855  6936  7083
COMMEN    469   487#
ENDCOM    472   487#
ERROR     487#  2670  2699  2714  2949  2987  3019  3041  3061  3085  3106  3130  3168  3227  3260
         3291  3315  3371  3417  3462  3487  3507  3531  3557  3583  3608  3635  3662  3684  3712
         3736  3768  3801  3870  3873  3937  4008  4029  4090  4181  4241  4309  4441  4443  4543
         4547  4580  4582  4687  4695  4735  4737  4846  4855  4894  4896  4998  5007  5019  5057
         5059  5144  5148  5182  5184  5289  5297  5334  5336  5443  5452  5489  5491  5596  5605
         5617  5653  5655  5733  5737  5771  5773  5871  5879  5915  5917  6026  6035  6073  6075
         6178  6187  6199  6238  6240  6344  6353  6365  6402  6404  6427  6433  6454  6682  6689
         6693  6706  6711  6714  6734  6742  6748  6753  6787  7127  7129  7153  7155  7170  7181
         7191  7201  7262  7457  7463  7523
ESCAPE    487#
GETPRI    487#
GETSWR    467#   487#  2643
HCOMPR    484#
HCOMPW    484#
MAKECL    484#  2956  3806
MSG      2656#  2657  2692#  2693  2810#  2811  2892#  2901#  2902  3812#  3882#  3883  4037#  4038  4189#
         4190  4317#  4468#  4469  4596#  4597  4745#  4746  4901#  4902  5069#  5070  5198#  5199  5344#
         5345  5496#  5497  5659#  5660  5782#  5783  5921#  5922  6080#  6081  6245#  6246  6411#  6412
         6436#  6437  6461#
MULT      487#
NEWTST    487#  2657  2693  2707  2718  2811  2892  2902  2956  2969  3005  3022  3045  3065  3089
         3110  3146  3184  3231  3268  3300  3353  3393  3445  3473  3493  3514  3539  3565  3591
         3615  3646  3668  3694  3718  3744  3777  3806  3812  3883  4038  4190  4317  4469  4597
         4746  4902  5070  5199  5345  5497  5660  5783  5922  6081  6246  6412  6437  6461
POP       487#  6567  6634  6819  6900  7023  7061  7231  7264  7482  7538  7566  7789  7951  8018
         8078  8139  8326  8485  8597  8628  8641  8642
PUSH      487#  6560  6624  6812  6872  6970  6972  7036  7214  7243  7348  7499  7546  7702  7841
         7972  8037  8100  8206  8356  8530  8610  8641  8642
REPORT    487#
RFORGC    484#  4580  4735  4894  5057  5182  5334  5489  5653  5771  5915  6073  6238  6402
RH7DCK    484#  2694  3493  3514  3591  3814
SAVE      484#  8642
SAVTST    484#  2658  2694  2707  2892  2903  2956  2969  3005  3022  3045  3065  3089  3110  3146
         3184  3231  3268  3300  3353  3393  3445  3473  3493  3514  3539  3565  3591  3615  3646
         3668  3694  3718  3744  3777  3806  3814  3884  4039  4191  4318  4471  4599  4748  4904
         5072  5201  5347  5499  5662  5784  5923  6082  6248  6413  6438
SCOPE     487#  2657  2693  2707  2718  2811  2892  2902  2956  2969  3005  3022  3045  3065  3089
         3110  3146  3184  3231  3268  3300  3353  3393  3445  3473  3493  3514  3539  3565  3591
         3615  3646  3668  3694  3718  3744  3777  3806  3812  3883  4038  4190  4317  4469  4597
         4746  4902  5070  5199  5345  5497  5660  5783  5922  6081  6246  6412  6437  6461  6497
SETPRI    487#  8641
SETTRA   8642#
SETUP     487#  2625
SKIP      484#   487#  2649  2712  2808  2894  3019  3041  3061  3085  3106  3130  3224  3258  3291
         3319  3375  3421  3487  3507  3531  3557  3583  3608  3635  3636  3662  3684  3712  3801
         3859  3880  4431  4461  4537  4684  4843  4995  5138  5286  5440  5593  5728  5868  6023
         6175  6341  6429  6433  6451
SLASH     487#
SMORE     484#  8641
SPACE     487#
STARS     478   483   487#  489   511  2221  2222  2255  2256  2657  2693  2707  2718  2773  2782
```

```
                2811    2840    2849    2859    2885    2892    2902    2956    2969    3005    3022    3045    3065    3089    3110
                3146    3184    3231    3268    3300    3353    3393    3445    3473    3493    3514    3539    3565    3591    3615
                3646    3668    3694    3718    3744    3777    3806    3812    3883    4038    4190    4317    4469    4597    4746
                4902    5070    5199    5345    5497    5660    5783    5922    6081    6246    6412    6437    6461    6497    6502
                6520    6552    6559    6571    6573    6611    6623    6667    6670    6723    6726    6760    6764    6804    6811
                6826    6828    6864    6871    6904    6906    6944    6969    7026    7035    7065    7079    7207    7213    7235
                7242    7270    7272    7486    7490    7542    7544    7570    7572    7617    7675    7677    7793    7816    7822
                7823    7961    7962    8025    8026    8085    8086    8151    8176    8177    8329    8330    8495    8496    8601
                8602    8630    8634    8641    8642
SWRSU            487#   2625#
TJUMP            484#   2694    3493    3514    3591    3814    3938    4092    4243
TRMTRP          8642#
TSCLR            484#
TSCLR1           484#
TSCLR2           484#
TSCLR3           484#
TSCLR4           484#
TSCLR5           484#
TYPBIN           487#
TYPDEC           487#   6497    8642
TYPNAM           487#
TYPNUM           487#
TYPOCS           487#
TYPOCT           487#   2854    8641    8642
TYPTXT           487#   2637    2639    2640    2649    2673    2683    2686    2723    2745    2746    2787    2789    2791    2852
                2854    2880    2882    2884    2895    2898    3195    3196    6464    6466    6526    6528    6531    6534    7573
                7576    7588    7590    7592    7594    7596    7598    7600    8630
WDATAR           484#
XREAD            484#
$$CMRE           511#
$$CMTM           511#
$$ESCA           487#
$$NEWT           487#   2657    2693    2707    2718    2811    2892    2902    2956    2969    3005    3022    3045    3065    3089
                3110    3146    3184    3231    3268    3300    3353    3393    3445    3473    3493    3514    3539    3565    3591
                3615    3646    3668    3694    3718    3744    3777    3806    3812    3883    4038    4190    4317    4469    4597
                4746    4902    5070    5199    5345    5497    5660    5783    5922    6081    6246    6412    6437    6461
$$SET           8642#
$$SKIP           487#
.EQUAT           467#    487
.FLOAT           484#
.HEADE           467#
.KT11            467#    509
.SETUP           467#   2623
.SWRHI           467#    485
.SWRLO           467#    485#
.$ACTI           467#    489
.$CATC           467#    488
.$CMTA           467#    511
.$EOP            467#   6497
.$ERRO           467#   8642
.$ERRT           467#   8642
.$POWE           467#   8642
.$RDOC           467#   8642
.$READ           467#   8641
.$SCOP           467#   8641
.$TRAP           467#   8642
```

```
.$TYPD    467#   8641
.$TYPE    467#   8641
.$TYPO    467#   8642

. ABS.  062630     000

 ERRORS DETECTED:  0

 RM03:CZRJHB,CZRJHB.SEQ/CRF/SOL/NL:MC:ME:CND=RM03:CZRJHB.P11,CZRJHB.P12
 RUN-TIME:  31  25  2 SECONDS
 RUN-TIME RATIO:  484/59=8.1
 CORE USED:  29K  (57 PAGES)
```

20N