RP04/5/6

RP04/5/6 RD/WT

CZRJADO

AH 9182D MC

FICHE 1 OF 2

NOV 1979

digital

MADE IN USA

RP04/5/6

RP04/5/6 RD/WT

CZRJAD0

AH 9182D MC

FICHE 2 OF 2

NOV 1979

digital

MADE IN USA

.REM @

I D E N T I F I C A T I O N

PRODUCT CODE: AC-9180D-MC

PRODUCT NAME: CZRJADO RP04/5/6 MECHANICAL AND READ-WRITE TEST

DATE: MAY, 1979

MAINTAINER: DIAGNOSTIC ENGINEERING

AUTHOR: C. HESS

CONTENTS

1. ABSTRACT

THIS PROGRAM CONTAINS A SERIES OF TESTS THAT WILL VERIFY THAT THE DISK IS CAPABLE OF PERFORMING SEEKS, THAT THE ACCESS TIMES ARE WITHIN TOLERANCE, THAT THE TRACK AND SECTOR ADDRESSING CIRCUITRY OPERATES PROPERLY, AND THAT THE DATA STORAGE AND RETRIEVAL CAPABILITIES ARE FUNCTIONING.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP-11 PROCESSOR
16K MEMORY
TELETYPE
PROGRAM LOADING DEVICE
KW11-L OR KW11-P (THE KW11-P IS REQUIRED FOR THE TIMING TESTS)
RH11 OR RH70 WITH 1 - 8 RP04/5/6 DISK DRIVES

2.2 PRELIMINARY PROGRAMS

RP04/5/6 DISKLESS CONTROLLER TEST
PART 1 (MAINDEC-11-DZRJG)
PART 2 (MAINDEC-11-DZRJH)

RP04/5/6 FUNCTIONAL CONTROLLER TEST
PART 1 (MAINDEC-11-DZRJI)
PART 2 (MAINDEC-11-DZRJJ)

2.3 MEDIA

THE PROGRAM REQUIRES THAT EACH DRIVE TO BE TESTED HAS A FORMATTED DISK PACK. THE PACK MAY BE FORMATTED IN EITHER 16-BIT OR 18-BIT MODE, DEPENDING ON THE TESTING REQUIREMENTS. NOTE THAT THE PROGRAM WILL NOT TEST A MIXTURE OF DRIVES WITH BOTH 16 AND 18 BIT MODE PACKS.

2.4 PROGRAMMABLE DRIVES (DUAL PORT ENABLED)

THIS REV INCORPORATES A SAFEGUARD TO PREVENT INADVERTENT CORRUPTION OF DISK PACKS IN PROGRAMMABLE DRIVES.
THIS IS A POTENTIAL HAZARD IN RUNNING THIS PROGRAM IN A MULTIPROCESSOR SYSTEM. FOR THE STANDARD STARTING ADDRESS OF 200 THE PROGRAM HAS BEEN MODIFIED TO PREVENT INITIALIZING DRIVES FOUND TO BE PROGRAMMABLE. THIS MODIFICATION APPLIES ONLY TO THE FIELD ENVIRONMENT (XXDP CHAIN, STANDALONE) WHERE LOCATION 42 DOES NOT EQUAL LOCATION 46. FOR THE MANUFACTURING ENVIRONMENT (WHERE LOCATION 42 EQUALS LOCATION 46) PROGRAMMABLE DRIVES WILL NOT BE INHIBITED. IF THE OPERATOR DESIRES TO RUN THIS PROGRAM USING PROGRAMMABLE DRIVES IN A FIELD ENVIRONMENT USE STARTING ADDRESS 220, WHERE 220 IS

THE SAME AS 200 WITHOUT INHIBITING PROGRAMMABLE DRIVES.
SEE SECTION 4.1 FOR A SUMMARY OF ALL STARTING ADDRESSES.

3. LOADING PROCEDURE

THE PROGRAM MAY BE LOADED FROM PAPER TAPE USING THE ABSOLUTE LOADER OR IT MAY BE LOADED FROM THE APPROPRIATE 'XXDP' MEDIA USING THE ASSOCAITED LOADER. THE PROGRAM MAY BE INCLUDED IN AN 'XXDP' CHAIN. IF THE PROGRAM IS BEING RUN ON A PROCESSOR WITH 16K, THE 'XXDP' LOADER WILL NOT BE PRESERVED. THE PROGRAM MUST BE RUN ON A SYSTEM WITH 20K OR MORE TO PRESERVE THE 'XXDP' LOADER. THE 'ABSOLUTE' LOADER WILL BE PRESERVED IN A 16K SYSTEM, HOWEVER.

4. STARTING PROCEDURE

4.1 STARTING ADDRESSES

200 NORMAL STARTING ADDRESS (INHIBIT PROGRAMMABLE DRIVES-SEE SECTION 2.4)
204 SELECT OPERATING PARAMETERS (INHIBIT PROGRAMMABLE DRIVES)
210 SELECT RH11-RH70 ADDRESSES (DO NOT INHIBIT PROGRAMMABLE DRIVES)
214 COMBINATION OF 204 AND 210 (DO NOT INHIBIT PROGRAMMAGBLE DRIVES)
220 SAME AS 200 BUT WITH NO INHIBITIONS
224 SAME AS 204 BUT WITH NO INHIBITIONS

NOTE: STARTING ADDRESSES 210 AND 214 ARE AVAILABLE WHEN THE PROGRAM IS INITIALLY STARTED; THESE STARTING ADDRESSES ARE TREATED AS ADDRESSES 200 OR 204 RESPECTIVELY ON RESTARTS.

4.2 OPERATOR ACTION

1. LOAD PROGRAM INTO MEMORY (SEE SECTION 3.)
2. LOAD A FORMATTED PACK INTO DRIVE(S) TO BE TESTED
3. BRING DRIVE(S) TO ONLINE STATE, WRITE ENABLED, AND LOCKED ON PORT.
4. LOAD ADDRESS 200.
5. SET SWITCHES (SEE SECTION 5.)
6. PRESS START.
7. THE PROGRAM WILL TYPEOUT THE STATUS OF THE DRIVES ATTACHED TO THE SELECTED MASSBUS SUBSYSTEM. TO INHIBIT THIS TYPEOUT, DO NOT RESTART THE PROGRAM FROM ANY OF THE STARTING ADDRESSES; INSTEAD TYPE A 'CONTROL C' ON THE KEYBOARD TO RETURN THE PROGRAM TO COMMAND ENTRY MODE.

4.3 PROGRAM ACTION

IN AN EFFORT TO ALLOW CONVERSATION WITH A PROGRAM FOR THE PURPOSE OF CONTROLLING ITS OPERATION AND PARAMETERS THE FOLLOWING CONSTRUCTIONS HAVE BEEN ADOPTED.

NOTE1: IN ALL EXAMPLES BRACKETS ARE USED FOR CLARITY AND ARE NOT TYPED BY THE USER.

NOTE2: THE CARRRIAGE RETURN TYPED BY THE USER IS INDICATED BY <CR> AND WILL BE ECHOED AS A 'CARRIAGE RETURN-LINE FEED'.

<.><CR> PERIOD

A STATEMENT TERMINATOR: WHEN TYPED AT THE END OF A LINE (LEGAL ON ALL LINES) IT TELLS THE PARAMETER STRING INTERPRETER (PSI) THIS IS THE END OF CHANGES TO THE CURRENT PARAMETER STRING.

<..><CR> PERIOD PERIOD

THE 'PERIOD PERIOD' TERMINATOR IS TYPED TO INDICATE THE END OF TEST PARAMETER MODIFICATION AND TO SIGNAL THE START OF TEST EXECUTION.

<,><CR> COMMA

THE COMMA IS USED AS A SEPARATOR BETWEEN DRIVE NUMBERS AND TEST NUMBERS.

</> SLASH

A MODIFICATION INDICATOR: IF A SLASH FOLLOWS A TEST NUMBER, THE PROGRAM WILL OPEN THAT TEST FOR PARAMETER MODIFICATION.

<^U> CONTROL-U

DELETE THE PRESENT INPUT STRING AND START A NEW LINE. TYPED BY DEPRESSING THE "CONTROL KEY" (CTRL) AND THEN STRIKING THE "U".

<\> RUBOUT

DELETE THE LAST CHARACTER FROM THE INPUT STRING. TYPED BY STRIKING THE "RUBOUT" KEY. WHICH WILL BE ECHOED BY A BACKSLASH (\) FOLLOWED BY THE CHARACTER DELETED.

4.3.1 CONTROL SWITCH SELECTION

STARTING THE PROGRAM AT ANY OF THE POSSIBLE STARTING ADDRESSES WITH SW<07>=1 WILL RESULT IN ENTERING THE "CONTROL SWITCH SETTING" MODE. THUS, ALLOWING THE OPERATOR TO SPECIFY THE DESIRED STATE OF "C.SWR".

CONTROL SWITCH SELECTION EXAMPLES:

EXAMPLE #1

```
SET SW<07>=0
C.SWR=000000 / 400..
```

EXAMPLE #2

```
SET SW<07>=0
C.SWR=000000 / 220.
C.SWR=000000 / 220..
```

4.3.2 RH11 - RH70 ADDRESS SELECTION

STARTING THE PROGRAM AT 200 WILL RESULT IN AUTOMATIC SELECT OF THE DEFAULT VALUES OF BUS ADDRESS (RPCS1), VECTOR ADDRESS, AND PRIORITY LEVEL OF THE RH11-RH70. IF THE DEFAULT VAULE OF THE BUS ADDRESS DOES NOT RESPOND (TIMES OUT) WHEN ADDRESSED, AN ERROR IS REPORTED. AFTER THE ERROR IS REPORTED ONE OF TWO COURSES OF ACTION WILL BE TAKEN:

1. IF THERE IS A MONITOR -- RETURN TO THE MONITOR

2. IF THERE ISN'T A MONITOR -- ASK FOR NEW ADDRESSES

STARTING THE PROGRAM AT 210 OR 214 ALLOWS THE OPERATOR TO CHANGE THE ADDRESS OF THE RH11 OR RH70 AND THE VECTOR ADDRESS.

THE PROGRAM ALLOWS THE ADDRESSES TO BE CHANGED ON WHEN THE PROGRAM IS FIRST STARTED. STARTING ADDRESSES 210(8) AND 214(8) ARE TREATED AS ADDRESSES 200(8) OR 204(8) RESPECTIVELY.

ADDRESS SELECTION EXAMPLES

EXAMPLE #1

```
RPCS1=176700 / 177200.
```

EXAMPLE #2

```
RPCS1=176700 / 176300<CR>
RHVEC=254 / 260<CR>
RHPRIO=5 / 6.
```

EXAMPLE #3

```
RPCS1=176700<CR>
RHVEC=254 / 260.
```

EXAMPLE #4

```
RH11/RP04 FAILED TO RESPOND TO ADDRESSING
RPCS1   ERR PC
176300  XXXXXX
RPCS1=176300 / 176700.
```

EXAMPLE #5

```
RPCS1=176700 / 1776\67\6300<CR>
RHVEC=254<CR>
RHPRIO=5<CR>
RPCS1=176300.
```

4.3.3 DRIVE AND PARAMETER SELECTION

STARTING THE PROGRAM AT 200 OR 210 WILL RESULT IN AUTOMATIC SELECTION OF THE DRIVES TO TEST AND THE TESTS TO RUN.

STARTING THE PROGRAM AT 204 OR 214 ALLOWS THE OPERATOR TO SELECT THE DRIVE(S) TO BE TESTED, THE TESTS TO BE EXECUTED, AND THE PARAMETERS TO USE.

EACH TEST CONTAINS TWO SETS OF CYLINDER LIMIT PARAMETERS. PARAMETERS 'LC' AND 'FC' ARE USED BY RP04/5 DRIVES AND PARAMETERS 'LC'' AND 'FC'' ARE USED BY RP06 DRIVES. THE PROGRAM DETERMINES WHICH DRIVE IS BEING TESTED AND SELECTS THE CORRECT SET OF CYLINDER LIMIT VALUES. IF THE PROGRAM IS BEING USED TO TEST A SUBSYSTEM WHICH CONTAINS BOTH RP04/5 AND RP06 DRIVES, THE OPERATOR MUST CHANGE BOTH SETS OF CYLINDER LIMITS IF THE TESTS ARE TO BE MODIFIED FOR ALL DRIVES TESTED.

4.3.3.1 DRIVE AND PARAMETER SELECTION DESCRIPTION

THE FOLLOWING IS A TABLE OF TERMS USED BY THE PSI.

''R''	REPEATS (ITERATIONS)
''FC''	FIRST CYLINDER ADDRESS FOR RP04/5'S
''LC''	LAST CYLINDER ADDRESS FOR RP04/5'S
''FC'''	FIRST CYLINDER ADDRESS FOR RP06'S
''LC'''	LAST CYLINDER ADDRESS FOR RP06'S
''IC''	INCREMENT CYLINDER
''FT''	FIRST TRACK ADDRESS
''LT''	LAST TRACK ADDRESS
''IT''	INCREMENT TRACK
''FS''	FIRST SECTOR ADDRESS
''LS''	LAST SECTOR ADDRESS
''PAT''	PATTERN (USED FOR DATA TEST)
''WDX''	WORD OF PATTERN 0 WHERE X IS 1 TO 16
*''S''	ALL SEEK TESTS (TESTS 0 - 10)
*''T''	ALL TIMING TESTS (TESTS 12 - 15)
*''A''	ALL ADDRESS TESTS (TESTS 16 - 17)
*''D''	THE DATA TEST (TEST 20)
*''E''	THE EXERCISER (TEST 21)

* USED BY THE OPERATOR TO SELECT TEST GROUPS

NOTE: ALL NUMBERS WILL BE IN DECIMAL EXCEPT FOR THE PATTERN (PAT) AND WORDS (WDX) SELECTION. ''PAT'' WILL BE SELECTED BY A BIT (I.E. 001000(8)=PATTERN 9) AND ''WDX'' WILL BE IN OCTAL.

SPECIAL CASES OF CONTROL CHARACTERS

IF <..> IS TYPED WHILE A TEST IS OPEN FOR MODIFICATION
(</>) AND OTHER TESTS IN THE "TEST COMMAND" STRING
ARE TO BE MODIFIED, THE REMAINING TESTS WILL BE UNCHANGED.

WHEN THE PROGRAM IS STARTED FROM LOCATION 200 OR 210, TESTS 0-10,
12-20 WILL BE RUN USING ALL AVAILABLE, ONLINE DRIVES. IF THE OPERATOR
WISHES TO SELECT THE DRIVES TO BE TESTED, THE TESTS TO BE PERFORMED,
OR THE PARAMETERS TO BE USED, THE CONVERSATION MODE MAY BE ENTERED
BY TYPING A 'CONTROL C' OR BY STARTING THE PROGRAM FROM EITHER
LOCATION 204 OR 214.

THE PROGRAM WILL THEN RESPOND WITH:

```
DRIVE(S)=
```

THE FOLLOWING EXAMPLES ASSUME THAT THE OPERATOR IS TO TEST
DRIVE #3 USING TESTS 2 THRU 7 AND TEST 11 AND DOES NOT DESIRE TO CHANGE
THE PARAMETERS (INITIAL CYLINDER ADDRESS, FINAL CYLINDER ADDRESS, ETC.).
THE USER WOULD TYPE '3<CR>' WHICH SAYS 'THIS IS THE END OF DRIVE ENTRY'.
THE PROGRAM WILL THEN REQUEST TEST NUMBERS.

THE TRANSACTION APPEARS AS FOLLOWS:

```
DRIVE(S)=3<CR>
TEST=
```

THE OPERATOR MAY NOW ENTER DESIRED TEST NUMBERS. IN THE EXAMPLE,
HE WANTS TESTS 2 THRU 7 AND TEST 11 SO HE TYPES 2-7<,> (THE 'COMMA'
SEPARATES ENTRIES), 11<.><CR> ('PERIOD' 'CARRIAGE RETURN' -
END OF CHANGES, START TEST EXECUTION.)

IT NOW LOOKS LIKE THIS

```
DRIVE(S)=3<CR>
TEST=2-7,11.<CR>
```

THE NEXT EXAMPLE, IT IS ASSUMED THAT THE OPERATOR WISHES TO TEST
DRIVE 4 AND TO RUN TESTS 1 AND 3 THRU 11, MODIFYING THE PARAMETERS
FOR TESTS 3 AND 10.

THE TRANSACTION WOULD BE AS FOLLOWS:

```
DRIVE(S)=4<CR>
TEST=
```

THE OPERATOR NOW ENTERS THE TEST NUMBERS. THE TRANSACTION IS
GIVEN BELOW:

```
DRIVE(S)=4<CR>
TEST=1,3/4-7,10/11<CR>
```

NOTICE THIS SAYS SELECT TEST 1, CONTINUE<,>; SELECT TEST 3, OPEN</>;
SELECT TESTS 4-7, CONTINUE<,>; SELECT TEST 10, OPEN</>; SELECT TEST
11, END OF INPUT <.>.

THE PROGRAM SCANS THE TEST NUMBER INPUT AND DETERMINES THAT THE
PARAMETERS FOR TEST 3 AND TEST 10 ARE TO BE CHANGED. THE OTHER
TESTS WILL NOT BE ALTERED.

(THE ENTIRE TRANSACTION IS REPEATED FOR CLARITY)

```
DRIVE(S)=4<CR>
TEST=1,3/4-7,10/11<CR>
  TEST 3
    R=X /                ;WHERE X IS ITERATION
```

THE NEW VALUE FOR 'R' MAY BE ENTERED. TERMINATING THE ENTRY WITH
A <.> (PERIOD) WILL TERMINATE THE CHANGES FOR THIS TEST; TYPING A
<CR> OR TERMINATING THE ENTRY WITH A <CR> WILL CAUSE THE PROGRAM
TO MOVE TO THE NEXT PARAMETER.

```
DRIVE(S)=4<CR>
TEST=1,3/4-7,10/11<CR>
  TEST 3
    R=1 / <CR>           ;DO NOT ALTER-BUT CONTINUE
    FC=N /               ;WHERE 'N' IS FIRST CYLINDER ADDRESS
```

IF THE OPERATOR DOES NOT WISH TO CHANGE 'FC', THE FOLLOWING OCCURS:

```
DRIVE(S)=4<CR>
TEST=1,3/4-7,10/11.<CR>
  TEST 3
    R=1 / <CR>           ;DO NOT ALTER THIS LINE BUT CONTINUE
    FC=0 / <CR>          ;DO NOT ALTER THIS LINE BUT CONTINUE
    LC=410 /
```

THE PROGRAM RESPONDS WITH THE PREVIOUSLY ASSIGNED PARAMETER FOR LAST
CYLINDER ADDRESS IN THIS CASE USING 410 AS THE EXAMPLE. THIS IS
WHAT THE OPERATOR INTENDED TO MODIFY AND IS WHY TEST 3 WAS
OPENED. TO CHANGE THE VALUE TO '20', THE NEW VALUE IS TYPED
FOLLOWED BY A 'PERIOD' TERMINATOR (<.><CR>).

THE TOTAL TRANSACTION AND RESPONSE:

```
DRIVE(S)=4<CR>
TEST=1,3/4-7,10/11<CR>
  TEST 3
    R=1 / <CR>
    FC=0 / <CR>
    LC= 410 / 20.<CR>
  TEST 10
    R=1 /
```

THE PROGRAM HAS LOADED TEST 3 WITH ITS NEW PARAMETERS AND THE
PROGRAM IS WAITING FOR CHANGES TO TEST 10'S PARAMETERS.

```
DRIVE(S)=4<CR>
TEST=1,3/4-7,10/11<CR>
  TEST 3
    R=1 / <CR>
```

```
                    FC=0 / <CR>
                    LC= 410 / 20.<CR>
                  TEST 10
                    R=1 / 10.<CR>
```

THE OPERATOR TYPES THE NEW VALUE (10) AND TERMINATES THE ENTRY WITH A 'PERIOD' 'CARRIAGE RETURN'.

THE PROGRAM NOW LOADS TEST 10 WITH THE NEW PARAMETERS (TEST 11 RETAINS THE PREVIOUSLY ASSIGNED PARAMETERS) AND RESPONDS WITH:

```
        DRIVE(S)=
```

SINCE THE USER DID NOT END THE CONVERSATION MODE WITH A 'PERIOD PERIOD', THE PROGRAM HAS LOOPED BACK TO THE BEGINNING LOOKING FOR MORE CHANGES. THAT IS TO SAY, AFTER THE ENTRY FOR DRIVE SELECTION, A <,><CR> WILL CAUSE THE TEST MESSAGE TO BE REPEATED AND FURTHER CHANGES CAN BE MADE. HOWEVER, AT SOME POINT IN ORDER TO EXECUTE THE PROGRAM, A 'PERIOD PERIOD' MUST BE TYPED.

IF A SINGLE 'PERIOD' IS TYPED WHILE DRIVE OR TEST NUMBERS ARE BEING ENTERED, THE PROGRAM WILL START EXECUTION IMMEDIATELY. A 'PERIOD PERIOD' MUST BE TYPED BEFORE THE PROGRAM WILL EXIT TEST PARAMETER CHANGE MODE TO GO TO EXECUTION.

4.3.3.2 DRIVE AND PARAMETER SELECTION EXAMPLES

EXAMPLE #1

```
DRIVE=4.<CR>              ;SELECT DRIVE #4, TERMINATE AND
                          ;BEGIN EXECUTION USING PREVIOUSLY ASSIGNED
                          ;PARAMETERS
```

EXAMPLE #2

```
DRIVE=0<CR>               ;SELECT DRIVE #0 AND MAKE CHANGES ","
TEST=1-5.<CR>             ;RUN TEST 1 THRU 5 ONLY, USE DEFAULT
                          ;PARAMETERS AND TERMINATE AND EXECUTE."
```

EXAMPLE #3

```
DRIVE=2<CR>               ;SELECT DRIVE #2 AND MAKE CHANGES ","
TEST=1-5,6/7/10/<CR>      ;RUN TEST 1-5 WITH DEFAULT PARAMETERS, OPEN
  TEST 6                  ;TEST 6,7 AND 10 FOR CHANGES
    R=1 / <CR>            ;LEAVE 'R' AS IS AND MOVE TO NEXT PARAMETER
    FC=0 / 10.<CR>        ;SET 'FC' CYLINDER ADDRESS TO 10, END CHANGES
                          ;TO TEST 6.

  TEST 7
    R=1 / 50<CR>          ;50 ITERATIONS, MOVE TO NEXT PARAMETER
    FC=0 / <CR>           ;DO NOT CHANGE 'FC' CYLINDER ADDRESS BUT CONTINUE
    LC=410 / 50..<CR>     ;TEST 10 IS STILL PENDING AND WILL BE
```

```
                                    ;RETAIN ITS PRESENT PARAMETERS.


                    EXAMPLE #4
                    ----------

                    DRIVE=0<CR>             ;SELECT DRIVE #0 AND MAKE CHANGES
                    TEST=S,E.<CR>           ;RUN ALL SEEK TESTS AND THE EXERCISER

                    EXAMPLE #5
                    ----------

                    DRIVE=1<CR>
                    TEST=S/D<CR>            ;RUN ALL SEEK TESTS (OPEN FOR CHANGES) AND
                      TEST 0                ;THE DATA TEST (WITH DEFAULT PARAMETERS).
                        R=10 / <CR>         ;RUN WITH 10 ITERATIONS
                        FC=0 / 10..<CR>     ;CHANGE FIRST CYLINDER ADDRESS
                                            ;AND START EXECUTION
                                            ;TESTS 1 - 10 WILL RETAIN THEIR PREVIOUSLY
                                            ;ASSIGNED PARAMETERS.

                    EXAMPLE #6
                    ----------

                    DRIVE=1<CR>
                    TEST=S/<CR>             ;OPEN THE SEEK TESTS (TESTS 0-10)
                      TEST 0
                        R=10 / 100.<CR>     ;CHANGE TO 100 ITERATIONS, TO TO THE NEXT TEST
                      TEST 1
                        R=100 / 1000.<CR>   ;CHANGE 'R' TO 1000 ITERATIONS, MOVE TO NEXT TEST
                      TEST 2
                        R=1 / 10<CR>        ;CHANGE 'R' TO 10 ITERATIONS, GO TO NEXT PARAMETER
                        FC=0 / 50<CR>       ;CHANGE 'FC' TO 50, GO TO NEXT PARAMETER
                        LC=410 / 51.<CR>    ;CHANGE 'LC' TO 51, GO TO THE NEXT TEST
                      TEST 3
                        R=1.<CR>            ;MOVE TO NEXT TEST
                      TEST 4
                        R=1..<CR>           ;USE TEST 4'S PARAMETERS AND START PROGRAM EXECUTION

                    EXAMPLE #7
                    ----------

                    DRIVE=1<CR>
                    TEST=D/<CR>             ;SELECT AND OPEN THE DATA TEST
                      TEST 20
                        R=1 / 1000<CR>      ;DO 1000 ITERATION OF TEST PATTERN
                        FC=0 / 10<CR>       ;#8 ON CYLINDER 10, TRACK 2, SECTOR 4
                        LC=410 / 10<CR>
                        FC'=0 / <CR>        ;RP06 PARAMETER
                        LC'=814 / <CR>      ;RP06 PARAMETER
                        IC=64 / 0<CR>
                        FT=0 / 2<CR>
                        LT=18 / 2<CR>
                        IT=1 / <CR>
                        FS=0 / 4<CR>
                        LS=22 / 4<CR>
```

```
        PAT=177777 / 400..<CR> ;RUN WITH PATTERN #8

EXAMPLE #8
----------

DRIVE=1<CR>             ;USE THE SAME PARAMETERS AS IN EXAMPLE
TEST=D/<CR>             ;#7, BUT ALSO SPECIFY A DATA PATTERN (PAT #0).
  TEST 20
    R=1000 / <CR>
    FC=10 / <CR>
    LC=10 / <CR>
    FC'=0 / <CR>
    LC'=814 / <CR>
    IC=0 / <CR>
    FT=2 / <CR>
    LT=2 / <CR>
    IT=1 / <CR>
    FS=4 / <CR>
    LS=4 / <CR>
    PAT=000400 / 401<CR> ;RUN WITH PATTERNS #8 & #0 (0=OPERATOR INPUT)
      WD1=165555 / 125252<CR> ;FIRST WORD OF PATTERN 0
      WD2=133333 / 52525..<CR> ;SECOND WORD OF PATTERN 0
                        ;<..> START EXECUTION

EXAMPLE #9
----------

DRIVE=0,1,4<CR>         ;TEST DRIVES 0,1, AND 4 IN SEQUENCE
TEST=0-5/<CR>           ;CHANGE TEST 5
  TEST 0
    R=10 / <CR>
    FC=0 / <CR>
    LC=410 / 1<CR>      ;CHANGE LAST CYLINDER FROM 410 TO 1
    FC'=0 / <CR>
    LC'=814 / 2..<CR>   ;CHANGE THE LAST CYLINDER FOR ALL RP06'S TO
                        ;2.  START PROGRAM EXECUTION.
```

5. SWITCH SETTINGS

5.1 OPERATIONAL SWITCH SETTINGS

WITH SW<15:0>=0 THE PROGRAM WILL PRINT OUT ON
ERRORS AND CONTINUE IN TEST.
THE SWITCH SETTINGS ARE:

SW<15>=1...HALT ON ERROR
SW<14>=1...LOOP ON TEST
SW<13>=1...INHIBIT ERROR TYPEOUTS
SW<11>=1...INHIBIT ITERATIONS
SW<10>=1...RING BELL ON ERROR
SW<09>=1...LOOP ON ERROR
SW<08>=1...PRINT ERROR MESSAGE ON LINE PRINTER

SW<07>=1...READ CONTROL SWITCH SETTINGS FROM TTY
SW<06>=1...INHIBIT TIME REPORTS (TESTS 12-15)
SW<05>=1...REPORT ONE ERROR PER SECTOR (TESTS 16 & 17)
SW<04>=1...INHIBIT WRITES (TEST 20)
SW<03>=1...INHIBIT WRITE CHECKS (TEST 20)
SW<02>=1...INHIBIT READ AND SOFTWARE COMPARES (TEST 20)
SW<01>=1...INHIBIT SOFTWARE COMPARES (TEST 20)
SW<00>=1...PERFORM READ AFTER WRITE CHECK ERROR (TEST 20)

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 A 'SOFTWARE' SWITCH REGISTER. THE 'SOFTWARE' SWITCH REGISTER IS LOCATED AT LOCATION 176 (8). THE SETTINGS OF THE 'SOFTWARE' SWITCHES ARE CONTROLLED THROUGH A KEYBOARD ROUTINE WHICH IS CALLED BY TYPING A 'CONTROL G'. THE PROGRAM WILL RECOGNIZE THE 'CONTROL G' AT ANY TIME EXCEPT WHEN THE PROGRAM IS AT A HIGHER PRIORITY PROCESSING AN RP04/5/6 INTERRUPT. THE 'SOFTWARE' SWITCH VALUES ARE ENTERED AS AN OCTAL NUMBER IN RESPONSE TO THE PROMPT FROM THE SWITCH ENTRY ROUTINE:

'SWR = NNNNNN   NEW ='

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

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

5.2 CONTROL SWITCH SETTINGS

THE CONTROL SWITCH SETTINGS ARE ENTERED THROUGH THE KEYBOARD.

TO ENTER THE CONTROL SWITCH SETTING MODE PLACE SW<07>=1 BEFORE PRESSING START. THEN UPON STARTING THE PROGRAM IT WILL TYPE THE PRESENT CONTENTS OF THE CONTROL SWITCH REGISTER (C.SWR) AND WAIT FOR THE NEW SETTING TO BE INPUT. THE INPUT STRING MUST CONSIST OF 1 TO 6 OCTAL DIGITS, TWO PERIODS (..), AND A CARRIAGE RETURN.

THE C.SWR SETTINGS ARE:

C.SWR<15>=0...WRITE PACK BEFORE TESTING (TEST16)
=1...INHIBIT WRITE PACK BEFORE TESTING (TEST16)
C.SWR<14>=0...NO STALL BETWEEN DRIVE FUNCTIONS
=1...STALL AFTER EVERY DRIVE FUNCTION
C.SWR<13>=0...USE SPECIFIC STALL TIMES
=1...USE RANDOM STALL TIMES
C.SWR<12>=0...NO INCREMENTING STALLS IN TEST4
=1...PERFORM INCREMENTING STALLS IN TEST4
C.SWR<08>=0...DO IMPLIED SEEKS WITH DATA TRANSFERS
=1...DO EXPLICIT SEEKS BEFORE DATA TRANSFERS

```
C.SWR<07>=0...DO READ HEADER AND  DATA COMMANDS IN TESTS 0-6
         =1...DO EXPLICIT SEEK COMMANDS IN TESTS 0-6
C.SWR<06>=0...60 HZ POWER SOURCE
         =1...50 HZ POWER SOURCE
C.SWR<05>=0...ALLOW SOFTWARE TIMEOUTS(ENABLE WATCHDOG TIMER)
         =1...INHIBIT SOFTWARE TIMEOUTS(DISABLE WATCHDOG TIMER)
C.SWR<00>=0...OPERATE IN 22 SECTOR (16 BIT) MODE
         =1...OPERATE IN 20 SECTOR (18 BIT) MODE

THE DEFAULT CONDITION OF C.SWR<15:00>=0.

REFER TO 4.3.1 FOR C.SWR SELECTION
```

6. ERRORS

THERE ARE ANUMBER OF ERRORS THAT CAN OCCUR IN THIS PROGRAM. WHEN AN ERROR IS ENCOUNTERED, THE CALL TO THE ERROR ROUTINE IS MADE AND IF SW<13> IS NOT SET, AN ERROR MESSAGE PERTAINING TO THE ERROR WILL BE TYPED. EACH ERROR TYPEOUT WILL CONTAIN THE FOLLOWING:

1. AN ERROR MESSAGE
2. A DATA HEADER
3. A DATA STRING

REFER TO THE FOLLOWING SECTION FOR THE DIFFERENT ERRORS THAT CAN OCCUR.

6.1 ERROR TYPES

THE ERRORS THAT OCCUR IN THIS PROGRAM FALL INTO THREE (3) CATEGORIES DEFINED AND EXPLAINED AS FOLLOWS:

6.1.1 DRIVER ERROR

THESE ERRORS WILL BE DETECTED BY THE RH11/RP04/5/6 DRIVER. THERE ARE TWO CLASSES OF DRIVER ERRORS; THOSE THAT CAN NOT BE IDENTIFIED IN A MANNER THAT ALLOWS THE INFORMATION TO BE RETURNED TO A "DATA PARAMETER BLOCK" (DPB) AND THOSE THAT CAN. THE FIRST CLASS WILL BE REPORTED BY ERROR CALLS (EMT'S) 1-5 WITHIN THE DRIVER. THE SECOND CLASS WILL PASS THE ERROR CODES TO THE STATUS/ERROR WORD (DPB+16) OF THE PROPER DPB.

6.1.2 NON-FATAL ERRORS

THESE ERRORS WILL BE DUE TO "DISK" OR "DATA" FAILURES WHICH WILL BE REPORTED AS THEY OCCUR. AFTER REPORTING THE ERROR THE PROGRAM WILL CONTINUE TESTING.

6.1.3 FATAL ERRORS

THIS TYPE OF ERROR WILL BE THE RESULT OF ANY KIND
OF ERROR THAT INHIBITS THE PROGRAM FROM TESTING THE DISK.

THIS ERROR WILL BE REPORTED WHEN IT OCCURS, THEN THE PROGRAM
WILL ABORT THE TEST AND GO TO THE END OF PROGRAM.

6.2 ERROR RECOVERY

6.2.1 PRETEST ERROR

WHEN THIS TYPE OF ERROR OCCURS IT WILL BE REPORTED. THEN DEPENDING
ON HOW THE PROGRAM WAS STARTED IT WILL ASK FOR THE DRIVES AND
ADDRESSES FOR TESTING OR RETURN TO MONITOR.

6.2.2 NON-FATAL ERROR

WHEN THIS TYPE OF ERROR OCCURS IT WILL BE REPORTED AND
THE PROGRAM WILL CONTINUE IN TEST.

6.2.3 FATAL ERROR

WHEN THIS TYPE OF ERROR OCCURS IT WILL BE REPORTED. THE
PROGRAM WILL ABORT THE TEST AND GO TO THE END OF PROGRAM.

7. RESTRICTIONS

THE PROGRAM WILL TEST THE DRIVES IN EITHER 16 BIT MODE OR IN 18
BIT MODE DEPENDING ON THE SETTING OF 'S.SWR<00>'. IF 'C.SWR<00>'
IS 0, ALL OF THE DRIVES WILL BE TESTED IN 16 BIT MODE; IF 'C.SWR<00>
IS 1, ALL OF THE DRIVES WILL BE TESTED IN 18 BIT MODE. THE PROGRAM
HAS NO PROVISIONS FOR TESTING DRIVES WITH INTERMIXED PACKS OR TESTING
BOTH 16 BIT MODE AND 18 BIT MODE DRIVES ON THE SAME SYSTEM.
ACT11 AUTOMATIC MODE ASSUMES 16 BIT MODE.

BEFORE THE PROGRAM IS STARTED, PROPERLY FORMATTED PACKS MUST BE MOUNTED
ON THE DRIVES WHICH WILL BE TESTED. THE PROGRAM ASSUMES A
PROPERLY FORMATTED PACK. THE FORMAT OF THE PACK IS NOT ALTERED
BY THE PROGRAM.

THERE IS A RESTRICTION ON PROGRAMMABLE DRIVES IN CERTAIN SITUATIONS SEE SECTIONS 2.4 AND 4.1.

8. MISCELLANEOUS

8.1 EXECUTION TIME

THE PROGRAM REQUIRES APPROXIMATELY 15 MINUTES TO MAKE ONE PASS WITH
RP04/5 DRIVES AND APPROXIMATELY 16.5 MINUTES TO A PASS WITH RP06
DRIVES. THIS ASSUMES THE DEFAULT TEST SEQUENCE (TESTS 0-10, 12-20)

AND DEFAULT TEST PARAMETERS.

8.2 STACK POINTER

THE STACK POINTER IS INITIALLY SET TO 1100.

8.3 TIMING TESTS (TESTS 12-15) PRINTOUTS

AT THE COMPLETION OF EACH OF THE TIMING TESTS THE TIME OF THE MINIMUM SEEK, MAXIMUM SEEK, AND THE AVERAGE OF ALL OF THE SEEKS PERFORMED ARE TYPED ON THE TTY. THE NUMBER OF SEEKS THAT HAD TIMES BELOW THE MINIMUM TIME ALLOWED WILL BE TYPED ON THE SAME LINE AS THE MINIMUM TIME. THE NUMBER ABOVE THE MAXIMUM WILL BE TYPED ON THE SAME LINE AS THE MAXIMUM TIME, AND THE TOTAL NUMBER OF SEEKS PERFORMED WILL BE ON THE SAME LINE AS THE AVERAGE.

NOTE: THE PROGRAM STALLS FOR 2 MILLISECONDS BETWEEN SEEK ORDERS, THIS STALL TIME IS NOT INCLUDED IN THE CALCULATED SEEK TIMES. THE 2 MILLISECOND STALL BETWEEN SEEK ORDERS IS SPECIFIED BY THE RP04 VENDOR. THE SEEK TIMES SPECIFIED FOR THE RP04 ARE POSITIONER MOVEMENT TIMES ONLY AND ARE NOT A MEASUREMENT OF EFFECTIVE SEEK TIME.

8.3.1 TIMING TOLERANCES

1. TEST 12 -- ROTATIONAL SPEED TIMES

   60 HZ
   MINIMUM=16340 US
   MAXIMUM=17000 US
   NOMINAL=16670 US

   50 HZ
   MINIMUM=16250 US
   MAXIMUM=17090 US
   NOMINAL=16670 US

2. TEST 13 -- ONE CYLINDER SEEK TIMES

   MAXIMUM=10000 US
   NOMINAL=7000 US

3. TEST 14 -- ACCESS TIME MEASUREMENT

   MAXIMUM=30000 US
   NOMINAL=28000 US

4. TEST 15 -- MAXIMUM SEEK TIMES

   MAXIMUM=52000 US
   NOMINAL=50000 US

8.3.2. TIMING TESTS PRINTOUT EXAMPLES

EXAMPLE #1

ROTATIONAL SPEED TIMES
MIN=16670 US
MAX=16690 US
AVG=16680 US 10 SEEKS TIMED

ONE CYLINDER SEEK TIMES
 * FORWARD
MIN=5350 US
MAX=6920 US
AVG=5550 US 409 SEEKS TIMED
 * REVERSE
MIN=5140 US
MAX=5960 US
AVG=5430 US 410 SEEKS TIMED

ACCESS TIME MEASUREMENTS
 * FORWARD
MIN=27770 US
MAX=28640 US
AVG=28230 US 128 SEEKS TIMED
 * REVERSE
MIN=27990 US
MAX=28550 US
AVG=28220 US 128 SEEKS TIMED

MAXIMUM SEEK TIMES
 * FORWARD
MIN=49990 US
MAX=51980 US
AVG=51010 US 128 SEEKS TIMED
 * REVERSE
MIN=48120 US
MAX=50650 US
AVG=49340 US 128 SEEKS TIMED

EXAMPLE #2

ROTATIONAL SPEED TIMES
MIN=16670 US
MAX=16690 US
AVG=16680 US 10 SEEKS TIMED

ONE CYLINDER SEEK TIMES
 * FORWARD
MIN=5470 US
MAX=10940 US 3 ABOVE THE MAXIMUM OF 10000 US
AVG=5830 US 409 SEEKS TIMED
 * REVERSE
MIN=5040 US
MAX=5970 US
AVG=5330 US 410 SEEKS TIMED

```
                ACCESS TIME MEASURMENTS
                 * FORWARD
                MIN=29730 US
                MAX=31620 US  73 ABOVE THE MAXIMUM OF 30000 US
                AVG=30320 US  128 SEEKS TIMED
                 * REVERSE
                MIN=28620
                MAX=31230 US  128 ABOVE THE MAXIMUM OF 30000 US
                AVG=30800 US  128 SEEKS TIMED

                MAXIMUM SEEK TIMES
                 * FORWARD
                MIN=53510 US
                MAX=54240 US  128 ABOVE THE MAXIMUM OF 52000 US
                AVG=54020 US  128 SEEKS TIMED
                 * REVERSE
                MIN=52050 US
                MAX=54550 US  128 ABOVE THE MAXIMUM OF 52000 US
                AVG=52210 US  128 SEEKS TIMED


        8.4     END OF TEST

                WITH ALL SWITCHES ON A "0" AN "END OF PASS" MESSAGE WILL BE
                TYPED AT THE COMPLETION OF TESTING A DRIVE AND THE "END OF TEST"
                TYPEOUT WILL OCCUR WHEN ALL DRIVES HAVE BEEN TESTED.
```

9. PROGRAM DESCRIPTION

```
                THIS PROGRAM CONTAINS NINTEEN TESTS NUMBERED 0-22 IN OCTAL.
                TESTS 0-7 & 11 WILL READ THE CYLINDER, TRACK, AND SECTOR INFORMATION
                FROM THE HEADER, USING A "READ HEADER AND DATA" COMMAND, AND
                THEN CHECK THE INFORMATION FOR VALIDITY. THUS, INSURING
                THE SEEK OPERATION FUNCTIONS PROPERLY. TESTS 12-15 WILL MEASURE
                THE ROTATIONAL SPEED, THE ONE CYLINDER SEEK, THE ACCESS TIME,
                AND THE MAXIMUM SEEK TIMES TO ENSURE THEY ARE ALL WITHIN THE
                TOLERANCES ALLOWED. TEST 16 AND 17 ENSURES THE SECTOR AND
                TRACK ADDRESSING CIRCUITRY WORKS PROPERLY. TEST 20 VERIFIES
                THE DATA STORAGE AND RETRIEVAL CAPABILITIES ARE FUNCTIONAL.
                AND TEST 21 WILL STRESS AND CHECK THE READ/WRITE AND SERVO
                SYSTEMS.

                THE PROGRAM WILL START BY IDENTIFING ITSELF AND DETERMINING ALL
                DRIVES THAT ARE AVAILABLE FOR TESTING. THEN BEGINNING WITH
                THE LOWEST NUMBERICAL DRIVE AND PROCEEDING IN SEQUENTIAL ORDER.
                ALL OF THE DRIVES WILL BE TESTED. ONE PASS THROUGH THE TEST
                SEQUENCE (TESTS 0-10,12-20) WILL BE PERFORMED ON EACH DRIVE BEFORE
                MOVING TO THE NEXT DRIVE IN SEQUENCE.  DRIVE TO BE
                TESTED WILL BE TYPED AT THE BEGINNING OF EACH PASS, AN "END
                OF PASS" MESSAGE WILL BE TYPED AT THE COMPLETION OF EACH PASS,
                AND AN "END OF TEST" MESSAGE WILL BE TYPED AFTER TESTING ALL
```

DRIVES.

REFER TO THE FOLLOWING SECTIONS FOR DETAILED DESCRIPTIONS
OF EACH TEST.

9.1 TEST 0 - RECAL/SEEK TEST

THIS TEST WILL CAUSE THE DRIVE TO EXECUTE A RECALIBRATE
COMMAND CYCLE AND THEN SEEK FORWARD TO CYLINDER "LC". AT
THE COMPLETION OF BOTH COMMANDS, STATUS INDICATIONS ARE
CHECKED TO ENSURE NO ERRORS OCCURRED.

THE PARAMETERS USED BY THE TEST ARE GIVEN BELOW:

R	-	200
LC	-	410
LC'	-	814
FT	-	0
FS	-	0

9.2 TEST 1 - SEEK/SEEK TEST

THIS TEST WILL CAUSE THE DRIVE TO EXECUTE A FORWARD SEEK
CYCLE TO "LC", "LT", "LS" FOLLOWED BY A REVERSE SEEK CYCLE TO
"FC", "FT", "FS". AT THE COMPLETION OF EACH SEEK, THE PROPER
INDICATORS ARE EXAMINED TO ENSURE PROPER OPERATION.

THE PARAMETERS USED BY THE TEST ARE GIVEN BELOW:

R	-	100
FC	-	0
LC	-	128
FC'	-	0
LC'	-	256
IC	-	0
FT	-	0
LT	-	0
FS	-	0
LS	-	0

9.3 TEST 2 - INCREMENTAL SEEK TEST

THIS TEST WILL COMMAND FORWARD SEEK CYCLES TO ADVANCE THE
CYLINDER ADDRESS FROM "FC" TO "LC" BY THE INCREMENT "IC".
WHEN THE RESULTANT CYLINDER ADDRESS (NC) EXCEEDS
"LC" REVERSE SEEK CYCLES ARE INITIATED; STARTING
AT THE LAST LEGAL "NC" AND DECREMENTING BY "IC"
UNTIL "NC" IS LESS THAN "FC". AT THE COMPLETION OF EACH
SEEK COMMAND THE PROPER INDICATORS ARE EXAMINED TO
ENSURE PROPER OPERATION.

THE PARAMETERS USED BY THE TEST ARE GIVEN BELOW:

R	-	1
FC	-	0

```
LC      -       410
FC'     -         0
LC'     -       814
IC      -         1
FT      -         0
FS      -         0
```

9.4 TEST 3 - STEPPING SEEK TEST

THIS TEST WILL COMMAND SEEK CYCLES TO CYLINDER 0,1,2,4, 8,16,32,64,128, AND 256. AT THE COMPLETION OF EACH SEEK COMMAND THE PROPER INDICATORS ARE EXAMINED TO VERIFY PROPER OPERATION.

THE PARAMETERS USED BY THE TEST ARE GIVEN BELOW:

```
R       -         8
FC      -         0
LC      -       256
FC'     -         0
LC'     -       256
IC      -         1
FT      -         0
FS      -         0
```

9.5 TEST 4 - OSCILLATING SEEK TEST

THIS TEST WILL COMMAND SEEK CYCLES FROM "FC" TO "NC" AND BACK TO "FC". "NC" STARTS AT "FC" AND INCREMENTS BY "IC" UP TO CYLINDER "LC", THEN IS DECREMENTED BY "IC" BACK TO CYLINDER "FC". AT THE COMPLETION OF EVERY SEEK COMMAND THE PROPER INDICATORS ARE EXAMINED TO ENSURE PROPER OPERATION.

THE FOLLOWING PARAMETERS ARE USED BY THE TEST:

```
R       -         1
FC      -         0
LC      -       410
FC'     -         0
LC'     -       814
IC      -         1
FT      -         0
FS      -         0
```

9.6 TEST 5 - CONVERGING/DIVERGING SEEK TEST

- THIS TEST WILL CAUSE THE DRIVE TO EXECUTE FORWARD AND REVERSE

SEEKS FROM "NC1" AND "NC2" RESPECTIVELY, "NC1" WILL BE INCREMENTED
BY "IC" AND "NC2" WILL BE DECREMENTED BY "IC" UNTIL "NC1" IS
GREATER THAN THE INITIAL VALUE OF "NC2" AND "NC2" IS
LESS THAN THE INITIAL VALUE OF "NC1". AT THE COMPLETION OF
EACH SEEK COMMAND THE PROPER INDICATORS ARE EXAMINED TO
ENSURE PROPER OPERATION. "NC1" AND "NC2" DEFAULT TO

"FC" AND "LC" RESPECTIVELY.
THE FOLLOWING PARAMETERS ARE USED BY THE TEST:

R	-	1
FC	-	0
LC	-	410
FC'	-	0
LC'	-	814
IC	-	1
FT	-	0
FS	-	0

9.7 TEST 6 - SERVO ADDRESSING LOGIC NOISE GENERATOR TEST

IN THIS TEST A SEEK IS DONE TO CYL "NC" THEN A SEEK TO
NC+4 THEN NC+1 THEN NC+3 THEN NC+2 THEN NC+5. NOW "NC" IS UPDATED
BY "IC" AND THE ABOVE SEQUENCE IS REPEATED UNITL "LC" IS
EXCEEDED BY ANY OF THE ABOVE VALUES. THE INITIAL VALUE OF "NC"
IS "FC". AT THE COMPLETION OF EACH SEEK COMMAND THE
PROPER INDICATORS ARE EXAMINED TO ENSURE PROPER OPERATION.

THE FOLLOWING PARAMETERS ARE USED BY THE TEST:

R	-	1
FC	-	0
LC	-	410
FC'	-	0
LC'	-	814
IC	-	1
FT	-	0
FS	-	0

9.8 TEST 7 - RANDOM SEEK TEST

THIS TEST PERFORMS RANDOM SEEK OPERATIONS BETWEEN CYLINDERS 'FC'
'LC'. AFTER EACH SEEK, THE POSITION OF THE DRIVE IS VERIFIED BY
READING A SECTOR FROM THE CURRENTLY ADDRESSED CYLINDER AND TRACK.
THE TRACK ADDRESS IS INCREMENTED FOR EACH SEEK SO THAT VERIFICATION
OF POSITIONING OCCURS USING EACH HEAD. TRACK ADDRESSES ARE INCREMENTED
BETWEEN PARAMTERS 'FT' AND 'LT'.

THE FOLLOWING PARAMETERS ARE USED BY THE TEST:

R	-	5000
FC	-	0
LC	-	410
FC'	-	0
LC'	-	814
FT	-	0
LT	-	18

9.9 TEST 10 - SERVO SETTLE DOWN TEST

THIS TEST VERIFIES THAT THE SERVO HAS SETTLED DOWN AND THAT
THE DRIVE IS ON CYLINDER WHEN THE DRIVE INDICATES SEEK COMPLETE.
RANDOM SEEKS ARE ISSUED BETWEEN CYLINDERS 'NC1' AND 'NC1+IC'
('NC1' STARTS AT VALUE 'FC'). AT THE COMPLETION OF 1000 (10) SEEKS,
'NC1' IS INCREMENTED BY VALUE 'IC' AND THE SEQUENCE IS REPEATED.
THE TEST IS COMPLETED WHEN 'NC1' HAS BEEN INCREMENTED BEYOND 'LC'.

WHEN THE SEEK COMPLETES, THE PROGRAM READS THE DRIVE'S LOOK-AHEAD
REGISTER (RPLA) TO DETERMINE THE ADDRESS OF THE SECTOR ROTATING INTO
POSITION. THE PROGRAM THEN ISSUES A WRITE HEADER AND DATA COMMAND
FOR THAT SECTOR. IF THE DRIVE'S POSITIONER HAS NOT SETTLED DOWN OR
IF THE POSITIONER IS NOT ON CYLINDER (IF THE DRIVE IS AN RP04, THE
OFF CYLINDER CONDITION MUST LAST FOR AT LEAST 800 US), THE DRIVE
WILL REPORT A 'WRU' ERROR. (RP05/6'S MAY ALSO REPORT 'NHS' ERROR UNDER
ERRORS IN THIS TEST INDICATE THAT THE SERVO SYSTEM MAY NOT BE ADJUSTED
CORRECTLY, THAT THE DRIVE IS MALFUNCTIONING, OR THAT A PCAK WITH
MARGINAL SERVO TRACKS IS MOUNTED ON THE DRIVE.

THIS TEST USES THE EXTENTION BITS IN THE LOOK-AHEAD REGISTER TO DETERMINE WHETHER OR NOT IT CAN PICK UP THE SECTOR ROTATING INTO POSITION. THE TEST IS OPTIMIZED SUCH THAT IF THE DRIVE SIGNALS SEEK DONE WITHIN THE FIRST 80% OF THE SECTOR CURRENTLY UNDER THE HEAD, THE TEST WILL TRY TO ADDRESS THE NEXT SECTOR. BASED ON OBSERVATION, THE PROGRAM IS ABLE TO START THE OPERATION WITHOUT LOSING A REVOLUTION MOST OF THE TIME.

THIS TEST IS VALID ONLY IF THE OPERATION IS STARTED WITHIN A FEW HUNDRED MICRO-SECONDS AFTER SEEK DONE OCCURS. THE NECESSARY TIME DEPENDENT PARAMETERS OCCUR WITHIN THE REQUIRED TIME RANGE FREQUENTLY ENOUGH TO PERMIT THIS TEST TO BE EFFECTIVE.

THE FOLLOWING PARAMETERS ARE USED BY THE TEST:

R	-	1
FC	-	0
LC	-	410
FC'	-	0
LC'	-	814
IC	-	100
FT	-	0

9.10 TEST 11 - ALL SEEKS TEST

THIS TEST VERIFIES THAT THE DISK DRIVE CAN SEEK FROM EACH CYLINDER TO ALL OTHER CYLINDERS.

BEGINNING WITH CYLINDER 'FC', THE TEST SEEKS TO EACH CYLINDER BETWEEN 'FC' AND 'LC' FROM CYINDER 'FC'. THE BEGINNING CYLINDER ADDRESS IS INCREMENTED AND THE TEST SEEKS BETWEEN THE NEW CYLINDER ADDRESS AND ALL CYLINDERS BETWEEN 'FC' AND 'LC'. THE SEQUENCE CONTINUES UNTIL ALL CYLINDERS HAVE BEEN CHECKED.

THE FOLLOWING PARAMETERS ARE USED BY THIS TEST:

R	-	1
FC	-	0
LC	-	410
FC'	-	0
LC'	-	814
IC	-	1
FT	-	0
FS	-	0

9.11 TEST 12 - ROTATIONAL SPEED TIMING TEST

THIS TEST WILL START A SEARCH TO CYLINDER 0, TRACK 0, SECTOR 0. AS SOON AS THE INTERRUPT OCCURS, THE GO BIT IS SET AGAIN AND THE OPERATION IS TIMED. THIS PROCEDURE IS REPEATED 10 TIMES THEN THE AVERAGE TIME IS CALCULATED AND CHECKED TO ENSURE IT IS WITHIN TOLERANCE:

16.67 MS/REV + OR - 2% IF 60HZ
16.67 MS/REV + OR - 2.5% IF 50HZ.

THE FOLLOWING PARAMETERS ARE USED BY THE TEST:

R	-	1
FC	-	0
FC'	-	0
FT	-	0
FS	-	0

9.12 TEST 13 - ONE CYLINDER SEEK TIMING TEST

THIS TEST WILL COMMAND FORWARD SEEK CYCLES TO ADVANCE THE CYLINDER BY ONE UNTIL THE INCREMENT IS GREATER THAN THE CYLINDER 'LC', THEN REVERSE SEEK TO CYLINDER 'FC'. THE TIME TO PERFORM EACH SEEK IS CHECKED TO ENSURE IT DOES NOT EXCEED THE MAXIMUM TIME PERMITTED FOR A ONE CYLINDER SEEK. THE TIME MUST BE LESS THAN 10MS.

THE TEST USES THE FOLLOWING PARAMETERS:

R	-	1
FC	-	0
LC	-	410
FC'	-	0
LC'	-	814

9.13 TEST 14 - ACCESS TIME MEASUREMENT

THIS TEST WILL COMMAND A FORWARD SEEK FROM CYLINDER 0 TO CYLINDER 'LC', THEN A REVERSEK FROM CYLINDER 'LC' TO CYLINDER 0. BOTH SEEKS ARE TIMED AND CHECKED TO ENSURE THEY ARE WITHIN THE TOLERANCE ALLOWED FOR THE ACCESS TIME MEASURMENT. THIS SEQUENCE IS REPEATED 128 TIMES (FOR A TOTAL OF 256 SEEKS). THE AVERAGE ACCESS TIME MUST BE LESS THAN 30 MS. CYLINDER 'LC' DEFAULTS TO 136 (10) FOR AN RP04/5 OR TO 255 (10) FOR AN RP06.

THE TEST USES THE FOLLOWING PARAMETERS:

R	-	1
FC	-	0
LC	-	136
FC'	-	0
LC'	-	255

9.14 TEST 15 - MAXIMUM SEEK TIMING TEST

THIS TEST WILL COMMAND A FORWARD SEEK FROM CYLINDER 0 TO CYLINDER 'LC', THEN A REVERSE SEEK FROM CYLINDER 'LC' TO CYLINDER 0. BOTH SEEKS ARE TIMED AND CHECKED TO ENSURE THEY ARE WITHIN THE TOLERANCE ALLOWED FOR THE MAXIMUM SEEK TIME. THIS SEQUENCE IS REPEATED 128 TIMES (FOR A TOTAL OF 256 SEEKS). THE MAXIMUM SEEK TIME MUST BE LESS THAN 54 MS. 'LC' DEFAULTS TO 410 (10) FOR RP04/5'S AND TO 814 (10) FOR RP06'S.

THE TEST USES THE FOLLOWING PARAMETERS:

R	-	1
FC	-	0
LC	-	410
FC'	-	0
LC'	-	814

9.15 TEST 16 - SECTOR ADDRESSING TEST

THIS TEST WRITES DATA INTO ALL SECTORS OF TRACK "FT". THE DATA WILL BE 256 WORDS OF THE SECTOR ADDRESS OF THE SECTOR BEING WRITTEN. A WRITE CHECK IS PERFORMED, THE BUFFER IS CLEARED (TO 177400) AND THE DATA IS READ AND COMPARED. THEN SECTOR 0 IS REWRITTEN AND SECTORS 0 - 21 ARE WRITE CHECKED. THEN SECTOR 1 IS REWRITTEN AND SECTORS 0 - 21 ARE WRITE CHECKED. THIS REWRITE AND WRITE CHECK PROCEDURE IS CONTINUED UP THROUGH REWRITE SECTOR 21 AND WRITE CHECK SECTORS 0-21.

THE TEST USES THE FOLLOWING PARAMETERS:

R	-	1
FC	-	0
FC'	-	0
FT	-	0

9.16 TEST 17 - TRACK ADDRESSING TEST

THIS TEST WILL WRITE DATA IN THE FORM OF TRACK ADDRESSES IN CYLINDER "FC" SECTOR "FS" OF EVERY TRACK WITH EACH TRACK GETTING ITS OWN TRACK ADDRESS.
A WRITE CHECK IS THEN PERFORMED ON EACH TRACK TO INSURE THE DATA IS VALID. THEN TRACK 0 IS REWRITTEN AND TRACK 1 THROUGH TRACK 18 IS WRITE CHECKED. THEN TRACK 1 IS REWRITTEN AND TRACK 2 THROUGH TRACK 18 IS WRITE CHECKED. THIS PROCEDURE IS CONTINUED UP THROUGH REWRITING TRACK 17 AND WRITE CHECKING TRACK 18.

THE TEST USES THE FOLLOWING PARAMETERS:

R	-	1
FC	-	0
FC'	-	0
FS	-	0

9.17 TEST 20 - DATA TEST

THIS TEST PERFORMS DATA STORAGE AND RETRIEVAL ON CYLINDERS "FC" THROUGH "LC" BY THE INCREMENT "IC" USING THE DATA PATTERNS SPECIFIED. THE FOLLOWING SEQUENCE OCCURS FOR EACH CYLINDER:

1. SET "NT" TO "FT" THEN REPEAT 2-4 UNTIL "NT" > "LT"
2. WRITE THEN WRITE CHECK "FS" THROUGH "LS" OF TRACK "NT"
3. READ THEN SOFTWARE COMPARE "FS" THROUGH "LS" OF TRACK "NT"
4. INCREMENT "NT" BY "IT"
5. REPEAT STEPS 1-4 FOR EACH DATA PATTERN
6. REPEAT STEPS 1-5 FOR "FC" THROUGH "LC" ADVANCING BY "IC"

IF A WRITE CHECK ERROR OCCURS THE ERROR IS REPORTED AND
THE TRACK IN ERROR IS REWRITTEN AND CHECKED. THIS CHECK IS
ACCOMPLISHED BY PERFORMING TWO(2) SUCCESSIVE ERROR FREE
WRITE CHECKS. IF THE CHECK FAILS THE ERROR IS REPORTED AS
FATAL AND NO READ OCCURS.
FS DEFAULTS TO 1 AND LS DEFAULTS TO 0
PAT DEFAULTS TO 177777 (ALL POSSIBLE PATTERNS)
THE POSSIBLE PATTERNS ARE:

PAT 0	PAT 1	PAT 2	PAT 3	PAT 4	PAT 5	PAT 6	PAT 7
165555	000001	177776	000000	000000	052525	007417	026455
133333	000003	177774	000000	010421	052525	007417	026455
165555	000007	177770	000000	021042	052525	007417	026455
133333	000017	177760	177777	031463	125252	170360	151322
165555	000037	177740	177777	042104	125252	170360	151322
133333	000077	177700	177777	052525	125252	170360	151322
165555	000177	177600	000000	063146	052525	007417	026455
133333	000377	177400	000000	073567	052525	007417	026455
165555	000777	177000	177777	104210	125252	170360	151322
133333	001777	176000	177777	114631	125252	170360	151322
165555	003777	174000	000000	125252	052525	007417	026455
133333	007777	170000	177777	135673	125252	170360	151322
165555	017777	160000	000000	146314	052525	007417	026455
133333	037777	140000	177777	156735	125252	170360	151322
165555	077777	100000	000000	167356	052525	007417	026455
133333	177777	000000	177777	177777	125252	170360	151322

PAT 8	PAT 9	PAT 10	PAT 11	PAT 12	PAT 13	PAT 14	PAT 15
165555	000001	177776	172666	077777	153333	000000	177777
133333	000002	177775	155555	137777	066667	177777	000000
165555	000004	177773	172666	157777	153333	177777	000000
133333	000010	177767	155555	167777	066667	177777	000000
165555	000020	177757	172666	173777	153333	177777	000000
133333	000040	177737	155555	175777	066667	177777	000000
165555	000100	177677	172666	176777	153333	177777	000000
133333	000200	177577	155555	177377	066667	177777	000000
165555	000400	177377	172666	177577	153333	177777	000000
133333	001000	176777	155555	177677	066667	177777	000000
165555	002000	175777	172666	177737	153333	177777	000000
133333	004000	173777	155555	177757	066667	177777	000000
165555	010000	167777	172666	177767	153333	177777	000000
133333	020000	157777	155555	177773	066667	177777	000000
165555	040000	137777	172666	177775	153333	177777	000000
133333	100000	077777	155555	177776	066667	177777	000000

THE TEST USES THE FOLLOWING PARAMETERS:

R	-	1
FC	-	0
LC	-	410
FC'	-	0
LC'	-	814
IC	-	64

FT	-	0
LT	-	18
IT	-	1
FS	-	1
LS	-	0
PAT	-	177777

9.18 TEST 21 - RANDOM ADDRESS AND RANDOM PATTERN TEST

STARTING AT "FC" AND GOING THROUGH "LC" THE DISK PACK
IS WRITTEN WITH A RANDOM PATTERN. THE FIRST TWO WORDS
OF EACH SECTOR WILL BE THE BASE OF THE RANDOM GENERATOR
FOR THAT SECTOR.
THE TEST THEN PERFORMS THE FOLLOWING SEQUENCE "R" TIMES
"R" DEFAULTS TO 20,000.

1) GENERATE A RANDOM ADDRESS
2) WRITE A RANDOM PATTERN AT THE ADDRESS
GENERATED IN 1.
3) GENERATE A RANDOM ADDRESS
4) READ THE SECTOR AT THE ADDRESS
GENERATED IN 3.
5) DO A SOFTWARE CHECK OF THE DATA READ IN 4.
6) DO A WRITE CHECK OF THE DATA WRITTEN IN 2
7) GENERATE A RANDOM ADDRRESS
8) READ THE SECTOR AT THE ADDRESS
GENERATED IN 7.
9) DO A SOFTWARE CHECK OF THE DATA READ IN 8
10) DO A WRITE CHECK OF THE DATA WRITTEN IN 2

THE TEST USES THE FOLLOWING PARAMETERS:

R	-	20000
FC	-	0
LC	-	410
FC'	-	0
LC'	-	814

9.19 TEST 22 - RP04 ACCESS TIME ADJUSTMENT TEST

THIS TEST PERFORMS SEEKS BETWEEN CYLINDERS 0 & 136 TO ALLOW THE
OPERATOR TO ADJUST THE ACCESS TIME ON AN RP04 USING THE
DDU. THE PROGRAM STALLS APPROXIMATELY 5 SECONDS BETWEEN SEEKS
SO THAT THE ACCESS TIME INDICATORS ON THE DDU MAY BE OBSERVED.

THE TEST USES THE FOLLOWING PARAMETERS:

R	-	5000
FC	-	0
LC	-	136
FC'	-	0
LC'	-	255

10. PROGRAM LISTING

```
1449                            .TITLE  CZRJAD RP04/5/6 MECHANICAL AND READ/WRITE TEST
1450                            ;*COPYRIGHT (C) 1976,1978
1451                            ;*DIGITAL EQUIPMENT CORP.
1452                            ;*MAYNARD, MASS. 01754
1453                            ;*
1454                            ;*PROGRAM BY C. HESS
1455                            ;*
1456                            ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
1457                            ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
1458                            ;*
1459
1460                            .SBTTL  CONTROL SWITCH SETTINGS
1461                            ;*
1462                            ;*    SWITCH  STATE              USE
1463                            ;*    ------  -----    ------------------------------------
1464                            ;*      15      0      WRITE PACK BEFORE TESTING (TEST 21)
1465                            ;*              1      INHIBIT WRITING PACK BEFORE TESTING (TEST 21)
1466                            ;*      14      0      NO STALL BETWEEN DRIVE FUNCTIONS
1467                            ;*              1      STALL AFTER EVERY DRIVE FUNCTION
1468                            ;*      13      0      USE SPECIFIC STALL TIME
1469                            ;*              1      USE RANDOM STALL
1470                            ;*      12      0      NO INCREMENTING STALL IN TEST 4
1471                            ;*              1      DO INCREMENTING STALL IN TEST 4
1472                            ;*       8      0      DO IMPLIED SEEKS WITH DATA TRANSFERS
1473                            ;*              1      DO EXPLICIT SEEKS BEFORE DATA TRANSFERS
1474                            ;*       7      0      DO "READ HEADER AND DATA" IN TESTS 0-11
1475                            ;*              1      DO EXPLICIT SEEKS IN TESTS 0-11
1476                            ;*       6      0      60 HZ
1477                            ;*              1      50 HZ
1478                            ;*       5      0      RUN WATCHDOG TIMER
1479                            ;*              1      INHIBIT WATCHDOG TIMER
1480                            ;*       0      0      TEST DRIVE(S) IN 22 SECTOR (16 BIT) MODE
1481                            ;*              1      TEST DRIVE(S) IN 20 SECTOR (18 BIT) MODE
1482
1483                            .SBTTL  OPERATIONAL SWITCH SETTINGS
1484                            ;*
1485                            ;*      SWITCH              USE
1486                            ;*      ------    --------------------
1487                            ;*        15      HALT ON ERROR
1488                            ;*        14      LOOP ON TEST
1489                            ;*        13      INHIBIT ERROR TYPEOUTS
1490                            ;*        10      BELL ON ERROR
1491                            ;*         9      LOOP ON ERROR
1492                            ;*         8      PRINT ERROR MESSAGE ON LINE PRINTER
1493                            ;*         7      READ CONTROL SWITCH SETTINGS FROM TTY
1494                            ;*         6      INHIBIT TIME REPORTS (TESTS 12-15)
1495                            ;*         5      REPORT ONE ERROR PER SECTOR (TESTS 16 & 17)
1496                            ;*         4      INHIBIT WRITES (TEST 15)
1497                            ;*         3      INHIBIT WRITE CHECKS (TEST 20)
1498                            ;*         2      INHIBIT READ AND SOFTWARE COMPARES (TEST 20)
1499                            ;*         1      INHIBIT SOFTWARE COMPARES (TEST 20)
1500                            ;*         0      PERFORM READ AFTER WRITE CHECK ERROR (TEST 20)
1501
1502                            .SBTTL  TRAP CATCHER
```

```
1503
1504            000000                          .=0
1505                                            ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
1506                                            ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
1507                                            ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
1508            000174                          .=174
1509  000174   000000                DISPREG: .WORD  0               ;;SOFTWARE DISPLAY REGISTER
1510  000176   000000                SWREG:   .WORD  0               ;;SOFTWARE SWITCH REGISTER
1511
1512                                            .SBTTL  ACT11 HOOKS
1513
1514                                            ;;**************************************************************
1515                                            ;HOOKS REQUIRED BY ACT11
1516            000200                          $SVPC=.                 ;SAVE PC
1517            000046                          .=46
1518  000046    017732                          $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
1519            000052                          .=52
1520  000052    000000                          .WORD   0               ;;2)SET LOC.52 TO ZERO
1521            000200                          .=$SVPC                 ;; RESTORE PC
1522
1523                                            .SBTTL  STARTING ADDRESSES
1524            000200                          .=200
1525                                            ;*200 = NORMAL START, INHIBIT PROGRAMMABLE DRIVES STANDALONE OR XXDP CHAIN
1526  000200    000137  004674                        JMP     @#START1
1527                                            ;*204 = SELECT OPERATING PARAMETERS, INHIBIT PROGRAMMABLE DRIVES
1528  000204    000137  004730                        JMP     @#START2
1529                                            ;*210 = SELECT RH11/RP04/5/6 ADDRESSES, DO NOT INHIBIT PROGRAMMABLE DRIVES
1530  000210    000137  004656                        JMP     @#START3
1531                                            ;*214 = COMBINATION OF 204 AND 210, DO NOT INHIBIT PROGRAMMABLE DRIVES
1532  000214    000137  004712                        JMP     @#START4
1533                                            ;*220 = SAME AS 200 BUT DO NOT INHIBIT PROGRAMMABLE DRIVES
1534  000220    000137  004642                        JMP     @#START5
1535                                            ;*224 = SAME AS 204 BUT DO NOT INHIBIT PROGRAMMABLE DRIVES
1536  000224    000137  004626                        JMP     @#START6
1537
1538                                            .SBTTL  BASIC DEFINITIONS
1539
1540                                            ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
1541            001100                          STACK=  1100
1542                                            .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
1543                                            .EQUIV  TOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL
1544
1545                                            ;*MISCELLANEOUS DEFINITIONS
1546            000011                          HT=     11              ;;CODE FOR HORIZONTAL TAB
1547            000012                          LF=     12              ;;CODE FOR LINE FEED
1548            000015                          CR=     15              ;;CODE FOR CARRIAGE RETURN
1549            000200                          CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
1550            177776                          PS=     177776          ;;PROCESSOR STATUS WORD
1551                                            .EQUIV  PS,PSW
1552            177774                          STKLMT= 177774          ;;STACK LIMIT REGISTER
1553            177772                          PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
1554            177570                          DSWR=   177570          ;;HARDWARE SWITCH REGISTER
1555            177570                          DDISP=  177570          ;;HARDWARE DISPLAY REGISTER
1556
1557                                            ;*GENERAL PURPOSE REGISTER DEFINITIONS
1558            000000                          R0=     %0              ;;GENERAL REGISTER
```

```
1559            000001                  R1=     %1              ;;GENERAL REGISTER
1560            000002                  R2=     %2              ;;GENERAL REGISTER
1561            000003                  R3=     %3              ;;GENERAL REGISTER
1562            000004                  R4=     %4              ;;GENERAL REGISTER
1563            000005                  R5=     %5              ;;GENERAL REGISTER
1564            000006                  R6=     %6              ;;GENERAL REGISTER
1565            000007                  R7=     %7              ;;GENERAL REGISTER
1566            000006                  SP=     %6              ;;STACK POINTER
1567            000007                  PC=     %7              ;;PROGRAM COUNTER
1568
1569                                    ;*PRIORITY LEVEL DEFINITIONS
1570            000000                  PR0=    0               ;;PRIORITY LEVEL 0
1571            000040                  PR1=    40              ;;PRIORITY LEVEL 1
1572            000100                  PR2=    100             ;;PRIORITY LEVEL 2
1573            000140                  PR3=    140             ;;PRIORITY LEVEL 3
1574            000200                  PR4=    200             ;;PRIORITY LEVEL 4
1575            000240                  PR5=    240             ;;PRIORITY LEVEL 5
1576            000300                  PR6=    300             ;;PRIORITY LEVEL 6
1577            000340                  PR7=    340             ;;PRIORITY LEVEL 7
1578
1579                                    ;*"SWITCH REGISTER" SWITCH DEFINITIONS
1580            100000                  SW15=   100000
1581            040000                  SW14=   40000
1582            020000                  SW13=   20000
1583            010000                  SW12=   10000
1584            004000                  SW11=   4000
1585            002000                  SW10=   2000
1586            001000                  SW09=   1000
1587            000400                  SW08=   400
1588            000200                  SW07=   200
1589            000100                  SW06=   100
1590            000040                  SW05=   40
1591            000020                  SW04=   20
1592            000010                  SW03=   10
1593            000004                  SW02=   4
1594            000002                  SW01=   2
1595            000001                  SW00=   1
1596                                    .EQUIV  SW09,SW9
1597                                    .EQUIV  SW08,SW8
1598                                    .EQUIV  SW07,SW7
1599                                    .EQUIV  SW06,SW6
1600                                    .EQUIV  SW05,SW5
1601                                    .EQUIV  SW04,SW4
1602                                    .EQUIV  SW03,SW3
1603                                    .EQUIV  SW02,SW2
1604                                    .EQUIV  SW01,SW1
1605                                    .EQUIV  SW00,SW0
1606
1607                                    ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
1608            100000                  BIT15=  100000
1609            040000                  BIT14=  40000
1610            020000                  BIT13=  20000
1611            010000                  BIT12=  10000
1612            004000                  BIT11=  4000
1613            002000                  BIT10=  2000
1614            001000                  BIT09=  1000
```

```
 1615          000400                  BIT08=  400
 1616          000200                  BIT07=  200
 1617          000100                  BIT06=  100
 1618          000040                  BIT05=  40
 1619          000020                  BIT04=  20
 1620          000010                  BIT03=  10
 1621          000004                  BIT02=  4
 1622          000002                  BIT01=  2
 1623          000001                  BIT00=  1
 1624                                  .EQUIV  BIT09,BIT9
 1625                                  .EQUIV  BIT08,BIT8
 1626                                  .EQUIV  BIT07,BIT7
 1627                                  .EQUIV  BIT06,BIT6
 1628                                  .EQUIV  BIT05,BIT5
 1629                                  .EQUIV  BIT04,BIT4
 1630                                  .EQUIV  BIT03,BIT3
 1631                                  .EQUIV  BIT02,BIT2
 1632                                  .EQUIV  BIT01,BIT1
 1633                                  .EQUIV  BIT00,BIT0
 1634
 1635                                  ;*BASIC "CPU" TRAP VECTOR ADDRESSES
 1636          000004                  ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
 1637          000010                  RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
 1638          000014                  TBITVEC=14              ;;"T" BIT
 1639          000014                  TRTVEC= 14              ;;TRACE TRAP
 1640          000014                  BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
 1641          000020                  IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
 1642          000024                  PWRVEC= 24              ;;POWER FAIL
 1643          000030                  EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
 1644          000034                  TRAPVEC=34              ;;"TRAP" TRAP
 1645          000060                  TKVEC=  60              ;;TTY KEYBOARD VECTOR
 1646          000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
 1647          000240                  PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
 1648                                  ;;**************************************************************
 1649
 1650                                  .SBTTL  RH11 REGISTERS
 1651
 1652                                  ;;**************************************************************
 1653
 1654                                  ;CONTROL AND STATUS REGISTER 1 (RPCS1)
 1655
 1656          000100                  IE=     100                     ;INTERRUPT ENABLE (BIT #6)
 1657          000200                  RDY=    200                     ;READY (BIT #7)
 1658          000400                  A16=    400                     ;HIGH ORDER BUS ADDRESS BIT (BIT #8)
 1659          001000                  A17=    1000                    ;HIGH ORDER BUS ADDRESS BIT (BIT #9)
 1660          002000                  PSEL=   2000                    ;PORT SELECT (BIT #10)
 1661          020000                  MCPE=   20000                   ;MASSBUSS PARITY ERROR (BIT #13)
 1662          040000                  TRE=    40000                   ;TRANSFER ERROR (BIT #14)
 1663                                  ;SC=    100000                  ;SPECIAL CONDITION (BIT #15)
 1664
 1665                                  ;WORD COUNT REGISTER (RPWC)
 1666                                  ;(EACH BIT IS CALLED BY BIT NUMBER)
 1667
 1668                                  ;BUS ADDRESS REGISTER (RPBA)
 1669                                  ;(EACH BIT IS CALLED BY BIT NUMBER)
 1670
```

```
                        ;CONTROL AND STATUS REGISTER 2 (RPCS2)

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)
                ;PAT=   20              ;MASSBUS PARITY TEST (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 EXISTENT MEMORY (BIT #11)
010000          NED=    10000           ;NON EXISTENT 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 (RPDB)
                ;(EACH BIT IS CALLED BY BIT NUMBER)


                ;;*****************************************************************

                .SBTTL  RP04/5/6 REGISTERS

                ;;*****************************************************************

                ;CONTROL AND STATUS 1 REGISTER. (#00)

000001          GO=     1               ;GO BIT .BIT #0)
000002          F1=     2               ;FUNCTION CODE BIT #1
000004          F2=     4               ;FUNCTION CODE BIT #2
000010          F3=     10              ;FUNCTION CODE BIT #3
000020          F4=     20              ;FUNCTION CODE BIT #4
000040          F5=     40              ;FUNCTION CODE BIT #5
004000          DVA=    4000            ;DEVICE AVAILABLE (BIT #11)

                ;DRIVE STATUS REGISTER (RPDS1) (#01)

                ;DF5=    1              DRIVE FORWARD 5"/SEC.  (BIT #0)
000002          DFF20=  2               ;DRIVE FORWARD 20"/SEC.  (BIT #1)
000004          DIGB=   4               ;DRIVE TO INNER GUARD 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          PGM=    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           ;COMPOSITE ERROR (BIT #14)
```

```
1727            100000          ATA=    100000                  ;ATTENTION ACTIVE (BIT #15)
1728
1729                            ;ERROR REGISTER #01 (RPER1) (#02)
1730
1731            000001          ILF=    1                       ;ILLEGAL FUNCTION (BIT #0)
1732            000002          ILR=    2                       ;ILLEGAL REGISTER (BIT #1)
1733            000004          RMR=    4                       ;REGISTER MODIFICATION REFUSED (BIT #2)
1734            000010          PAR=    10                      ;PARITY ERROR (BIT #3)
1735            000020          FER=    20                      ;FORMAT ERROR (BIT #4)
1736            000040          WCF=    40                      ;WRITE CLOCK FAIL (BIT #5)
1737            000100          ECH=    100                     ;ECC HARD ERROR (BIT #6)
1738            000200          HCE=    200                     ;HEADER COMPARE ERROR (BIT #7)
1739            000400          HCRC=   400                     ;HEADER CRC ERROR (BIT #8)
1740            001000          AOE=    1000                    ;ADDRESS OVERFLOW ERROR (BIT #9)
1741            002000          IAE=    2000                    ;INVALID ADDRESS ERROR (BIT #10)
1742            004000          WLE=    4000                    ;WRITE LOCK ERROR (BIT #11)
1743            010000          DTE=    10000                   ;DRIVE TIMING ERROR (BIT #12)
1744            020000          OPI=    20000                   ;OPERATION INCOMPLETE (BIT #13)
1745            040000          UNS=    40000                   ;DRIVE UNSAFE (BIT #14)
1746            100000          DCK=    100000                  ;DATA CHECK ERROR (BIT 15)
1747
1748                            ;MAINTAINABILITY REGISTER (RPMR)(#03)
1749
1750            000001          DMD=    1                       ;DIAGINOSTIC MODE (BIT #0)
1751            000002          MCLK=   2                       ;MAINTAINABILITY CLOCK (BIT #1)
1752            000004          MINX=   4                       ;MAINTAINABILITY INDEX (BIT #2)
1753            000010          MSTCK=  10                      ;MAINTAINABILITY SECTOR CLOCK (BIT #3)
1754            000020          MRD=    20                      ;MAINTAINABILITY READ (BIT #4)
1755            000040          MWR=    40                      ;MAINTAINABILITY WRITE (BIT #5)
1756            000200          DTSY=   200                     ;MAINTAINABILITY SYNC DETECTED (BIT #7)
1757
1758                            ;ATTENTION SUMMARY PSEUDO-REGISTER (RPAS) (#04)
1759
1760            000001          AT0=    1                       ;DEVICE 0 (BIT #0)
1761            000002          AT1=    2                       ;DEVICE 1 (BIT #1)
1762            000004          AT2=    4                       ;DEVICE 2 (BIT #2)
1763            000010          AT3=    10                      ;DEVICE 3 (BIT #3)
1764            000020          AT4=    20                      ;DEVICE 4 (BIT #4)
1765            000040          AT5=    40                      ;DEVICE 5 (BIT #5)
1766            000100          AT6=    100                     ;DEVICE 6 (BIT #6)
1767            000200          AT7=    200                     ;DEVICE 7 (BIT #7)
1768
1769                            ;DESIRED SECTOR/TRACK ADDRESS REGISTER (RPDA) (#05)
1770                            ;(EACH BIT IS CALLED BY BIT NUMBER)
1771
1772                            ;DRIVE TYPE REGISTER (RPDT) (#06)
1773
1774            000001          DT00=   1                       ;DRIVE TYPE NUMBER BIT 1
1775            000002          DT01=   2                       ;DRIVE TYPE NUMBER BIT 2
1776            000004          DT02=   4                       ;DRIVE TYPE NUMBER BIT 3
1777            000010          DT03=   10                      ;DRIVE TYPE NUMBER BIT 4
1778            000020          DT04=   20                      ;DRIVE TYPE NUMBER BIT 5
1779            000040          DT05=   40                      ;DRIVE TYPE NUMBER BIT 6
1780            000100          DT06=   100                     ;DRIVE TYPE NUMBER BIT 7
1781            000200          DT07=   200                     ;DRIVE TYPE NUMBER BIT 8
1782            000400          DT08=   400                     ;DRIVE TYPE NUMBER BIT 9
```

```
1783    004000          DRQ=    4000            ;DRIVE REQUEST REQUIRED (BIT #11)
1784    020000          MOH=    20000           ;MOVING HEAD (BIT #13)
1785    040000          TAP=    40000           ;TAPE DRIVE (BIT #14)
1786    100000          NBA=    100000          ;NOT BLOCK ADDRESSED (BIT #15)
1787
1788                    ;LOOK-AHEAD REGISTER (RPLA) (#07)
1789
1790    000001          EXT1=   1               ;EXTENSION 1 (BIT #0)
1791    000002          EXT2=   2               ;EXTENSION 2 (BIT #1)
1792    000004          EXT4=   4               ;EXTENSION 3 (BIT #2)
1793    000010          EXT10=  10              ;EXTENSION 4 (BIT #3)
1794    000020          EXT20=  20              ;EXTENSION 5 (BIT #4)
1795    000040          EXT40=  40              ;EXTENSION 6 (BIT #5)
1796    000100          SC1=    100             ;SECTOR COUNT FIELD 0 (BIT #6)
1797    000200          SC2=    200             ;SECTOR COUNT FIELD 1 (BIT #7)
1798                    ;SC4=   400             ;SECTOR COUNT FIELD 2 (BIT #8)
1799    001000          SC10=   1000            ;SECTOR COUNT FIELD 3 (BIT #9)
1800    002000          SC20=   2000            ;SECTOR COUNT FIELD 4 (BIT #10)
1801    004000          TRK1=   4000            ;TRACK FIELD 1 (BIT #11)
1802    010000          TRK2=   10000           ;TRACK FIELD 2 (BIT #12)
1803    020000          TRK4=   20000           ;TRACK FIELD 3 (BIT #13)
1804    040000          TRK10=  40000           ;TRACK FIELD 4 (BIT #14)
1805    100000          TRK20=  100000          ;TRACK FIELD 5 (BIT #15)
1806
1807                    ;RP04 ERROR REGISTER #2 (RPER2) (#10)
1808
1809    000001          WCU=    1               ;WRITE CURRENT UNSAFE (BIT #0)
1810    000002          CSF=    2               ;CURRENT SINK FAILURE (BIT #1)
1811    000004          WSU=    4               ;WRITE SELECT UNSAFE (BIT #2)
1812    000010          CSU=    10              ;CURRENT SWITCH UNSAFE (BIT #3)
1813    000020          MSE=    20              ;MOTOR SEQUENCE ERROR (BIT #4)
1814    000040          TDF=    40              ;TRANSITIONS DETECTOR FAILURE (BIT #5)
1815    000100          TUF=    100             ;TRANSITIONS UNSAFE (BIT #6)
1816    000200          FEN=    200             ;FAILSAFE ENABLED (BIT #7)
1817    000400          WRU=    400             ;WRITE READY UNSAFE (BIT #8)
1818    001000          MHS=    1000            ;MULTIPLE HEAD SELECT (BIT #9)
1819    002000          NHS=    2000            ;NO HEAD SELECTION (BIT #10)
1820    004000          IXE=    4000            ;INDEX ERROR (BIT #11)
1821    010000          VU30=   10000           ;30VOLT UNSAFE (BIT #12)
1822    020000          PLU=    20000           ;PLO UNSAFE (BIT #13)
1823    100000          ACU=    100000          ;AC UNSAFE (BIT #15)
1824
1825                    ;RP05/6 ERROR REGISTER #02 (RPER2) (#10)
1826
1827    000001          WCU=    1               ;WRITE CURRENT UNSAFE (BIT #0)
1828    000002          CSF=    2               ;CURRENT SINK FAILURE (BIT #1)
1829    000004          WSU=    4               ;WRITE SELECT UNSAFE (BIT #2)
1830    000010          CSU=    10              ;CURRENT SWITCH UNSAFE (BIT #3)
1831    000020          RAW=    20              ;READ AND WRITE (BIT #4)
1832    000040          TDF=    40              ;TRANSITIONS DETECTOR FAILURE (BIT #5)
1833    000100          TUF=    100             ;TRANSITIONS UNSAFE (BIT #6)
1834    000200          ABS=    200             ;ABNORMAL STOP (BIT #7)
1835    000400          WRU=    400             ;WRITE READY UNSAFE (BIT #8)
1836    001000          MHS=    1000            ;MUTLTIPLE HEAD SELECT (BIT #9)
1837    002000          NHS=    2000            ;NO HEAD SELECTION (BIT #10)
1838    004000          IXE=    4000            ;INDEX ERROR (BIT #11)
```

```
1839            020000                  PLU=    20000                   ;PLO UNSAFE (BIT #12)
1840
1841                                    ;OFFSET REGISTER (RPOF) (#11)
1842
1843            000001                  OF25=   1                       ;OFFSET 25 MICRO INCHES (BIT #0)
1844            000002                  OF50=   2                       ;OFFSET 50 MICRO INCHES (BIT #1)
1845            000004                  OF100=  4                       ;OFFSET 100 MICRO INCHES (BIT #2)
1846            000010                  OF200=  10                      ;OFFSET 200 MICRO INCHES (BIT #3)
1847            000020                  OF400=  20                      ;OFFSET 400 MICRO INCHES (BIT #4)
1848            000040                  OF800=  40                      ;OFFSET 800 MICRO INCHES (BIT #5)
1849            000200                  OFREV=  200                     ;OFFSET NEGATIVE (REVERSE) (BIT #5)
1850            002000                  HCI=    2000                    ;HEADER COMPARE INHIBIT (BIT #10)
1851            004000                  ECI=    4000                    ;ERROR CORRECTION CODE INHIBIT (BIT #11)
1852            010000                  FMT22=  10000                   ;FORMAT BIT (BIT #12
1853
1854                                    ;DESIRED CYLINDER ADDRESS (RPCA) (#12)
1855                                    ;(EACH BIT IS CALLED BY BIT NUMBER)
1856
1857                                    ;CURRENT CYLINDER ADDRESS (RPCC) (#13)
1858                                    ;(EACH BIT IS CALLED BY BIT NUMBER)
1859
1860                                    ;SERIAL NUMBER REGISTER (RPSN) (#14)
1861                                    ;(EACH IS CALLED BY BIT NUMBER)
1862
1863                                    ;RP04 ERROR REGISTER #03 (RPER3) (#15)
1864
1865            000001                  PSU=    1                       ;PACK SPEED UNSAFE (BIT #0)
1866            000002                  VUF=    2                       ;VELOCITY UNSAFE (BIT #1)
1867            000010                  UWR=    10                      ;ANY UNSAFE EXCEPT READ/WRITE (BIT #3)
1868            000040                  ACL=    40                      ;AC LOW (BIT #5)
1869            000100                  DCL=    100                     ;DC LOW (BIT #6)
1870            040000                  SKI=    40000                   ;SEEK INCOMPLETE (BIT #14)
1871            100000                  OCYL=   100000                  ;OFF CYLINDER (BIT #15)
1872
1873                                    ;RP05/6 ERROR REGISTER #03 (RPER3) (#15)
1874
1875            000001                  DCU=    1                       ;DC UNSAFE (BIT #0)
1876            000002                  WAO=    2                       ;WRITE AND OFFSET (BIT #1)
1877            000040                  ACL=    40                      ;AC LOW (BIT #5)
1878            000100                  DCL=    100                     ;DC LOW (BIT #6)
1879            020000                  OPE=    20000                   ;OPERATOR PLUG ERROR (BIT #13)
1880            040000                  SKI=    40000                   ;SEEK INCOMPLETE (BIT #14)
1881            100000                  OCYL=   100000                  ;OFF CYLINDER ERROR (BIT #15)
1882
1883                                    ;ECC POSITION REGISTER (RPEC1) (#16)
1884                                    ;(EACH BIT IS CALLED BY BIT NUMBER)
1885
1886                                    ;ECC PATTERN REGISTER (RPEC2) (#17)
1887                                    ;(EACH BIT IS CALLED BY BIT NUMBER)
1888
1889                                    ;;*****************************************************************
1890
1891                                    ;OP CODE DEFINITIONS
1892            000101                  NOOP=101
1893            000103                  UNLOAD=103
1894            000105                  SEEK=105
```

```
1895            000107                  RECAL=107
1896            000111                  DRVCLR=111
1897            000113                  RELEASE=113
1898            000115                  OFFSET=115
1899            000117                  RTC=117
1900            000121                  READIN=121
1901            000123                  PACK=123
1902            000131                  SEARCH=131
1903            000151                  WRCKD=151
1904            000153                  WRCKHD=153
1905            000161                  WRITE=161
1906            000163                  WRTHD=163
1907            000171                  READ=171
1908            000173                  READHD=173
1909            000141                  GETREG=141
1910            000143                  SETFORM=143
1911            000145                  SELDRV=145
1912
1913                                    ;OTHER EQUATES
1914
1915            177400                  SCTRWC = -256.                  ;WORD COUNT FOR SECTOR
1916            010000                  FMT22=10000                     ;FORMAT 22 BIT
1917
```

```
 1918                                   .SBTTL  COMMON TAGS
 1919
 1920                                   ;;*****************************************************************
 1921                                   ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
 1922                                   ;*USED IN THE PROGRAM.
 1923
 1924           001100                          .=1100
 1925  001100                           $CMTAG:                 ;;START OF COMMON TAGS
 1926  001100  000000                  $PASS:  .WORD   0       ;;CONTAINS PASS COUNT
 1927  001102     000                  $TSTNM: .BYTE   0       ;;CONTAINS THE TEST NUMBER
 1928  001103     000                  $ERFLG: .BYTE   0       ;;CONTAINS ERROR FLAG
 1929  001104  000000                  $ICNT:  .WORD   0       ;;CONTAINS SUBTEST ITERATION COUNT
 1930  001106  000000                  $LPADR: .WORD   0       ;;CONTAINS SCOPE LOOP ADDRESS
 1931  001110  000000                  $LPERR: .WORD   0       ;;CONTAINS SCOPE RETURN FOR ERRORS
 1932  001112  000000                  $ERTTL: .WORD   0       ;;CONTAINS TOTAL ERRORS DETECTED
 1933  001114     000                  $ITEMB: .BYTE   0       ;;CONTAINS ITEM CONTROL BYTE
 1934  001115     001                  $ERMAX: .BYTE   1       ;;CONTAINS MAX. ERRORS PER TEST
 1935  001116  000000                  $ERRPC: .WORD   0       ;;CONTAINS PC OF LAST ERROR INSTRUCTION
 1936  001120  000000                  $GDADR: .WORD   0       ;;CONTAINS ADDRESS OF 'GOOD' DATA
 1937  001122  000000                  $BDADR: .WORD   0       ;;CONTAINS ADDRESS OF 'BAD' DATA
 1938  001124  000000                  $GDDAT: .WORD   0       ;;CONTAINS 'GOOD' DATA
 1939  001126  000000                  $BDDAT: .WORD   0       ;;CONTAINS 'BAD' DATA
 1940  001130  000000                          .WORD   0       ;;RESERVED--NOT TO BE USED
 1941  001132  000000                          .WORD   0
 1942  001134     000                  $AUTOB: .BYTE   0       ;;AUTOMATIC MODE INDICATOR
 1943  001135     000                  $INTAG: .BYTE   0       ;;INTERRUPT MODE INDICATOR
 1944  001136  000000                          .WORD   0
 1945  001140  177570                  SWR:    .WORD   DSWR    ;;ADDRESS OF SWITCH REGISTER
 1946  001142  177570                  DISPLAY: .WORD  DDISP   ;;ADDRESS OF DISPLAY REGISTER
 1947  001144  177560                  $TKS:   177560          ;;TTY KBD STATUS
 1948  001146  177562                  $TKB:   177562          ;;TTY KBD BUFFER
 1949  001150  177564                  $TPS:   177564          ;;TTY PRINTER STATUS REG. ADDRESS
 1950  001152  177566                  $TPB:   177566          ;;TTY PRINTER BUFFER REG. ADDRESS
 1951  001154     000                  $NULL:  .BYTE   0       ;;CONTAINS NULL CHARACTER FOR FILLS
 1952  001155     002                  $FILLS: .BYTE   2       ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
 1953  001156     012                  $FILLC: .BYTE   12      ;;INSERT FILL CHARS. AFTER A "LINE FEED"
 1954  001157     000                  $TPFLG: .BYTE   0       ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
 1955  001160  000000                  $REGAD: .WORD   0       ;;CONTAINS THE ADDRESS FROM
 1956                                                          ;;WHICH  ($REG0) WAS OBTAINED
 1957  001162  000000                  $REG0:  .WORD   0       ;;CONTAINS (($REGAD)+0)
 1958  001164  000000                  $REG1:  .WORD   0       ;;CONTAINS (($REGAD)+2)
 1959  001166  000000                  $REG2:  .WORD   0       ;;CONTAINS (($REGAD)+4)
 1960  001170  000000                  $REG3:  .WORD   0       ;;CONTAINS (($REGAD)+6)
 1961  001172  000000                  $REG4:  .WORD   0       ;;CONTAINS (($REGAD)+10)
 1962  001174  000000                  $REG5:  .WORD   0       ;;CONTAINS (($REGAD)+12)
 1963  001176  000000                  $TMP0:  .WORD   0       ;;USER DEFINED
 1964  001200  000000                  $TMP1:  .WORD   0       ;;USER DEFINED
 1965  001202  000000                  $TMP2:  .WORD   0       ;;USER DEFINED
 1966  001204  000000                  $TIMES: 0               ;;MAX. NUMBER OF ITERATIONS
 1967  001206  000000                  $ESCAPE:0               ;;ESCAPE ON ERROR ADDRESS
 1968  001210  177607  000377          $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
 1969  001214     077                  $QUES:  .ASCII  /?/     ;;QUESTION MARK
 1970  001215     015                  $CRLF:  .ASCII  <15>    ;;CARRIAGE RETURN
 1971  001216  000012                  $LF:    .ASCIZ  <12>    ;;LINE FEED
 1972                                   ;;*****************************************************************
 1973           000015                  CR      =       15
```

```
1974            000012                  LF      =       12
1975  001220    000000                  C.SWR:  .WORD   0               ;CONTROL SWITCHES
1976  001222    000000                  SAVCSW: .WORD   0               ;PREVIOUS CONTENTS OF 'C.SWR'
1977  001224    000000                  CNTRLC: .WORD   0               ;CONTROL "C" FLAG
1978  001226    000000                  BUSADR: .WORD   0               ;GET ADDRESSES FROM THE TTY FLAG (0=NO, -1=YES)
1979  001230    000000                  LPTAVL: .WORD   0               ;LPT AVAILABLE STATUS (0=NO,1=YES)
1980  001232    000000                  DRVSEL: .WORD   0               ;DRIVES SELECTED FOR TESTING
1981  001234    037777  000000          TSTNMS: .WORD   37777,0         ;RUN TESTS 0-15
1982  001240    000000  000000          OPNFLG: .WORD   0,0             ;MODIFY TEST PARAMETER FLAGS
1983  001244    000000                  CLKSTA: .WORD   0               ;CLOCK STATUS (0=NO CLOCK,+1=KW11-P,
1984                                                                    ;AND -1=KW11-L)
1985  001246    000020                  TICKMS: .WORD   16.             ;16 MILLISECONDS PER CLOCK TICK
1986  001250    040432                  TICKUS: .WORD   16666.          ;16666 MIRCOSECONDS PER CLOCK TICK
1987  001252    000000                  BYPASS: .WORD   0
1988  001254    000000                  CHKDRV: .WORD   0               ;DRIVE UNDER TEST
1989  001256    000000                  DRVMSK: .WORD   0               ;DRIVE MASK BIT
1990  001260    000000                  SVSTAT: .WORD   0               ;STATUS/ERROR INDICATOR IS
1991                                                                    ;SAVED HERE ON AN ERROR
1992  001262    000000                  CYL.RD: .WORD   0               ;CYLINDER READ
1993  001264    000000                  TRK.RD: .WORD   0               ;TRACK READ
1994  001266    000000                  SEC.RD: .WORD   0               ;SECTOR READ
1995  001270    000000                  CYL.DS: .WORD   0               ;CYLINDER DESIRED
1996  001272    000000                  SEC.DS: .WORD   0               ;SECTOR DESIRED
1997  001274    000000                  TRK.DS: .WORD   0               ;TRACK DESIRED
1998  001276    000000                  TIM.UP: .WORD   0               ;MINIMUM TIME
1999  001300    000000                          .WORD   0               ;NUMBER OF COUNTS BELOW MIN. LIMIT
2000  001302    000000                          .WORD   0               ;MAXIMUM TIME
2001  001304    000000                          .WORD   0               ;NUMBER OF COUNTS ABOVE MAX. LIMIT
2002  001306    000000  000000                  .WORD   0,0             ;TOTAL TIME OF ALL SEEKS
2003  001312    000000                          .WORD   0               ;NUMBER OF SEEKS PERFORMED
2004  001314    000000                  TIM.DN: .WORD   0               ;MINIMUM TIME
2005  001316    000000                          .WORD   0               ;NUMBER OF COUNTS BELOW MIN. LIMIT
2006  001320    000000                          .WORD   0               ;MAXIMUM TIME
2007  001322    000000                          .WORD   0               ;NUMBER OF COUNTS ABOVE MAX. LIMIT
2008  001324    000000  000000                  .WORD   0,0             ;TOTAL TIME OF ALL SEEKS
2009  001330    000000                          .WORD   0               ;NUMBER OF SEEKS PERFORMED
2010  001332    000000                  TIM.PT: .WORD   0               ;POINTS TO TABLE OF TIMES
2011  001334    000000                  WCEFLG: .WORD   0               ;FATAL WRITE CHECK ERROR FLAG (TEST 20)
2012  001336    000000                  STALL0: .WORD   0               ;VARIABLE STALL (TEST 4)
2013  001340    000000  000000          SVADR:  .WORD   0,0             ;SAVE DISK ADDRESS (TEST 22)
2014  001344    000000                  SEKTMR: .WORD   0               ;SEEK TIMER  (TEST 10)
2015  001346    000000                  SEKCNT: .WORD   0               ;SEEK COUNTER
2016  001350    000000                  DELTA:  .WORD   0               ;TESTING RANGE FOR SERVO SETTLE DOWN TEST
2017  001352    165000                  TRCKWC: .WORD   -<256.*22.>      ;WORD COUNT FOR A FULL TRACK IN 16 BIT MODE
2018  001354    000012                  STALL1: .WORD   10.             ;10 MILLISECONDS STALL (TEST 0-11)
2019  001356    000012                  STALL2: .WORD   10.             ;10 MILLISECONDS STALL (TEST 16-21)
2020  001360    011610                  STALL3: .WORD   5000.           ;5 SEC STALL (TEST 22)
2021  001362    000031                  MXSTAL: .WORD   25.             ;MAX. INCREMENTING STALL ALLOWED IN TEST 4
2022  001364       144                  ERR.CT: .BYTE   100.            ;NUMBER OF ERRORS ALLOWED IN TESTS 16 - 21
2023                                                                    ;BEFORE GOING TO THE NEXT TEST
2024  001365       000                          .BYTE   0               ;RESERVED
2025
2026                                    ;ADDRESSES AND VECTORS
2027  001366    176700                  RH.ADR: .WORD   176700          ;RH11-RH70 UNIBUS ADDRESS
2028  001370    000254  000240          RHVEC:  .WORD   254,5*32.        ;RH11-RH70 VECTOR ADDRESS AND PRIORITY
2029  001374    000104  000106          PKV:    .WORD   104,106         ;KW11-P VECTOR ADDRESS
```

```
 2030  001400  172540                 PKCS:   .WORD   172540          ;KW11-P CONTROL AND STATUS REG.
 2031  001402  172542                 PKB:    .WORD   172542          ;KW11-P COUNT SET BUFFER
 2032  001404  172544                 PKC:    .WORD   172544          ;KW11-P COUNTER
 2033  001406  000100  000102         LKV:    .WORD   100,102         ;KW11-L VECTOR ADDRESS
 2034  001412  177546                 LKS:    .WORD   177546          ;KW11-L STATUS REGISTER
 2035  001414  177564                 TPS:    .WORD   177564          ;TTY PRINTER STATUS
 2036  001416  177566                 TPB:    .WORD   177566          ;TTY PRINTER BUFFER
 2037  001420  177514                 LPS:    .WORD   177514          ;LINE PRINTER STATUS
 2038  001422  177516                 LPB:    .WORD   177516          ;LINE PRINTER BUFFER
 2039
 2040                                 ;BIT TABLE
 2041  001424  000001                 BITS:   .WORD   BIT00
 2042  001426  000002                         .WORD   BIT01
 2043  001430  000004                         .WORD   BIT02
 2044  001432  000010                         .WORD   BIT03
 2045  001434  000020                         .WORD   BIT04
 2046  001436  000040                         .WORD   BIT05
 2047  001440  000100                         .WORD   BIT06
 2048  001442  000200                         .WORD   BIT07
 2049  001444  000400                         .WORD   BIT08
 2050  001446  001000                         .WORD   BIT09
 2051  001450  002000                         .WORD   BIT10
 2052  001452  004000                         .WORD   BIT11
 2053  001454  010000                         .WORD   BIT12
 2054  001456  020000                         .WORD   BIT13
 2055  001460  040000                         .WORD   BIT14
 2056  001462  100000                         .WORD   BIT15
 2057  001464  000001                         .WORD   BIT00
 2058  001466  000002                         .WORD   BIT01
 2059  001470  000004                         .WORD   BIT02
 2060  001472  000010                         .WORD   BIT03
 2061  001474  000020                         .WORD   BIT04
 2062  001476  000040                         .WORD   BIT05
 2063  001500  000100                         .WORD   BIT06
 2064  001502  000200                         .WORD   BIT07
 2065
 2066
 2067                                 ;COMMON STORAGE FOR TEST PARAMETER
 2068  001504  000000                 PRM:    .WORD   0
 2069  001506  000000                 RPT:    .WORD   0               ;REPEAT COUNTS FOR ALL TESTS
 2070  001510  000000                 FC:     .WORD   0               ;FIRST CYLINDER
 2071  001512  000000                 LC:     .WORD   0               ;LAST CYLINDER
 2072  001514  000000                 IC:     .WORD   0               ;INCREMENT CYLINDER
 2073  001516  000000                 FT:     .WORD   0               ;FIRST TRACK
 2074  001520  000000                 LT:     .WORD   0               ;LAST TRACK
 2075  001522  000000                 IT:     .WORD   0               ;INCREMENT TRACK
 2076  001524  000000                 FS:     .WORD   0               ;FIRST SECTOR
 2077  001526  000000                 LS:     .WORD   0               ;LAST SECTOR
 2078  001530  000000                 PAT:    .WORD   0               ;PATTERN CODE
 2079  001532  000000                 NC1:    .WORD   0               ;NEW CYLINDER ADDRESS
 2080  001534  000000                 NC2:    .WORD   0               ;NEW CYLINDER ADDRESS
 2081
 2082                                 ;TABLE OF PARAMETER POINTERS
 2083  001536  002330                 PRMPT:  .WORD   PRM0
 2084  001540  002344                         .WORD   PRM1
 2085  001542  002372                         .WORD   PRM2
```

```
 2086  001544  002414                          .WORD   PRM3
 2087  001546  002436                          .WORD   PRM4
 2088  001550  002460                          .WORD   PRM5
 2089  001552  002502                          .WORD   PRM6
 2090  001554  002524                          .WORD   PRM7
 2091  001556  002544                          .WORD   PRM10
 2092  001560  002564                          .WORD   PRM11
 2093  001562  002606                          .WORD   PRM12
 2094  001564  002622                          .WORD   PRM13
 2095  001566  002636                          .WORD   PRM14
 2096  001570  002652                          .WORD   PRM15
 2097  001572  002666                          .WORD   PRM16
 2098  001574  002700                          .WORD   PRM17
 2099  001576  002712                          .WORD   PRM20
 2100  001600  002744                          .WORD   PRM21
 2101  001602  002760                          .WORD   PRM22
 2102  001604  000000                          .WORD   0               ;TERMINATOR
 2103
 2104                                  ;PARAMETER UPPER LIMIT
 2105  001606  032767                  PRMLMT: .WORD   32767           ;"R"
 2106  001610  000632                          .WORD   410.            ;"FC"
 2107  001612  000632                          .WORD   410.            ;"LC"
 2108  001614  001456                          .WORD   814.            ;"FC'"
 2109  001616  001456                          .WORD   814.            ;"LC'"
 2110  001620  001456                          .WORD   814.            ;"IC"
 2111  001622  000022                          .WORD   18.             ;"FT"
 2112  001624  000022                          .WORD   18.             ;"LT"
 2113  001626  000022                          .WORD   18.             ;"IT"
 2114  001630  000025                          .WORD   21.             ;"FS"
 2115  001632  000025                          .WORD   21.             ;"LS"
 2116  001634  177777                          .WORD   177777          ;"PAT"
 2117
 2118                                  ;TABLE OF MESSAGE POINTERS
 2119  001636  043146                  PRMMSG: .WORD   MSG.R
 2120  001640  043150                          .WORD   MSG.FC
 2121  001642  043153                          .WORD   MSG.LC
 2122  001644  043156                          .WORD   MSGFCP
 2123  001646  043162                          .WORD   MSGLCP
 2124  001650  043166                          .WORD   MSG.IC
 2125  001652  043171                          .WORD   MSG.FT
 2126  001654  043174                          .WORD   MSG.LT
 2127  001656  043177                          .WORD   MSG.IT
 2128  001660  043202                          .WORD   MSG.FS
 2129  001662  043205                          .WORD   MSG.LS
 2130
 2131                                  ;STATUS/ERROR INDICATOR MESSAGES POINTER TABLE
 2132                                  ;DEFAULT VALUES OF TEST PARAMETERS
 2133  001664  001125  000310  000632  DFLT:   .WORD   1125,200.,410.,814.,0,0 ;RECAL/SEEK (T0)
 2134  001672  001456  000000  000000
 2135  001700  003377  000144  000000          .WORD   3377,100.,0,128.,0,256.,0,0,0,0,0 ;SEEK/SEEK (T1)
 2136  001706  000200  000000  000400
 2137  001714  000000  000000  000000
 2138  001722  000000  000000
 2139  001726  001177  000001  000000          .WORD   1177,1,0,410.,0,814.,1,0,0 ;INCREMENT SEEK (T2)
 2140  001734  000632  000000  001456
 2141  001742  000001  000000  000000
```

```
2142  001750  001177  000010  000000          .WORD   1177,10,0,256.,0,256.,1,0,0 ;STEPPING SEEK (T3)
2143  001756  000400  000000  000400
2144  001764  000001  000000  000000
2145  001772  001177  000001  000000          .WORD   1177,1,0,410.,0,814.,1,0,0 ;OSCILLATING SEEK (T4)
2146  002000  000632  000000  001456
2147  002006  000001  000000  000000
2148  002014  001177  000001  000000          .WORD   1177,1,0,410.,0,814.,1,0,0 ;CONVERGING/DIVERGING SEEK (T5)
2149  002022  000632  000000  001456
2150  002030  000001  000000  000000
2151  002036  001177  000001  000000          .WORD   1177,1,0,410.,0,814.,1,0,0 ;SERVO ADDRESSING LOGIC NOISE (T6)
2152  002044  000632  000000  001456
2153  002052  000001  000000  000000
2154  002060  000337  011610  000000          .WORD   337,5000.,0,410.,0,814.,0,18.  ;RANDOM SEEK TEST (T7)
2155  002066  000632  000000  001456
2156  002074  000000  000022
2157  002100  000177  000001  000000          .WORD   177,1,0,410.,0,814.,100.,0  ;SERVO SETTLE DOWN TEST (T10)
2158  002106  000632  000000  001456
2159  002114  000144  000000
2160  002120  001177  000001  000000          .WORD   1177,1,0,410.,0,814.,1,0,0      ;ALL SEEKS TEST (T11)
2161  002126  000632  000000  001456
2162  002134  000001  000000  000000
2163  002142  001113  000001  000000          .WORD   1113,1,0,0,0,0  ;ROTATIONAL SPEED TIMING TEST (T12)
2164  002150  000000  000000  000000
2165  002156  000037  000001  000000          .WORD   37,1,0,410.,0,814.      ;ONE CYLINDER SEEK TIMING TEST (T13)
2166  002164  000632  000000  001456
2167  002172  000037  000001  000000          .WORD   37,1,0,136.,0,255.      ;ACCESS TIME MEASURMENT TEST (T14)
2168  002200  000210  000000  000377
2169  002206  000037  000001  000000          .WORD   37,1,0,410.,0,814.      ;MAXIMUM SEEK TIMING TEST (T15)
2170  002214  000632  000000  001456
2171  002222  000113  000001  000000          .WORD   113,1,0,0,0     ;SECTOR ADDRESSING TEST (T16)
2172  002230  000000  000000
2173  002234  001013  000001  000000          .WORD   1013,1,0,0,0    ;TRACK ADDRESSING TEST (T17)
2174  002242  000000  000000
2175  002246  007777  000001  000000          .WORD   7777,1,0,410.,0,814.,64.,0,18.,1,1,0,177777 ;DATA TEST (T20)
2176  002254  000632  000000  001456
2177  002262  000100  000000  000022
2178  002270  000001  000001  000000
2179  002276  177777
2180  002300  000037  047040  000000          .WORD   37,20000.,0,410.,0,814. ;EXERCISER (T21)
2181  002306  000632  000000  001456
2182  002314  000037  011610  000000          .WORD   37,5000.,0,136.,0,255.  ;RP04 ACCESS TIME ADJUSTMENT TEST (T22)
2183  002322  000210  000000  000377
2184
2185                                    ;PARAMETER TABLES
2186
2187                                    ;RECAL/SEEK (T0)
2188  002330  001125                    PRM0:   .WORD   1125
2189  002332  000310                            .WORD   200.
2190  002334  000632                            .WORD   410.
2191  002336  001456                            .WORD   814.
2192  002340  000000                            .WORD   0
2193  002342  000000                            .WORD   0
2194
2195                                    ;SEEK/SEEK (T1)
2196  002344  003377                    PRM1:   .WORD   3377
2197  002346  000144                            .WORD   100.
```

```
 2198  002350  000000                          .WORD   0
 2199  002352  000200                          .WORD   128.
 2200  002354  000000                          .WORD   0
 2201  002356  000400                          .WORD   256.
 2202  002360  000000                          .WORD   0
 2203  002362  000000                          .WORD   0
 2204  002364  000000                          .WORD   0
 2205  002366  000000                          .WORD   0
 2206  002370  000000                          .WORD   0
 2207
 2208                                  ;INCREMENT SEEK (T2)
 2209  002372  001177                 PRM2:    .WORD   1177
 2210  002374  000001                          .WORD   1
 2211  002376  000000                          .WORD   0
 2212  002400  000632                          .WORD   410.
 2213  002402  000000                          .WORD   0
 2214  002404  001456                          .WORD   814.
 2215  002406  000001                          .WORD   1
 2216  002410  000000                          .WORD   0
 2217  002412  000000                          .WORD   0
 2218
 2219                                  ;STEPPING SEEK (T3)
 2220  002414  001177                 PRM3:    .WORD   1177
 2221  002416  000001                          .WORD   1
 2222  002420  000000                          .WORD   0
 2223  002422  000400                          .WORD   256.
 2224  002424  000000                          .WORD   0
 2225  002426  001000                          .WORD   512.
 2226  002430  000001                          .WORD   1
 2227  002432  000000                          .WORD   0
 2228  002434  000000                          .WORD   0
 2229
 2230                                  ;OSCILLATING SEEK (T4)
 2231  002436  001177                 PRM4:    .WORD   1177
 2232  002440  000001                          .WORD   1
 2233  002442  000000                          .WORD   0
 2234  002444  000632                          .WORD   410.
 2235  002446  000000                          .WORD   0
 2236  002450  001456                          .WORD   814.
 2237  002452  000001                          .WORD   1
 2238  002454  000000                          .WORD   0
 2239  002456  000000                          .WORD   0
 2240
 2241                                  ;CONVERGING/DIVERGING SEEK (T5)
 2242  002460  001177                 PRM5:    .WORD   1177
 2243  002462  000001                          .WORD   1
 2244  002464  000000                          .WORD   0
 2245  002466  000632                          .WORD   410.
 2246  002470  000000                          .WORD   0
 2247  002472  001456                          .WORD   814.
 2248  002474  000001                          .WORD   1
 2249  002476  000000                          .WORD   0
 2250  002500  000000                          .WORD   0
 2251
 2252                                  ;SERVO ADDRESSING LOGIC NOISE GENERATOR (T6)
 2253  002502  001177                 PRM6:    .WORD   1177
```

```
 2254  002504  000001                        .WORD   1
 2255  002506  000000                        .WORD   0
 2256  002510  000632                        .WORD   410.
 2257  002512  000000                        .WORD   0
 2258  002514  001456                        .WORD   814.
 2259  002516  000001                        .WORD   1
 2260  002520  000000                        .WORD   0
 2261  002522  000000                        .WORD   0
 2262
 2263                                  ;RANDOM SEEK TEST (T7)
 2264  002524  000337                 PRM7:   .WORD   337
 2265  002526  011610                        .WORD   5000.
 2266  002530  000000                        .WORD   0
 2267  002532  000632                        .WORD   410.
 2268  002534  000000                        .WORD   0
 2269  002536  001456                        .WORD   814.
 2270  002540  000000                        .WORD   0
 2271  002542  000022                        .WORD   18.
 2272
 2273                                  ;SERVO SETTLE DOWN TEST (T10)
 2274  002544  000177                 PRM10:  .WORD   177
 2275  002546  000001                        .WORD   1
 2276  002550  000000                        .WORD   0
 2277  002552  000632                        .WORD   410.
 2278  002554  000000                        .WORD   0
 2279  002556  001456                        .WORD   814.
 2280  002560  000144                        .WORD   100.
 2281  002562  000000                        .WORD   0
 2282
 2283                                  ;ALL SEEKS TEST (T11)
 2284  002564  001177                 PRM11:  .WORD   1177
 2285  002566  000001                        .WORD   1
 2286  002570  000000                        .WORD   0
 2287  002572  000632                        .WORD   410.
 2288  002574  000000                        .WORD   0
 2289  002576  001456                        .WORD   814.
 2290  002600  000001                        .WORD   1
 2291  002602  000000                        .WORD   0
 2292  002604  000000                        .WORD   0
 2293
 2294                                  ;ROTATIONAL SPEED TIMING TEST (T12)
 2295  002606  001113                 PRM12:  .WORD   1113
 2296  002610  000001                        .WORD   1
 2297  002612  000000                        .WORD   0
 2298  002614  000000                        .WORD   0
 2299  002616  000000                        .WORD   0
 2300  002620  000000                        .WORD   0
 2301
 2302                                  ;ONE CYLINDER SEEK TIMING TEST (T13)
 2303  002622  000037                 PRM13:  .WORD   37
 2304  002624  000001                        .WORD   1
 2305  002626  000000                        .WORD   0
 2306  002630  000632                        .WORD   410.
 2307  002632  000000                        .WORD   0
 2308  002634  001456                        .WORD   814.
 2309
```

```
 2310                                   ;ACCESS TIME MEASURMENT TEST (T14)
 2311  002636  000037                   PRM14:  .WORD   37
 2312  002640  000001                           .WORD   1
 2313  002642  000000                           .WORD   0
 2314  002644  000210                           .WORD   136.
 2315  002646  000000                           .WORD   0
 2316  002650  000377                           .WORD   255.
 2317
 2318                                   ;MAXIMUM SEEK TIMING TEST (T15)
 2319  002652  000037                   PRM15:  .WORD   37
 2320  002654  000001                           .WORD   1
 2321  002656  000000                           .WORD   0
 2322  002660  000632                           .WORD   410.
 2323  002662  000000                           .WORD   0
 2324  002664  001456                           .WORD   814.
 2325
 2326                                   ;SECTOR ADDRESSING TEST (T16)
 2327  002666  000113                   PRM16:  .WORD   113
 2328  002670  000001                           .WORD   1
 2329  002672  000000                           .WORD   0
 2330  002674  000000                           .WORD   0
 2331  002676  000000                           .WORD   0
 2332
 2333                                   ;TRACK ADDRESSING TEST (T17)
 2334  002700  001013                   PRM17:  .WORD   1013
 2335  002702  000001                           .WORD   1
 2336  002704  000000                           .WORD   0
 2337  002706  000000                           .WORD   0
 2338  002710  000000                           .WORD   0
 2339
 2340                                   ;DATA TEST (T20)
 2341  002712  007777                   PRM20:  .WORD   7777
 2342  002714  000001                           .WORD   1
 2343  002716  000000                           .WORD   0
 2344  002720  000632                           .WORD   410.
 2345  002722  000000                           .WORD   0
 2346  002724  001456                           .WORD   814.
 2347  002726  000100                           .WORD   64.
 2348  002730  000000                           .WORD   0
 2349  002732  000022                           .WORD   18.
 2350  002734  000001                           .WORD   1
 2351  002736  000001                           .WORD   1
 2352  002740  000000                           .WORD   0
 2353  002742  177777                   PTRN15: .WORD   177777
 2354
 2355                                   ;EXERCISER (T21)
 2356  002744  000037                   PRM21:  .WORD   37
 2357  002746  047040                           .WORD   20000.
 2358  002750  000000                           .WORD   0
 2359  002752  000632                           .WORD   410.
 2360  002754  000000                           .WORD   0
 2361  002756  001456                           .WORD   814.
 2362
 2363                                   ;RP04 ACCESS TIME ADJUSTMENT TEST (T22)
 2364  002760  000037                   PRM22:  .WORD   37
 2365  002762  011610                           .WORD   5000.
```

```
2366  002764  000000                    .WORD   0
2367  002766  000210                    .WORD   136.
2368  002770  000000                    .WORD   0
2369  002772  000377                    .WORD   255.
2370
2371
2372
2373                              ;SEEK TIMING LIMITS
2374
2375  002774  043710              T7A:  .WORD   MSG7
2376  002776  000000                    .WORD   0
2377  003000  003142                    .WORD   1634.           ;(16.67-2%)**2
2378  003002  003244                    .WORD   1700.           ;(16.67+2%)**2
2379
2380  003004  043710              T7B:  .WORD   MSG7
2381  003006  000000                    .WORD   0
2382  003010  003131                    .WORD   1625.           ;(16.67-2.5%)**2
2383  003012  003255                    .WORD   1709.           ;(16.67+2.5%)**2
2384
2385  003014  043742              T10:  .WORD   MSG10A
2386  003016  044011                    .WORD   MSG10B
2387  003020  000000                    .WORD   0               ;NO LOWER LIMIT
2388  003022  001750                    .WORD   1000.           ;(7+3)**2
2389
2390  003024  044026              T11:  .WORD   MSG11A
2391  003026  044074                    .WORD   MSG11B
2392  003030  000000                    .WORD   0               ;NO LOWER LIMIT
2393  003032  012574                    .WORD   5500.           ;(28+2)**2
2394
2395  003034  044111              T12:  .WORD   MSG12A
2396  003036  044153                    .WORD   MSG12B
2397  003040  000000                    .WORD   0               ;NO LOWER LIMIT
2398  003042  012574                    .WORD   5500.           ;(50+2)**2
2399
2400  003044  003104              PAT.PT: .WORD PAT0            ;TABLE OF POINTERS WHICH POINT TO THE
2401  003046  003144                    .WORD   PAT1            ;PATTERNS USED BY THE DATA TEST
2402  003050  003204                    .WORD   PAT2
2403  003052  003244                    .WORD   PAT3
2404  003054  003304                    .WORD   PAT4
2405  003056  003344                    .WORD   PAT5
2406  003060  003404                    .WORD   PAT6
2407  003062  003444                    .WORD   PAT7
2408  003064  003504                    .WORD   PAT8
2409  003066  003544                    .WORD   PAT9
2410  003070  003604                    .WORD   PAT10
2411  003072  003644                    .WORD   PAT11
2412  003074  003704                    .WORD   PAT12
2413  003076  003744                    .WORD   PAT13
2414  003100  004004                    .WORD   PAT14
2415  003102  004044                    .WORD   PAT15
2416
2417                              ;PATTERNS 0 THRU 15
2418
2419  003104  165555              PAT0: .WORD   165555          ;PATTERN 0
2420  003106  133333                    .WORD   133333
2421  003110  165555                    .WORD   165555
```

```
 2422  003112  133333                          .WORD   133333
 2423  003114  165555                          .WORD   165555
 2424  003116  133333                          .WORD   133333
 2425  003120  165555                          .WORD   165555
 2426  003122  133333                          .WORD   133333
 2427  003124  165555                          .WORD   165555
 2428  003126  133333                          .WORD   133333
 2429  003130  165555                          .WORD   165555
 2430  003132  133333                          .WORD   133333
 2431  003134  165555                          .WORD   165555
 2432  003136  133333                          .WORD   133333
 2433  003140  165555                          .WORD   165555
 2434  003142  133333                          .WORD   133333
 2435
 2436  003144  000001                  PAT1:   .WORD   000001          ;PATTERN 1
 2437  003146  000003                          .WORD   000003
 2438  003150  000007                          .WORD   000007
 2439  003152  000017                          .WORD   000017
 2440  003154  000037                          .WORD   000037
 2441  003156  000077                          .WORD   000077
 2442  003160  000177                          .WORD   000177
 2443  003162  000377                          .WORD   000377
 2444  003164  000777                          .WORD   000777
 2445  003166  001777                          .WORD   001777
 2446  003170  003777                          .WORD   003777
 2447  003172  007777                          .WORD   007777
 2448  003174  017777                          .WORD   017777
 2449  003176  037777                          .WORD   037777
 2450  003200  077777                          .WORD   077777
 2451  003202  177777                          .WORD   177777
 2452
 2453  003204  177776                  PAT2:   .WORD   177776          ;PATTERN 2
 2454  003206  177774                          .WORD   177774
 2455  003210  177770                          .WORD   177770
 2456  003212  177760                          .WORD   177760
 2457  003214  177740                          .WORD   177740
 2458  003216  177700                          .WORD   177700
 2459  003220  177600                          .WORD   177600
 2460  003222  177400                          .WORD   177400
 2461  003224  177000                          .WORD   177000
 2462  003226  176000                          .WORD   176000
 2463  003230  174000                          .WORD   174000
 2464  003232  170000                          .WORD   170000
 2465  003234  160000                          .WORD   160000
 2466  003236  140000                          .WORD   140000
 2467  003240  100000                          .WORD   100000
 2468  003242  000000                          .WORD   000000
 2469
 2470  003244  000000                  PAT3:   .WORD   000000          ;PATTERN 3
 2471  003246  000000                          .WORD   000000
 2472  003250  000000                          .WORD   000000
 2473  003252  177777                          .WORD   177777
 2474  003254  177777                          .WORD   177777
 2475  003256  177777                          .WORD   177777
 2476  003260  000000                          .WORD   000000
 2477  003262  000000                          .WORD   000000
```

```
 2478  003264  177777                          .WORD   177777
 2479  003266  177777                          .WORD   177777
 2480  003270  000000                          .WORD   000000
 2481  003272  177777                          .WORD   177777
 2482  003274  000000                          .WORD   000000
 2483  003276  177777                          .WORD   177777
 2484  003300  000000                          .WORD   000000
 2485  003302  177777                          .WORD   177777
 2486
 2487  003304  000000                PAT4:     .WORD   000000          ;PATTERN 4
 2488  003306  010421                          .WORD   010421
 2489  003310  021042                          .WORD   021042
 2490  003312  031463                          .WORD   031463
 2491  003314  042104                          .WORD   042104
 2492  003316  052525                          .WORD   052525
 2493  003320  063146                          .WORD   063146
 2494  003322  073567                          .WORD   073567
 2495  003324  104210                          .WORD   104210
 2496  003326  114631                          .WORD   114631
 2497  003330  125252                          .WORD   125252
 2498  003332  135673                          .WORD   135673
 2499  003334  146314                          .WORD   146314
 2500  003336  156735                          .WORD   156735
 2501  003340  167356                          .WORD   167356
 2502  003342  177777                          .WORD   177777
 2503
 2504  003344  052525                PAT5:     .WORD   052525          ;PATTERN 5
 2505  003346  052525                          .WORD   052525
 2506  003350  052525                          .WORD   052525
 2507  003352  125252                          .WORD   125252
 2508  003354  125252                          .WORD   125252
 2509  003356  125252                          .WORD   125252
 2510  003360  052525                          .WORD   052525
 2511  003362  052525                          .WORD   052525
 2512  003364  125252                          .WORD   125252
 2513  003366  125252                          .WORD   125252
 2514  003370  052525                          .WORD   052525
 2515  003372  125252                          .WORD   125252
 2516  003374  052525                          .WORD   052525
 2517  003376  125252                          .WORD   125252
 2518  003400  052525                          .WORD   052525
 2519  003402  125252                          .WORD   125252
 2520
 2521  003404  007417                PAT6:     .WORD   007417          ;PATTERN 6
 2522  003406  007417                          .WORD   007417
 2523  003410  007417                          .WORD   007417
 2524  003412  170360                          .WORD   170360
 2525  003414  170360                          .WORD   170360
 2526  003416  170360                          .WORD   170360
 2527  003420  007417                          .WORD   007417
 2528  003422  007417                          .WORD   007417
 2529  003424  170360                          .WORD   170360
 2530  003426  170360                          .WORD   170360
 2531  003430  007417                          .WORD   007417
 2532  003432  170360                          .WORD   170360
 2533  003434  007417                          .WORD   007417
```

```
 2534  003436  170360                          .WORD   170360
 2535  003440  007417                          .WORD   007417
 2536  003442  170360                          .WORD   170360
 2537
 2538  003444  026455                  PAT7:   .WORD   026455          ;PATTERN 7
 2539  003446  026455                          .WORD   026455
 2540  003450  026455                          .WORD   026455
 2541  003452  151322                          .WORD   151322
 2542  003454  151322                          .WORD   151322
 2543  003456  151322                          .WORD   151322
 2544  003460  026455                          .WORD   026455
 2545  003462  026455                          .WORD   026455
 2546  003464  151322                          .WORD   151322
 2547  003466  151322                          .WORD   151322
 2548  003470  026455                          .WORD   026455
 2549  003472  151322                          .WORD   151322
 2550  003474  026455                          .WORD   026455
 2551  003476  151322                          .WORD   151322
 2552  003500  026455                          .WORD   026455
 2553  003502  151322                          .WORD   151322
 2554
 2555  003504  165555                  PAT8:   .WORD   165555          ;PATTERN 8
 2556  003506  133333                          .WORD   133333
 2557  003510  165555                          .WORD   165555
 2558  003512  133333                          .WORD   133333
 2559  003514  165555                          .WORD   165555
 2560  003516  133333                          .WORD   133333
 2561  003520  165555                          .WORD   165555
 2562  003522  133333                          .WORD   133333
 2563  003524  165555                          .WORD   165555
 2564  003526  133333                          .WORD   133333
 2565  003530  165555                          .WORD   165555
 2566  003532  133333                          .WORD   133333
 2567  003534  165555                          .WORD   165555
 2568  003536  133333                          .WORD   133333
 2569  003540  165555                          .WORD   165555
 2570  003542  133333                          .WORD   133333
 2571
 2572  003544  000001                  PAT9:   .WORD   000001          ;PATTERN 9
 2573  003546  000002                          .WORD   000002
 2574  003550  000004                          .WORD   000004
 2575  003552  000010                          .WORD   000010
 2576  003554  000020                          .WORD   000020
 2577  003556  000040                          .WORD   000040
 2578  003560  000100                          .WORD   000100
 2579  003562  000200                          .WORD   000200
 2580  003564  000400                          .WORD   000400
 2581  003566  001000                          .WORD   001000
 2582  003570  002000                          .WORD   002000
 2583  003572  004000                          .WORD   004000
 2584  003574  010000                          .WORD   010000
 2585  003576  020000                          .WORD   020000
 2586  003600  040000                          .WORD   040000
 2587  003602  100000                          .WORD   100000
 2588
 2589  003604  177776                  PAT10:  .WORD   177776          ;PATTERN 10
```

```
 2590  003606  177775                          .WORD   177775
 2591  003610  177773                          .WORD   177773
 2592  003612  177767                          .WORD   177767
 2593  003614  177757                          .WORD   177757
 2594  003616  177737                          .WORD   177737
 2595  003620  177677                          .WORD   177677
 2596  003622  177577                          .WORD   177577
 2597  003624  177377                          .WORD   177377
 2598  003626  176777                          .WORD   176777
 2599  003630  175777                          .WORD   175777
 2600  003632  173777                          .WORD   173777
 2601  003634  167777                          .WORD   167777
 2602  003636  157777                          .WORD   157777
 2603  003640  137777                          .WORD   137777
 2604  003642  077777                          .WORD   077777
 2605
 2606  003644  172666                PAT11:    .WORD   172666          ;PATTERN 11
 2607  003646  155555                          .WORD   155555
 2608  003650  172666                          .WORD   172666
 2609  003652  155555                          .WORD   155555
 2610  003654  172666                          .WORD   172666
 2611  003656  155555                          .WORD   155555
 2612  003660  172666                          .WORD   172666
 2613  003662  155555                          .WORD   155555
 2614  003664  172666                          .WORD   172666
 2615  003666  155555                          .WORD   155555
 2616  003670  172666                          .WORD   172666
 2617  003672  155555                          .WORD   155555
 2618  003674  172666                          .WORD   172666
 2619  003676  155555                          .WORD   155555
 2620  003700  172666                          .WORD   172666
 2621  003702  155555                          .WORD   155555
 2622
 2623  003704  077777                PAT12:    .WORD   077777          ;PATTERN 12
 2624  003706  137777                          .WORD   137777
 2625  003710  157777                          .WORD   157777
 2626  003712  167777                          .WORD   167777
 2627  003714  173777                          .WORD   173777
 2628  003716  175777                          .WORD   175777
 2629  003720  176777                          .WORD   176777
 2630  003722  177377                          .WORD   177377
 2631  003724  177577                          .WORD   177577
 2632  003726  177677                          .WORD   177677
 2633  003730  177737                          .WORD   177737
 2634  003732  177757                          .WORD   177757
 2635  003734  177767                          .WORD   177767
 2636  003736  177773                          .WORD   177773
 2637  003740  177775                          .WORD   177775
 2638  003742  177776                          .WORD   177776
 2639
 2640  003744  153333                PAT13:    .WORD   153333          ;PATTERN 13
 2641  003746  066667                          .WORD   066667
 2642  003750  153333                          .WORD   153333
 2643  003752  066667                          .WORD   066667
 2644  003754  153333                          .WORD   153333
 2645  003756  066667                          .WORD   066667
```

```
 2646  003760  153333                          .WORD   153333
 2647  003762  066667                          .WORD   066667
 2648  003764  153333                          .WORD   153333
 2649  003766  066667                          .WORD   066667
 2650  003770  153333                          .WORD   153333
 2651  003772  066667                          .WORD   066667
 2652  003774  153333                          .WORD   153333
 2653  003776  066667                          .WORD   066667
 2654  004000  153333                          .WORD   153333
 2655  004002  066667                          .WORD   066667
 2656
 2657  004004  000000                  PAT14:  .WORD   000000          ;PATTERN 14
 2658  004006  177777                          .WORD   177777
 2659  004010  177777                          .WORD   177777
 2660  004012  177777                          .WORD   177777
 2661  004014  177777                          .WORD   177777
 2662  004016  177777                          .WORD   177777
 2663  004020  177777                          .WORD   177777
 2664  004022  177777                          .WORD   177777
 2665  004024  177777                          .WORD   177777
 2666  004026  177777                          .WORD   177777
 2667  004030  177777                          .WORD   177777
 2668
 2669  004032  177777                          .WORD   177777
 2670  004034  177777                          .WORD   177777
 2671  004036  177777                          .WORD   177777
 2672  004040  177777                          .WORD   177777
 2673
 2674  004042  177777                          .WORD   177777
 2675
 2676  004044  177777                  PAT15:  .WORD   177777          ;PATTERN 15
 2677  004046  000000                          .WORD   000000
 2678  004050  000000                          .WORD   000000
 2679  004052  000000                          .WORD   000000
 2680  004054  000000                          .WORD   000000
 2681  004056  000000                          .WORD   000000
 2682  004060  000000                          .WORD   000000
 2683  004062  000000                          .WORD   000000
 2684  004064  000000                          .WORD   000000
 2685  004066  000000                          .WORD   000000
 2686  004070  000000                          .WORD   000000
 2687  004072  000000                          .WORD   000000
 2688  004074  000000                          .WORD   000000
 2689  004076  000000                          .WORD   000000
 2690  004100  000000                          .WORD   000000
 2691  004102  000000                          .WORD   000000
 2692
 2693                                  ;DPB (DATA PARAMETER BLOCK)
 2694
 2695  004104      000                 DPB.A:  .BYTE   0               ;(0) DRIVE NUMBER
 2696  004105      000                         .BYTE   0               ;(1) OFFSET VALUE OR FMT22, ECI, AND HCI
 2697  004106      000                         .BYTE   0               ;(2) COMMAND
 2698  004107      000                         .BYTE   0               ;(3) PSEL AND A17 AND A16
 2699  004110  000000                          .WORD   0               ;(4) WORD COUNT (MUST BE NEG.)
 2700  004112  050202                          .WORD   BUFFER          ;(6) BUFFER ADDRESS OR
 2701                                                                  ;REGISTER TABLE POINTER
```

```
2702  004114      000                        .BYTE   0               ;(10) SECTOR ADDRESS OR
2703                                                                  ;FIRST REG. INDEX
2704  004115      000                        .BYTE   0               ;(11) TRACK ADDRESS OR
2705                                                                  ;LAST REG. INDEX
2706  004116  000000                         .WORD   0               ;(12) CYLINDER ADDRESS
2707  004120  004204                         .WORD   RP.REG          ;(14) ERROR TABLE POINTER
2708                                                                  ;POINTS TO THE FIRST OF TWENTY
2709                                                                  ;LOCATIONS OF WHERE THE DRIVER
2710                                                                  ;IS TO STORE THE RH11/RP04
2711                                                                  ;REGISTERS ON AN ERROR. IF LEFT
2712                                                                  ;ZERO REGISTERS ARE NOT SAVED.
2713  004122  000000                         .WORD   0               ;(16) STATUS/ERROR INDICATOR
2714                                                                  ;BIT15=1=>ERROR OCCURRED
2715                                                                  ;BIT07=1=>DONE
2716                                                                  ;BIT14-BIT09 AND BIT06-BIT03
2717                                                                  ;INDICATE TYPE OF ERROR
2718
2719  004124      000              DPB.B:    .BYTE   0               ;(0) DRIVE NUMBER
2720  004125      000                        .BYTE   0               ;(1) OFFSET VALUE OR FMT22, ECI, AND HCI
2721  004126      000                        .BYTE   0               ;(2) COMMAND
2722  004127      000                        .BYTE   0               ;(3) PSEL AND A17 AND A16
2723  004130  177776                         .WORD   -2              ;(4) WORD COUNT (MUST BE NEG.)
2724  004132  050202                         .WORD   BUFFER          ;(6) BUFFER ADDRESS OR
2725                                                                  ;REGISTER TABLE POINTER
2726  004134      000                        .BYTE   0               ;(10) SECTOR ADDRESS OR
2727                                                                  ;FIRST REG. INDEX
2728  004135      000                        .BYTE   0               ;(11) TRACK ADDRESS OR
2729                                                                  ;LAST REG. INDEX
2730  004136  000000                         .WORD   0               ;(12) CYLINDER ADDRESS
2731  004140  004204                         .WORD   RP.REG          ;(14) ERROR TABLE POINTER
2732                                                                  ;POINTS TO THE FIRST OF TWENTY
2733                                                                  ;LOCATIONS OF WHERE THE DRIVER
2734                                                                  ;IS TO STORE THE RH11/RP04
2735                                                                  ;REGISTERS ON AN ERROR. IF LEFT
2736                                                                  ;ZERO REGISTERS ARE NOT SAVED.
2737  004142  000000                         .WORD   0               ;(16) STATUS/ERROR INDICATOR
2738                                                                  ;BIT15=1=>ERROR OCCURRED
2739                                                                  ;BIT07=1=>DONE
2740                                                                  ;BIT14-BIT09 AND BIT06-BIT03
2741                                                                  ;INDICATE TYPE OF ERROR
2742
2743  004144      000              DPB.C:    .BYTE   0               ;(0) DRIVE NUMBER
2744  004145      000                        .BYTE   0               ;(1) OFFSET VALUE OR FMT22, ECI, AND HCI
2745  004146      000                        .BYTE   0               ;(2) COMMAND
2746  004147      000                        .BYTE   0               ;(3) PSEL AND A17 AND A16
2747  004150  177776                         .WORD   -2              ;(4) WORD COUNT (MUST BE NEG.)
2748  004152  050202                         .WORD   BUFFER          ;(6) BUFFER ADDRESS OR
2749                                                                  ;REGISTER TABLE POINTER
2750  004154      000                        .BYTE   0               ;(10) SECTOR ADDRESS OR
2751                                                                  ;FIRST REG. INDEX
2752  004155      000                        .BYTE   0               ;(11) TRACK ADDRESS OR
2753                                                                  ;LAST REG. INDEX
2754  004156  000000                         .WORD   0               ;(12) CYLINDER ADDRESS
2755  004160  004204                         .WORD   RP.REG          ;(14) ERROR TABLE POINTER
2756                                                                  ;POINTS TO THE FIRST OF TWENTY
2757                                                                  ;LOCATIONS OF WHERE THE DRIVER
```

```
                                                        ;IS TO STORE THE RH11/RP04
                                                        ;REGISTERS ON AN ERROR. IF LEFT
                                                        ;ZERO REGISTERS ARE NOT SAVED.
004162  000000                  .WORD   0              ;(16) STATUS/ERROR INDICATOR
                                                        ;BIT15=1=>ERROR OCCURRED
                                                        ;BIT07=1=>DONE
                                                        ;BIT14-BIT09 AND BIT06-BIT03
                                                        ;INDICATE TYPE OF ERROR

004164      000         DTADPB: .BYTE   0              ;(0) DRIVE NUMBER
004165      000                 .BYTE   0              ;(1) OFFSET VALUE OR FMT22, ECT, AND MCI
004166      000                 .BYTE   0              ;(2) COMMAND
004167      000                 .BYTE   0              ;(3) PSEL AND A17 AND A16
004170  000000                  .WORD   0              ;(4) WORD COUNT (MUST BE NEG.)
004172  050202                  .WORD   BUFFER         ;(6) BUFFER ADDRESS OR
                                                        ;REGISTER TABLE POINTER
004174      000                 .BYTE   0              ;(10) SECTOR ADDRESS OR
                                                        ;FIRST REG. INDEX
004175      000                 .BYTE   0              ;(11) TRACK ADDRESS OR
                                                        ;LAST REG. INDEX
004176  000000                  .WORD   0              ;(12) CYLINDER ADDRESS
004200  004204                  .WORD   RP.REG         ;(14) ERROR TABLE POINTER
                                                        ;POINTS TO THE FIRST OF TWENTY
                                                        ;LOCATIONS OF WHERE THE DRIVER
                                                        ;IS TO STORE THE RH11/RP04
                                                        ;REGISTERS ON AN ERROR. IF LEFT
                                                        ;ZERO REGISTERS ARE NOT SAVED.
004202  000000                  .WORD   0              ;(16) STATUS/ERROR INDICATOR
                                                        ;BIT15=1=>ERROR OCCURRED
                                                        ;BIT07=1=>DONE
                                                        ;BIT14-BIT09 AND BIT06-BIT03
                                                        ;INDICATE TYPE OF ERROR


                        ;SAVE RH11/RP04 REGISTERS HERE ON ERROR

004204  000000          RP.REG: .WORD   0              ;RPCS1 (776700) CONTROL & STATUS #1
004206  000000                  .WORD   0              ;RPWC (776702) WORD COUNT
004210  000000                  .WORD   0              ;RPBA (776704) BUS ADDRESS
004212  000000                  .WORD   0              ;RPDA (776706) DESIRED SECTOR/TRACK
004214  000000                  .WORD   0              ;RPCS2 (776710) CONTROL & STATUS #2
004216  000000                  .WORD   0              ;RPDS1 (776712) DISK STATUS
004220  000000                  .WORD   0              ;RPER1 (776714) ERROR REG. #1
004222  000000                  .WORD   0              ;RPAS (776716) ATTENTION SUMMARY
004224  000000                  .WORD   0              ;RPLA (776720) LOOK AHEAD
004226  000000                  .WORD   0              ;RPDB (776722) DATA BUFFER
004230  000000                  .WORD   0              ;RPMR (776724) MAINTAINABILITY
004232  000000                  .WORD   0              ;RPDT (776726) DRIVE TYPE
004234  000000                  .WORD   0              ;RPSN (776730) SERIAL NUMBER
004236  000000                  .WORD   0              ;RPOF (776732) OFFSET
004240  000000                  .WORD   0              ;RPCA (776734) DESIRED CYLINDER
004242  000000                  .WORD   0              ;RPCC (776736) CURRENT CYLINDER
004244  000000                  .WORD   0              ;RPER2 (776740) ERROR REG #2
004246  000000                  .WORD   0              ;RPER3 (776742) ERROR REG #3
004250  000000                  .WORD   0              ;RPEC1 (776744) ECC POSITION
```

```
 2814  004252  000000                          .WORD   0               ;RPEC2 (776746) ECC PATTERN
 2815
 2816                                  ;STATUS/ERROR MESSAGE POINTER TABLE
 2817  004254  045212                  STATBL: .WORD   MSGB14          ;OFFLINE OR UNSAFE DRIVE REQUESTED
 2818  004256  045254                          .WORD   MSGB13          ;UNLOAD DRIVE REQUESTED
 2819  004260  045305                          .WORD   MSGB12          ;PERSISTENT UNSAFE
 2820  004262  045327                          .WORD   MSGB11          ;PARITY ERROR OCCURRED
 2821  004264  045355                          .WORD   MSGB10          ;FATAL PARITY ERROR
 2822  004266  045400                          .WORD   MSGB09          ;SOFTWARE TIMEOUT ON THIS DRIVE
 2823  004270  045437                          .WORD   MSGB08          ;SOFTWARE TIMEOUT ON ANOTHER DRIVE
 2824  004272  045501                          .WORD   MSGB06          ;ERROR OCCURRED DURING I/O OPERATION
 2825  004274  045545                          .WORD   MSGB05          ;ERROR OCCURRED DURING NON-I/O OPERATION
 2826  004276  045615                          .WORD   MSGB04          ;UNSAFE OCCURRED
 2827  004300  045635                          .WORD   MSGB03          ;AUTOMATIC RECALIBRATE SEQUENCE OCCURRED
 2828  004302  045705                          .WORD   MSGB02          ;DRIVE HAS NOT RESPONDED TO PORT REQUEST
 2829  004304  045755                          .WORD   MSGB01          ;DRIVE HAS BECOME NONEXISTENT
```

```
                                .SBTTL  ERROR POINTER TABLE

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

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


004306                          $ERRTB:
                                ;*EM AND DH ARE ASCIZ MESSAGES, DT IS A STRING OF WORDS THAT POINT TO THE
                                ;*DATA TO BE TYPED AND DF IS A STRING OF DATA THAT TELL HOW THE DT WORDS
                                ;*ARE TO BE TYPED.  IF ANY OF THE POINTERS ARE NOT NEEDED FOR A PARTICULAR
                                ;*ERROR IT IS REPLACED WITH A ZERO.
                                ;*EACH OF THE ITEMS BELOW REFER TO THE ERROR NUMBER AND INDICATE
                                ;*THE INFORMATION THAT WILL BE TYPED WHEN THE ERROR OCCURS.
                                ;*UNLESS STATED OTHER ALL NUMBERS ARE OCTAL

                                ;*  ERROR ITEM 1
                                ;*    RH11 INTERRUPT OCCURED (RPAS = 0)
                                ;*    ERR PC  RPAS
                                ;*    $ERRPC  $REG3

004306  044333                          EM1
004310  046013                          DH1
004312  047374                          DT1
004314  050026                          DF1

                                ;*  ERROR ITEM 2
                                ;*    UNEXPECTED ATTENTION OCCURRED
                                ;*    ERR PC  DRIVE   RPAS    RPDS1   RPER1     RPER2     RPER3
                                ;*    $ERRPC  $REG1   $REG3   RPERRS  RPERRS+2  RPERRS+4  RPERRS+6

004316  044376                          EM2
004320  046030                          DH2
004322  047400                          DT2
004324  050032                          DF2

                                ;*  ERROR ITEM 3
                                ;*    MASSBUS PARITY ERROR (MCPE=1)
                                ;*    TEST    ERR PC  ADDRESS DATA
                                ;*    $TMP0   $ERRPC  RD.ADR  RD.WRD

004326  044434                          EM3
004330  046116                          DH3
004332  047416                          DT3
004334  050036                          DF3

                                ;*  ERROR ITEM 4
                                ;*    MASSBUS PARITY ERROR (PAR=1)
                                ;*    TEST    ERR PC  ADDRESS GDDATA  BDDATA
```

```
2886                                    ;*    $TMPO   $ERRPC  WRT.ADR WRT.WD  RD.WRD
2887
2888  004336  044471                        EM4
2889  004340  046153                        DH4
2890  004342  047426                        DT4
2891  004344  050042                        DF4
2892
2893                                    ;*  ERROR ITEM 5
2894                                    ;*    ADDRESS PLUG CHANGE BIT SET
2895                                    ;*    ERR PC  DRIVE   RPAS    RPDS1   RPER1     RPER2    RPER3
2896                                    ;*    $ERRPC  $REG1   $REG3   RPERRS  RPERRS+2  RPERRS+4 RPERRS+6
2897
2898  004346  044525                        EM5
2899  004350  046030                        DH2
2900  004352  047400                        DT2
2901  004354  050032                        DF2
2902
2903                                    ;*  ERROR ITEM 6 -- NOT USED
2904
2905  004356  000000                        0
2906  004360  000000                        0
2907  004362  000000                        0
2908  004364  000000                        0
2909
2910                                    ;*  ERROR ITEM 7 -- NOT USED
2911
2912  004366  000000                        0
2913  004370  000000                        0
2914  004372  000000                        0
2915  004374  000000                        0
2916
2917                                    ;*  ERROR ITEM 10
2918                                    ;*    RH11/RP04/5/6 FAILED TO RESPOND TO ADDRESSING
2919                                    ;*    RPCS1   ERR PC
2920                                    ;*    RH.ADR  $ERRPC
2921
2922  004376  044561                        EM10
2923  004400  046222                        DH10
2924  004402  047460                        DT10
2925  004404  050046                        DF10
2926
2927                                    ;*  ERROR ITEM 11
2928                                    ;*    DRIVE SELECTED IS NOT ONLINE
2929                                    ;*    DRIVE   ERR PC
2930                                    ;*    $REG2   $ERRPC
2931
2932  004406  044637                        EM11
2933  004410  046241                        DH11
2934  004412  047464                        DT11
2935  004414  050052                        DF11
2936
2937                                    ;*  ERROR ITEM 12
2938                                    ;*    IMPROPER HEADER DATA
2939                                    ;*    TEST    ERR PC  TST PC  DRIVE   CYLNDR  TRACK   SECTOR
2940                                    ;*    $TMPO   $ERRPC  $REG0   CHKDRV  CYL.DS  TRK.DS  SEC.DS
2941                                    ;*        GDCYL   GDTRK   GDSCTR  BDCYL   BDTRK   BDSCTR
```

```
                                ;*        CYL.DS  TRK.DS  SEC.DS  CYL.RD  TRK.RD  SEC.RD
                                ;*    CYLNDR, TRACK, AND SECTOR ARE DECIMAL

004416  044674                      EM12
004420  046260                      DH12
004422  047470                      DT12
004424  050056                      DF12

                                ;*  ERROR ITEM 13
                                ;*    DATA COMPARE FAILURE
                                ;*    TEST    ERR PC  TST PC  DRIVE   CYLNDR  TRACK   SECTOR
                                ;*    $TMP0   $ERRPC  $REG0   CHKDRV  CYL.DS  TRK.DS  SEC.DS
                                ;*        GDDAT   BDDAT   WRDCNT  GDADR   BDADR
                                ;*        $GDDAT  $BDDAT  $REG4   $GDADR  $BDADR
                                ;*      CYLNDR, TRACK, SECTOR, AND WRDCNT  ARE DECIMAL

004426  044721                      EM13
004430  046260                      DH12
004432  047522                      DT13
004434  050066                      DF13

                                ;*  ERROR ITEM 14 -- FOLLOWS #13
                                ;*        $GDDAT  $BDDAT  $REG4   $GDADR  $BDADR

004436  000000                      0
004440  000000                      0
004442  047540                      DT13A
004444  050076                      DF14

                                ;*  ERROR ITEM 15
                                ;*    DATA COMPARE FAILURE
                                ;*    TEST    ERR PC  TST PC  DRIVE   CYLNDR  TRACK   SECTOR
                                ;*    $TMP0   $ERRPC  $REG0   CHKDRV  CYL.DS  TRK.DS  SEC.DS
                                ;*        GDDAT   BDDAT   WRDCNT  GDADR   BDADR
                                ;*        $GDDAT  $BDDAT  $REG4   $GDADR  $BDADR
                                ;*        CYLNDR, TRACK, SECTOR, AND WRDCNT ARE DECIMAL

004446  044721                      EM13
004450  046260                      DH12
004452  047522                      DT13
004454  050066                      DF13

                                ;*  ERROR ITEM 16 -- FOLLOWS #15
                                ;*        $GDDAT  $BDDAT  $REG4   $GDADR  $BDADR

004456  000000                      0
004460  000000                      0
004462  047540                      DT13A
004464  050076                      DF14

                                ;*  ERROR ITEM 17
                                ;*    DISK ERROR IN TIMING TEST
                                ;*    TEST    ERR PC  DRIVE   RPCS1   RPDS1     RPER1     RPER2     RPER3
                                ;*    $TMP0   $ERRPC  CHKDRV  RP.REG  RP.REG+12 RP.REG+14 RP.REG+40 RP.REG+42

004466  044746                      EM17
```

```
 2998  004470  046474                        DH17
 2999  004472  047552                        DT17
 3000  004474  050102                        DF17
 3001
 3002                                  ;*  ERROR ITEM 20
 3003                                  ;*    CLOCK (KW11-P) OVERFLOW IN TIMING TEST
 3004                                  ;*    TEST    ERR PC  DRIVE   RPCS1   RPDS1     RPER1     RPER2     RPER3
 3005                                  ;*    $TMPO   $ERRPC  CHKDRV  RP.REG  RP.REG+12 RP.REG+14 RP.REG+40 RP.REG+42
 3006
 3007  004476  045000                        EM20
 3008  004500  046474                        DH17
 3009  004502  047552                        DT17
 3010  004504  050102                        DF17
 3011
 3012                                  ;*  ERROR ITEM 21
 3013                                  ;*    DATA COMPARE FAILURE
 3014                                  ;*    TEST      ERR PC  TST PC  DRIVE   CYLNDR  TRACK
 3015                                  ;*    $TMPO     $ERRPC  $REGO   CHKDRV  CYL.DS  TRK.DS
 3016                                  ;*        GDDAT   BDDAT   WRDCNT  SECTOR
 3017                                  ;*        $REG1   $BDDAT  $REG4   $REG1
 3018                                  ;*        CYLINDR, TRACK, WRDCNT, AND SECTOR ARE DECIMAL
 3019
 3020  004506  044721                        EM13
 3021  004510  046572                        DH21
 3022  004512  047572                        DT21
 3023  004514  050106                        DF21
 3024
 3025                                  ;*  ERROR ITEM 22--FOLLOWS #21
 3026                                  ;*        $REG1   $BDDAT   $REG4   $REG1
 3027
 3028  004516  000000                          0
 3029  004520  000000                          0
 3030  004522  047606                        DT21A
 3031  004524  050116                        DF22
 3032
 3033                                  ;*  ERROR ITEM 23
 3034                                  ;*      DISK ERROR DURING SEEK
 3035                                  ;*      TEST    ERR PC  DRIVE   CYLNDR  RPCS1   RPCS2     RPDS1
 3036                                  ;*      $TMPO   $ERRPC  CHKDRV  CYL.DS  RP.REG RP.REG+10 RP.REG+12
 3037                                  ;*          RPER1     RPER2     RPER3     RPCA      RPCC
 3038                                  ;*          RP.REG+14 RP.REG+40 RP.REG+42 RP.REG+34 RP.REG+36
 3039
 3040  004526  045047                        EM23
 3041  004530  046707                        DH23
 3042  004532  047616                        DT23
 3043  004534  050122                        DF23
 3044
 3045                                  ;*  ERROR ITEM 24
 3046                                  ;*      SEEK NOT COMPLETE WITHIN 120 MS
 3047                                  ;*      TEST    ERR PC  DRIVE   CYLNDR  RPCS1   RPCS2     RPDS1
 3048                                  ;*      $TMPO   $ERRPC  CHKDRV  CYL.DS  RP.REG RP.REG+10 RP.REG+12
 3049                                  ;*          RPER1     RPER2     RPER3     RPCA      RPCC
 3050                                  ;*          RP.REG+14 RP.REG+40 RP.REG+42 RP.REG+34 RP.REG+36
 3051
 3052  004536  045076                        EM24
 3053  004540  046707                        DH23
```

```
3054  004542  047616                        DT23
3055  004544  050122                        DF23
3056
3057
3058
3059                                ;;*****************************************************************
3060                                ;;*****************************************************************
3061                                ;*  ERROR ITEMS 23-40 NOT USED
3062                                ;*  ERROR ITEMS 41-46 WILL HAVE AN EM THAT
3063                                ;*  VARIES DEPENDING ON THE ERROR, IT WILL BE IN THE FORM:
3064                                ;*  RH11/RP04/5/6 ERROR (MESSAGE)
3065                                ;*  WHERE MESSAGE WILL BE ONE OR MORE OF THE FOLLOWING:
3066                                ;*  1)  OFFLINE OR UNSAFE DRIVE REQUESTED
3067                                ;*  2)  UNLOADED DRIVE REQUESTED
3068                                ;*  3)  PERSISTENT UNSAFE
3069                                ;*  4)  PARITY ERROR OCCURRED
3070                                ;*  5)  FATAL PARITY ERROR
3071                                ;*  6)  SOFTWARE TIMEOUT ON THIS DRIVE
3072                                ;*  7)  SOFTWARE TIMEOUT ON ANOTHER DRIVE
3073                                ;*  8)  ERROR OCCURRED DURING I/O OPERATION
3074                                ;*  9)  ERROR OCCURRED DURING NON-I/O OPERATION
3075                                ;*  10) UNSAFE OCCURRED
3076                                ;*  11) AUTOMATIC RECALIBRATE SEQUENCE OCCURRED
3077
3078  004546                        ITEM41:
3079
3080                                ;*  ERROR ITEM 41
3081                                ;*    RH11/RP04/5/6 ERROR (MESSAGE)
3082                                ;*    TEST    ERR PC  TST PC  DRIVE
3083                                ;*    $TMP0   $ERRPC  $REG0   CHKDRV
3084
3085  004546  045136                    EM41
3086  004550  047042                    DH41
3087  004552  047646                    DT41
3088  004554  050132                    DF41
3089
3090                                ;*  ERROR ITEM 42
3091                                ;*    RH11/RP04/5/6 ERROR (MESSAGE)
3092                                ;*    TEST    ERR PC  TST PC  DRIVE   RPCS1   RPCS2   RPDS1
3093                                ;*    $TMP0   $ERRPC  $REG0   CHKDRV  RP.REG  RP.REG+10 RP.REG+12
3094
3095  004556  045136                    EM41
3096  004560  047100                    DH42
3097  004562  047656                    DT42
3098  004564  050136                    DF42
3099
3100                                ;*  ERROR ITEM 43
3101                                ;*    RH11/RP04/5/6 ERROR (MESSAGE)
3102                                ;*    TEST    ERR PC  TST PC  DRIVE   RPCS1   RPCS2   RPDS1
3103                                ;*    $TMP0   $ERRPC  $REG0   CHKDRV  RP.REG  RP.REG+10 RP.REG+12
3104                                ;*       RPER1     RPER2     RPER3
3105                                ;*       RP.REG+14 RP.REG+40 RP.REG+42
3106
3107  004566  045136                    EM41
3108  004570  047100                    DH42
3109  004572  047674                    DT43
```

```
3110  004574  050142                          DF43
3111
3112                                  ;*  ERROR ITEM 44
3113                                  ;*    RH11/RP04/5/6 ERROR (MESSAGE)
3114                                  ;*    TEST    ERR PC  TST PC  DRIVE   CYLNDR  TRACK     SECTOR
3115                                  ;*    $TMP0   $ERRPC  $REG0   CHKDRV  CYL.DS  TRK.DS    SEC.DS
3116                                  ;*       RPCS1     RPCS2     RPDS1     RPCC    RPCA     RPDA
3117                                  ;*       RP.REG    RP.REG+10 RP.REG+12 RP.REG+36 RP.REG+34 RP.REG+06
3118                                  ;*      RPER1      RPER2      RPER3
3119                                  ;*      RP.REG+14  RP.REG+40  RP.REG+42
3120                                  ;*      CYLNDR,TRACK, AND SECTOR ARE DECIMAL
3121
3122  004576  045136                          EM41
3123  004600  046260                          DH12
3124  004602  047720                          DT44
3125  004604  050152                          DF44
3126
3127                                  ;*  ERROR ITEM 45
3128                                  ;*    RH11/RP04/5/6 ERROR (MESSAGE)
3129                                  ;*    TEST    ERR PC  TST PC  DRIVE   CYLNDR  TRACK     SECTOR
3130                                  ;*    $TMP0   $ERRPC  $REG0   CHKDRV  CYL.DS  TRK.DS    SEC.DS
3131                                  ;*       RPCS1     RPCS2     RPDS1     RPCC      RPCA      RPDA
3132                                  ;*       RP.REG    RP.REG+10 RP.REG+12 RP.REG+36 RP.REG+34 RP.REG+06
3133                                  ;*      RPER1      RPER2      RPER3      RPWC      RPBA     RPDB
3134                                  ;*      RP.REG+14  RP.REG+40  RP.REG+42  RP.REG+2  RP.REG+4 RP.REG+22
3135                                  ;*      CYLNDR,TRACK, AND SECTOR ARE DECIMAL
3136
3137  004606  045136                          EM41
3138  004610  046260                          DH12
3139  004612  047760                          DT45
3140  004614  050166                          DF45
3141
3142                                  ;*  ERROR ITEM 46
3143                                  ;*    FATAL WRITE CHECK ERROR (MESSAGE)
3144                                  ;*    TEST    ERR PC  TST PC  DRIVE   CYLNDR  TRACK     SECTOR
3145                                  ;*    $TMP0   $ERRPC  $REG0   CHKDRV  CYL.DS  TRK.DS    SEC.DS
3146                                  ;*       RPCS1     RPCS2     RPDS1     RPCC      RPCA      RPDA
3147                                  ;*       RP.REG    RP.REG+10 RP.REG+12 RP.REG+36 RP.REG+34 RP.REG+06
3148                                  ;*      RPER1      RPER2      RPER3      RPWC      RPBA     RPDB
3149                                  ;*      RP.REG+14  RP.REG+40  RP.REG+42  RP.REG+2  RP.REG+4 RP.REG+22
3150                                  ;*      CYLNDR,TRACK, AND SECTOR ARE DECIMAL
3151
3152  004616  045162                          EM46
3153  004620  046260                          DH12
3154  004622  047760                          DT45
3155  004624  050166                          DF45
3156
```

```
3157
3158                                    .SBTTL  START OF PROGRAM
3159
3160
3161  004626  005037  001226            START6: CLR     @#BUSADR        ;CLR BUSADR FLAG
3162  004632  012737  000001  035446            MOV     #1,     TSTPGM  ;ALLOW PROGRAMMABLE DRIVES
3163  004640  000437                            BR      STRT2A          ;
3164  004642  005037  001226            START5: CLR     @#BUSADR        ;CLEAR BUSADR FLAG
3165  004646  012737  000001  035446            MOV     #1,     TSTPGM  ;ALLOW PROGRAMMBLE DRIVES
3166  004654  000413                            BR      STRT1A
3167  004656  012737  177777  001226  START3: MOV     #-1,@#BUSADR    ;GET BUSADR FLAG
3168  004664  012737  000001  035446            MOV     #1,     TSTPGM  ;ALLOW PROGRAMMABLE DRIVES
3169  004672  000404                            BR      STRT1A
3170  004674  005037  001226            START1: CLR     @#BUSADR        ;CLR BUSADR FLAG
3171  004700  005037  035446                    CLR     TSTPGM          ;DISABLE PROGRAMMABLE DRIVES
3172  004704  005037  001224            STRT1A: CLR     @#CNTRLC        ;NO CONTROL "C"
3173  004710  000416                            BR      START
3174  004712  012737  177777  001226  START4: MOV     #-1,@#BUSADR    ;SET BUSADR FLAG
3175  004720  012737  000001  035446            MOV     #1,     TSTPGM  ;ALLOW PROGRAMMABLE DRIVES
3176  004726  000404                            BR      STRT2A
3177  004730  005037  001226            START2: CLR     @#BUSADR        ;CLR BUSADR FLAG
3178  004734  005037  035446                    CLR     TSTPGM          ;DISABLE PROGRAMMABLE DRIVES
3179  004740  012737  177777  001224  STRT2A: MOV     #-1,@#CNTRLC    ;SET CONTROL "C" FLAG
3180  004746  000005                    START:  RESET
3181                                    .SBTTL  INITIALIZE THE COMMON TAGS
3182                                    ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
3183  004750  012706  001100                    MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
3184  004754  005026                            CLR     (R6)+           ;;CLEAR MEMORY LOCATION
3185  004756  022706  001140                    CMP     #SWR,R6 ;;DONE?
3186  004762  001374                            BNE     .-6             ;;LOOP BACK IF NO
3187  004764  012706  001100                    MOV     #STACK,SP       ;;SETUP THE STACK POINTER
3188                                    ;;INITIALIZE A FEW VECTORS
3189  004770  012737  022724  000020            MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
3190  004776  012737  000340  000022            MOV     #340,@#IOTVEC+2 ;;LEVEL 7
3191  005004  012737  017752  000030            MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
3192  005012  012737  000340  000032            MOV     #340,@#EMTVEC+2 ;;LEVEL 7
3193  005020  012737  023252  000034            MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
3194  005026  012737  000340  000036            MOV     #340,@#TRAPVEC+2;LEVEL 7
3195  005034  012737  176543  023724            MOV     #176543,$HINUM  ;;PRIME THE RANDOM NUMBER GENERATOR
3196  005042  012737  123456  023726            MOV     #123456,$LONUM  ;;BOTH HIGH AND LOW WORDS
3197  005050  005037  001204                    CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
3198  005054  005037  001206                    CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
3199  005060  112737  000001  001115            MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
3200  005066  012737  005066  001106            MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
3201  005074  012737  005074  001110            MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
3202                                    ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
3203                                    ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
3204  005102  013746  000004                    MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
3205  005106  012737  005142  000004            MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
3206  005114  012737  177570  001140            MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
3207  005122  012737  177570  001142            MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
3208  005130  022777  177777  174002            CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
3209  005136  001012                            BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
3210                                                                    ;;AND  THE HARDWARE SWR IS NOT = -1
3211  005140  000403                            BR      65$             ;;BRANCH IF NO TIMEOUT
3212  005142  012716  005150            64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
```

```
3213  005146  000002                        RTI
3214  005150  012737  000176  001140  65$:  MOV   #SWREG,SWR       ;;POINT TO SOFTWARE SWR
3215  005156  012737  000174  001142        MOV   #DISPREG,DISPLAY
3216  005164  012637  000004        66$:    MOV   (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
3217
3218  005170  012700  001160                MOV   #$REGAD,R0      ;FIRST ADDRESS
3219  005174  005020                  1$:   CLR   (R0)+           ;CLEAR VARIABLE STORAGE
3220  005176  022700  001210                CMP   #$BELL,R0       ;DONE?
3221  005202  001374                        BNE   1$              ;NO--BRANCH
3222  005204  013737  001414  001150        MOV   @#TPS,@#$TPS    ;SETUP THE STATUS AND BUFFER REG'S
3223  005212  013737  001416  001152        MOV   @#TPB,@#$TPB    ;FOR THE TYPE ROUTINE
3224  005220  005227  177777                INC   #-1             ;FIRST START ?
3225  005224  001043                        BNE   3$              ;BR IF NOT
3226  005226  023737  000042  000046        CMP   @#42,@#46       ;ACT11 AUTOMATIC MODE?
3227  005234  001004                        BNE   4$              ;BRANCH IF NO
3228  005236  012737  000001  035446        MOV   #1,     TSTPGM  ;ENABLE PROGRAMMABLE DRIVES
3229  005244  000402                        BR    5$
3230  005246  104401  050202          4$:   TYPE  ,TITLE          ;TYPE THE PROGRAM'S TITLE
3231  005252  005737  035446          5$:   TST   TSTPGM          ;CAN WE USE PROGRAMMABLE DRIVES?
3232  005256  001402                        BEQ   6$              ;BRANCH IF NO
3233  005260  104401  043242                TYPE  ,USE            ;TYPE MSG
3234  005264                           6$:
3235  005264  005737  000042                TST   42              ;AUTO ACCEPT OR CHAIN MODE ?
3236  005270  001006                        BNE   2$              ;BR IF EITHER
3237  005272  122737  000011  000041        CMPB  #11,41          ;LOADED FROM AN RP04/5/6 ?
3238  005300  001002                        BNE   2$              ;BR IF NOT
3239  005302  104401  050266                TYPE  ,LOADRV         ;INSTRUCT THE OPERATOR TO REMOVE THE PACK
3240                                                              ;ON DRIVE 0 IF DRIVE 0 IS TO BE TESTED
3241  005306  105737  000041          2$:   TSTB  @#41            ;LOADED FROM PAPER TAPE ?
3242  005312  001410                        BEQ   3$              ;BR IF NOT
3243  005314  004737  050562                JSR   PC,$SIZE        ;SIZE THE MEMORY
3244  005320  023727  050656  100000        CMP   $LSTAD,#100000  ;16K OR MORE ON THE SYSTEM ?
3245  005326  103002                        BHIS  3$              ;BR IF YES
3246  005330  104401  050464                TYPE  ,NOLOAD         ;INFORM THE OPEATOR THAT THE 'XXDP' LOADER
3247                                                              ;WILL BE OVERWRITTEN
3248  005334  004737  021462          3$:   JSR   PC,$TKINT       ;TURN ON THE TTY KEYBOARD INTERRUPT
3249                                   .SBTTL GET VALUE FOR SOFTWARE SWITCH REGISTER
3250  005340  005737  000042                TST   @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
3251  005344  001006                        BNE   67$             ;;BRANCH IF YES
3252  005346  023727  001140  000176        CMP   SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
3253  005354  001005                        BNE   68$             ;;BRANCH IF NO
3254  005356  104406                        GTSWR                 ;;GET SOFT-SWR SETTINGS
3255  005360  000403                        BR    68$
3256  005362  112737  000001  001134  67$:  MOVB  #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
3257  005370                           68$:
3258  005370  005227  177777                INC   #-1             ;SEE IF FIRST START
3259  005374  001002                        BNE   SRTINT          ;BR IF NOT
3260  005376  004737  050660                JSR   PC,GETADR       ;GET OR CHECK THE RH11 ADDRESS
3261  005402  104401  001215          SRTINT: TYPE  ,$CRLF        ;CR-LF
3262  005406  004737  024150                JSR   PC,@#LP.AVL     ;CHECK FOR A LINE PRINTER
3263  005412  005037  177776                CLR   @#PS            ;ENSURE THE PRIORITY = 0
3264  005416  012737  000001  001104        MOV   #1,$ICNT        ;SET ITERATION COUNT TO 1
3265  005424  004737  031542                JSR   PC,@#GETSWR     ;GO CHECK FOR CONTROL SWITCHES
3266  005430  004737  024212                JSR   PC,@#ST.CLK     ;INITIALIZE THE CLOCK
3267  005434  004737  034650          SETVEC: JSR   PC,RPINIT     ;CHECK THE DRIVE STATUS
3268  005440  012737  177777  034572        MOV   #-1,SAVEFG      ;SET THE SAVE REGISTERS FLAG
```

```
3269  005446  062727  177777  000000          ADD     #-1,#0          ;FIRST START ?
3270  005454  103003                          BCC     11$             ;BR IF YES
3271  005456  005737  001224                  TST     CNTRLC          ;CONTROL 'C' SWITCH SET ?
3272  005462  001111                          BNE     SRTDRV          ;CONTINUE IF YES
3273  005464  012737  000340  177776  11$:    MOV     #PR7,PS         ;SET PRIORITY TO 7
3274  005472  005004                          CLR     R4              ;DRIVE TABLE POINTER
3275  005474  104401  001215                  TYPE    ,$CRLF          ;CR-LF
3276  005500  104401  043370                  TYPE    ,SYSTAT         ;TYPE STATUS HEADING
3277  005504                          1$:
3278  005504  010446                          MOV     R4,-(SP)        ;;SAVE R4 FOR TYPEOUT
3279                                                                  ;;TYPE DRIVE NUMBER
3280  005506  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
3281  005510     002                          .BYTE   2               ;;TYPE 2 DIGIT(S)
3282  005511     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
3283  005512  104401  044330                  TYPE    ,MSG.SP         ;SPACES
3284  005516  104401  044330                  TYPE    ,MSG.SP         ;SPACES
3285  005522  105764  034504                  TSTB    DRVSTA(R4)      ;CHECK DRIVE'S STATUS
3286  005526  100425                          BMI     4$              ;BR IF UNSAFE
3287  005530  001027                          BNE     5$              ;BR IF ONLINE
3288  005532  105764  034514                  TSTB    DRVTYP(R4)      ;SEE IF OFFLINE OR NONEXISTENT
3289  005536  001404                          BEQ     2$              ;BR IF NONEXISTENT
3290  005540  100006                          BPL     3$              ;BR IF OFFLINE
3291  005542  104401  043464                  TYPE    ,NOTRP          ;DRIVE NOT AN RP04/5/6
3292  005546  000447                          BR      9$              ;CHECK NEXT DRIVE
3293  005550  104401  043437          2$:     TYPE    ,NOTPRS         ;DRIVE NOT PRESENT
3294  005554  000444                          BR      9$              ;CHECK NEXT DRIVE
3295  005556  132764  000010  034514  3$:     BITB    #BIT03, DRVTYP(R4)      ;DRIVE PROGRAMMABLE?
3296  005564  001403                          BEQ     12$             ;BRANCH IF NO
3297  005566  104401  043317                  TYPE    ,NOUSE          ;PRINT MSG
3298  005572  000410                          BR      6$              ;PRINT DRIVE TYPE
3299  005574  104401  043416          12$:    TYPE    ,UNTOFF         ;DRIVE OFFLINE
3300  005600  000405                          BR      6$              ;PRINT DRIVE TYPE
3301  005602  104401  043454          4$:     TYPE    ,NOTSAF         ;DRIVE UNSAFE
3302  005606  000402                          BR      6$              ;PRINT DRIVE TYPE
3303  005610  104401  043427          5$:     TYPE    ,UNTON          ;DRIVE ONLINE
3304  005614  104401  044330          6$:     TYPE    ,MSG.SP         ;SPACES
3305  005620  012737  043502  005664          MOV     #RP04B,8$       ;ADDRESS OF RP04 MESSAGE
3306  005626  132764  000001  034514          BITB    #BIT00,DRVTYP(R4)  ;RP04 ?
3307  005634  001012                          BNE     7$              ;BR IF YES
3308  005636  012737  043507  005664          MOV     #RP05,8$        ;ADDRESS OF RP05 MESSAGE
3309  005644  132764  000002  034514          BITB    #BIT01,DRVTYP(R4)  ;RP05 ?
3310  005652  001003                          BNE     7$              ;BR IF YES
3311  005654  012737  043514  005664          MOV     #RP06,8$        ;ADDRESS OF RP06 MESSAGE
3312  005662  104401                  7$:     TYPE                    ;TYPE THE DRIVE TYPE MESSAGE
3313  005664  000000                  8$:     .WORD   0               ;MESSAGE ADDRESS HERE
3314  005666  104401  001215          9$:     TYPE    ,$CRLF          ;CR-LF
3315  005672  005204                          INC     R4              ;INCREMENT DRIVE NUMBER/TABLE POINTER
3316  005674  020427  000010                  CMP     R4,#8.          ;FINISHED ?
3317  005700  001301                          BNE     1$              ;BR IF NOT
3318  005702  005037  177776                  CLR     PS              ;SET PRIORITY BACK TO '0'
3319  005706  005737  001224          SRTDRV: TST     @#CNTRLC        ;CONTROL "C" START/RESTART?
3320  005712  001417                          BEQ     1$              ;NO--BRANCH
3321  005714  013746  001222                  MOV     SAVCSW,-(SP)    ;GET THE PREVIOUS 'C.SWR' CONTENTS
3322  005720  063716  001220                  ADD     C.SWR,(SP)      ;SET UP TO SEE IF 'BIT00' IS DIFFERENT
3323  005724  032726  000001                  BIT     #BIT00,(SP)+    ;IS 'BIT00' DIFFERENT ?
3324  005730  001405                          BEQ     9$              ;BR IF NOT
```

```
3325  005732  013737  001220  001222          MOV     C.SWR,SAVCSW    ;STORE PRESENT 'C.SWR' VALUE
3326  005740  004737  024470                  JSR     PC,LODFLT       ;RESET PARAMETERS TO THEIR DEFAULT VALUES
3327  005744  004737  031772          9$:     JSR     PC,@#GT.PRM     ;GET PARAMETERS
3328  005750  000420                          BR      4$
3329  005752  004737  024470          1$:     JSR     PC,LODFLT       ;SETUP DEFAULT PARAMETERS
3330  005756  005037  001232                  CLR     DRVSEL          ;NO DRIVES SELECTED
3331  005762  005000                          CLR     R0              ;DETERMINE THE DRIVES THAT
3332  005764  012701  000001                  MOV     #1,R1           ;ARE AVAILABLE FOR TESTING
3333  005770  105760  034504          2$:     TSTB    DRVSTA(R0)
3334  005774  003403                          BLE     3$
3335  005776  156037  034620  001232          BISB    ATABIT(R0),@#DRVSEL
3336  006004  005200                  3$:     INC     R0
3337  006006  106301                          ASLB    R1
3338  006010  001367                          BNE     2$
3339  006012  005037  034574          4$:     CLR     @#SEEKFG        ;CLEAR SEEK FLAG
3340  006016  032737  000400  001220          BIT     #SW08,@#C.SWR   ;DO SEEK BEFORE DATA TRANSFER?
3341  006024  001002                          BNE     5$              ;YES--BRANCH
3342  006026  005137  034574                  COM     @#SEEKFG        ;NO
3343  006032  122737  000011  000041  5$:     CMPB    #11,41  ;LOADED FROM AN RP04/5/6 ?
3344  006040  001003                          BNE     10$             ;BR IF NOT
3345  006042  042737  000001  001232          BIC     #BIT00,DRVSEL   ;CLEAR THE DRIVE 0 SELECTION BIT
3346  006050  104401  043521          10$:    TYPE    ,DRIVES         ;'DRIVES(S) TO BE TESTED'
3347  006054  005037  017664                  CLR     @#$ENDCT        ;DETERMINE PASSES TO MAKE AND
3348  006060  005000                          CLR     R0              ;THE DRIVES TO BE TESTED
3349  006062  013701  001232                  MOV     @#DRVSEL,R1     ;ANY DRIVES SELECTED?
3350  006066  001004                          BNE     6$              ;YES--BRANCH
3351  006070  104401  043552                  TYPE    ,NONE           ;'NONE'
3352
3353  006074  000137  017502                  JMP     @#$EOP          ;GO TO END OF PROGRAM
3354  006100  006201                  6$:     ASR     R1              ;REPORT THE DRIVES TO BE TESTED
3355  006102  103011                          BCC     7$
3356  006104  005237  017664                  INC     @#$ENDCT        ;GIVE THIS DRIVE A PASS
3357  006110  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
3358  006112  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
3359  006114     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
3360  006115     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
3361  006116  005701                          TST     R1              ;MORE DRIVES?
3362  006120  001404                          BEQ     8$              ;NO--BRANCH
3363  006122  104401  043557                  TYPE    ,COMMA          ;','
3364  006126  005200                  7$:     INC     R0              ;FORM DRIVE NUMBER
3365  006130  000763                          BR      6$
3366  006132  013737  017664  017656  8$:     MOV     @#$ENDCT,@#$EOPCT
3367  006140  005737  001244                  TST     @#CLKSTA        ;KW11-P AVAILABLE
3368  006144  003006                          BGT     RSTRT1          ;YES--BRANCH
3369  006146  032737  036000  001234          BIT     #36000,@#TSTNMS ;NO--ANY TIMING TESTS TO BE PERFORMED? *
3370  006154  001402                          BEQ     RSTRT1          ;NO--BRANCH
3371  006156  104401  043561                  TYPE    ,NOCLOK         ;'NO KW11-P CLOCK, TIMING TESTS WILL NOT BE PERFORMED'
3372  006162  005737  001232          RSTRT1: TST     DRVSEL          ;ANY DRIVES SELECTED ?
3373  006166  001002                          BNE     1$              ;BR IF YES
3374  006170  000137  004730                  JMP     START2          ;GET DRIVE SELECTION ENTRY
3375  006174  005037  001254          1$:     CLR     CHKDRV          ;INIT. THE CHECK DRIVE KEY
3376  006200  012737  000001  001256          MOV     #1,DRVMSK       ;START TO CHECK DESIRED DRIVES
3377  006206  033737  001256  001232  RSTRT2: BIT     DRVMSK,DRVSEL   ;IS THIS DRIVE SELECTED?
3378  006214  001010                          BNE     DRVOK           ;YES--GO CHECK IF DRIVE IS READY FOR TESTING
3379  006216  012706  001100          RESTART: MOV    #STACK,SP       ;SETUP THE STACK POINTER
3380  006222  005237  001254                  INC     CHKDRV          ;MOVE  TO NEXT DRIVE NUMBER
```

```
3381  006226  106337  001256            ASLB    DRVMSK          ;POSITION THE MASK
3382  006232  103753                    BCS     RSTRT1          ;BRANCH IF THE DRIVE NUMBER NEEDS INITIALIZED
3383  006234  000764                    BR      RSTRT2
3384
3385  006236  013702  001254    DRVOK:  MOV     CHKDRV,R2       ;PICKUP THE DRIVE NUMBER
3386  006242  105762  034504            TSTB    DRVSTA(R2)      ;IS DESIRED DRIVE ON-LINE?
3387  006246  003005                    BGT     1$              ;YES, BRANCH
3388  006250  104011                    ERROR   11              ;DRIVE SELECTED IS NOT ONLINE
3389  006252  043737  001256  001232    BIC     DRVMSK,DRVSEL   ;CLEAR DRIVE'S SELECTION BIT
3390  006260  000756                    BR      RESTART         ;RETURN
3391  006262  010237  004104    1$:     MOV     R2,@#DPB.A      ;SET THE DRIVE NUMBER INTO THE DPB'S
3392  006266  010237  004124            MOV     R2,@#DPB.B
3393  006272  010237  004144            MOV     R2,@#DPB.C
3394  006276  010237  004164            MOV     R2,@#DTADPB
3395  006302  004737  025104            JSR     PC,@#LDCMD      ;LOAD COMMAND INTO DPB.B AND DPB.C
3396  006306  012737  017502  001252    MOV     #$EOP,@#BYPASS  ;IF ERROR GO TO END OF PROGRAM
3397  006314  112737  000020  004105    MOVB    #FMT22/256.,DPB.A+1  ;ASSUME 16 BIT FORMAT
3398  006322  032737  000001  001220    BIT     #BIT00,C.SWR    ;16 BIT FORMAT REQUESTED ?
3399  006330  001402                    BEQ     2$              ;BR IF YES
3400  006332  105037  004105            CLRB    DPB.A+1         ;CLEAR THE 'FMT22' BIT
3401  006336  112737  000143  004106  2$:     MOVB    #SETFORM,DPB.A+2  ;SET THE FORMAT BIT PER DPB.A+1
3402  006344  004037  025150            JSR     R0,@#CALL.A     ;GO EXECUTE THE COMMAND
3403  006350  112737  000107  004106            MOVB    #RECAL,@#DPB.A+2        ;RECAL=COMMAND
3404  006356  004037  025150            JSR     R0,@#CALL.A     ;GO EXECUTE THE COMMAND
3405  006362  104401  043647            TYPE    ,TESTNG         ;'TESTING DRIVE '
3406  006366  010246                    MOV     R2,-(SP)        ;;SAVE R2 FOR TYPEOUT
3407  006370  104403                    TYPOS                   ;;GO TYPE--OCTAL ASCII
3408  006372     001                    .BYTE   1               ;;TYPE 1 DIGIT(S)
3409  006373     000                    .BYTE   0               ;;SUPPRESS LEADING ZEROS
3410  006374  104401  044330            TYPE    ,MSG.SP         ;TYPE SPACES
3411  006400  104401  043671            TYPE    ,SERIAL         ;'SERIAL NUMBER '
3412  006404  012700  000004            MOV     #4,R0           ;FOUR DIGITS TO TYPE
3413  006410  013701  004234            MOV     RP.REG+30,R1    ;SERIAL NUMBER
3414  006414  005002                3$: CLR     R2              ;ZERO
3415  006416  006101                    ROL     R1              ;PUT THE NEXT DIGIT
3416  006420  006102                    ROL     R2              ;INTO R2
3417  006422  006101                    ROL     R1
3418  006424  006102                    ROL     R2
3419  006426  006101                    ROL     R1
3420  006430  006102                    ROL     R2
3421  006432  006101                    ROL     R1
3422  006434  006102                    ROL     R2
3423  006436  062702  000060            ADD     #'0,R2          ;MAKE IT ASCII
3424  006442  010227                    MOV     R2,(PC)+        ;SAVE IT
3425  006444  000000                4$: .WORD   0
3426  006446  104401  006444            TYPE    ,4$             ;TYPE
3427  006452  005300                    DEC     R0              ;ALL DIGITS TYPED?
3428  006454  003357                    BGT     3$              ;NO -- BRANCH
3429  006456  104401  001215            TYPE    ,$CRLF
3430  006462  113737  001364  001115    MOVB    ERR.CT,$ERMAX   ;SETUP MAX ERROR COUNT
```

```
                                .SBTTL  #### TESTS ####

;/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:
;/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:
;/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:

;*IN THE DESCRIPTIONS OF THE BELOW TESTS THE VARIABLES USED
;*AND THEIR DEFAULT VALUES (UNLESS SPECIFIED OTHERWISE) ARE:
;*
;*MNEMONIC       VALUE           VARIABLE
;*--------       -----           --------
;*
;*R              1               ITERATIONS (REPEATS)
;*FC             0               FIRST CYLINDER ADDRESS
;*LC             410 OR 814      LAST CYLINDER ADDRESS
;*IC             1               INCREMENT VALUE
;*NC OF NC1      FC+IC           NEW OR MODIFIED CYLINDER
;*                               ADDRRESS
;*NC2            LC-IC           NEW OR MODIFIED CYLINDER
;*                               ADDRESS
;*
;*FT             0               FIRST TRACK ADDRESS
;*LT             18              LAST TRACK ADDRESS
;*IT             1               INCREMENT VALUE
;*NT             FT+IT           NEW OR MODIFIED TRACK ADDRESS
;*
;*FS             0               FIRST SECTOR ADDRESS
;*LS             21              LAST SECTOR ADDRESS

;/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*
;/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*
;/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*

                                .SBTTL  *** SEEK TESTS ***

;/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:
;/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:
;/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:

;*THE SEEK TESTS WILL BE EXCUTED USING IMPLIED SEEKS. THESE
;*IMPLIED SEEKS WILL BE PERFORMED BY "READ HEADER AND
;*DATA" COMMANDS TO TRACK "FT" SECTOR "FS" OF THE DESIRED CYLINDER.
;*THE WORD COUNT WILL BE SET SUCH THAT ONLY THE CYLINDER AND
;*TRACK/SECTOR WORDS OF THE HEADER ARE READ.

;/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*
;/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*
;/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*

                                ;;*****************************************************************
                                ;*TEST 0         RECAL/SEEK TEST
```

```
                                        ;*      THIS TEST WILL CAUSE THE DRIVE TO EXECUTE A RECALIBRATE
                                        ;*      COMMAND CYCLE AND THEN SEEK FORWARD TO CYLINDER "LC" AT
                                        ;*      THE COMPLETION OF BOTH COMMANDS STATUS INDICATIONS ARE
                                        ;*      CHECKED TO ENSURE NO ERRORS OCCURRED.
                                        ;;*****************************************************************
006470                                  TST0:
006470  000240                                  NOP
006472  033737  001424  001234                  BIT     @#BITS+<0*2>,TSTNMS ;DO THIS TEST?
006500  001002                                  BNE     64$             ;YES--BRANCH
006502  000137  006632                          JMP     TST1            ;NO--GO TO THE NEXT TEST
006506  012737  000000  001102  64$:           MOV     #0,@#$TSTNM     ;SET UP TEST NUMBER AND
                                                                        ;CLEAR THE ERROR FLAG ($ERFLG)
006514  004737  024726                          JSR     PC,LODPRM       ;LOAD THE PARMETERS FOR THE TEST
006520  012737  006614  001110                  MOV     #TEST0,@#$LPERR  ;SETUP THE LOOP ON ERROR ADDRESS
006526  013777  001102  172406                  MOV     $TSTNM,@DISPLAY  ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
006534  013737  001506  001204                  MOV     @#RPT,$TIMES    ;GET THE ITERATION COUNT
006542  112737  000031  001115                  MOVB    #25.,$ERMAX     ;MAX ERRORS ALLOWED FOR TEST
006550  112737  000107  004106                  MOVB    #RECAL,@#DPB.A+2        ;RECAL=COMMAND
006556  113737  001524  004134                  MOVB    @#FS,@#DPB.B+10 ;FS
006564  113737  001516  004135                  MOVB    @#FT,@#DPB.B+11 ;FT
006572  013737  001512  004136                  MOV     @#LC,@#DPB.B+12 ;LC
006600  012737  006630  001252                  MOV     #EXIT0,@#BYPASS ;GO TO EXIT0 ON ERROR
006606  012737  006614  001106                  MOV     #TEST0,$LPADR   ;SETUP LOOP ADDRESS
006614  012706  001100          TEST0:  MOV     #STACK,SP       ;SET UP STACK POINTER
006620  004037  025150                  JSR     R0,@#CALL.A     ;GO EXECUTE THE COMMAND
006624  004037  025262                  JSR     R0,@#CALL.B     ;GO EXECUTE THE COMMAND
006630  000004                  EXIT0:  SCOPE                   ;LOOP
                                        ;;*****************************************************************
                                        ;*TEST 1         SEEK/SEEK TEST
                                        ;*      THIS TEST WILL CAUSE THE DRIVE TO EXECUTE A FORWARD SEEK
                                        ;*      CYCLE TO "LC", "LT", "LS" FOLLOWED BY A REVERSE SEEK CYCLE TO
                                        ;*      "FC", "FT", "FS".  AT THE COMPLETION OF EACH SEEK, THE PROPER
                                        ;*      INDICATORS ARE EXAMINED TO ENSURE PROPER OPERATION.
                                        ;*      "LC" WILL DEFAULT TO 128 AND "FC", "FT", "LT", "FS", AND "LS"
                                        ;*      WILL DEFAULT TO 0
                                        ;;*****************************************************************
006632                                  TST1:
006632  000240                                  NOP
006634  033737  001426  001234                  BIT     @#BITS+<1*2>,TSTNMS ;DO THIS TEST?
006642  001002                                  BNE     64$             ;YES--BRANCH
006644  000137  007010                          JMP     TST2            ;NO--GO TO THE NEXT TEST
006650  012737  000001  001102  64$:           MOV     #1,@#$TSTNM     ;SET UP TEST NUMBER AND
                                                                        ;CLEAR THE ERROR FLAG ($ERFLG)
006656  004737  024726                          JSR     PC,LODPRM       ;LOAD THE PARMETERS FOR THE TEST
006662  012737  006772  001110                  MOV     #TEST1,@#$LPERR  ;SETUP THE LOOP ON ERROR ADDRESS
006670  013777  001102  172244                  MOV     $TSTNM,@DISPLAY  ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
006676  013737  001506  001204                  MOV     @#RPT,$TIMES    ;GET THE ITERATION COUNT
006704  112737  000031  001115                  MOVB    #25.,$ERMAX     ;MAX ERRORS ALLOWED FOR TEST
006712  113737  001524  004134                  MOVB    @#FS,@#DPB.B+10 ;FS
006720  113737  001526  004154                  MOVB    @#LS,@#DPB.C+10 ;LS
006726  113737  001516  004135                  MOVB    @#FT,@#DPB.B+11 ;FT
```

```
3543  006734  113737  001520  004155          MOVB    @#LT,@#DPB.C+11 ;LT
3544  006742  013737  001510  004136          MOV     @#FC,@#DPB.B+12 ;FC
3545  006750  013737  001512  004156          MOV     @#LC,@#DPB.C+12 ;LC
3546  006756  012737  007006  001252          MOV     #EXIT1,@#BYPASS ;GO TO EXIT1 ON ERROR
3547  006764  012737  006772  001106          MOV     #TEST1,$LPADR   ;SETUP LOOP ADDRESS
3548  006772  012706  001100          TEST1:  MOV     #STACK,SP       ;SET THE STACK POINTER
3549  006776  004037  025452                  JSR     R0,@#CALL.C     ;GO EXECUTE THE COMMAND
3550  007002  004037  025262                  JSR     R0,@#CALL.B     ;GO EXECUTE THE COMMAND
3551  007006  000004                  EXIT1:  SCOPE                   ;LOOP
3552
3553                                  ;;*****************************************************************
3554                                  ;*TEST 2         INCREMENT/SEEK TEST
3555
3556                                  ;*      THIS TEST WILL COMMAND FORWARD SEEK CYCLES TO ADVANCE THE
3557                                  ;*      CYLINDER ADDRESS FROM "FC" TO "LC" BY THE INCREMENT "IC".
3558                                  ;*      WHEN THE RESULTANT CYLINDER ADDRESS (NC) EXCEEDS
3559                                  ;*      "LC" REVERSE SEEK CYCLES ARE INITIATED; STARTING
3560                                  ;*      AT THE LAST LEGAL "NC" AND DECREMENTING BY "IC"
3561                                  ;*      UNTIL "NC" IS LESS THAN "FC". AT THE COMPLETION OF EACH
3562                                  ;*      SEEK COMMAND THE PROPER INDICATORS ARE EXAMINED TO
3563                                  ;*      ENSURE PROPER OPERATION.
3564
3565                                  ;;*****************************************************************
3566  007010                  TST2:
3567  007010  000240                          NOP
3568  007012  033737  001430  001234          BIT     @#BITS+<2*2>,TSTNMS ;DO THIS TEST?
3569  007020  001002                          BNE     64$             ;YES--BRANCH
3570  007022  000137  007226                  JMP     TST3            ;NO--GO TO THE NEXT TEST
3571  007026  012737  000002  001102  64$:    MOV     #2,@#$TSTNM     ;SET UP TEST NUMBER AND
3572                                                                  ;CLEAR THE ERROR FLAG ($ERFLG)
3573  007034  004737  024726                  JSR     PC,LODPRM       ;LOAD THE PARMETERS FOR THE TEST
3574  007040  012737  007120  001110          MOV     #TEST2,@#$LPERR  ;SETUP THE LOOP ON ERROR ADDRESS
3575  007046  013777  001102  172066          MOV     $TSTNM,@DISPLAY  ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
3576  007054  013737  001506  001204          MOV     @#RPT,$TIMES    ;GET THE ITERATION COUNT
3577  007062  112737  000031  001115          MOVB    #25.,$ERMAX     ;MAX ERRORS ALLOWED FOR TEST
3578  007070  012737  007076  001106          MOV     #1$,$LPADR      ;SETUP LOOP ADDRESS
3579  007076  113737  001524  004134  1$:     MOVB    @#FS,@#DPB.B+10 ;FS
3580  007104  113737  001516  004135          MOVB    @#FT,@#DPB.B+11 ;FT
3581  007112  012737  007224  001252          MOV     #EXIT2,@#BYPASS ;GO TO EXIT2 ON ERROR
3582  007120  013737  001510  004136  TEST2:  MOV     @#FC,@#DPB.B+12 ;FC
3583  007126  012737  007126  001110          MOV     #.,$LPERR       ;SETUP THE ERROR LOOP ADDRESS
3584  007134  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
3585  007140                  INCSK:
3586  007140  004037  025262                  JSR     R0,@#CALL.B     ;GO EXECUTE THE COMMAND
3587  007144  063737  001514  004136          ADD     @#IC,@#DPB.B+12 ;MOVE TO NEXT CYLINDER
3588  007152  023737  001512  004136          CMP     @#LC,@#DPB.B+12 ;OUT OF CYLINDERS?
3589  007160  002367                          BGE     INCSK           ;NO--BRANCH
3590  007162  013737  001512  004136          MOV     @#LC,@#DPB.B+12
3591  007170  012737  007170  001110          MOV     #.,$LPERR       ;SETUP THE ERROR LOOP ADDRESS
3592  007176  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
3593  007202                  DECSK:
3594  007202  004037  025262                  JSR     R0,@#CALL.B     ;GO EXECUTE THE COMMAND
3595  007206  163737  001514  004136          SUB     @#IC,@#DPB.B+12
3596  007214  023737  001510  004136          CMP     @#FC,@#DPB.B+12
3597  007222  003767                          BLE     DECSK
3598  007224  000004                  EXIT2:  SCOPE                   ;LOOP
```

```
3599
3600                                    ;;*****************************************************************
3601                                    ;*TEST 3        STEPPING SEEK TEST
3602
3603                                    ;*      THIS TEST WILL COMMAND SEEK CYCLES TO CYLINDER 0,1,2,4,
3604                                    ;*      8,16,32,64,128, AND 256.  AT THE COMPLETION OF EACH SEEK
3605                                    ;*      COMMAND THE PROPER INDICATORS ARE EXAMINED TO ENSURE PROPER
3606                                    ;*      OPERATION.
3607
3608                                    ;;*****************************************************************
3609  007226                            TST3:
3610  007226  000240                           NOP
3611  007230  033737  001432  001234           BIT     @#BITS+<3*2>,TSTNMS ;DO THIS TEST?
3612  007236  001002                           BNE     64$                 ;YES--BRANCH
3613  007240  000137  007422                   JMP     TST4                ;NO--GO TO THE NEXT TEST
3614  007244  012737  000003  001102  64$:    MOV     #3,@#$TSTNM         ;SET UP TEST NUMBER AND
3615                                                                       ;CLEAR THE ERROR FLAG ($ERFLG)
3616  007252  004737  024726                   JSR     PC,LODPRM           ;LOAD THE PARMETERS FOR THE TEST
3617  007256  012737  007336  001110           MOV     #TEST3,@#$LPERR      ;SETUP THE LOOP ON ERROR ADDRESS
3618  007264  013777  001102  171650           MOV     $TSTNM,@DISPLAY      ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
3619  007272  013737  001506  001204           MOV     @#RPT,$TIMES        ;GET THE ITERATION COUNT
3620  007300  112737  000031  001115           MOVB    #25.,$ERMAX         ;MAX ERRORS ALLOWED FOR TEST
```

```
 3621  007306  012737  007314  001106          MOV     #1$,$LPADR      ;SETUP TEST LOOP ADDRESS
 3622  007314  113737  001524  004134  1$:     MOVB    @#FS,@#DPB.B+10 ;FS
 3623  007322  113737  001516  004135          MOVB    @#FT,@#DPB.B+11 ;FT
 3624  007330  012737  007420  001252          MOV     #EXIT3,@#BYPASS ;GO TO BYPASS ON ERROR
 3625  007336  013737  001510  004136  TEST3:  MOV     @#FC,@#DPB.B+12 ;FC
 3626  007344  012737  007344  001110          MOV     #.,$LPERR       ;SETUP THE ERROR LOOP ADDRESS
 3627  007352  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
 3628  007356  004037  025262                  JSR     R0,@#CALL.B     ;GO EXECUTE THE COMMAND
 3629  007362  013701  001514                  MOV     1C,R1           ;CYLINDER 1
 3630  007366  012737  007366  001110          MOV     #.,$LPERR       ;SETUP THE ERROR LOOP ADDRESS
 3631  007374  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
 3632  007400  010137  004136          1$:     MOV     R1,@#DPB.B+12   ;DESIRED CYLINDER
 3633  007404  004037  025262                  JSR     R0,@#CALL.B     ;GO EXECUTE THE COMMAND
 3634  007410  006301                          ASL     R1              ;MOVE TO NEXT CYLINDER
 3635  007412  020137  001512                  CMP     R1,@#LC         ;DONE?
 3636  007416  003770                          BLE     1$              ;NO--LOOP
 3637  007420  000004                  EXIT3:  SCOPE
 3638
```

```
3639                                    ;;*****************************************************************
3640                                    ;*TEST 4        OSCILLATING SEEK TEST
3641
3642                                    ;*      THIS TEST WILL COMMAND SEEK CYCLES FROM "FC" TO "NC" AND BACK
3643                                    ;*      TO "FC".  "NC" STARTS AT "FC" AND INCREMENTS BY "IC" UP TO CYLINDER
3644
3645                                    ;*      "LC", THEN IS DECREMENTED BY "IC" BACK TO CYLINDER "FC".  AT THE
3646                                    ;*      COMPLETION OF EVERY SEEK COMMAND THE PROPER INDICATORS ARE
3647                                    ;*      EXAMINED TO ENSURE PROPER OPERATION.
3648
3649                                    ;;*****************************************************************
3650  007422                            TST4:
3651  007422  000240                           NOP
3652  007424  033737  001434  001234           BIT     @#BITS+<4*2>,TSTNMS ;DO THIS TEST?
3653  007432  001002                           BNE     64$             ;YES--BRANCH
3654  007434  000137  010010                   JMP     TST5            ;NO--GO TO THE NEXT TEST
3655  007440  012737  000004  001102  64$:     MOV     #4,@#$TSTNM     ;SET UP TEST NUMBER AND
3656                                                                    ;CLEAR THE ERROR FLAG ($ERFLG)
3657  007446  004737  024726                   JSR     PC,LODPRM       ;LOAD THE PARMETERS FOR THE TEST
3658  007452  012737  007546  001110           MOV     #TEST4,@#$LPERR  ;SETUP THE LOOP ON ERROR ADDRESS
3659  007460  013777  001102  171454           MOV     $TSTNM,@DISPLAY  ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
3660  007466  013737  001506  001204           MOV     @#RPT,$TIMES    ;GET THE ITERATION COUNT
3661  007474  112737  000031  001115           MOVB    #25.,$ERMAX     ;MAX ERRORS ALLOWED FOR TEST
3662  007502  012737  007510  001106           MOV     #1$,$LPADR      ;SETUP LOOP ADDRESS
3663  007510  113737  001524  004134  1$:      MOVB    @#FS,@#DPB.B+10 ;FS
3664  007516  113737  001516  004135           MOVB    @#FT,@#DPB.B+11 ;FT
3665  007524  012737  010006  001252           MOV     #EXIT4,@#BYPASS ;GO TO EXIT4 ON ERROR
3666  007532  005002                           CLR     R2              ;CLEAR STALL SWITCH (NO STALL)
3667  007534  032737  010000  001220           BIT     #SW12,@#C.SWR   ;STALL REQUIRED?
3668  007542  001401                           BEQ     TEST4           ;NO--BRANCH
3669  007544  005102                           COM     R2              ;YES--SET SWITCH
3670  007546  013701  001510          TEST4:   MOV     @#FC,R1         ;SET NC TO FC
3671  007552  005037  001336                   CLR     @#STALL0        ;START AT ZERO IF STALLS REQUIRED
3672  007556  012737  007556  001110           MOV     #.,$LPERR       ;SETUP THE ERROR LOOP ADDRESS
3673  007564  012706  001100                   MOV     #STACK,SP       ;LOAD THE STACK POINTER
3674  007570  010137  004136          1$:      MOV     R1,@#DPB.B+12   ;NC
3675  007574  004037  025262                   JSR     R0,@#CALL.B     ;GO EXECUTE THE COMMAND
3676  007600  005702                           TST     R2              ;STALL?
3677  007602  001403                           BEQ     2$              ;NO--BRANCH
3678  007604  004037  026502                   JSR     R0,@#STALL      ;YES-GO TO STALL ROUTINE
3679  007610  001336                           .WORD   STALL0          ;TIME POINTER
3680  007612  013737  001510  004136  2$:      MOV     FC,@#DPB.B+12   ;FC
3681  007620  004037  025262                   JSR     R0,@#CALL.B     ;GO EXECUTE THE COMMAND
3682  007624  005702                           TST     R2              ;STALL?
3683  007626  001413                           BEQ     3$              ;NO--BRANCH
3684  007630  004037  026502                   JSR     R0,@#STALL      ;YES--GO TO STALL ROUTINE
3685  007634  001336                           .WORD   STALL0          ;TIME POINTER
3686  007636  005237  001336                   INC     @#STALL0        ;UPDATE THE TIME
3687  007642  023737  001362  001336           CMP     @#MXSTAL,@#STALL0 ;TIME TO BIG?
3688  007650  003347                           BGT     1$              ;NO--BRANCH
3689  007652  005037  001336                   CLR     @#STALL0        ;YES--START OVER AT ZERO
3690  007656  063701  001514          3$:      ADD     @#IC,R1         ;MOVE TO NEXT CYLINDER
3691  007662  020137  001512                   CMP     R1,@#LC         ;LAST CYLINDER COMPLETED?
3692  007666  003740                           BLE     1$              ;NO--BRANCH
3693  007670  013701  001512                   MOV     @#LC,R1         ;SET NC TO LC
3694  007674  012737  007674  001110           MOV     #.,$LPERR       ;SETUP THE ERROR LOOP ADDRESS
```

```
3695  007702  012706  001100            MOV     #STACK,SP       ;LOAD THE STACK POINTER
3696  007706  010137  004136      4$:   MOV     R1,@#DPB.B+12   ;NC
3697  007712  004037  025262            JSR     R0,@#CALL.B     ;GO EXECUTE THE COMMAND
3698  007716  005702                    TST     R2              ;STALL?
3699  007720  001403                    BEQ     5$              ;NO--BRANCH
3700  007722  004037  026502            JSR     R0,@#STALL      ;YES--GO TO STALL ROUTINE
3701  007726  001336                    .WORD   STALL0          ;TIME POINTER
3702  007730  013737  001512  004136  5$:   MOV     @#LC,@#DPB.B+12 ;LC
3703  007736  004037  025262            JSR     R0,@#CALL.B     ;GO EXECUTE THE COMMAND
3704  007742  005702                    TST     R2              ;STALL?
3705  007744  001413                    BEQ     6$              ;NO--BRANCH
3706  007746  004037  026502            JSR     R0,@#STALL      ;YES--GO TO STALL ROUTINE
3707  007752  001336                    .WORD   STALL0          ;TIME POINTER
3708  007754  005237  001336            INC     @#STALL0        ;UPDATE STALL TIME
3709  007760  023737  001362  001336    CMP     @#MXSTAL,@#STALL0 ;TIME TOO BIG?
3710  007766  003347                    BGT     4$              ;NO--BRANCH
3711  007770  005037  001336            CLR     @#STALL0        ;YES--SET STALL TIME BACK TO ZERO
3712  007774  163701  001514      6$:   SUB     @#IC,R1         ;NEXT CYLINDER
3713  010000  020137  001510            CMP     R1,@#FC         ;DONE?
3714  010004  002340                    BGE     4$              ;NO--BRANCH
3715  010006  000004                EXIT4:  SCOPE                   ;LOOP
3716
3717                                    ;;*****************************************************************
3718                                    ;*TEST 5        CONVERGING/DIVERGING SEEK TEST
3719
3720                                    ;*      THIS TEST WILL CAUSE THE DRIVE TO EXECUTE FORWARD AND REVERSE
3721                                    ;*      SEEKS FROM "NC1" AND "NC2" RESPECTIVELY, "NC1" WILL BE INCREMENTED
3722                                    ;*      BY "IC" AND "NC2" WILL BE DECREMENTED BY "IC" UNTIL "NC1" IS
3723                                    ;*      GREATER THAN THE INITIAL VALUE OF "NC2" AND "NC2" IS
3724                                    ;*      LESS THAN THE INITIAL VALUE OF "NC1".  AT THE COMPLETION OF
3725                                    ;*      EACH SEEK COMMAND THE PROPER INDICATORS ARE EXAMINED TO
3726                                    ;*      ENSURE PROPER OPERATION.  "NC1" AND "NC2" DEFAULT TO
3727                                    ;*      "FC" AND "LC" RESPECTIVELY.
3728
3729                                    ;;*****************************************************************
3730  010010                        TST5:
3731  010010  000240                        NOP
3732  010012  033737  001436  001234        BIT     @#BITS+<5*2>,TSTNMS ;DO THIS TEST?
3733  010020  001002                        BNE     64$             ;YES--BRANCH
3734  010022  000137  010210                JMP     TST6            ;NO--GO TO THE NEXT TEST
3735  010026  012737  000005  001102  64$:   MOV     #5,@#$TSTNM     ;SET UP TEST NUMBER AND
3736                                                                ;CLEAR THE ERROR FLAG ($ERFLG)
3737  010034  004737  024726                JSR     PC,LODPRM       ;LOAD THE PARMETERS FOR THE TEST
3738  010040  012737  010120  001110        MOV     #TEST5,@#$LPERR  ;SETUP THE LOOP ON ERROR ADDRESS
3739  010046  013777  001102  171066        MOV     $TSTNM,@DISPLAY  ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
3740  010054  013737  001506  001204        MOV     @#RPTL,$TIMES   ;GET THE ITERATION COUNT
3741  010062  112737  000031  001115        MOVB    #25.,$ERMAX     ;MAX ERRORS ALLOWED FOR TEST
3742  010070  012737  010076  001106        MOV     #1$,$LPADR      ;SETUP LOOP ADDRESS
3743  010076  113737  001524  004134  1$:   MOVB    @#FS,@#DPB.B+10 ;FS
3744  010104  113737  001516  004135        MOVB    @#FT,@#DPB.B+11 ;FT
3745  010112  012737  010206  001252        MOV     #EXIT5,@#BYPASS ;GO TO EXIT5 ON ERROR
3746  010120  013701  001510            TEST5:  MOV     @#FC,R1         ;START NC1 AT FC
3747  010124  013702  001512                    MOV     @#LC,R2         ;START NC2 AT LC
3748  010130  012737  010130  001110            MOV     #.,$LPERR       ;SETUP THE ERROR LOOP ADDRESS
3749  010136  012706  001100                    MOV     #STACK,SP       ;LOAD THE STACK POINTER
3750  010142  010137  004136      1$:   MOV     R1,@#DPB.B+12   ;NC1
```

```
3751  010146  004037  025262                    JSR     R0,@#CALL.B     ;GO EXECUTE THE COMMAND
3752  010152  010237  004136                    MOV     R2,@#DPB.B+12   ;NC2
3753  010156  004037  025262                    JSR     R0,@#CALL.B     ;GO EXECUTE THE COMMAND
3754  010162  063701  001514                    ADD     @#IC,R1         ;NEXT NC1
3755  010166  163702  001514                    SUB     @#IC,R2         ;NEXT NC2
3756  010172  020137  001512                    CMP     R1,@#LC         ;DONE?
3757  010176  003003                            BGT     EXIT5           ;YES--BRANCH
3758  010200  020237  001510                    CMP     R2,@#FC ;?
3759  010204  002356                            BGE     1$              ;NO--BRANCH
3760  010206  000004                EXIT5:      SCOPE                   ;LOOP
3761
3762                                            ;;*****************************************************************
3763                                            ;*TEST 6        SERVO ADDRESSING LOGIC NOISE GENERATOR
3764
3765                                            ;*      IN THIS TEST A SEEK IS DONE TO CYL "NC" THEN A SEEK TO
3766                                            ;*      NC+4 THEN NC+1 THEN NC+3 THEN NC+2 THEN NC+5.  NOW "NC" IS UPDATED
3767                                            ;*      BY "IC" AND THE ABOVE SEQUENCE IS REPEATED UNITL "LC" IS
3768                                            ;*      EXCEEDED BY ANY OF THE ABOVE VALUES.  THE INITIAL VALUE OF "NC"
3769                                            ;*      IS "FC".  AT THE COMPLETION OF EACH SEEK COMMAND THE
3770                                            ;*      PROPER INDICATORS ARE EXAMINED TO ENSURE PROPER OPERATION.
3771
3772                                            ;;*****************************************************************
3773  010210                                    TST6:
3774  010210  000240                            NOP
3775  010212  033737  001440  001234            BIT     @#BITS+<6*2>,TSTNMS ;DO THIS TEST?
3776  010220  001002                            BNE     64$             ;YES--BRANCH
3777  010222  000137  010454                    JMP     TST7            ;NO--GO TO THE NEXT TEST
3778  010226  012737  000006  001102  64$:      MOV     #6,@#$TSTNM     ;SET UP TEST NUMBER AND
3779                                            ;CLEAR THE ERROR FLAG ($ERFLG)
3780  010234  004737  024726                    JSR     PC,LODPRM       ;LOAD THE PARMETERS FOR THE TEST
3781  010240  012737  010320  001110            MOV     #TEST6,@#$LPERR  ;SETUP THE LOOP ON ERROR ADDRESS
3782  010246  013777  001102  170666            MOV     $TSTNM,@DISPLAY  ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
3783  010254  013737  001506  001204            MOV     @#RPT,$TIMES    ;GET THE ITERATION COUNT
3784  010262  112737  000031  001115            MOVB    #25.,$ERMAX     ;MAX ERRORS ALLOWED FOR TEST
3785  010270  012737  010276  001106            MOV     #1$,$LPADR      ;SETUP LOOP ADDRESS
3786  010276  113737  001524  004134  1$:       MOVB    @#FS,@#DPB.B+10 ;FS
3787  010304  113737  001516  004135            MOVB    @#FT,@#DPB.B+11 ;FT
3788  010312  012737  010452  001252            MOV     #EXIT6,@#BYPASS ;GO TO EXIT6 ON ERROR
3789  010320  013701  001510          TEST6:    MOV     @#FC,R1         ;PICKUP "FC"
3790  010324  013702  001512                    MOV     @#LC,R2         ;FORM LAST CYLINDER THAT
3791  010330  162702  000005                    SUB     #5,R2           ;IS AVAILABLE FOR TESTING
3792  010334  012737  010334  001110            MOV     #.,$LPERR       ;SETUP THE ERROR LOOP ADDRESS
3793  010342  012706  001100                    MOV     #STACK,SP       ;LOAD THE STACK POINTER
3794  010346  020102                  1$:       CMP     R1,R2           ;LAST CYLINDER
3795  010350  003040                            BGT     EXIT6           ;YES--BRANCH
3796  010352  010137  004136                    MOV     R1,@#DPB.B+12   ;NC
3797  010356  004037  025262                    JSR     R0,@#CALL.B     ;GO EXECUTE THE COMMAND
3798  010362  062737  000004  004136            ADD     #4,@#DPB.B+12   ;NC+4
3799  010370  004037  025262                    JSR     R0,@#CALL.B     ;GO EXECUTE THE COMMAND
3800  010374  162737  000003  004136            SUB     #3,@#DPB.B+12   ;NC+1
3801  010402  004037  025262                    JSR     R0,@#CALL.B     ;GO EXECUTE THE COMMAND
3802  010406  062737  000002  004136            ADD     #2,@#DPB.B+12   ;NC+3
3803  010414  004037  025262                    JSR     R0,@#CALL.B     ;GO EXECUTE THE COMMAND
3804  010420  162737  000001  004136            SUB     #1,@#DPB.B+12   ;NC+2
3805  010426  004037  025262                    JSR     R0,@#CALL.B     ;GO EXECUTE THE COMMAND
3806  010432  062737  000003  004136            ADD     #3,@#DPB.B+12   ;NC+5
```

```
3807  010440  004037  025262                    JSR     R0,@#CALL.B     ;GO EXECUTE THE COMMAND
3808  010444  063701  001514                    ADD     @#IC,R1
3809  010450  000736                           BR      1$
3810  010452  000004                    EXIT6: SCOPE                   ;LOOP
3811
3812                                    ;;*****************************************************************
3813                                    ;*TEST 7        RANDOM SEEK TEST
3814
3815                                    ;*      THIS TEST PERFORMS RANDOM SEEK OPERATIONS BETWEEN CYLINDERS 'FC'
3816                                    ;*      'LC'.  AFTER EACH SEEK, THE POSITION OF THE DRIVE IS VERIFIED BY
3817                                    ;*      READING A SECTOR FROM THE CURRENTLY ADDRESSED CYLINDER AND TRACK.
3818                                    ;*      THE TRACK ADDRESS IS INCREMENTED FOR EACH SEEK SO THAT VERIFICATION
3819                                    ;*      OF POSITIONING OCCURS USING EACH HEAD.  TRACK ADDRESSES ARE INCREMENTED
3820                                    ;*      BETWEEN PARAMTERS 'FT' AND 'LT'.
3821
3822                                    ;;*****************************************************************
3823  010454                            TST7:
3824  010454  000240                           NOP
3825  010456  033737  001442  001234           BIT     @#BITS+<7*2>,TSTNMS ;DO THIS TEST?
3826  010464  001002                           BNE     64$             ;YES--BRANCH
3827  010466  000137  011034                   JMP     TST10           ;NO--GO TO THE NEXT TEST
3828  010472  012737  000007  001102  64$:    MOV     #7,@#$TSTNM     ;SET UP TEST NUMBER AND
3829                                                                   ;CLEAR THE ERROR FLAG ($ERFLG)
3830  010500  004737  024726                   JSR     PC,LODPRM       ;LOAD THE PARMETERS FOR THE TEST
3831  010504  012737  010576  001110           MOV     #TEST7,@#$LPERR  ;SETUP THE LOOP ON ERROR ADDRESS
3832  010512  013777  001102  170422           MOV     $TSTNM,@DISPLAY  ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
3833  010520  013737  001506  001204           MOV     @#RPT,$TIMES    ;GET THE ITERATION COUNT
3834  010526  112737  000031  001115           MOVB    #25.,$ERMAX     ;MAX ERRORS ALLOWED FOR TEST
3835  010534  113737  001516  004135           MOVB    FT,DPB.B+11     ;LOAD STARTING TRACK ADDRESS
3836  010542  112737  000105  004106           MOVB    #SEEK,@#DPB.A+2 ;SEEK=COMMAND
3837  010550  112737  000173  004126           MOVB    #READHD,DPB.B+2  ;READ HEADER & DATA COMMAND
3838  010556  013704  034632                   MOV     RPADR,R4        ;UNIBUS ADDRESS OF THE RH11
3839  010562  012737  011032  001252           MOV     #EXIT7,BYPASS   ;ERROR TERMINATION ADDRESS
3840  010570  012737  010576  001106           MOV     #TEST7,$LPADR   ;SETUP THE LOOP ON TEST ADDRESS
3841  010576  012706  001100           TEST7:  MOV     #STACK,SP       ;SETUP THE STACK POINTER
3842  010602  013737  001510  004136           MOV     FC,DPB.B+12     ;INITIAL CYLINDER ADDRESS
3843  010610  023737  001510  001512           CMP     FC,LC           ;CYLINDER LIMITS THE SAME ?
3844  010616  001422                           BEQ     1$              ;BR IF THEY ARE
3845  010620  004737  023626                   JSR     PC,$RAND        ;CYCYLE THE RANDOM NUMBER GENERATOR
3846  010624  013746  023724                   MOV     $HINUM,-(SP)    ;USE THE HIGH RANDOM NUMBER
3847  010630  005046                           CLR     -(SP)           ;UPPER DIVIDEND
3848  010632  013746  001512                   MOV     LC,-(SP)        ;FORM THE DIVISOR
3849  010636  005216                           INC     (SP)            ;INCREMENT
3850  010640  163716  001510                   SUB     FC,(SP)         ;SUBTRACT THE LOWER LIMIT
3851  010644  004737  023730                   JSR     PC,$DIV         ;DIVIDE
3852  010650  062637  004136                   ADD     (SP)+,DPB.B+12  ;ADD THE REMAINDER TO THE INITIAL CYLINDER
3853  010654  005726                           TST     (SP)+           ;DISCARD THE QUOTENT
3854  010656  013737  004136  004116           MOV     DPB.B+12,DPB.A+12  ;COPY NEW CYLINDER ADDRESS
3855  010664                            1$:
3856  010664  012737  010664  001110           MOV     #.,$LPERR       ;SETUP THE ERROR LOOP ADDRESS
3857  010672  012706  001100                   MOV     #STACK,SP       ;LOAD THE STACK POINTER
3858  010676  004037  025150                   JSR     R0,@#CALL.A     ;GO EXECUTE THE COMMAND
3859  010702  012737  010702  001110           MOV     #.,$LPERR       ;SETUP THE ERROR LOOP ADDRESS
3860  010710  012706  001100                   MOV     #STACK,SP       ;LOAD THE STACK POINTER
3861  010714  113764  004104  000010           MOVB    DPB.A,RPCS2(R4) ;SELECT THE DRIVE
3862  010722  016446  000020                   MOV     RPLA(R4),-(SP)  ;GET THE LOOK AHEAD REGISTER
```

```
3863  010726  006316                          ASL     (SP)              ;ALIGN THE SECTOR ADDRESS
3864  010730  006316                          ASL     (SP)              ;ALIGN THE SECTOR ADDRESS
3865  010732  000316                          SWAB    (SP)              ;PUT ADDRESS IN LOWER BYTE
3866  010734  105766  000001                  TSTB    1(SP)             ;IN THE 1ST 20% OF SECTOR ?
3867  010740  001401                          BEQ     2$                ;BR IF YES
3868  010742  105216                          INCB    (SP)              ;INCREMENT THE SECTOR ADDRESS
3869  010744  105216                  2$:     INCB    (SP)              ;INCREMENT THE SECTOR ADDRESS
3870  010746  112637  004174                  MOVB    (SP)+,DTADPB+10   ;LOAD THE DPB
3871  010752  013746  001630                  MOV     PRMLMT+22,-(SP)   ;PUT LAST SECTOR ADDRESS ON THE STACK
3872  010756  005216                          INC     (SP)              ;INCREMENT IT
3873  010760  122637  004174                  CMPB    (SP)+,DTADPB+10   ;NEW SECTOR ADDRESS TOO LARGE ?
3874  010764  103007                          BHIS    4$                ;BR IF NOT
3875  010766  103403                          BLO     3$                ;BR IF ADDRESS IS 2 GREATER
3876  010770  105037  004174                  CLRB    DTADPB+10         ;RESET TO SECTOR ADDRESS 0
3877  010774  000403                          BR      4$                ;CONTINUE
3878  010776  112737  000001  004174  3$:     MOVB    #1,DTADPB+10      ;RESET ADDRESS TO SECTOR 1
3879  011004                          4$:
3880  011004  004037  025262                  JSR     R0,@#CALL.B       ;GO EXECUTE THE COMMAND
3881  011010  105237  004135                  INCB    DPB.B+11          ;INCREMENT THE TRACK ADDRESS
3882  011014  123737  004135  001520          CMPB    DPB.B+11,LT       ;MAXIMUM ?
3883  011022  101403                          BLOS    EXIT7             ;BR IF NOT
3884  011024  113737  001516  004135          MOVB    FT,DPB.B+11       ;RELOAD STARTING TRACK ADDRESS
3885  011032  000004                  EXIT7:  SCOPE                     ;LOOP ?
3886
3887
3888                                  ;;*****************************************************************
3889                                  ;*TEST 10       SERVO SETTLE DOWN TEST
3890
3891                                  ;*      THIS TEST VERIFIES THAT THE SERVO HAS SETTLED DOWN AND THAT
3892                                  ;*      THE DRIVE IS ON CYLINDER WHEN THE DRIVE INDICATES SEEK COMPLETE.
3893                                  ;*      RANDOM SEEKS ARE ISSUED BETWEEN CYLINDERS 'NC1' AND 'NC1+1C'
3894                                  ;*      ('NC1' STARTS AT VALUE 'FC').  AT THE COMPLETION OF 1000 (10) SEEKS,
3895                                  ;*      'NC1' IS INCREMENTED BY VALUE '1C' AND THE SEQUENCE IS REPEATED.
3896                                  ;*      THE TEST IS COMPLETED WHEN 'NC1' HAS BEEN INCREMENTED BEYOND 'LC'.
3897                                  ;*
3898                                  ;*      WHEN THE SEEK COMPLETES, THE PROGRAM READS THE DRIVE'S LOOK-AHEAD
3899                                  ;*      REGISTER (RPLA) TO DETERMINE THE ADDRESS OF THE SECTOR ROTATING  INTO
3900                                  ;*      POSITION.  THE PROGRAM THEN ISSUES A WRITE HEADER AND DATA COMMAND
3901                                  ;*      FOR THAT SECTOR.  IF THE DRIVE'S POSITIONER HAS NOT SETTLED DOWN OR
3902                                  ;*      IF THE POSITIONER IS NOT ON CYLINDER (IF THE DRIVE IS AN RP04, THE
3903                                  ;*      OFF CYLINDER CONDITION MUST LAST FOR AT LEAST 800 US), THE DRIVE
3904                                  ;*      WILL REPORT A 'WRU' ERROR.  (RP05/6'S MAY ALSO REPORT 'NHS' ERROR UNDER
3905                                  ;*      ERRORS IN THIS TEST INDICATE THAT THE SERVO SYSTEM MAY NOT BE ADJUSTED
3906                                  ;*      CORRECTLY, THAT THE DRIVE IS MALFUNCTIONING, OR THAT A PACK WITH
3907                                  ;*      MARGINAL SERVO TRACKS IS MOUNTED ON THE DRIVE.
3908                                  ;*
3909                                  ;*      THIS TEST USES THE EXTENTION BITS IN THE LOOK-AHEAD REGISTER TO DETERMINE
3910                                  ;*      WHETHER OR NOT IT CAN PICK UP THE SECTOR ROTATING INTO POSITION.  THE
3911                                  ;*      TEST IS OPTIMIZED SUCH THAT IF THE DRIVE SIGNALS SEEK DONE WITHIN
3912                                  ;*      THE FIRST 80% OF THE SECTOR CURRENTLY UNDER THE HEAD, THE TEST WILL
3913                                  ;*      TRY TO ADDRESS THE NEXT SECTOR.  BASED ON OBSERVATION, THE PROGRAM
3914                                  ;*      IS ABLE TO START THE OPERATION WITHOUT LOSING A REVOLUTION MOST OF
3915                                  ;*      THE TIME.
3916                                  ;*
3917                                  ;*      THIS TEST IS VALID ONLY IF THE OPERATION IS STARTED WITHIN A FEW
3918                                  ;*      HUNDRED MICRO-SECONDS AFTER SEEK DONE OCCURS.  THE NECESSARY TIME
```

```
                                        ;*      TIME DEPENDENT PARAMETERS OCCUR FREQUENTLY ENOUGH WITHIN THE REQUIRED
                                        ;*      RANGE TO PERMIT THIS TEST TO BE EFFECTIVE.
                                        ;
                                        ;;*****************************************************************
011034                                  TST10:
011034  000240                                NOP
011036  033737  001444  001234                BIT     @#BITS+<10*2>,TSTNMS ;DO THIS TEST?
011044  001002                                BNE     64$             ;YES--BRANCH
011046  000137  012140                        JMP     TST11           ;NO--GO TO THE NEXT TEST
011052  012737  000010  001102  64$:          MOV     #10,@#$TSTNM    ;SET UP TEST NUMBER AND
                                                                      ;CLEAR THE ERROR FLAG ($ERFLG)
011060  004737  024726                        JSR     PC,LODPRM       ;LOAD THE PARMETERS FOR THE TEST
011064  012737  011242  001110                MOV     #TEST10,@#$LPERR  ;SETUP THE LOOP ON ERROR ADDRESS
011072  013777  001102  170042                MOV     $TSTNM,@DISPLAY  ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
011100  013737  001506  001204                MOV     @#RPT,$TIMES    ;GET THE ITERATION COUNT
011106  112737  000031  001115                MOVB    #25.,$ERMAX     ;MAX ERRORS ALLOWED FOR TEST
011114  012737  011122  001106                MOV     #1$,$LPADR      ;SETUP THE LOOP ADDRESS
011122                                  1$:
011122  112737  000105  004106                MOVB    #SEEK,@#DPB.A+2 ;SEEK=COMMAND
011130  112737  000161  004166                MOVB    #WRITE,DTADPB+2 ;COMMAND
011136  113737  001516  004175                MOVB    FT,DTADPB+11    ;TRACK ADDRESS FOR THE WRITE
011144  013737  001510  004116                MOV     FC,DPB.A+12     ;CYLINDER ADDRESS FOR THE SEEK
011152  013737  001510  004176                MOV     FC,DTADPB+12    ;CYLINDER ADDRESS FOR THE WRITE
011160  013737  001510  001532                MOV     FC,NC1          ;STARTING CYLINDER
011166  013737  001514  001350                MOV     IC,DELTA        ;CYLINDER INCREMENT VALUE
011174  012737  176000  004170                MOV     #-<256.*4.>,DTADPB+4  ;WORD COUNT
011202  012737  050202  004172                MOV     #BUFFER,DTADPB+6  ;BUFFER ADDRESS
011210  005000                                CLR     R0              ;PATTERN POINTER (WC PATTERN)
011212  004737  030476                        JSR     PC,SETBUF       ;LOAD THE WRITE BUFFER
011216  005001                                CLR     R1              ;CLEAR REGISTER
011220  113701  004104                        MOVB    DPB.A,R1        ;LOAD DRIVE ADDRESS
011224  013704  034632                        MOV     RPADR,R4        ;UNIBUS ADDRESS OF THE RH11
011230  004737  042664                        JSR     PC,CLRQUE       ;CLEAR THE OPERATION QUEUES
011234  012737  012136  001252                MOV     #EXIT10,BYPASS  ;ERROR EXIT FROM TEST
011242                                  TEST10:
011242  012737  011242  001110                MOV     #.,$LPERR       ;SETUP THE ERROR LOOP ADDRESS
011250  012706  001100                        MOV     #STACK,SP       ;LOAD THE STACK POINTER
011254  012737  000340  177776                MOV     #PR7,@#PS       ;SET PRIORITY TO 7
011262  005737  001244                        TST     CLKSTA          ;SEE WHICH CLOCK ON SYSTEM
011266  001415                                BEQ     3$              ;BR IF NO CLOCK
011270  100405                                BMI     1$              ;BR IF KW11-L CLOCK
011272  017746  170076                        MOV     @PKV,-(SP)      ;SAVE THE VECTOR
011276  013746  001374                        MOV     PKV,-(SP)       ;SAVE THE VECTOR ADDRESS
011302  000404                                BR      2$              ;CONTINUE
011304  017746  170076                1$:     MOV     @LKV,-(SP)      ;SAVE THE 'L' CLOCK VECTOR
011310  013746  001406                        MOV     LKV,-(SP)       ;SAVE THE VECTOR ADDRESS
011314  012776  012072  000000        2$:     MOV     #TST10B,@(SP)   ;CHANGE THE VECTOR
011322  012777  027170  023304        3$:     MOV     #DORTI,@RPVEC   ;CHANGE THE RP04/RP05 VECTOR
011330  012737  000010  001344                MOV     #8., SEKTMR     ;LOAD THE SEEK TIMER
011336  012764  000040  000010                MOV     #BIT05,RPCS2(R4)  ;INIT THE MASSBUS
011344  110164  000010                        MOVB    R1,RPCS2(R4)    ;RESELECT THE DRIVE
011350  013764  004116  000034                MOV     DPB.A+12,RPCA(R4)  ;LOAD THE CYLINDER ADDRESS
011356  013737  004116  001270                MOV     DPB.A+12,CYL.DS ;CYLINDER ADDRESS FOR ERROR MESSAGE
011364  112764  000105  000000                MOVB    #SEEK,RPCS1(R4) ;START THE SEEK
011372  005037  177776                        CLR     @#PS            ;CLEAR THE PRIORITY
011376  105764  000012                4$:     TSTB    RPDS1(R4)       ;HAS THE DRIVE FINISHED ?
```

```
3975  011402  100402                        BMI     5$              ;BR IF IT HAS
3976  011404  00000i                        WAIT                    ;WAIT FOR THE OPERATION TO COMPLETE
3977  011406  000773                        BR      4$              ;CONTINUE
3978  011410  012737  000340  177776  5$:   MOV     #PR7,@#PS       ;CHANGE PRIORITY TO MAX
3979  011416  032764  040000  000012        BIT     #BIT14,RPDS1(R4)  ;ERROR ?
3980  011424  001412                        BEQ     6$              ;BR IF NOT
3981  011426  012702  004104                MOV     #DPB.A,R2       ;DPB POINTER
3982  011432  004737  042202                JSR     PC,SVRH11       ;SAVE THE REGISTERS
3983  011436  104023                        ERROR   23              ;ERROR DURING SEEK
3984  011440  012764  000040  000010        MOV     #BIT05,RPCS2(R4)  ;INIT THE MASSBUS
3985  011446  110164  000010                MOVB    R1,RPCS2(R4)    ;RESELECT THE DRIVE
3986  011452  012777  037544  023154  6$:   MOV     #ISR,@RPVEC     ;SETUP THE RP04/RP05 VECTOR
3987  011460  005737  001244                TST     CLKSTA          ;WHICH CLOCK
3988  011464  001405                        BEQ     TST10A          ;BR IF NONE
3989  011466  016676  000002  000000        MOV     2(SP),@(SP)     ;RELOAD THE CLOCK VECTOR
3990  011474  062706  000004                ADD     #4,SP           ;CORRECT THE STACK POINTER
3991  011500                        TST10A:
3992  011500  012737  011500  001110        MOV     #.,$LPERR       ;SETUP THE ERROR LOOP ADDRESS
3993  011506  012706  001100                MOV     #STACK,SP       ;LOAD THE STACK POINTER
3994  011512  110164  000010                MOVB    R1,RPCS2(R4)    ;SELECT THE DRIVE
3995  011516  016446  000020                MOV     RPLA(R4),-(SP)  ;GET THE LOOK AHEAD REGISTER
3996  011522  006316                        ASL     (SP)            ;ALIGN THE SECTOR ADDRESS
3997  011524  006316                        ASL     (SP)            ;ALIGN THE SECTOR ADDRESS
3998  011526  000316                        SWAB    (SP)            ;PUT ADDRESS IN LOWER BYTE
3999  011530  122766  000300  000001        CMPB    #300,1(SP)      ;IN THE LAST 20% OR SECTOR ?
4000  011536  001001                        BNE     2$              ;BR IF NOT
4001  011540  105216                        INCB    (SP)            ;INCREMENT THE SECTOR ADDRESS
4002  011542  105216                2$:     INCB    (SP)            ;INCREMENT THE SECTOR ADDRESS
4003  011544  112637  004174                MOVB    (SP)+,DTADPB+10 ;LOAD THE DPB
4004  011550  013746  001630                MOV     PRMLMT+22,-(SP)  ;PUT MAXIMUM SECTOR ADDRESS ON THE STACK
4005  011554  005216                        INC     (SP)            ;INCREMENT PAST THE MAXIMUM ADDRESS
4006  011556  122637  004174                CMPB    (SP)+,DTADPB+10 ;NEW SECTOR ADDRESS TOO LARGE ?
4007  011562  101007                        BHI     4$              ;BR IF NOT
4008  011564  103403                        BLO     3$              ;BR IF ADDRESS IS 2 GREATER THAN MAXIMUM
4009  011566  105037  004174                CLRB    DTADPB+10       ;RESET TO SECTOR ADDRESS 0
4010  011572  000403                        BR      4$              ;CONTINUE
4011  011574  112737  000001  004174  3$:   MOVB    #1,DTADPB+10    ;RESET ADDRESS TO SECTOR 1
4012  011602  012703  004170          4$:   MOV     #DTADPB+4,R3    ;POINTER
4013  011606  012764  000111  000000        MOV     #DRVCLR,RPCS1(R4)  ;CLEAR THE DRIVE
4014  011614  012364  000002                MOV     (R3)+,RPWC(R4)  ;LOAD THE WORD COUNT
4015  011620  012364  000004                MOV     (R3)+,RPBA(R4)  ;LOAD THE BUFFER ADDRESS
4016  011624  012364  000006                MOV     (R3)+,RPDA(R4)  ;LOAD THE TRACK/SECTOR ADDR
4017  011630  005037  004202                CLR     DTADPB+16       ;RESET 'DONE' INDICATOR
4018  011634  012737  004164  034544        MOV     #DTADPB,TRNSWT  ;LOAD 'TRANSFER' DPB ADDRESS
4019  011642  010137  034616                MOV     R1,DTUW         ;ADDRESS OF DRIVE TRANSFERING
4020  011646  112761  000001  034474        MOVB    #1,DRVACT(R1)   ;SET DRIVE ACTIVE INDICATOR
4021  011654  006301                        ASL     R1              ;SHIFT DRIVE ADDRESS
4022  011656  012761  001750  034576        MOV     #1000.,TIMER(R1)  ;SETUP THE OPERATION TIMER
4023  011664  006201                        ASR     R1              ;RESTORE R1
4024  011666  013764  004166  000000        MOV     DTADPB+2,RPCS1(R4)  ;START THE OPERATION
4025  011674  005037  177776                CLR     @#PS            ;CLEAR THE PRIORITY
4026  011700  004037  025662                JSR     R0,DRVCL1       ;WAIT FOR OPERATION TO COMPLETE
4027  011704  023727  001346  001750  5$:   CMP     SEKCNT,#1000.   ;FINISHED SEEKS ?
4028  011712  001026                        BNE     6$              ;BR IF NOT
4029  011714  005037  001346                CLR     SEKCNT          ;CLEAR THE SEEK COUNT
4030  011720  063737  001514  001532        ADD     IC,NC1          ;ADD THE INCREMENT
```

```
 4031  011726  023737  001532  001512          CMP     NC1,LC          ;EXCEEDED THE CYLINDER LIMIT ?
 4032  011734  103100                          BHIS    EXIT10          ;BR IF IT HAS
 4033  011736  013737  001512  001350          MOV     LC,DELTA        ;GET THE NEXT 'ZONE' ADDRESS
 4034  011744  163737  001532  001350          SUB     NC1,DELTA       ;CHECK THE DIFFERENCE
 4035  011752  023737  001514  001350          CMP     IC,DELTA        ;DIFFERENCE GREATER THAN THE INCREMENT ?
 4036  011760  101003                          BHI     6$              ;BR IF IT IS
 4037  011762  013737  001514  001350          MOV     IC,DELTA        ;USE THE ICREMENT PARAMETER
 4038  011770  005237  001346          6$:     INC     SEKCNT          ;COUNT THE NEXT SEEK
 4039  011774  023737  001510  001512          CMP     FC,LC           ;BEGINNING AND ENDING CYLINDERS THE SAME ?
 4040  012002  001002                          BNE     7$              ;BR IF NOT
 4041  012004  000137  011242                  JMP     TEST10          ;BR IF THEY ARE
 4042  012010  013737  001532  004116  7$:     MOV     NC1,DPB.A+12    ;RESET THE CYLINDER ADDRESS
 4043  012016  004737  023626                  JSR     PC,$RAND        ;CYCLE THE RANDOM NUMBER GENERATOR
 4044  012022  013746  023724                  MOV     $HINUM,-(SP)    ;USE THE HIGH RANDOM NUMBER
 4045  012026  005046                          CLR     -(SP)           ;CLEAR THE UPPER DIVIDEND
 4046  012030  013746  001350                  MOV     DELTA,-(SP)     ;FORM THE DIVISOR
 4047  012034  005216                          INC     (SP)            ;INCREMENT
 4048  012036  004737  023730                  JSR     PC,$DIV         ;DIVIDE
 4049  012042  062637  004116                  ADD     (SP)+,DPB.A+12  ;ADD THE REMAINDER TO THE INITIAL CYLINDER
 4050  012046  005726                          TST     (SP)+           ;DISCARD THE QUOTENT
 4051  012050  023737  004116  004176          CMP     DPB.A+12,DTADPB+12  ;SAME CYLINDER SELECTED AS LAST TIME ?
 4052  012056  001754                          BEQ     7$              ;BR IF IT WAS
 4053  012060  013737  004116  004176          MOV     DPB.A+12,DTADPB+12  ;COPY NEW CYLINDER ADDRESS
 4054  012066  000137  011242                  JMP     TEST10          ;CONTINUE
 4055  012072  005337  001344          TST10B: DEC     SEKTMR          ;DECREMENT THE SEEK TIMER
 4056  012076  001016                          BNE     1$              ;CONTINUE IF NOT DONE
 4057  012100  012702  004104                  MOV     #DPB.A,R2       ;DPB ADDRESS
 4058  012104  004737  042202                  JSR     PC,SVRH11       ;SAVE THE REGISTERS
 4059  012110  104024                          ERROR   24              ;TIMEOUT DURING SEEK
 4060  012112  012764  000040  000010          MOV     #BIT05,RPCS2(R4)  ;INIT THE MASSBUS
 4061  012120  110164  000010                  MOVB    R1,RPCS2(R4)    ;RESELECT THE DRIVE
 4062  012124  016676  000002  000000          MOV     2(SP),@(SP)     ;RESTORE THE CLOCK VECTOR ADDRESS
 4063  012132  000401                          BR      EXIT10          ;ABORT THE TEST
 4064  012134  000002                  1$:     RTI                     ;RETURN
 4065  012136  000004                  EXIT10: SCOPE                   ;LOOP ?
 4066
 4067                                          ;;*****************************************************************
 4068                                          ;*TEST 11       ALL SEEKS TEST
 4069
 4070                                          ;*      THIS TEST VERIFIES THAT THE DISK DRIVE CAN SEEK FROM EACH CYLINDER
 4071                                          ;*      TO ALL OTHER CYLINDERS.
 4072                                          ;*
 4073                                          ;*      BEGINNING WITH CYLINDER 'FC', THE TEST SEEKS TO EACH CYLINDER
 4074                                          ;*      BETWEEN 'FC' AND 'LC' FROM CYINDER 'FC'.  THE BEGINNING CYLINDER
 4075                                          ;*      ADDRESS IS INCREMENTED AND THE TEST SEEKS BETWEEN THE NEW CYLINDER
 4076                                          ;*      ADDRESS AND ALL CYLINDERS BETWEEN 'FC' AND 'LC'.  THE SEQUENCE
 4077                                          ;*      CONTINUES UNTIL ALL CYLINDERS HAVE BEEN CHECKED.
 4078
 4079                                          ;;*****************************************************************
 4080  012140                          TST11:
 4081  012140  000240                          NOP
 4082  012142  033737  001446  001234          BIT     @#BITS+<11*2>,TSTNMS ;DO THIS TEST?
 4083  012150  001002                          BNE     64$             ;YES--BRANCH
 4084  012152  000137  012360                  JMP     TST12           ;NO--GO TO THE NEXT TEST
 4085  012156  012737  000011  001102  64$:     MOV     #11,@#$TSTNM    ;SET UP TEST NUMBER AND
 4086                                                                  ;CLEAR THE ERROR FLAG ($ERFLG)
```

```
 4087  012164  004737  024726                 JSR     PC,LODPRM       ;LOAD THE PARMETERS FOR THE TEST
 4088  012170  012737  012300  001110         MOV     #TEST11,@#$LPERR  ;SETUP THE LOOP ON ERROR ADDRESS
 4089  012176  013777  001102  166736         MOV     $TSTNM,@DISPLAY  ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
 4090  012204  013737  001506  001204         MOV     @#RPT,$TIMES    ;GET THE ITERATION COUNT
 4091  012212  112737  000031  001115         MOVB    #25.,$ERMAX     ;MAX ERRORS ALLOWED FOR TEST
 4092  012220  012737  012226  001106         MOV     #1$,$LPADR      ;SETUP THE LOOP ADDRESS
 4093  012226  113737  001524  004134  1$:    MOVB    FS,DPB.B+10     ;SECTOR ADDRESS
 4094  012234  113737  001524  004154         MOVB    FS,DPB.C+10     ;SECTOR ADDRESS
 4095  012242  113737  001516  004135         MOVB    FT,DPB.B+11     ;TRACK ADDRESS
 4096  012250  113737  001516  004155         MOVB    FT,DPB.C+11     ;TRACK ADDRESS
 4097  012256  013737  001510  004136         MOV     FC,DPB.B+12     ;STARTING CYLINDER ADDRESS
 4098  012264  013737  001510  004156         MOV     FC,DPB.C+12     ;STARTING CYLINDER ADDRESS
 4099  012272  012737  012356  001252         MOV     #EXIT11,BYPASS  ;TEST ABORT EXIT
 4100  012300  012706  001100         TEST11: MOV     #STACK,SP       ;SETUP THE STACK POINTER
 4101  012304                         1$:
 4102  012304  004037  025452                 JSR     R0,@#CALL.C     ;GO EXECUTE THE COMMAND
 4103  012310  004037  025262                 JSR     R0,@#CALL.B     ;GO EXECUTE THE COMMAND
 4104  012314  063737  001514  004156         ADD     IC,DPB.C+12     ;INCREMENT THE ENDING CYLINDER ADDRESS
 4105  012322  023737  001512  004156         CMP     LC,DPB.C+12     ;CHECK IF EXCEEDING MAXIMUM
 4106  012330  002365                 BGE     1$              ;BR IF NOT
 4107  012332  013737  001510  004156         MOV     FC,DPB.C+12     ;RESET ENDING CYLINDER ADDRESS
 4108  012340  063737  001514  004136         ADD     IC,DPB.B+12     ;INCREMENT THE STARTING ADDRESS
 4109  012346  023737  001512  004136         CMP     LC,DPB.B+12     ;EXCEEDING MAXIMUM ?
 4110  012354  002353                 BGE     1$              ;BR IF NOT
 4111  012356  000004         EXIT11: SCOPE                   ;LOOP ?
 4112
```

```
4113
4114                                    .SBTTL  *** TIMING TESTS ***
4115

                ;/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:
                ;/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:
                ;/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:

                ;*THE TIMING TESTS WILL ENSURE THAT THOSE FUNCTIONS BEING
                ;*TIMED ARE WITHIN THE TOLERANCES SPECIFIED IN THE "RP04
                ;*ENGINEERING SPECIFICATIONS".
                ;*THE SEEK TIMING WILL BE PERFORMED USING EXPLICIT SEEK
                ;*OPERATIONS. AT THE COMPLETION OF EACH OF THE TIMING
                ;*TESTS THE MINIMUM, MAXIMUM AND AVERAGE TIMES WILL BE
                ;*TYPED.

                ;/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*
                ;/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*
                ;/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*
4133
4134                                    ;;*****************************************************************
4135                                    ;*TEST 12       ROTATIONAL SPEED TIMING TEST
4136
4137                                    ;*      THIS TEST WILL START A SEARCH TO CYLINDER 0, TRACK 0, SECTOR
4138                                    ;*      0.  AS SOON AS THE INTERRUPT OCCURS, THE GO BIT IS SET AGAIN
4139                                    ;*      AND THE OPERATION IS TIMED.  THIS PROCEDURE IS REPEATED 10
4140                                    ;*      TIMES THEN THE AVERAGE TIME IS CALCULATED AND CHECKED TO
4141                                    ;*      ENSURE IT IS WITHIN TOLERANCE:
4142                                    ;*              16.67 MS/REV + OR - 2% IF 60HZ
4143                                    ;*              16.67 MS/REV + OR - 2.5% IF 50HZ.
4144
4145                                    ;;*****************************************************************
4146  012360                            TST12:
4147  012360  000240                           NOP
4148  012362  033737  001450  001234           BIT     @#BITS+<12*2>,TSTNMS ;DO THIS TEST?
4149  012370  001002                           BNE     64$                  ;YES--BRANCH
4150  012372  000137  013116                   JMP     TST13                ;NO--GO TO THE NEXT TEST
4151  012376  012737  000012  001102  64$:     MOV     #12,@#$TSTNM         ;SET UP TEST NUMBER AND
4152                                                                         ;CLEAR THE ERROR FLAG ($ERFLG)
4153  012404  004737  024726                   JSR     PC,LODPRM            ;LOAD THE PARMETERS FOR THE TEST
4154  012410  012737  012360  001110           MOV     #TST12,@#$LPERR ;SETUP THE ERROR LOOP ADDRESS
4155  012416  013777  001102  166516           MOV     $TSTNM,@DISPLAY  ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
4156  012424  013737  001506  001204           MOV     @#RPT,$TIMES         ;GET THE ITERATION COUNT
4157  012432  112737  000031  001115           MOVB    #25.,$ERMAX          ;MAX ERRORS ALLOWED FOR TEST
4158  012440  005737  001244                   TST     @#CLKSTA             ;KW11-P CLOCK?
4159  012444  003002                           BGT     1$                   ;YES--START TEST
4160  012446  000137  013116                   JMP     TST13                ;NO--GO TO NEXT TEST
4161  012452  012737  012452  001106  1$:      MOV     #1$,$LPADR           ;SETUP LOOP ADDRESS
4162  012460  004037  027006                   JSR     R0,@#SRCH00          ;DO A MASSBUS INIT & RECAL
4163  012464  000402                           BR      2$                   ;RETURN HERE IF NO ERROR
4164  012466  000137  013114                   JMP     EXIT12               ;RETURN HERE IF ERROR
4165  012472  013764  001510  000034  2$:      MOV     @#FC,RPCA(R4)        ;FC
4166  012500  013746  001524                   MOV     @#FS,-(SP)           ;FS
4167  012504  113766  001516  000001           MOVB    @#FT,1(SP)           ;FT
4168  012512  012664  000006                   MOV     (SP)+,RPDA(R4)       ;LOAD FT/FS
```

```
4169  012516  012737  013114  001206          MOV     #EXIT12,$ESCAPE ;;ESCAPE TO EXIT12 ON ERROR
4170  012524  005005                          CLR     R5              ;COUNT UP
4171  012526  012703  002774                  MOV     #T7A,R3         ;60HZ PARAMETERS
4172  012532  032737  000100  001220          BIT     #SW06,@#C.SWR   ;60 HZ?
4173  012540  001402                          BEQ     TEST12          ;YES--BRANCH
4174  012542  012703  003004                  MOV     #T7B,R3         ;NO--50 HZ PARAMETERS
4175  012546  012706  001100          TEST12: MOV     #STACK,SP       ;SETUP STACK
4176  012552  012701  000012                  MOV     #10.,R1         ;TIME 10 SEARCHES
4177  012556  004737  027172                  JSR     PC,@#STRTMR     ;INITIALIZE THE TIMERS
4178  012562  012777  012770  166604          MOV     #7$,@PKV        ;SETUP VECTOR IN CASE OF OVERFLOW
4179  012570  012777  027170  022036          MOV     #DORT1,@RPVEC   ;SETUP RP04/5/6 VECTOR
4180  012576  005077  166600          1$:     CLR     @PKB            ;START COUNTING AT ZERO
4181  012602  012777  000131  166570          MOV     #131,@PKCS      ;INT.EN., COUNT UP AT 100KHZ
4182  012610  012714  000131                  MOV     #SEARCH,(R4)    ;START A SEARCH
4183  012614  000001                          WAIT                    ;WAIT ON INTERRUPT
4184  012616  042777  000101  166554          BIC     #101,@PKCS      ;STOP THE CLOCK
4185  012624  032764  040000  000012          BIT     #BIT14,RPDS1(R4) ;ERROR?
4186  012632  001415                          BEQ     2$              ;NO--BRANCH
4187  012634  104412                          SAVREG                  ;SAVE R0-R5
4188  012636  012702  004164                  MOV     #DTADPB,R2      ;DPB POINTER
4189  012642  004737  042202                  JSR     PC,@#SVRH11     ;SAVE ALL THE RH11/RP04 REGISTERS
4190  012646  012764  000040  000010          MOV     #BIT05,RPCS2(R4) ;MASSBUS CLEAR
4191  012654  013764  004164  000010          MOV     @#DTADPB,RPCS2(R4) ;SELECT DRIVE
4192  012662  104413                          RESREG                  ;RESTORE R0-R5
4193  012664  104017                          ERROR   17
4194  012666  005077  166510          2$:     CLR     @PKB            ;START THE COUNT AT ZERO
4195  012672  012714  000131                  MOV     #SEARCH,(R4)    ;START A SEARCH
4196  012676  012777  000131  166474          MOV     #131,@PKCS      ;START THE CLOCK
4197  012704  000001                          WAIT                    ;WAIT ON INTERRUPT
4198  012706  042777  000101  166464          BIC     #101,@PKCS      ;STOP THE CLOCK
4199  012714  032764  040000  000012          BIT     #BIT14,RPDS1(R4) ;IS "ERR=1"?
4200  012722  001415                          BEQ     3$              ;NO--BRANCH
4201  012724  104412                          SAVREG                  ;SAVE R0-R5
4202  012726  012702  004164                  MOV     #DTADPB,R2      ;DPB POINTER
4203  012732  004737  042202                  JSR     PC,@#SVRH11     ;SAVE ALL THE RH11/RP04 REGISTERS
4204  012736  012764  000040  000010          MOV     #BIT05,RPCS2(R4) ;MASSBUS CLEAR
4205  012744  013764  004164  000010          MOV     @#DTADPB,RPCS2(R4) ;SELECT DRIVE
4206  012752  104413                          RESREG                  ;RESTORE R0-R5
4207  012754  104017                          ERROR   17              ;DISK ERROR OCCURRED
4208  012756  004737  027236          3$:     JSR     PC,@#COUNT      ;UPDATE THE COUNT
4209  012762  005301                          DEC     R1              ;DONE?
4210  012764  003304                          BGT     1$              ;NO--BRANCH
4211  012766  000424                          BR      8$              ;YES--GO TO THE EXIT
4212  012770  042777  000101  166402  7$:     BIC     #101,@PKCS      ;STOP THE CLOCK
4213  012776  005037  177776                  CLR     @#PS            ;DROP THE PRIORITY
4214  013002  012600                          MOV     (SP)+,R0        ;PC OF WAIT+2
4215  013004  005726                          TST     (SP)+           ;POP THE PS FROM THE STACK
4216  013006  104412                          SAVREG                  ;SAVE R0-R5
4217  013010  012702  004164                  MOV     #DTADPB,R2      ;DPB POINTER
4218  013014  004737  042202                  JSR     PC,@#SVRH11     ;SAVE ALL THE RH11/RP04 REGISTERS
4219  013020  012764  000040  000010          MOV     #BIT05,RPCS2(R4) ;MASSBUS CLEAR
4220  013026  013764  004164  000010          MOV     @#DTADPB,RPCS2(R4) ;SELECT DRIVE
4221  013034  104413                          RESREG                  ;RESTORE R0-R5
4222  013036  104020                          ERROR   20              ;CLOCK OVERFLOWED
4223  013040                          8$:
4224  013040  012764  000040  000010          MOV     #BIT05,RPCS2(R4) ;MASSBUS INIT.
```

```
 4225  013046  013764  004164  000010          MOV     @#DTADPB,RPCS2(R4) ;SELECT DRIVE
 4226  013054  004737  024212                  JSR     PC,@#ST.CLK        ;INITIALIZE THE CLOCK
 4227  013060  012777  037544  021546          MOV     #ISR,@RPVEC        ;RESTORE RH11/RP04/5/6 INT. VECTOR
 4228  013066  032737  000100  001220          BIT     #SW06,@#C.SWR      ;60 HZ?
 4229  013074  001004                          BNE     9$                 ;NO -- BRANCH
 4230  013076  004037  027370                  JSR     R0,@#TYPTIM        ;GO TYPE THE TIMES
 4231  013102  002774                          T7A                        ;POINTER
 4232  013104  000403                          BR      EXIT12             ;GO TO EXIT
 4233  013106                          9$:
 4234  013106  004037  027370                  JSR     R0,@#TYPTIM        ;GO TYPE THE TIMES
 4235  013112  003004                          T7B                        ;POINTER
 4236  013114  000004                  EXIT12: SCOPE                      ;LOOP ?
 4237
 4238
 4239                                  ;;*****************************************************************
 4240                                  ;*TEST 13       ONE CYLINDER SEEK TIMING TEST
 4241
 4242                                  ;*      THIS TEST WILL COMMAND FORWARD SEEK CYCLES TO ADVANCE THE
 4243                                  ;*      CYLINDER BY ONE UNTIL THE INCREMENT IS GREATER THAN THE
 4244                                  ;*      CYLINDER 'LC', THEN REVERSE SEEK TO CYLINDER 'FC'.  THE
 4245                                  ;*      TIME TO PERFORM EACH SEEK IS CHECKED TO ENSURE IT DOES NOT
 4246                                  ;*      EXCEED THE MAXIMUM TIME PERMITTED FOR A ONE CYLINDER SEEK.
 4247                                  ;*      THE TIME MUST BE LESS THAN 10MS.
 4248
 4249                                  ;;*****************************************************************
 4250  013116                          TST13:
 4251  013116  000240                          NOP
 4252  013120  033737  001452  001234          BIT     @#BITS+<13*2>,TSTNMS ;DO THIS TEST?
 4253  013126  001002                          BNE     64$                ;YES--BRANCH
 4254  013130  000137  013562                  JMP     TST14              ;NO--GO TO THE NEXT TEST
 4255  013134  012737  000013  001102  64$:    MOV     #13,@#$TSTNM       ;SET UP TEST NUMBER AND
 4256                                                                     ;CLEAR THE ERROR FLAG ($ERFLG)
 4257  013142  004737  024726                  JSR     PC,LODPRM          ;LOAD THE PARMETERS FOR THE TEST
 4258  013146  012737  013116  001110          MOV     #TST13,@#$LPERR    ;SETUP THE ERROR LOOP ADDRESS
 4259  013154  013777  001102  165760          MOV     $TSTNM,@DISPLAY    ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
 4260  013162  013737  001506  001204          MOV     @#RPT,$TIMES       ;GET THE ITERATION COUNT
 4261  013170  112737  000031  001115          MOVB    #25.,$ERMAX        ;MAX ERRORS ALLOWED FOR TEST
 4262  013176  005737  001244                  TST     @#CLKSTA           ;KW11-P CLOCK?
 4263  013202  003002                          BGT     1$                 ;YES--START TEST
 4264  013204  000137  013562                  JMP     TST14              ;NO--GO TO NEXT TEST
 4265  013210  012737  013210  001106  1$:     MOV     #1$,$LPADR         ;SETUP THE LOOP ADDRESS
 4266  013216  004037  027006                  JSR     R0,@#SRCH00        ;DO A MASSBUS INIT. AND RECAL
 4267  013222  000402                          BR      2$                 ;NO ERROR RETURN
 4268  013224  000137  013560                  JMP     EXIT13             ;ERROR RETURN--SCOPE LOOP CALL
 4269  013230  012703  003014          2$:     MOV     #T10,R3            ;PARAMETER POINTER
 4270  013234  012737  013560  001206          MOV     #EXIT13,$ESCAPE    ;;ESCAPE TO EXIT13 ON ERROR
 4271  013242  012706  001100          TEST13: MOV     #STACK,SP          ;SETUP STACK
 4272  013246  013737  001510  004176          MOV     FC,@#DTADPB+12     ;START WITH BEGINNING CYLINDER
 4273  013254  005237  004176                  INC     DTADPB+12          ;INCREMENT THE BEGINNING CYLINDER
 4274  013260  005005                          CLR     R5                 ;SET THE UP/DOWN SWITCH TO UP
 4275  013262  004737  027172                  JSR     PC,@#STRTMR        ;INITIALIZE THE TIMERS
 4276  013266  012777  013454  166100          MOV     #7$,@PKV           ;SETUP INCASE OF OVERFLOW
 4277  013274  012777  027170  021332          MOV     #DORTI,@RPVEC      ;SET RP04/5/6 VECTOR
 4278  013302  005077  166074          1$:     CLR     @PKB               ;START THE COUNTER AT ZERO
 4279  013306  013764  004176  000034          MOV     @#DTADPB+12,RPCA(R4) ;LOAD DESIRED CYLINDER
 4280  013314  012714  000105                  MOV     #SEEK,(R4)         ;START A SEEK
```

```
4281  013320  012777  000131  166052          MOV     #131,@PKCS      ;START THE CLOCK
4282  013326  000001                          WAIT                    ;WAIT ON INTERRUPT
4283  013330  042777  000101  166042          BIC     #101,@PKCS      ;STOP THE CLOCK
4284  013336  032764  040000  000012          BIT     #BIT14,RPDS1(R4) ;ANY DISK ERRORS?
4285  013344  001415                          BEQ     2$              ;NO--BRANCH
4286  013346  104412                          SAVREG                  ;SAVE R0-R5
4287  013350  012702  004164                  MOV     #DTADPB,R2      ;DPB POINTER
4288  013354  004737  042202                  JSR     PC,@#SVRH11     ;SAVE ALL THE RH11/RP04 REGISTERS
4289  013360  012764  000040  000010          MOV     #BIT05,RPCS2(R4) ;MASSBUS CLEAR
4290  013366  013764  004164  000010          MOV     @#DTADPB,RPCS2(R4) ;SELECT DRIVE
4291  013374  104413                          RESREG                  ;RESTORE R0-R5
4292  013376  104017                          ERROR   17              ;REPORT THE ERROR
4293  013400  004737  027236          2$:     JSR     PC,@#COUNT      ;COUNT THIS SEEKS TIME
4294  013404  004737  026564                  JSR     PC,@#TWOMS      ;STALL FOR 2 MILLISECONDS
4295  013410  005705                          TST     R5              ;UP OR DOWN?
4296  013412  001011                          BNE     4$              ;DOWN--BRANCH
4297  013414  005237  004176          3$:     INC     @#DTADPB+12     ;MOVE TO NEXT CYLINDER
4298  013420  023737  004176  001512          CMP     @#DTADPB+12,LC ;OUT OF CYLINDERS?
4299  013426  002725                          BLT     1$              ;NO--GO DO THE NEXT SEEK
4300  013430  012705  177777                  MOV     #-1,R5          ;SET UP/DOWN SWITCH TO DOWN
4301  013434  000722                          BR      1$              ;GO DO THE NEXT SEEK
4302  013436  005337  004176          4$:     DEC     @#DTADPB+12     ;MOVE TO NEXT CYLINDER
4303  013442  023727  004176  000000          CMP     @#DTADPB+12,#0  ;OUT OF CYLINDERS?
4304  013450  003314                          BGT     1$              ;NO--GO DO THE NEXT SEEK
4305  013452  000424                          BR      8$              ;GO TO THE EXIT
4306  013454  042777  000101  165716  7$:     BIC     #101,@PKCS      ;STOP THE CLOCK
4307  013462  005037  177776                  CLR     @#PS            ;DROP THE PRIORITY
4308  013466  012600                          MOV     (SP)+,R0        ;PC OF WAIT+2
4309  013470  005726                          TST     (SP)+           ;POP THE PS FROM THE STACK
4310  013472  104412                          SAVREG                  ;SAVE R0-R5
4311  013474  012702  004164                  MOV     #DTADPB,R2      ;DPB POINTER
4312  013500  004737  042202                  JSR     PC,@#SVRH11     ;SAVE ALL THE RH11/RP04 REGISTERS
4313  013504  012764  000040  000010          MOV     #BIT05,RPCS2(R4) ;MASSBUS CLEAR
4314  013512  013764  004164  000010          MOV     @#DTADPB,RPCS2(R4) ;SELECT DRIVE
4315  013520  104413                          RESREG                  ;RESTORE R0-R5
4316  013522  104020                          ERROR   20              ;REPORT CLOCK OVERFLOW
4317  013524                          8$:
4318  013524  012764  000040  000010          MOV     #BIT05,RPCS2(R4) ;MASSBUS INIT.
4319  013532  013764  004164  000010          MOV     @#DTADPB,RPCS2(R4) ;SELECT DRIVE
4320  013540  004737  024212                  JSR     PC,@#ST.CLK     ;INITIALIZE THE CLOCK
4321  013544  012777  037544  021062          MOV     #ISR,@RPVEC     ;RESTORE RH11/RP04/5/6 INT. VECTOR
4322  013552  004037  027370                  JSR     R0,@#TYPTIM     ;GO TYPE THE TIMES
4323  013556  003014                          T10                     ;POINTER
4324  013560  000004                  EXIT13: SCOPE                   ;LOOP ?
4325
4326
4327                                  ;;*****************************************************************
4328                                  ;*TEST 14       ACCESS TIME MEASURMENT TEST
4329
4330                                  ;*      THIS TEST WILL COMMAND A FORWARD SEEK FROM CYLINDER 0 TO
4331                                  ;*      CYLINDER 'LC', THEN A REVERSEK FROM CYLINDER 'LC' TO
4332                                  ;*      CYLINDER 0.  BOTH SEEKS ARE TIMED AND CHECKED TO ENSURE THEY
4333                                  ;*      ARE WITHIN THE TOLERANCE ALLOWED FOR THE ACCESS TIME MEASURMENT.
4334                                  ;*      THIS SEQUENCE IS REPEATED 128 TIMES (FOR A TOTAL
4335                                  ;*      OF 256 SEEKS).  THE AVERAGE ACCESS TIME MUST BE LESS THAN 30 MS.
4336                                  ;*      CYLINDER 'LC' DEFAULTS TO 136 (10) FOR AN RP04/5 OR TO 255 (10)
```

```
                                        ;*      FOR AN RP06.
                                        ;*
                                        ;;*****************************************************************
013562                                  TST14:
013562  000240                                  NOP
013564  033737  001454  001234                  BIT     @#BITS+<14+2>,TSTNMS ;DO THIS TEST?
013572  001002                                  BNE     64$                 ;YES--BRANCH
013574  000137  014300                          JMP     TST15               ;NO--GO TO THE NEXT TEST
013600  012737  000014  001102  64$:            MOV     #14,@#$TSTNM        ;SET UP TEST NUMBER AND
                                                                            ;CLEAR THE ERROR FLAG ($ERFLG)
013606  004737  024726                          JSR     PC,LODPRM           ;LOAD THE PARMETERS FOR THE TEST
013612  012737  013562  001110                  MOV     #TST14,@#$LPERR     ;SETUP THE ERROR LOOP ADDRESS
013620  013777  001102  165314                  MOV     $TSTNM,@DISPLAY     ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
013626  013737  001506  001204                  MOV     @#RPT,$TIMES        ;GET THE ITERATION COUNT
013634  112737  000031  001115                  MOVB    #25.,$ERMAX         ;MAX ERRORS ALLOWED FOR TEST
013642  005737  001244                          TST     @#CLKSTA            ;KW11-P CLOCK?
013646  003002                                  BGT     1$                  ;YES--START TEST
013650  000137  014300                          JMP     TST15               ;NO--GO TO NEXT TEST
013654  012737  013654  001106  1$:             MOV     #1$,$LPADR          ;SET THE LOOP ADDRESS
013662  004037  027006                          JSR     R0,@#SRCH00         ;DO A MASSBUS INIT & RECAL
013666  000402                                  BR      2$                  ;RETURN HERE IF NO ERROR
013670  000137  014276                          JMP     EXIT14              ;RETURN HERE ON ERROR
013674  012703  003024          2$:             MOV     #T11,R3             ;PARAMETER POINTER
013700  012737  014276  001206                  MOV     #EXIT14,$ESCAPE     ;;ESCAPE TO EXIT14 ON ERROR
013706  012706  001100          TEST14: MOV     #STACK,SP           ;SETUP STACK
013712  012701  000200                          MOV     #128.,R1            ;REPEAT "0-136-0" 128 TIMES
013716  004737  027172                          JSR     PC,@#STRTMR         ;INIT. THE COUNTERS
013722  012777  014172  165444                  MOV     #7$,@PKV            ;SET UP VECTOR IN CASE OF OVERFLOW
013730  012777  027170  020676                  MOV     #DORTI,@RPVEC       ;SETUP RP04/5/6 VECTOR
013736  005077  165440          1$:             CLR     @PKB                ;START COUNT AT ZERO
013742  013764  001512  000034                  MOV     LC,RPCA(R4)         ;'MIDDLE' CYLINDER
013750  012764  000105  000000                  MOV     #SEEK,RPCS1(R4)     ;START A SEEK
013756  012777  000131  165414                  MOV     #131,@PKCS          ;START THE CLOCK
013764  000001                                  WAIT                        ;WAIT ON INTERRUPT
013766  042777  000101  165404                  BIC     #101,@PKCS          ;STOP CLOCK
013774  032764  040000  000012                  BIT     #BIT14,RPDS1(R4)    ;ERR=1?
014002  001415                                  BEQ     2$                  ;NO--BRANCH
014004  104412                                  SAVREG                      ;SAVE R0-R5
014006  012702  004164                          MOV     #DTADPB,R2          ;DPB POINTER
014012  004737  042202                          JSR     PC,@#SVRH11         ;SAVE ALL THE RH11/RP04 REGISTERS
014016  012764  000040  000010                  MOV     #BIT05,RPCS2(R4)    ;MASSBUS CLEAR
014024  013764  004164  000010                  MOV     @#DTADPB,RPCS2(R4)  ;SELECT DRIVE
014032  104413                                  RESREG                      ;RESTORE R0-R5
014034  104017                                  ERROR   17
014036  005005                          2$:     CLR     R5                  ;SET UP/DOWN SWITCH TO UP
014040  004737  027236                          JSR     PC,@#COUNT          ;UPDATE THE COUNT
014044  004737  026564                          JSR     PC,@#TWOMS          ;STALL FOR 2 MILLISECONDS
014050  005077  165326                          CLR     @PKB                ;START THE COUNT AT ZERO
014054  013764  001510  000034                  MOV     FC,RPCA(R4)         ;BEGINNING CYLINDER
014062  012764  000105  000000                  MOV     #SEEK,RPCS1(R4)     ;START A SEEK
014070  012777  000131  165302                  MOV     #131,@PKCS          ;START THE CLOCK
014076  000001                                  WAIT                        ;WAIT ON INTERRUPT
014100  042777  000101  165272                  BIC     #101,@PKCS          ;STOP THE CLOCK
014106  032764  040000  000012                  BIT     #BIT14,RPDS1(R4)    ;ERR=1?
014114  001415                                  BEQ     3$                  ;NO--BRANCH
014116  104412                                  SAVREG                      ;SAVE R0-R5
```

```
4393  014120  012702  004164                    MOV     #DTADPB,R2      ;DPB POINTER
4394  014124  004737  042202                    JSR     PC,@#SVRH11     ;SAVE ALL THE RH11/RP04 REGISTERS
4395  014130  012764  000040  000010            MOV     #BIT05,RPCS2(R4) ;MASSBUS CLEAR
4396  014136  013764  004164  000010            MOV     @#DTADPB,RPCS2(R4) ;SELECT DRIVE
4397  014144  104413                            RESREG                  ;RESTORE R0-R5
4398  014146  104017                            ERROR   17
4399  014150  012705  177777          3$:       MOV     #-1,R5          ;SET UP/DOWN SWITCH TO DOWN
4400  014154  004737  027236                    JSR     PC,@#COUNT      ;UPDATE THE COUNT
4401  014160  004737  026564                    JSR     PC,@#TWOMS      ;STALL FOR 2 MILLISECONDS
4402  014164  005301                            DEC     R1              ;DONE?
4403  014166  003263                            BGT     1$              ;NO--BRANCH
4404  014170  000424                            BR      8$              ;YES--EXIT
4405  014172  042777  000101  165200  7$:       BIC     #101,@PKCS      ;STOP THE CLOCK
4406  014200  005037  177776                    CLR     @#PS            ;DROP THE PRIORITY
4407  014204  012600                            MOV     (SP)+,R0        ;PC OF WAIT+2
4408  014206  005726                            TST     (SP)+           ;POP THE PS FROM THE STACK
4409  014210  104412                            SAVREG                  ;SAVE R0-R5
4410  014212  012702  004164                    MOV     #DTADPB,R2      ;DPB POINTER
4411  014216  004737  042202                    JSR     PC,@#SVRH11     ;SAVE ALL THE RH11/RP04 REGISTERS
4412  014222  012764  000040  000010            MOV     #BIT05,RPCS2(R4) ;MASSBUS CLEAR
4413  014230  013764  004164  000010            MOV     @#DTADPB,RPCS2(R4) ;SELECT DRIVE
4414  014236  104413                            RESREG                  ;RESTORE R0-R5
4415  014240  104020                            ERROR   20              ;CLOCK OVERFLOWED
4416  014242                          8$:
4417  014242  012764  000040  000010            MOV     #BIT05,RPCS2(R4) ;MASSBUS INIT.
4418  014250  013764  004164  000010            MOV     @#DTADPB,RPCS2(R4) ;SELECT DRIVE
4419  014256  004737  024212                    JSR     PC,@#ST.CLK     ;INITIALIZE THE CLOCK
4420  014262  012777  037544  020344            MOV     #ISR,@RPVEC     ;RESTORE RH11/RP04/5/6 INT. VECTOR
4421  014270  004037  027370                    JSR     R0,@#TYPTIM     ;GO TYPE THE TIMES
4422  014274  003024                            T11                     ;POINTER
4423  014276  000004                  EXIT14:   SCOPE                   ;LOOP ?
4424
4425                                            ;;*********************************************************************
4426                                            ;*TEST 15       MAXIMUM SEEK TIMING TEST
4427
4428                                            ;*      THIS TEST WILL COMMAND A FORWARD SEEK FROM CYLINDER 0 TO
4429                                            ;*      CYLINDER 'LC', THEN A REVERSE SEEK FROM CYLINDER 'LC' TO
4430                                            ;*      CYLINDER 0.  BOTH SEEKS ARE TIMED AND CHECKED TO ENSURE
4431                                            ;*      THEY ARE WITHIN THE TOLERANCE ALLOWED FOR THE MAXIMUM SEEK
4432                                            ;*      TIME.  THIS SEQUENCE IS REPEATED 128 TIMES (FOR
4433                                            ;*      A TOTAL OF 256 SEEKS).  THE MAXIMUM SEEK TIME MUST BE LESS THAN
4434                                            ;*      54 MS.  'LC' DEFAULTS TO 410 (10) FOR RP04/5'S AND TO 814 (10)
4435                                            ;*      FOR RP06'S.
4436
4437                                            ;;*********************************************************************
4438  014300                          TST15:
4439  014300  000240                            NOP
4440  014302  033737  001456  001234            BIT     @#BITS+<15*2>,TSTNMS ;DO THIS TEST?
4441  014310  001002                            BNE     64$             ;YES--BRANCH
4442  014312  000137  015016                    JMP     TST16           ;NO--GO TO THE NEXT TEST
4443  014316  012737  000015  001102  64$:     MOV     #15,@#$TSTNM    ;SET UP TEST NUMBER AND
4444                                                                    ;CLEAR THE ERROR FLAG ($ERFLG)
4445  014324  004737  024726                    JSR     PC,LODPRM       ;LOAD THE PARMETERS FOR THE TEST
4446  014330  012737  014300  001110            MOV     #TST15,@#$LPERR ;SETUP THE ERROR LOOP ADDRESS
4447  014336  013777  001102  164576            MOV     $TSTNM,@DISPLAY  ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
4448  014344  013737  001506  001204            MOV     @#RPT,$TIMES    ;GET THE ITERATION COUNT
```

```
 4449  014352  112737  000031  001115          MOVB    #25.,$ERMAX      ;MAX ERRORS ALLOWED FOR TEST
 4450  014360  005737  001244                  TST     @#CLKSTA         ;KW11-P CLOCK
 4451  014364  003002                          BGT     1$               ;YES--START TEST
 4452  014366  000137  015016                  JMP     TST16            ;NO--GO TO NEXT TEST
 4453  014372  012737  014372  001106  1$:     MOV     #1$,$LPADR       ;SETUP THE LOOP ADDRESS
 4454  014400  004037  027006                  JSR     R0,@#SRCH00      ;DO A MASSBUS INIT & RECAL
 4455  014404  000402                          BR      2$               ;RETURN HERE IF NO ERROR
 4456  014406  000137  015014                  JMP     EXIT15           ;RETURN HERE ON ERROR
 4457  014412  012703  003034          2$:     MOV     #T12,R3          ;PARAMETER POINTER
 4458  014416  012737  015014  001206          MOV     #EXIT15,$ESCAPE  ;;ESCAPE TO EXIT15 ON ERROR
 4459  014424  012706  001100          TEST15: MOV     #STACK,SP        ;SETUP STACK
 4460  014430  012701  000200                  MOV     #128.,R1         ;REPEAT "0-'LC'-0" 128 TIMES
 4461  014434  004737  027172                  JSR     PC,@#STRTMR      ;INIT. THE TIMERS
 4462  014440  012777  014710  164726          MOV     #7$,@PKV         ;SETUP VECTOR IN CASE OF OVERFLOW
 4463  014446  012777  027170  020160          MOV     #DORTI,@RPVEC    ;SETUP RP04/5/6 VECTOR
 4464  014454  005077  164722          1$:     CLR     @PKB             ;START COUNTING FROM ZERO
 4465  014460  013764  001512  000034          MOV     LC,RPCA(R4)      ;MAXIMUM CYLINDER
 4466  014466  012764  000105  000000          MOV     #SEEK,RPCS1(R4)  ;START A SEEK
 4467  014474  012777  000131  164676          MOV     #131,@PKCS       ;START THE CLOCK
 4468  014502  000001                          WAIT                     ;WAIT ON INTERRUPT
 4469  014504  042777  000101  164666          BIC     #101,@PKCS       ;STOP THE CLOCK
 4470  014512  032764  040000  000012          BIT     #BIT14,RPDS1(R4) ;ERR=1?
 4471  014520  001415                          BEQ     2$               ;NO--BRANCH
 4472  014522  104412                          SAVREG                   ;SAVE R0-R5
 4473  014524  012702  004164                  MOV     #DTADPB,R2       ;DPB POINTER
 4474  014530  004737  042202                  JSR     PC,@#SVRH11      ;SAVE ALL THE RH11/RP04 REGISTERS
 4475  014534  012764  000040  000010          MOV     #BIT05,RPCS2(R4) ;MASSBUS CLEAR
 4476  014542  013764  004164  000010          MOV     @#DTADPB,RPCS2(R4) ;SELECT DRIVE
 4477  014550  104413                          RESREG                   ;RESTORE R0-R5
 4478  014552  104017                          ERROR   17
 4479  014554  005005                  2$:     CLR     R5               ;SET THE UP/DOWN SWITCH TO UP
 4480  014556  004737  027236                  JSR     PC,@#COUNT       ;UP THE COUNT
 4481  014562  004737  026564                  JSR     PC,@#TWOMS       ;STALL FOR 2 MILLISECONDS
 4482  014566  005077  164610                  CLR     @PKB             ;START COUNT AT ZERO
 4483  014572  013764  001510  000034          MOV     FC,RPCA(R4)      ;BEGINNING CYLINDER
 4484  014600  012764  000105  000000          MOV     #SEEK,RPCS1(R4)  ;START A SEEK
 4485  014606  012777  000131  164564          MOV     #131,@PKCS       ;START THE CLOCK
 4486  014614  000001                          WAIT                     ;WAIT ON INTERRUPT
 4487  014616  042777  000101  164554          BIC     #101,@PKCS       ;STOP THE CLOCK
 4488  014624  032764  040000  000012          BIT     #BIT14,RPDS1(R4) ;"ERR"=1?
 4489  014632  001415                          BEQ     3$               ;NO--BRANCH
 4490  014634  104412                          SAVREG                   ;SAVE R0-R5
 4491  014636  012702  004164                  MOV     #DTADPB,R2       ;DPB POINTER
 4492  014642  004737  042202                  JSR     PC,@#SVRH11      ;SAVE ALL THE RH11/RP04 REGISTERS
 4493  014646  012764  000040  000010          MOV     #BIT05,RPCS2(R4) ;MASSBUS CLEAR
 4494  014654  013764  004164  000010          MOV     @#DTADPB,RPCS2(R4) ;SELECT DRIVE
 4495  014662  104413                          RESREG                   ;RESTORE R0-R5
 4496  014664  104017                          ERROR   17               ;REPORT THE ERROR
 4497  014666  012705  177777          3$:     MOV     #-1,R5           ;SET THE UP/DOWN SWITCH TO DOWN
 4498  014672  004737  027236                  JSR     PC,@#COUNT       ;UPDATE THE COUNT
 4499  014676  004737  026564                  JSR     PC,@#TWOMS       ;STALL FOR 2 MILLISECONDS
 4500  014702  005301                          DEC     R1               ;DONE?
 4501  014704  003263                          BGT     1$               ;NO--BRANCH
 4502  014706  000424                          BR      8$               ;YES--EXIT
 4503  014710  042777  000101  164462  7$:     BIC     #101,@PKCS       ;STOP THE CLOCK
 4504  014716  005037  177776                  CLR     @#PS             ;DROP THE PRIORITY
```

```
 4505  014722  012600                          MOV     (SP)+,R0        ;PC OF WAIT+2
 4506  014724  005726                          TST     (SP)+           ;POP THE PS FROM THE STACK
 4507  014726  104412                          SAVREG                  ;SAVE R0-R5
 4508  014730  012702  004164                  MOV     #DTADPB,R2      ;DPB POINTER
 4509  014734  004737  042202                  JSR     PC,@#SVRH11     ;SAVE ALL THE RH11/RP04 REGISTERS
 4510  014740  012764  000040  000010          MOV     #BIT05,RPCS2(R4) ;MASSBUS CLEAR
 4511  014746  013764  004164  000010          MOV     @#DTADPB,RPCS2(R4) ;SELECT DRIVE
 4512  014754  104413                          RESREG                  ;RESTORE R0-R5
 4513  014756  104020                          ERROR   20              ;CLOCK OVERFLOWED
 4514  014760                          8$:
 4515  014760  012764  000040  000010          MOV     #BIT05,RPCS2(R4) ;MASSBUS INIT.
 4516  014766  013764  004164  000010          MOV     @#DTADPB,RPCS2(R4) ;SELECT DRIVE
 4517  014774  004737  024212                  JSR     PC,@#ST.CLK     ;INITIALIZE THE CLOCK
 4518  015000  012777  037544  017626          MOV     #ISR,@RPVEC     ;RESTORE RH11/RP04/5/6 INT. VECTOR
 4519  015006  004037  027370                  JSR     R0,@#TYPTIM     ;GO TYPE THE TIMES
 4520  015012  003034                          T12                     ;POINTER
 4521  015014  000004                  EXIT15: SCOPE                   ;LOOP ?
```

```
4522
4523                                    .SBTTL  *** ADDRESSING TESTS ***
4524

        ;/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:
        ;/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:
        ;/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:

        ;*THE ADDRESSING TESTS ENSURES PROPER OPERATION OF THE TRACK
        ;*AND SECTOR ADDRESS CIRCUITRY.  BOTH ADDRESSING TESTS
        ;*WILL BE PERFORMED ON CYLINDER FC.

        ;/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*
        ;/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*
        ;/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*

4538
4539                                    ;;*****************************************************************
4540                                    ;*TEST 16       SECTOR ADDRESSING TEST
4541
4542                                    ;*      THIS TEST WRITES DATA INTO ALL SECTORS OF TRACK "FT".  THE
4543                                    ;*      DATA WILL BE 256 WORDS OF THE SECTOR ADDRESS OF THE SECTOR
4544                                    ;*      BEING WRITTEN.  A WRITE CHECK IS PERFORMED, THE BUFFER IS
4545                                    ;*      CLEARED (TO 177400) AND THE DATA IS READ AND COMPARED.  THEN SECTOR 0
4546                                    ;*      IS REWRITTEN AND SECTORS 0 - 21 ARE WRITE CHECKED.  THEN
4547                                    ;*      SECTOR 1 IS REWRITTEN AND SECTORS 0 - 21 ARE WRITE CHECKED.
4548                                    ;*      THIS REWRITE AND WRITE CHECK PROCEDURE IS CONTINUED UP THROUGH
4549                                    ;*      REWRITE SECTOR 21 AND WRITE CHECK SECTORS 0-21.
4550
4551                                    ;;*****************************************************************
4552  015016                            TST16:
4553  015016  000240                           NOP
4554  015020  033737  001460  001234           BIT     @#BITS+<16*2>,TSTNMS ;DO THIS TEST?
4555  015026  001002                           BNE     64$             ;YES--BRANCH
4556  015030  000137  015414                   JMP     TST17           ;NO--GO TO THE NEXT TEST
4557  015034  012737  000016  001102  64$:     MOV     #16,@#$TSTNM    ;SET UP TEST NUMBER AND
4558                                                                    ;CLEAR THE ERROR FLAG ($ERFLG)
4559  015042  004737  024726                   JSR     PC,LODPRM       ;LOAD THE PARMETERS FOR THE TEST
4560  015046  012737  015112  001110           MOV     #TEST16,@#$LPERR  ;SETUP THE LOOP ON ERROR ADDRESS
4561  015054  013777  001102  164060           MOV     $TSTNM,@DISPLAY  ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
4562  015062  013737  001506  001204           MOV     @#RPT,$TIMES    ;GET THE ITERATION COUNT
4563  015070  113737  001364  001115           MOVB    ERR.CT,$ERMAX   ;MAX ERRORS ALLOWED FOR TEST
4564  015076  012737  015412  001252           MOV     #EXIT16,@#BYPASS
4565  015104  012737  015112  001106           MOV     #TEST16,$LPADR  ;SETUP THE LOOP ADDRESS
4566  015112  012706  001100          TEST16:  MOV     #STACK,SP       ;SET THE STACK POINTER
4567  015116  004737  030022                   JSR     PC,@#FILBUF     ;FILL THE BUFFER WITH SECTOR ADDRESSES
4568  015122  013737  001510  004176           MOV     @#FC,@#DTADPB+12 ;CYLINDER
4569  015130  113737  001516  004175           MOVB    @#FT,@#DTADPB+11 ;TRACK
4570  015136  105037  004174                   CLRB    @#DTADPB+10     ;SECTOR
4571  015142  013737  001352  004170           MOV     TRCKWC,@#DTADPB+4 ;WORD COUNT
4572  015150  012737  050202  004172           MOV     #BUFFER,@#DTADPB+6 ;BUFFER ADDRESS
4573  015156  012737  015156  001110           MOV     #.,$LPERR       ;SETUP THE ERROR LOOP ADDRESS
4574  015164  012706  001100                   MOV     #STACK,SP       ;LOAD THE STACK POINTER
4575  015170  012737  000161  004166           MOV     #WRITE,@#DTADPB+2 ;COMMAND=WRITE DATA
4576  015176  004037  025642                   JSR     R0,@#DRVCAL     ;START A DATA TRANSFER
4577  015202  012737  000151  004166           MOV     #WRCKD,@#DTADPB+2 ;COMMAND=WRITE CHECK DATA
```

```
 4578  015210  012737  015210  001110          MOV     #.,$LPERR       ;SETUP THE ERROR LOOP ADDRESS
 4579  015216  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
 4580  015222  004037  025642                  JSR     R0,@#DRVCAL     ;START A DATA TRANSFER
 4581  015226  012737  015226  001110          MOV     #.,$LPERR       ;SETUP THE ERROR LOOP ADDRESS
 4582  015234  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
 4583  015240  004037  030060                  JSR     R0,@#CLRBUF     ;CLEAR BUFFER
 4584  015244  012737  000171  004166          MOV     #READ,@#DTADPB+2 ;COMMAND = READ
 4585  015252  004037  025642                  JSR     R0,@#DRVCAL     ;START A DATA TRANSFER
 4586  015256  004037  030126                  JSR     R0,@#CKSCTR     ;CHECK THE SECTOR DATA READ
 4587  015262  012700  050202                  MOV     #BUFFER,R0      ;BUFFER ADDRESS
 4588  015266  005001                          CLR     R1              ;FIRST SECTOR
 4589  015270  012737  015270  001110          MOV     #.,$LPERR       ;SETUP THE ERROR LOOP ADDRESS
 4590  015276  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
 4591  015302  012737  000161  004166  1$:     MOV     #WRITE,@#DTADPB+2 ;COMMAND=WRITE DATA
 4592  015310  012737  177400  004170          MOV     #SCTRWC,@#DTADPB+4 ;WORD COUNT
 4593  015316  010037  004172                  MOV     R0,@#DTADPB+6   ;BUFFER ADDRESS
 4594  015322  110137  004174                  MOVB    R1,@#DTADPB+10  ;SECTOR
 4595  015326  004037  025642                  JSR     R0,@#DRVCAL     ;START A DATA TRANSFER
 4596  015332  012737  015332  001110          MOV     #.,$LPERR       ;SETUP THE ERROR LOOP ADDRESS
 4597  015340  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
 4598  015344  012737  000151  004166          MOV     #WRCKD,@#DTADPB+2 ;COMMAND=WRITE CHECK DATA
 4599  015352  013737  001352  004170          MOV     TRCKWC,@#DTADPB+4 ;WORD COUNT
 4600  015360  012737  050202  004172          MOV     #BUFFER,@#DTADPB+6 ;BUFFER ADDRESS
 4601  015366  105037  004174                  CLRB    @#DTADPB+10     ;SECTOR
 4602  015372  004037  025642                  JSR     R0,@#DRVCAL     ;START A DATA TRANSFER
 4603  015376  062700  001000                  ADD     #512.,R0        ;MOVE TO NEXT SECTOR
 4604  015402  005201                          INC     R1
 4605  015404  023701  001630                  CMP     PRMLMT+22,R1    ;DONE?
 4606  015410  103334                          BHIS    1$              ;NO--BRANCH
 4607  015412  000004                  EXIT16: SCOPE                   ;LOOP ?
 4608
 4609                                          ;;*******************************************************************
 4610                                          ;*TEST 17       TRACK ADDRESSING TEST
 4611
 4612                                          ;*      THIS TEST WILL WRITE DATA IN THE FORM OF TRACK ADDRESSES
 4613                                          ;*      IN CYLINDER "FC" SECTOR "FS" OF EVERY TRACK WITH EACH TRACK
 4614                                          ;*      GETTING ITS OWN TRACK ADDRESS.
 4615                                          ;*      A WRITE CHECK IS THEN PERFORMED ON EACH TRACK TO ENSURE
 4616                                          ;*      THE DATA IS VALID.  THEN TRACK 0 IS REWRITTEN AND TRACK 1
 4617                                          ;*      THROUGH TRACK 18 IS WRITE CHECKED.  THEN TRACK 1 IS
 4618                                          ;*      REWRITTEN AND TRACK 2 THROUGH TRACK 18 IS WRITE CHECKED.
 4619                                          ;*      THIS PROCEDURE IS CONTINUED UP THROUGH REWRITING TRACK 17
 4620                                          ;*      AND WRITE CHECKING TRACK 18.
 4621
 4622                                          ;;*******************************************************************
 4623  015414                          TST17:
 4624  015414  000240                          NOP
 4625  015416  033737  001462  001234          BIT     @#BITS+<17*2>,TSTNMS ;DO THIS TEST?
 4626  015424  001002                          BNE     64$             ;YES--BRANCH
 4627  015426  000137  016034                  JMP     TST20           ;NO--GO TO THE NEXT TEST
 4628  015432  012737  000017  001102  64$:     MOV     #17,@#$TSTNM    ;SET UP TEST NUMBER AND
 4629                                                                  ;CLEAR THE ERROR FLAG ($ERFLG)
 4630  015440  004737  024726                  JSR     PC,LODPRM       ;LOAD THE PARMETERS FOR THE TEST
 4631  015444  012737  015510  001110          MOV     #TEST17,@#$LPERR  ;SETUP THE LOOP ON ERROR ADDRESS
 4632  015452  013777  001102  163462          MOV     $TSTNM,@DISPLAY  ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
 4633  015460  013737  001506  001204          MOV     @#RPT,$TIMES    ;GET THE ITERATION COUNT
```

```
 4634  015466  113737  001364  001115          MOVB    ERR.CT,$ERMAX   ;MAX ERRORS ALLOWED FOR TEST
 4635  015474  012737  016032  001252          MOV     #EXIT17,@#BYPASS
 4636  015502  012737  015510  001106          MOV     #TEST17,$LPADR  ;SETUP THE LOOP ADDRESS
 4637  015510  012706  001100          TEST17: MOV     #STACK,SP       ;SET THE STACK POINTER
 4638  015514  004737  030022                  JSR     PC,@#FILBUF     ;FILL THE BUFFER WITH TRACK ADDRESS
 4639  015520  012737  000161  004166          MOV     #WRITE,@#DTADPB+2 ;COMMAND=WRITE DATA
 4640  015526  013737  001510  004176          MOV     @#FC,@#DTADPB+12 ;CYLINDER
 4641  015534  113737  001524  004174          MOVB    @#FS,@#DTADPB+10 ;SECTOR
 4642  015542  012737  177400  004170          MOV     #SCTRWC,@#DTADPB+4 ;WORD COUNT
 4643  015550  012737  050202  004172          MOV     #BUFFER,@#DTADPB+6 ;BUFFER ADDRESS
 4644  015556  005000                          CLR     R0              ;TRACK=0
 4645  015560  012737  015560  001110          MOV     #.,$LPERR       ;SETUP THE ERROR LOOP ADDRESS
 4646  015566  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
 4647  015572  110037  004175          1$:     MOVB    R0,@#DTADPB+11  ;TRACK ADDRESS
 4648  015576  004037  025642                  JSR     R0,@#DRVCAL     ;START A DATA TRANSFER
 4649  015602  062737  001000  004172          ADD     #256.*2.,@#DTADPB+6 ;UPDATE BUFFER ADDRESS
 4650  015610  005200                          INC     R0              ;UPDATE TRACK NUMBER
 4651  015612  022700  000023                  CMP     #19.,R0         ;OUT OF TRACKS?
 4652  015616  003365                          BGT     1$              ;NO--BRANCH
 4653  015620  012737  050202  004172          MOV     #BUFFER,@#DTADPB+6 ;BUFFER ADDRESS
 4654  015626  005000                          CLR     R0
 4655  015630  012737  015630  001110          MOV     #.,$LPERR       ;SETUP THE ERROR LOOP ADDRESS
 4656  015636  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
 4657  015642  012737  000151  004166          MOV     #WRCKD,@#DTADPB+2 ;COMMAND=WRITE CHECK
 4658  015650  110037  004175          2$:     MOVB    R0,@#DTADPB+11  ;TRACK ADDRESS
 4659  015654  004037  025642                  JSR     R0,@#DRVCAL     ;START A DATA TRANSFER
 4660  015660  062737  001000  004172          ADD     #256.*2.,@#DTADPB+6 ;UPDATE BUFFER ADDRESS
 4661  015666  005200                          INC     R0              ;UPDATE TRACK NUMBER
 4662  015670  022700  000023                  CMP     #19.,R0 ;OUT OF TRACKS?
 4663  015674  003365                          BGT     2$              ;NO--BRANCH
 4664  015676  005000                          CLR     R0              ;FIRST TRACK ADDRESS
 4665  015700  110037  004175          3$:     MOVB    R0,@#DTADPB+11  ;TRACK
 4666  015704  010001                          MOV     R0,R1           ;FORM BUFFER ADDRESS
 4667  015706  012737  050202  004172          MOV     #BUFFER,@#DTADPB+6 ;BUFFER ADDRESS
 4668  015714  005301                  4$:     DEC     R1
 4669  015716  002411                          BLT     5$
 4670  015720  062737  001000  004172          ADD     #256.*2.,@#DTADPB+6
 4671  015726  000772                          BR      4$
 4672  015730  012737  015730  001110          MOV     #.,$LPERR       ;SETUP THE ERROR LOOP ADDRESS
 4673  015736  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
 4674  015742  012737  000161  004166  5$:     MOV     #WRITE,@#DTADPB+2 ;COMMAND=WRITE DATA
 4675  015750  004037  025642                  JSR     R0,@#DRVCAL     ;START A DATA TRANSFER
 4676  015754  062737  001000  004172  6$:     ADD     #256.*2.,@#DTADPB+6 ;UPDATE BUFFER ADDRESS
 4677  015762  105237  004175                  INCB    @#DTADPB+11     ;MOVE TO NEXT TRACK
 4678  015766  012737  015766  001110          MOV     #.,$LPERR       ;SETUP THE ERROR LOOP ADDRESS
 4679  015774  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
 4680  016000  012737  000151  004166          MOV     #WRCKD,@#DTADPB+2 ;COMMAND=WRITE CHECK DATA
 4681  016006  004037  025642                  JSR     R0,@#DRVCAL     ;START A DATA TRANSFER
 4682  016012  122737  000022  004175          CMPB    #18.,@#DTADPB+11 ;OUT OF TRACKS?
 4683  016020  003355                          BGT     6$              ;NO--BRANCH
 4684  016022  005200                          INC     R0              ;NEXT TRACK TO WRITE
 4685  016024  022700  000022                  CMP     #18.,R0         ;OUT OF TRACKS?
 4686  016030  003323                          BGT     3$              ;NO--BRANCH
 4687  016032  000004                  EXIT17: SCOPE
```

```
                        .SBTTL  *** DATA TEST ***

                        ;;*****************************************************************
                        ;*TEST 20        DATA TEST

                        ;*      THIS TEST PERFORMS DATA STORAGE AND RETRIEVAL ON CYLINDERS
                        ;*      "FC" THROUGH "LC" BY THE INCREMENT "IC" USING THE DATA PATTERNS
                        ;*      SPECIFIED. THE FOLLOWING SEQUENCE OCCURS FOR EACH CYLINDER:
                        ;*      1.      SET "NT" TO "FT" THEN REPEAT 2-4 UNTIL "NT" > "LT"
                        ;*      2.      WRITE THEN WRITE CHECK "FS" THROUGH "LS" OF TRACK "NT"
                        ;*      3.      READ THEN SOFTWARE COMPARE "FS" THROUGH "LS" OF TRACK "NT"
                        ;*      4.      INCREMENT "NT" BY "IT"
                        ;*      5.      REPEAT STEPS 1-4 FOR EACH DATA PATTERN
                        ;*      6.      REPEAT STEPS 1-5 FOR "FC" THROUGH "LC" ADVANCING BY "IC"
                        ;*
                        ;*      IF A WRITE CHECK ERROR OCCURS THE ERROR IS REPORTED AND
                        ;*      THE TRACK IN ERROR IS REWRITTEN AND CHECKED. THIS CHECK IS
                        ;*      ACCOMPLISHED BY PERFORMING TWO(2) SUCCESSIVE ERROR FREE
                        ;*      WRITE CHECKS. IF THE CHECK FAILS THE ERROR IS REPORTED AS
                        ;*      FATAL AND NO READ OCCURS.
                        ;*      FS DEFAULTS TO 1 AND LS DEFAULTS TO 0
                        ;*      PAT DEFAULTS TO 177777 (ALL POSSIBLE PATTERNS)
                        ;*      THE POSSIBLE PATTERNS ARE:
                        ;*
                        ;*      PAT 0   PAT 1   PAT 2   PAT 3   PAT 4   PAT 5   PAT 6   PAT 7
                        ;*      ------  ------  ------  ------  ------  ------  ------  ------
                        ;*      165555  000001  177776  000000  000000  052525  007417  026455
                        ;*      133333  000003  177774  000000  010421  052525  007417  026455
                        ;*      165555  000007  177770  000000  021042  052525  007417  026455
                        ;*      133333  000017  177760  177777  031463  125252  170360  151322
                        ;*      165555  000037  177740  177777  042104  125252  170360  151322
                        ;*      133333  000077  177700  177777  052525  125252  170360  151322
                        ;*      165555  000177  177600  000000  063146  052525  007417  026455
                        ;*      133333  000377  177400  000000  073567  052525  007417  026455
                        ;*      165555  000777  177000  177777  104210  125252  170360  151322
                        ;*      133333  001777  176000  177777  114631  125252  170360  151322
                        ;*      165555  003777  174000  000000  125252  052525  007417  026455
                        ;*      133333  007777  170000  177777  135673  125252  170360  151322
                        ;*      165555  017777  160000  000000  146314  052525  007417  026455
                        ;*      133333  037777  140000  177777  156735  125252  170360  151322
                        ;*      165555  077777  100000  000000  167356  052525  007417  026455
                        ;*      133333  177777  000000  177777  177777  125252  170360  151322
                        ;*
                        ;*      PAT 8   PAT 9   PAT 10  PAT 11  PAT 12  PAT 13  PAT 14  PAT 15
                        ;*      ------  ------  ------  ------  ------  ------  ------  ------
                        ;*      165555  000001  177776  172666  077777  153333  000000  177777
                        ;*      133333  000002  177775  155555  137777  066667  177777  000000
                        ;*      165555  000004  177773  172666  157777  153333  177777  000000
                        ;*      133333  000010  177767  155555  167777  066667  177777  000000
                        ;*      165555  000020  177757  172666  173777  153333  177777  000000
                        ;*      133333  000040  177737  155555  175777  066667  177777  000000
                        ;*      165555  000100  177677  172666  176777  153333  177777  000000
                        ;*      133333  000200  177577  155555  177377  066667  177777  000000
                        ;*      165555  000400  177377  172666  177577  153333  177777  000000
                        ;*      133333  001000  176777  155555  177677  066667  177777  000000
                        ;*      165555  002000  175777  172666  177737  153333  177777  000000
```

```
 4744                                   ;*      133333  004000  173777  155555  177757  066667  177777  000000
 4745                                   ;*      165555  010000  167777  172666  177767  153333  177777  000000
 4746                                   ;*      133333  020000  157777  155555  177773  066667  177777  000000
 4747                                   ;*      165555  040000  137777  172666  177775  153333  177777  000000
 4748                                   ;*      133333  100000  077777  155555  177776  066667  177777  000000
 4749                                   ;*
 4750
 4751                                   ;;*****************************************************************
 4752  016034                           TST20:
 4753  016034  000240                          NOP
 4754  016036  033737  001424  001236          BIT     BITS+<20*2-40>,TSTNMS+2  ;DO THIS TEST ?
 4755  016044  001002                          BNE     64$                      ;YES--BRANCH
 4756  016046  000137  016554                  JMP     TST21                    ;NO--GO TO THE NEXT TEST
 4757  016052  012737  000020  001102  64$:    MOV     #20,@#$TSTNM             ;SET UP TEST NUMBER AND
 4758                                                                           ;CLEAR THE ERROR FLAG ($ERFLG)
 4759  016060  004737  024726                  JSR     PC,LODPRM                ;LOAD THE PARMETERS FOR THE TEST
 4760  016064  012737  016236  001110          MOV     #TEST20,@#$LPERR         ;SETUP THE LOOP ON ERROR ADDRESS
 4761  016072  013777  001102  163042          MOV     $ISTNM,@DISPLAY          ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
 4762  016100  013737  001506  001204          MOV     @#RPT,$TIMES             ;GET THE ITERATION COUNT
 4763  016106  113737  001364  001115          MOVB    ERR.CT,$ERMAX            ;MAX ERRORS ALLOWED FOR TEST
 4764  016114  012737  016114  001106          MOV     #.,$LPADR                ;SETUP THE LOOP ADDRESS
 4765  016122  005000                          CLR     R0                       ;CLEAR SWITCH
 4766  016124  005004                          CLR     R4                       ;FORM WORD COUNT IN R4
 4767  016126  013701  001526                  MOV     @#LS,R1
 4768  016132  163701  001524                  SUB     @#FS,R1
 4769  016136  002004                          BGE     1$                       ;BRANCH IF FS < OR = LS
 4770  016140  063701  001630                  ADD     PRMLMT+22,R1             ;ADD MAXIMUM SECTOR ADDRESS TO
 4771  016144  005201                          INC     R1                       ;MAKE THE DIFFERENCE POSITIVE
 4772  016146  005100                          COM     R0                       ;SET SWITCH
 4773  016150  062704  000400          1$:     ADD     #256.,R4
 4774  016154  005301                          DEC     R1
 4775  016156  002374                          BGE     1$
 4776  016160  005404                          NEG     R4
 4777  016162  010405                          MOV     R4,R5                    ;COPY NORMAL WORD COUNT INTO SMALL WC
 4778  016164  005700                          TST     R0                       ;SWITCH SET?
 4779  016166  001412                          BEQ     3$                       ;NO--BRANCH
 4780  016170  005005                          CLR     R5                       ;FORM WORD COUNT FOR LS < FS
 4781  016172  013701  001630                  MOV     PRMLMT+22,R1
 4782  016176  163701  001524                  SUB     @#FS,R1
 4783  016202  062705  000400          2$:     ADD     #256.,R5
 4784  016206  005301                          DEC     R1
 4785  016210  002374                          BGE     2$
 4786  016212  005405                          NEG     R5
 4787  016214  113737  001524  004174  3$:     MOVB    @#FS,@#DTADPB+10 ;SECTOR
 4788  016222  012737  050202  004172          MOV     #BUFFER,@#DTADPB+6 ;DATA BUFFER
 4789  016230  012737  016552  001252          MOV     #EXIT20,@#BYPASS
 4790  016236  012706  001100          TEST20: MOV     #STACK,SP                ;LOAD THE STACK POINTER
 4791  016242  005037  001334                  CLR     @#WCEFLG                 ;CLEAR THE WRITE CHECK ERROR FLAG
 4792  016246  013701  001510                  MOV     @#FC,R1 ;PICKUP FIRST CYLINDER
 4793  016252  000407                          BR      2$
 4794  016254  005720                  1$:     TST     (R0)+                    ;MOVE TO NEXT DATA PATTERN
 4795  016256  022700  000040                  CMP     #16.*2.,R0               ;OUT OF PATTERNS?
 4796  016262  003004                          BGT     3$                       ;NO--BRANCH
 4797  016264  004037  027746                  JSR     R0,@#INCCYL              ;MOVE TO NEXT CYLINDER
 4798  016270  000530                          BR      EXIT20                   ;OUT OF CYLINDERS
 4799  016272  005000                  2$:     CLR     R0                       ;START WITH PATTERN 0
```

```
4800  016274  036037  001424  001530  3$:     BIT     BITS(R0),@#PAT  ;THIS PATTERN SELECTED?
4801  016302  001764                          BEQ     1$              ;NO--BRANCH
4802  016304  013702  001516                  MOV     @#FT,R2         ;FIRST TRACK
4803  016310  010137  004176                  MOV     R1,@#DTADPB+12  ;CYLINDER
4804  016314  110237  004175          4$:     MOVB    R2,@#DTADPB+11  ;TRACK
4805  016320  010437  004170                  MOV     R4,@#DTADPB+4   ;WORD COUNT
4806  016324  023701  001512                  CMP     LC,R1           ;LAST DISK CYLINDER?
4807  016330  003005                          BGT     5$              ;NO--BRANCH
4808  016332  022702  000022                  CMP     #18.,R2         ;LAST DISK TRACK?
4809  016336  003002                          BGT     5$              ;NO--BRANCH
4810  016340  010537  004170                  MOV     R5,@#DTADPB+4   ;SHORT WORD COUNT
4811  016344  017703  162570          5$:     MOV     @SWR,R3         ;INHIBIT WRITE AND
4812  016350  005103                          COM     R3              ;WRITE CHECK?
4813  016352  032703  000030                  BIT     #SW04!SW03,R3
4814  016356  001436                          BEQ     7$              ;YES--BRANCH
4815  016360  004737  030476                  JSR     PC,@#SETBUF     ;MOVE DATA PATTERN INTO THE BUFFER
4816  016364  032777  000020  162546          BIT     #SW04,@SWR      ;INHIBIT WRITE?
4817  016372  001012                          BNE     6$              ;YES--BRANCH
4818  016374  012737  016374  001110          MOV     #.,$LPERR       ;SETUP THE ERROR LOOP ADDRESS
4819  016402  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
4820  016406  012737  000161  004166          MOV     #WRITE,@#DTADPB+2 ;COMMAND=WRITE DATA
4821  016414  004037  025642                  JSR     R0,@#DRVCAL     ;START A DATA TRANSFER
4822  016420  032777  000010  162512  6$:     BIT     #SW03,@SWR      ;INHIBIT WRITE CHECK?
4823  016426  001012                          BNE     7$              ;YES--BRANCH
4824  016430  012737  016430  001110          MOV     #.,$LPERR       ;SETUP THE ERROR LOOP ADDRESS
4825  016436  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
4826  016442  012737  000151  004166          MOV     #WRCKD,@#DTADPB+2 ;COMMAND=WRITE CHECK DATA
4827  016450  004037  025642                  JSR     R0,@#DRVCAL     ;START A DATA TRANSFER
4828  016454  005737  001334          7$:     TST     @#WCEFLG        ;WRITE CHECK ERROR FLAG SET?
4829  016460  001404                          BEQ     8$              ;NO--BRANCH
4830  016462  032777  000001  162450          BIT     #SW00,@SWR      ;PERFORM READ AFTER FATAL "WCE"?
4831  016470  001424                          BEQ     9$              ;NO--BRANCH
4832  016472  032777  000004  162440  8$:     BIT     #SW02,@SWR      ;INHIBIT READ DATA AND SOFTWARE COMPARE?
4833  016500  001020                          BNE     9$              ;YES--BRANCH
4834  016502  012737  016502  001110          MOV     #.,$LPERR       ;SETUP THE ERROR LOOP ADDRESS
4835  016510  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
4836  016514  012737  000171  004166          MOV     #READ,@#DTADPB+2 ;COMMAND=READ
4837  016522  004037  025642                  JSR     R0,@#DRVCAL     ;START A DATA TRANSFER
4838  016526  032777  000002  162404          BIT     #SW01,@SWR      ;COMPARE THE DATA?
4839  016534  001002                          BNE     9$              ;NO--BRANCH
4840  016536  004737  030566                  JSR     PC,@#DATCMP     ;YES--DO IT
4841  016542  004037  027716          9$:     JSR     R0,@#INCTRK     ;MOVE TO NEXT TRACK
4842  016546  000642                          BR      1$              ;OUT OF TRACKS GO TO NEXT PATTERN
4843  016550  000661                          BR      4$              ;LOOP
4844  016552  000004          EXIT20: SCOPE                           ;SCOPE LOOP
```

```
                                        .SBTTL  *** EXERCISE TEST ***

                                        ;;*****************************************************************
                                        ;*TEST 21        RANDOM ADDRESS AND RANDOM PATTERN TEST

                                        ;*      STARTING AT "FC" AND GOING THROUGH "LC" THE DISK PACK
                                        ;*      IS WRITTEN WITH A RANDOM PATTERN. THE FIRST TWO WORDS
                                        ;*      OF EACH SECTOR WILL BE THE BASE OF THE RANDOM GENERATOR
                                        ;*      FOR THAT SECTOR.
                                        ;*      THE TEST THEN PERFORMS THE FOLLOWING SEQUENCE "R" TIMES
                                        ;*      "R" DEFAULTS TO 20,000.
                                        ;*
                                        ;*      1)      GENERATE A RANDOM ADDRESS
                                        ;*      2)      WRITE A RANDOM PATTERN AT THE ADDRESS
                                        ;*              GENERATED IN 1.
                                        ;*      3)      GENERATE A RANDOM ADDRESS
                                        ;*      4)      READ THE SECTOR AT THE ADDRESS
                                        ;*              GENERATED IN 3.
                                        ;*      5)      DO A SOFTWARE CHECK OF THE DATA READ IN 4.
                                        ;*      6)      DO A WRITE CHECK OF THE DATA WRITTEN IN 2
                                        ;*      7)      GENERATE A RANDOM ADDRRESS
                                        ;*      8)      READ THE SECTOR AT THE ADDRESS
                                        ;*              GENERATED IN 7.
                                        ;*      9)      DO A SOFTWARE CHECK OF THE DATA READ IN 8
                                        ;*      10)     DO A WRITE CHECK OF THE DATA WRITTEN IN 2

                                        ;;*****************************************************************
016554                                  TST21:
016554  000240                                  NOP
016556  033737  001426  001236                  BIT     BITS+<21*2-40>,TSTNMS+2  ;DO THIS TEST ?
016564  001002                                  BNE     64$                      ;YES--BRANCH
016566  000137  017332                          JMP     TST22                    ;NO--GO TO THE NEXT TEST
016572  012737  000021  001102  64$:           MOV     #21,@#$TSTNM             ;SET UP TEST NUMBER AND
                                                                                 ;CLEAR THE ERROR FLAG ($ERFLG)
016600  004737  024726                          JSR     PC,LODPRM                ;LOAD THE PARMETERS FOR THE TEST
016604  012737  017024  001110                  MOV     #TEST21,@#$LPERR  ;SETUP THE LOOP ON ERROR ADDRESS
016612  013777  001102  162322                  MOV     $TSTNM,@DISPLAY  ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
016620  013737  001506  001204                  MOV     @#RPT,$TIMES     ;GET THE ITERATION COUNT
016626  113737  001364  001115                  MOVB    ERR.CT,$ERMAX    ;MAX ERRORS ALLOWED FOR TEST
016634  012737  016634  001106                  MOV     #.,$LPADR        ;SETUP THE LOOP ADDRESS
016642  012737  017330  001252                  MOV     #EXIT21,@#BYPASS
016650  012737  176543  023724                  MOV     #176543,@#$HINUM ;PRIME THE RANDOM NUMBER GENERATOR
016656  012737  123456  023726                  MOV     #123456,@#$LONUM
016664  013737  001510  004176                  MOV     @#FC,@#DTADPB+12 ;CYLINDER
016672  013737  001352  004170                  MOV     TRCKWC,@#DTADPB+4 ;WORD COUNT
016700  012737  050202  004172                  MOV     #BUFFER,@#DTADPB+6 ;BUFFER ADDRESS
016706  012737  000161  004166                  MOV     #WRITE,@#DTADPB+2 ;COMMMAND
016714  032737  100000  001220                  BIT     #SW15,@#C.SWR    ;WRITE THE DISK PACK BEFORE TESTING?
016722  001027                                  BNE     3$               ;NO--BRANCH
016724  004037  031104                          JSR     R0,@#FILRAN      ;FILL DATA BUFFER WITH RANDOM DATA
016730  005037  004174          1$:             CLR     @#DTADPB+10      ;SECTOR AND TRACK
016734  012737  016734  001110                  MOV     #.,$LPERR        ;SETUP THE ERROR LOOP ADDRESS
016742  012706  001100                          MOV     #STACK,SP        ;LOAD THE STACK POINTER
016746                                  2$:
016746  004037  025642                          JSR     R0,@#DRVCAL      ;START A DATA TRANSFER
016752  105237  004175                          INCB    @#DTADPB+11      ;NEXT TRACK
```

```
 4901  016756  122737  000023  004175          CMPB    #19.,a#DTADPB+11 ;TIME FOR NEXT CYLINDER
 4902  016764  003370                          BGT     2$               ;NO--BRANCH
 4903  016766  005237  004176                  INC     a#DTADPB+12
 4904  016772  023737  001512  004176          CMP     a#LC,a#DTADPB+12 ;OUT OF CYLINDERS?
 4905  017000  002353                          BGE     1$               ;NO--BRANCH
 4906  017002  012737  177400  004170  3$:     MOV     #SCTRWC,a#DTADPB+4 ;WORD COUNT
 4907  017010  012737  017024  001106          MOV     #TEST21,a#$LPADR
 4908  017016  012737  017024  001110          MOV     #TEST21,a#$LPERR
 4909  017024  012706  001100          TEST21: MOV     #STACK,SP        ;SET STACK POINTER
 4910  017030  004037  031360                  JSR     R0,a#RANADR      ;GENERATE A RANDOM ADDRESS
 4911  017034  013737  004174  001340          MOV     a#DTADPB+10,a#SVADR ;SAVE THE TRACK/SECTOR
 4912  017042  013737  004176  001342          MOV     a#DTADPB+12,a#SVADR+2 ;SAVE THE CYLINDER
 4913  017050  012737  000161  004166          MOV     #WRITE,a#DTADPB+2 ;COMMAND=WRITE DATA
 4914  017056  012701  050202                  MOV     #BUFFER,R1       ;BUFFER ADDRESS
 4915  017062  010137  004172                  MOV     R1,a#DTADPB+6
```

```
 4916  017066  004037  031324                    JSR     R0,@#RANPAT     ;GENERATE RANDOM PATTERN
 4917  017072  012737  017072  001110            MOV     #.,$LPERR       ;SETUP THE ERROR LOOP ADDRESS
 4918  017100  012706  001100                    MOV     #STACK,SP       ;LOAD THE STACK POINTER
 4919  017104  004037  025642                    JSR     R0,@#DRVCAL     ;START A DATA TRANSFER
 4920  017110  004037  031360                    JSR     R0,@#RANADR
 4921  017114  012737  000171  004166            MOV     #READ,@#DTADPB+2 ;COMMAND=READ DATA
 4922  017122  012737  051202  004172            MOV     #BUFFER+512.,@#DTADPB+6 ;BUFFER ADDRESS
 4923  017130  012737  017130  001110            MOV     #.,$LPERR       ;SETUP THE ERROR LOOP ADDRESS
 4924  017136  012706  001100                    MOV     #STACK,SP       ;LOAD THE STACK POINTER
 4925  017142  004037  025642                    JSR     R0,@#DRVCAL     ;START A DATA TRANSFER
 4926  017146  004037  031126                    JSR     R0,@#RANCK      ;CHECK THE DATA
 4927  017152  013737  001340  004174            MOV     @#SVADR,@#DTADPB+10 ;GET ADDRESS OF WHERE THE LAST
 4928  017160  013737  001342  004176            MOV     @#SVADR+2,@#DTADPB+12 ;WRITE WAS PERFORMED
 4929  017166  012737  000151  004166            MOV     #WRCKD,@#DTADPB+2 ;COMMAND=WRITE CHECK DATA
 4930  017174  012737  050202  004172            MOV     #BUFFER,@#DTADPB+6 ;DATA BUFFER ADDRESS
 4931  017202  012737  017202  001110            MOV     #.,$LPERR       ;SETUP THE ERROR LOOP ADDRESS
 4932  017210  012706  001100                    MOV     #STACK,SP       ;LOAD THE STACK POINTER
 4933  017214  004037  025642                    JSR     R0,@#DRVCAL     ;START A DATA TRANSFER
 4934  017220  004037  031360                    JSR     R0,@#RANADR     ;GENERATE A RANDOM ADDRESS
 4935  017224  012737  000171  004166            MOV     #READ,@#DTADPB+2 ;COMMAND=READ
 4936  017232  012737  051202  004172            MOV     #BUFFER+512.,@#DTADPB+6 ;DATA BUFFER ADDRESS
 4937  017240  012737  017240  001110            MOV     #.,$LPERR       ;SETUP THE ERROR LOOP ADDRESS
 4938  017246  012706  001100                    MOV     #STACK,SP       ;LOAD THE STACK POINTER
 4939  017252  004037  025642                    JSR     R0,@#DRVCAL     ;START A DATA TRANSFER
 4940  017256  004037  031126                    JSR     R0,@#RANCK      ;CHECK THE DATA
 4941  017262  013737  001340  004174            MOV     @#SVADR,@#DTADPB+10 ;GET DISK ADDRESS OF THE
 4942  017270  013737  001342  004176            MOV     @#SVADR+2,@#DTADPB+12 ;LAST WRITE
 4943  017276  012737  000151  004166            MOV     #WRCKD,@#DTADPB+2 ;COMMAND=WRITE CHECK DATA
 4944  017304  012737  050202  004172            MOV     #BUFFER,@#DTADPB+6 ;DATA BUFFER ADDRESS
 4945  017312  012737  017312  001110            MOV     #.,$LPERR       ;SETUP THE ERROR LOOP ADDRESS
 4946  017320  012706  001100                    MOV     #STACK,SP       ;LOAD THE STACK POINTER
 4947  017324  004037  025642                    JSR     R0,@#DRVCAL     ;START A DATA TRANSFER
 4948  017330  000004                EXIT21: SCOPE                   ;LOOP ?
 4949
 4950                                .SBTTL  *** RP04 ACCESS TIME ADJUSTMENT TEST ***
 4951
 4952                                ;;*****************************************************************
 4953                                ;*TEST 22       RP04 ACCESS TIME ADJUSTMENT TEST
 4954
 4955                                ;*      THIS TEST PERFORMS SEEKS BETWEEN CYLINDERS 0 & 136 TO ALLOW THE
 4956                                ;*      OPERATOR TO ADJUST THE ACCESS TIME ON AN RP04 USING THE
 4957                                ;*      DDU.  THE PROGRAM STALLS APPROXIMATELY 5 SECONDS BETWEEN SEEKS
 4958                                ;*      SO THAT THE ACCESS TIME INDICATORS ON THE DDU MAY BE OBSERVED.
 4959
 4960                                ;;*****************************************************************
 4961  017332                        TST22:
 4962  017332  000240                            NOP
 4963  017334  033737  001430  001236            BIT     BITS+<22*2-40>,TSTNMS+2  ;DO THIS TEST ?
 4964  017342  001002                            BNE     64$             ;YES--BRANCH
 4965  017344  000137  017502                    JMP     $EOP            ;NO--GO TO THE END OF THE PROGRAM
 4966  017350  012737  000022  001102  64$:     MOV     #22,@#$TSTNM    ;SET UP TEST NUMBER AND
 4967                                                                    ;CLEAR THE ERROR FLAG ($ERFLG)
 4968  017356  004737  024726                    JSR     PC,LODPRM       ;LOAD THE PARMETERS FOR THE TEST
 4969  017362  012737  017420  001110            MOV     #TEST22,@#$LPERR  ;SETUP THE LOOP ON ERROR ADDRESS
 4970  017370  013777  001102  161544            MOV     $TSTNM,@DISPLAY  ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
 4971  017376  013737  001506  001204            MOV     @#RPT,$TIMES    ;GET THE ITERATION COUNT
```

```
 4972  017404  112737  000144  001115          MOVB    #100.,$ERMAX    ;MAX ERRORS ALLOWED FOR TEST
 4973  017412  012737  017420  001106          MOV     #TEST22,$LPADR  ;SETUP THE LOOP ADDRESS
 4974  017420  012706  001100          TEST22: MOV     #STACK,SP       ;SETUP THE STACK POINTER
 4975  017424  013737  001512  004116          MOV     LC,DPB.A+12     ;ENDING CYLINDER
 4976  017432  112737  000105  004106          MOVB    #SEEK,@#DPB.A+2 ;SEEK=COMMAND
 4977  017440  004037  025150                  JSR     R0,@#CALL.A     ;GO EXECUTE THE COMMAND
 4978  017444  004037  026502                  JSR     R0,STALL        ;STALL
 4979  017450  001360                          .WORD   STALL3          ;ADDRESS OF STALL VALUE
 4980  017452  013737  001510  004116          MOV     FC,DPB.A+12     ;STARTING CYLINDER
 4981  017460  112737  000105  004106          MOVB    #SEEK,@#DPB.A+2 ;SEEK=COMMAND
 4982  017466  004037  025150                  JSR     R0,@#CALL.A     ;GO EXECUTE THE COMMAND
 4983  017472  004037  026502                  JSR     R0,STALL        ;STALL
 4984  017476  001360                          .WORD   STALL3          ;ADDRESS OF STALL VALUE
 4985  017500  000004                  EXIT22: SCOPE                   ;LOOP ?
 4986
 4987
 4988
 4989                                  .SBTTL  END OF PASS ROUTINE
 4990
 4991                                  ;;*******************************************************************
 4992                                  ;*INCREMENT THE PASS NUMBER ($PASS)
 4993                                  ;*INDICATE END-OF-PROGRAM AFTER 8. PASSES THRU THE PROGRAM
 4994                                  ;*IF THERES A MONITOR GO TO IT
 4995                                  ;*IF THERE ISN'T JUMP TO RESTART
 4996
 4997  017502                          $EOP:
 4998  017502  104401  017510                  TYPE    ,65$            ;;TYPE ASCIZ STRING
 4999  017506  000410                          BR      64$             ;;GET OVER THE ASCIZ
 5000                                  ;;65$:  .ASCIZ  <CR><LF><LF>/END OF PASS/
 5001  017530                          64$:
 5002  017530  005737  001232                  TST     @#DRVSEL        ;ANY DRIVES SELECTED?
 5003  017534  001434                          BEQ     1$              ;NO--BRANCH
 5004  017536  104401  017544                  TYPE    ,67$            ;;TYPE ASCIZ STRING
 5005  017542  000405                          BR      66$             ;;GET OVER THE ASCIZ
 5006                                  ;;67$:  .ASCIZ  / ON DRIVE/
 5007  017556                          66$:
 5008  017556  013746  001254                  MOV     @#CHKDRV,-(SP)  ;;SAVE @#CHKDRV FOR TYPEOUT
 5009  017562  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
 5010  017564     002                          .BYTE   2               ;;TYPE 2 DIGIT(S)
 5011  017565     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
 5012  017566  104401  017574                  TYPE    ,69$            ;;TYPE ASCIZ STRING
 5013  017572  000412                          BR      68$             ;;GET OVER THE ASCIZ
 5014                                  ;;69$:  .ASCIZ  /   ERRORS DETECTED=/
 5015  017620                          68$:
 5016  017620  013746  001112                  MOV     @#$ERTTL,-(SP)  ;;SAVE @#$ERTTL FOR TYPEOUT
 5017  017624  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 5018  017626  005037  001112          1$:     CLR     @#$ERTTL        ;ZERO ERROR TOTAL
 5019  017632  005037  001102                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
 5020  017636  005037  001204                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
 5021  017642  005237  001100                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
 5022  017646  042737  100000  001100          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
 5023  017654  005327                          DEC     (PC)+           ;;LOOP?
 5024  017656  000010          $EOPCT: .WORD   8.
 5025  017660  003030                          BGT     $DOAGN          ;;YES
 5026  017662  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
 5027  017664  000010          $ENDCT: .WORD   8.
```

```
 5028  017666  017656                          $EOPCT
 5029  017670  10440i  017676                  TYPE    ,65$            ;;TYPE ASCIZ STRING
 5030  017674  000410                          BR      64$             ;;GET OVER THE ASCIZ
 5031                                  ;;65$:  .ASCIZ  <CR><LF>/END OF TEST /
 5032  017716                          64$:
 5033  017716  104401  017746                  TYPE    ,$ENULL         ;TYPE NULL CHARACTER
 5034  017722  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
 5035  017726  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
 5036  017730  000005                          RESET                   ;;CLEAR THE WORLD
 5037  017732  004710                  $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
 5038  017734  000240                          NOP                     ;;SAVE ROOM
 5039  017736  000240                          NOP                     ;;FOR
 5040  017740  000240                          NOP                     ;;ACT11
 5041  017742                          $DOAGN:
 5042  017742  000137                          JMP     @(PC)+          ;;RETURN
 5043  017744  006216                  $RTNAD: .WORD   RESTART
 5044  017746     377      377      000 $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
 5045          017752                          .EVEN
```

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

                                .SBTTL  ERROR HANDLER ROUTINE

                                ;;*******************************************************************
                                ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
                                ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
                                ;*AND GO TO TYPERR ON ERROR
                                ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                ;*SW15=1         HALT ON ERROR
                                ;*SW13=1         INHIBIT ERROR TYPEOUTS
                                ;*SW10=1         BELL ON ERROR
                                ;*SW09=1         LOOP ON ERROR
                                ;*CALL
                                ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER

017752                          $ERROR:
017752  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
017754  032777  000400  161156          BIT     #SW08,@SWR      ;SEND ERROR MESSAGE TO TTY?
017762  001411                          BEQ     7$              ;YES--BRANCH
017764  005737  001230                  TST     @#LPTAVL        ;IS THERE A LINE PRINTER AVAILABLE?
017770  001406                          BEQ     7$              ;NO--BRANCH
017772  013737  001420  001150          MOV     @#LPS,@#$TPS    ;YES--SETUP STATUS
020000  013737  001422  001152          MOV     @#LPB,@#$TPB    ;AND BUFFER REG.'S FOR LINE PRINTER
020006  105237  001103          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
020012  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
020014  013777  001102  161120          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
020022  032777  002000  161110          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
020030  001402                          BEQ     1$              ;;NO - SKIP
020032  104401  001210                  TYPE    ,$BELL          ;;RING BELL
020036  005237  001112          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
020042  011637  001116                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
020046  162737  000002  001116          SUB     #2,$ERRPC
020054  117737  161036  001114          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
020062  032777  020000  161050          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
020070  001004                          BNE     20$             ;;SKIP TYPEOUTS
020072  004737  020172                  JSR     PC,TYPERR       ;;GO TO USER ERROR ROUTINE
020076  104401  001215                  TYPE    ,$CRLF
020102                          20$:
020102  005777  161032          2$:     TST     @SWR            ;;HALT ON ERROR
020106  100002                          BPL     3$              ;;SKIP IF CONTINUE
020110  000000                          HALT                    ;;HALT ON ERROR:
020112  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
020114  032777  001000  161016  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
020122  001402                          BEQ     4$              ;;BR IF NO
020124  013716  001110                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
020130  005737  001206          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
020134  001402                          BEQ     5$              ;;BR IF NONE
020136  013716  001206                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
020142                          5$:
020142  023737  000042  000046          CMP     @#42,@#46               ;ACT11 AUTOMATIC MODE?
020150  001001                          BNE     6$              ;NO, CONTINUE
020152  000240                          NOP
020154  013737  001414  001150  6$:     MOV     @#TPS,@#$TPS    ;SET STATUS AND BUFFER REG.'S
020162  013737  001416  001152          MOV     @#TPB,@#$TPB    ;FOR TTY
```

```
5102  020170  000002                        RTI                     ;RETURN FROM ERROR CALL
5103
5104                                  ;;*****************************************************************
5105                                  .SBTTL  TYPERR - TYPE ERROR ROUTINE
5106                                  ;THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE
5107                                  ;WHICH ERROR IS TO BE REPORTED, IT THEN OBTAINS, FROM THE "ERROR
5108                                  ;TABLE" ($ERRTB), AND REPORTS THE APPROPIATE INFORMATION
5109                                  ;CONCERNING THE ERROR.
5110                                  ;CALL
5111                                  ;       JSR     PC,@#TYPERR
5112                                  ;       RETURN
5113
5114  020172  113737  001102  001176  TYPERR: MOVB    @#$TSTNM,@#$TMP0 ;SAVE THE TEST NUMBER
5115  020200  104412                        SAVREG                  ;SAVE R0 - R5
5116  020202  162700  000004                SUB     #4,R0           ;FORM TEST PC
5117  020206  010037  001162                MOV     R0,@#$REG0      ;COPY R0-R5 IN $REG0-$REG5
5118  020212  010137  001164                MOV     R1,@#$REG1
5119  020216  010237  001166                MOV     R2,@#$REG2
5120  020222  010337  001170                MOV     R3,@#$REG3
5121  020226  010437  001172                MOV     R4,@#$REG4
5122  020232  010537  001174                MOV     R5,@#$REG5
5123  020236  113700  001114                MOVB    @#$ITEMB,R0     ;PICKUP ERROR ITEM NUMBER
5124  020242  010001                        MOV     R0,R1           ;AND COPY IT INTO R1
5125  020244  005300                        DEC     R0              ;FORM INDEX FOR ERROR TABLE
5126  020246  106300                        ASLB    R0
5127  020250  106300                        ASLB    R0
5128  020252  106300                        ASLB    R0
5129  020254  103002                        BCC     1$              ;IS ERROR > 37?
5130  020256  062700  000240                ADD     #ITEM41-$ERRTB,R0 ;YES--FORM OFFSET
5131  020262  062700  004306        1$:     ADD     #$ERRTB,R0      ;FORM ADDRESS
5132  020266  012037  020302                MOV     (R0)+,2$        ;GET ERROR MESSAGE (EM) POINTER
5133  020272  001447                        BEQ     7$              ;BRANCH IF THERE ISN'T ONE
5134  020274  104401  001215                TYPE    ,$CRLF          ;"CARRIAGE RETURN - LINE FEED
5135  020300  104401                        TYPE
5136  020302  000000        2$:     .WORD   0                       ;"EM" POINTER GOES HERE
5137  020304  162701  000041                SUB     #41,R1          ;SPECIAL ERROR ITEM NUMBER?
5138  020310  100440                        BMI     7$              ;NO--BRANCH
5139  020312  013701  001260                MOV     @#SVSTAT,R1     ;GET STATUS/ERROR INDICATOR
5140  020316  106301                        ASLB    R1              ;STRIP "DONE" BIT (BIT07)
5141  020320  006301                        ASL     R1              ;STRIP "ERROR" BIT (BIT15)
5142  020322  012702  004254                MOV     #STATBL,R2      ;1ST ADDRESS ON STATUS MESSAGE POINTERS
5143  020326  005003                        CLR     R3              ;CARRIAGE RETURN-LINE FEED SWITCH
5144  020330  104401  020336                TYPE    ,65$            ;;TYPE ASCIZ STRING
5145  020334  000402                        BR      64$             ;;GET OVER THE ASCIZ
5146                        ;;65$:  .ASCIZ  / (/
5147  020342                64$:
5148  020342  012237  020364        3$:     MOV     (R2)+,5$        ;MESSAGE POINTER
5149  020346  006301                        ASL     R1              ;TYPE THIS MESSAGE?
5150  020350  103013                        BCC     6$              ;NO--BRANCH
5151  020352  005103                        COM     R3              ;YES--TYPE A "CR" & "LF"?
5152  020354  001002                        BNE     4$              ;NO--BRANCH
5153  020356  104401  001215                TYPE    ,$CRLF          ;YES
5154  020362  104401        4$:     TYPE
5155  020364  000000        5$:     .WORD   0                       ;MESSAGE POINTER GOES HERE
5156  020366  005701                        TST     R1              ;MORE TO TYPE?
5157  020370  001403                        BEQ     6$              ;NO--BRANCH
```

```
 5158  020372  104401  044330                TYPE    ,MSG.SP         ;YES--SPACES
 5159  020376  000761                        BR      3$              ;LOOP
 5160  020400  001360                6$:     BNE     3$              ;BRANCH IF NOT FINISHED
 5161  020402  104401  020410                TYPE    ,67$            ;;TYPE ASCIZ STRING
 5162  020406  000401                        BR      66$             ;;GET OVER THE ASCIZ
 5163                                ;;67$:  .ASCIZ  /!/
 5164  020412                        66$:
 5165  020412  012037  020426        7$:     MOV     (R0)+,8$        ;PICK UP DATA HEADER (DH) POINTER
 5166  020416  001404                        BEQ     9$              ;BRANCH IF NONE
 5167  020420  104401  001215                TYPE    ,$CRLF          ;CARRIAGE RETURN-LINE FEED
 5168  020424  104401                        TYPE
 5169  020426  000000                8$:     .WORD   0               ;"DH" POINTER GOES HERE
 5170  020430  012001                9$:     MOV     (R0)+,R1        ;PICKUP DATA TABLE (DT) POINTER
 5171  020432  001450                        BEQ     20$             ;BRANCH IF NONE
 5172  020434  005005                        CLR     R5              ;SET INDENT SWITCH
 5173  020436  012000                        MOV     (R0)+,R0        ;DATA FORMAT (DF) POINTER
 5174  020440  012002                        MOV     (R0)+,R2        ;NUMBER OF DH'S TO TYPE
 5175  020442  001441                        BEQ     17$             ;BRANCH IF DH NUMBER IS 0
 5176  020444  005105                        COM     R5              ;NO INDENT
 5177  020446  104401  001215                TYPE    ,$CRLF          ;CARRIAGE RETURN-LINE FEED
 5178  020452  112003                10$:    MOVB    (R0)+,R3        ;NUMBER OF DATA WORDS TO TYPE
 5179  020454  112004                        MOVB    (R0)+,R4        ;AND HOW TO TYPE THEM
 5180  020456  006004                11$:    ROR     R4              ;OCTAL OR DECIMAL?
 5181  020460  103403                        BCS     12$             ;DECIMAL--BRANCH
 5182  020462  013146                        MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
 5183  020464  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 5184  020466  000402                        BR      13$
 5185  020470                        12$:
 5186  020470  013146                        MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
 5187  020472  104405                        TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
 5188  020474  005303                13$:    DEC     R3              ;MORE NUMBERS TO TYPE?
 5189  020476  001403                        BEQ     14$             ;NO--BRANCH
 5190  020500  104401  044330                TYPE    ,MSG.SP         ;YES--TYPE SEPERATORS
 5191  020504  000764                        BR      11$             ;LOOP
 5192  020506  005302                14$:    DEC     R2              ;MORE DH'S?
 5193  020510  003421                        BLE     20$             ;NO--BRANCH
 5194  020512  104401  001215                TYPE    ,$CRLF          ;YES--START A NEW LINE
 5195  020516  005105                        COM     R5              ;INDENT?
 5196  020520  001002                        BNE     15$             ;NO--BRANCH
 5197  020522  104401  044330                TYPE    ,MSG.SP         ;YES--TYPE SPACES
 5198  020526  012037  020534        15$:    MOV     (R0)+,16$       ;GET NEXT DH
 5199  020532  104401                        TYPE                    ;AND TYPE IT
 5200  020534  000000                16$:    .WORD   0               ;DH POINTER GOES HERE
 5201  020536  104401  001215                TYPE    ,$CRLF          ;CARRIAGE RETURN-LINE FEED
 5202  020542  005705                        TST     R5              ;INDENT?
 5203  020544  001342                        BNE     10$             ;NO--BRANCH
 5204  020546  104401  044330        17$:    TYPE    ,MSG.SP         ;YES--TYPE SPACES
 5205  020552  000737                        BR      10$             ;LOOP
 5206  020554  104413                20$:    RESREG                  ;RESTORE R0 - R5
 5207  020556  000207                        RTS     PC              ;RETURN
 5208
 5209                                .SBTTL  TYPE ROUTINE
 5210
 5211                                ;;**************************************************************
 5212                                ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
 5213                                ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
```

```
                                ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
                                ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
                                ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
                                ;*
                                ;*CALL:
                                ;*1) USING A TRAP INSTRUCTION
                                ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                                ;*OR
                                ;*      TYPE
                                ;*      MESADR
                                ;*

020560  105737  001157          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
020564  100002                          BPL     1$              ;;BR IF YES
020566  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
020570  000407                          BR      3$              ;;LEAVE
020572  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
020574  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
020600  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
020602  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
020604  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
020606  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
020610  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
020614  000002                          RTI                     ;;RETURN
020616  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
020622  001430                          BEQ     8$
020624  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
020630  001006                          BNE     5$
020632  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
020634  104401                          TYPE                    ;;TYPE A CR AND LF
020636  001215                          $CRLF
020640  105037  020774                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
020644  000755                          BR      2$              ;;GET NEXT CHARACTER
020646  004737  020730          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
020652  123726  001156          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
020656  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
020660  013746  001154                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                                ;;AND THE NULL CHAR.
020664  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
020670  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
020672  004737  020730                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
020676  105337  020774                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
020702  000770                          BR      7$              ;;LOOP

                                ;HORIZONTAL TAB PROCESSOR

020704  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
020710  004737  020730          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
020714  132737  000007  020774          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
020722  001372                          BNE     9$              ;;TAB STOP
020724  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
020726  000724                          BR      2$              ;;GET NEXT CHARACTER
020730  105777  160214          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
020734  100375                          BPL     $TYPEC
020736  116677  000002  160206          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
020744  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
```

```
5270  020752  001003                        BNE     1$              ;;BRANCH IF NO
5271  020754  105037  020774                CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
5272  020760  000406                        BR      $TYPEX          ;;EXIT
5273  020762  122766  000012  000002  1$:   CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
5274  020770  001402                        BEQ     $TYPEX          ;;BRANCH IF YES
5275  020772  105227                        INCB    (PC)+           ;;COUNT THE CHARACTER
5276  020774  000000              $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
5277  020776  000207              $TYPEX: RTS     PC
5278
5279
5280                                .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
5281
5282                                ;;**********************************************************************
5283                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
5284                                ;*OCTAL (ASCII) NUMBER AND TYPE IT.
5285                                ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
5286                                ;*CALL:
5287                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
5288                                ;*      TYPOS                   ;;CALL FOR TYPEOUT
5289                                ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
5290                                ;*      .BYTE   M               ;;M=1 OR 0
5291                                ;*                                      ;;1=TYPE LEADING ZEROS
5292                                ;*                                      ;;0=SUPPRESS LEADING ZEROS
5293                                ;*
5294                                ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
5295                                ;*$TYPOS OR $TYPOC
5296                                ;*CALL:
5297                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
5298                                ;*      TYPON                   ;;CALL FOR TYPEOUT
5299                                ;*
5300                                ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
5301                                ;*CALL:
5302                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
5303                                ;*      TYPOC                   ;;CALL FOR TYPEOUT
5304
5305  021000  017646  000000        $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
5306  021004  116637  000001  021223        MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
5307  021012  112637  021225               MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
5308  021016  062716  000002               ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
5309  021022  000406                        BR      $TYPON
5310  021024  112737  000001  021223  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
5311  021032  112737  000006  021225        MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
5312  021040  112737  000005  021222  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
5313  021046  010346                        MOV     R3,-(SP)        ;;SAVE R3
5314  021050  010446                        MOV     R4,-(SP)        ;;SAVE R4
5315  021052  010546                        MOV     R5,-(SP)        ;;SAVE R5
5316  021054  113704  021225               MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
5317  021060  005404                        NEG     R4
5318  021062  062704  000006               ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
5319  021066  110437  021224               MOVB    R4,$OMODE       ;;SAVE IT FOR USE
5320  021072  113704  021223               MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
5321  021076  016605  000012               MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
5322  021102  005003                        CLR     R3              ;;CLEAR THE OUTPUT WORD
5323  021104  006105                1$:     ROL     R5              ;;ROTATE MSB INTO "C"
5324  021106  000404                        BR      3$              ;;GO DO MSB
5325  021110  006105                2$:     ROL     R5              ;;FORM THIS DIGIT
```

```
5326  021112  006105                          ROL     R5
5327  021114  006105                          ROL     R5
5328  021116  010503                          MOV     R5,R3
5329  021120  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
5330  021122  105337  021224                  DECB    $OMODE          ;;TYPE THIS DIGIT?
5331  021126  100016                          BPL     7$              ;;BR IF NO
5332  021130  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
5333  021134  001002                          BNE     4$              ;;TEST FOR 0
5334  021136  005704                          TST     R4              ;;SUPPRESS THIS 0?
5335  021140  001403                          BEQ     5$              ;;BR IF YES
5336  021142  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
5337  021144  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
5338  021150  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
5339  021154  110337  021220                  MOVB    R3,8$           ;;SAVE FOR TYPING
5340  021160  104401  021220                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
5341  021164  105337  021222          7$:     DECB    $OCNT           ;;COUNT BY 1
5342  021170  003347                          BGT     2$              ;;BR IF MORE TO DO
5343  021172  002402                          BLT     6$              ;;BR IF DONE
5344  021174  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
5345  021176  000744                          BR      2$              ;;GO DO THE LAST DIGIT
5346  021200  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
5347  021202  012604                          MOV     (SP)+,R4        ;;RESTORE R4
5348  021204  012603                          MOV     (SP)+,R3        ;;RESTORE R3
5349  021206  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
5350  021214  012616                          MOV     (SP)+,(SP)
5351  021216  000002                          RTI                     ;;RETURN
5352  021220     000                  8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
5353  021221     000                          .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
5354  021222     000                  $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
5355  021223     000                  $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
5356  021224  000000                  $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
5357
5358                                  .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
5359
5360                                  ;;**************************************************************************
5361                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
5362                                  ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
5363                                  ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
5364                                  ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
5365                                  ;*REPLACED WITH SPACES.
5366                                  ;*CALL:
5367                                  ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
5368                                  ;*      TYPDS                   ;;GO TO THE ROUTINE
5369
5370  021226                          $TYPDS:
5371  021226  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
5372  021230  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
5373  021232  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
5374  021234  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
5375  021236  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
5376  021240  012746  020200                  MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
5377  021244  016605  000020                  MOV     20(SP),R5       ;;GET THE INPUT NUMBER
5378  021250  100004                          BPL     1$              ;;BR IF INPUT IS POS.
5379  021252  005405                          NEG     R5              ;;MAKE THE BINARY NUMBER POS.
5380  021254  112766  000055  000001          MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
5381  021262  005000                  1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
```

```
5382  021264  012703  021442              MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
5383  021270  112723  000040              MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
5384  021274  005002                2$:   CLR     R2              ;;CLEAR THE BCD NUMBER
5385  021276  016001  021432              MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
5386  021302  160105                3$:   SUB     R1,R5           ;;FORM THIS BCD DIGIT
5387  021304  002402                      BLT     4$              ;;BR IF DONE
5388  021306  005202                      INC     R2              ;;INCREASE THE BCD DIGIT BY 1
5389  021310  000774                      BR      3$
5390  021312  060105                4$:   ADD     R1,R5           ;;ADD BACK THE CONSTANT
5391  021314  005702                      TST     R2              ;;CHECK IF BCD DIGIT=0
5392  021316  001002                      BNE     5$              ;;FALL THROUGH IF 0
5393  021320  105716                      TSTB    (SP)            ;;STILL DOING LEADING 0'S?
5394  021322  100407                      BMI     7$              ;;BR IF YES
5395  021324  106316                5$:   ASLB    (SP)            ;;MSD?
5396  021326  103003                      BCC     6$              ;;BR IF NO
5397  021330  116663  000001  177777      MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
5398  021336  052702  000060          6$: BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
5399  021342  052702  000040          7$: BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
5400  021346  110223                      MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
5401  021350  005720                      TST     (R0)+           ;;JUST INCREMENTING
5402  021352  020027  000010              CMP     R0,#10          ;;CHECK THE TABLE INDEX
5403  021356  002746                      BLT     2$              ;;GO DO THE NEXT DIGIT
5404  021360  003002                      BGT     8$              ;;GO TO EXIT
5405  021362  010502                      MOV     R5,R2           ;;GET THE LSD
5406  021364  000764                      BR      6$              ;;GO CHANGE TO ASCII
5407  021366  105726                8$:   TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
5408  021370  100003                      BPL     9$              ;;BR IF NO
5409  021372  116663  177777  177776      MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
5410  021400  105013                9$:   CLRB    (R3)            ;;SET THE TERMINATOR
5411  021402  012605                      MOV     (SP)+,R5        ;;POP STACK INTO R5
5412  021404  012603                      MOV     (SP)+,R3        ;;POP STACK INTO R3
5413  021406  012602                      MOV     (SP)+,R2        ;;POP STACK INTO R2
5414  021410  012601                      MOV     (SP)+,R1        ;;POP STACK INTO R1
5415  021412  012600                      MOV     (SP)+,R0        ;;POP STACK INTO R0
5416  021414  104401  021442              TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
5417  021420  016666  000002  000004      MOV     2(SP),4(SP)     ;;ADJUST THE STACK
5418  021426  012616                      MOV     (SP)+,(SP)
5419  021430  000002                      RTI                     ;;RETURN TO USER
5420  021432  023420                $DTBL:  10000.
5421  021434  001750                       1000.
5422  021436  000144                       100.
5423  021440  000012                       10.
5424  021442  000004                $DBLK:  .BLKW   4
5425
5426                                  .SBTTL  TTY INPUT ROUTINE
5427
5428                                  ;;**********************************************************************
5429                                  .ENABL  LSB
5430  021452  000000                $TKCNT: .WORD   0               ;;NUMBER OF ITEMS IN QUEUE
5431  021454  000000                $TKQIN: .WORD   0               ;;INPUT POINTER
5432  021456  000000                $TKQOUT: .WORD  0               ;;OUTPUT POINTER
5433  021460  000002                $TKQSRT: .BLKB  2               ;;TTY KEYBOARD QUEUE
5434          021462                $TKQEND=.
5435
5436                                  ;*TK INITIALIZE ROUTINE
5437                                  ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
```

```
                                        ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
                                        ;
                                        ;*CALL:
                                        ;*      JSR     PC,$TKINT
                                        ;*      RETURN
                                        ;
021462  005037  021452                  $TKINT: CLR     $TKCNT          ;;CLEAR COUNT OF ITEMS IN QUEUE
021466  012737  021460  021454                  MOV     #$TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
021474  013737  021454  021456                  MOV     $TKQIN,$TKQOUT  ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
021502  012737  021532  000060                  MOV     #$TKSRV,@#TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
021510  012737  000200  000062                  MOV     #200,@#TKVEC+2  ;;"BR" LEVEL 4
021516  005777  157424                          TST     @$TKB           ;;CLEAR DONE FLAG
021522  012777  000100  157414                  MOV     #100,@$TKS      ;;ENABLE TTY KEYBOARD INTERRUPT
021530  000207                                  RTS     PC              ;;RETURN TO CALLER

                                        ;*TK SERVICE ROUTINE
                                        ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
                                        ;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
                                        ;*IT IN THE QUEUE.
                                        ;*IF THE CHARACTER IS A "CONTROL-C" (^C) $TKINT IS CALLED AND
                                        ;*UPON RETURN EXIT IS MADE TO THE "CONTROL-C" RESTART ADDRESS (START2)
                                        ;
021532  117746  157410                  $TKSRV: MOVB    @$TKB,-(SP)     ;;PICKUP THE CHARACTER
021536  042716  177600                          BIC     #^C177,(SP)     ;;STRIP THE JUNK
021542  021627  000003                          CMP     (SP),#3         ;;IS IT A CONTROL C?
021546  001007                                  BNE     1$              ;;BRANCH IF NO
021550  104401  022662                          TYPE    ,$CNTLC         ;;TYPE A CONTROL-C (^C)
021554  004737  021462                          JSR     PC,$TKINT       ;;INIT THE KEYBOARD
021560  005726                                  TST     (SP)+           ;;CLEAN UP STACK
021562  000137  004730                          JMP     START2          ;;CONTROL C RESTART
021566  021627  000007                  1$:     CMP     (SP),#7         ;;IS IT A CONTROL G?
021572  001004                                  BNE     2$              ;;BRANCH IF NO
021574  022737  000176  001140                  CMP     #SWREG,SWR      ;;IS SOFT-SWR SELECTED?
021602  001500                                  BEQ     6$              ;;GO TO SWR CHANGE

021604                                  2$:
021604  022737  000002  021452                  CMP     #2,$TKCNT       ;;IS THE QUEUE FULL?
021612  001004                                  BNE     3$              ;;BRANCH IF NO
021614  104401  001210                          TYPE    ,$BELL          ;;RING THE TTY BELL
021620  005726                                  TST     (SP)+           ;;CLEAN CHARACTER OFF OF STACK
021622  000451                                  BR      5$              ;;EXIT
021624  021627  000023                  3$:     CMP     (SP),#23        ;;IS IT A CONTROL-S?
021630  001021                                  BNE     32$             ;;BRANCH IF NO
021632  005077  157306                          CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
021636  005726                                  TST     (SP)+           ;;CLEAN CHAR OFF STACK
021640  105777  157300                  31$:    TSTB    @$TKS           ;;WAIT FOR A CHAR
021644  100375                                  BPL     31$             ;;LOOP UNTIL ITS THERE
021646  117746  157274                          MOVB    @$TKB,-(SP)     ;;GET THE CHARACTER
021652  042716  177600                          BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
021656  022627  000021                          CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
021662  001366                                  BNE     31$             ;;BRANCH IF NO
021664  012777  000100  157252                  MOV     #100,@$TKS      ;;REENABLE TTY KEYBOARD INTERRUPTS
021672  000002                                  RTI                     ;;RETURN
021674  005237  021452                  32$:    INC     $TKCNT          ;;COUNT THIS CHARACTER
021700  021627  000140                          CMP     (SP),#140       ;;IS IT UPPER CASE?
021704  002405                                  BLT     4$              ;;BRANCH IF YES
```

```
5494  021706  021627  000175              CMP     (SP),#175       ;;IS IT A SPECIAL CHAR?
5495  021712  003002                      BGT     4$              ;;BRANCH IF YES
5496  021714  042716  000040              BIC     #40,(SP)        ;;MAKE IT UPPER CASE
5497  021720  112677  177530        4$:   MOVB    (SP)+,@$TKQIN   ;;AND PUT IT IN QUEUE
5498  021724  005237  021454              INC     $TKQIN          ;;UPDATE THE POINTER
5499  021730  023727  021454  021462      CMP     $TKQIN,#$TKQEND ;;GO OFF THE END?
5500  021736  001003                      BNE     5$              ;;BRANCH IF NO
5501  021740  012737  021460  021454      MOV     #$TKQSRT,$TKQIN ;;RESET THE POINTER
5502  021746  000002                5$:   RTI                     ;;RETURN
5503
5504                                ;;**********************************************************************
5505                                ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
5506                                ;*ROUTINE IS ENTERED  FROM THE TRAP HANDLER, AND WILL
5507                                ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
5508                                ;*CALL WHEN OPERATING IN TTY INTERRUPT MODE.
5509  021750  022737  000176  001140 $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED
5510  021756  001124                      BNE     15$             ;;EXIT IF NOT
5511  021760  105777  157160              TSTB    @$TKS           ;;IS A CHAR WAITING?
5512  021764  100121                      BPL     15$             ;;IF NOT, EXIT
5513  021766  117746  157154              MOVB    @$TKB,-(SP)     ;;YES
5514  021772  042716  177600              BIC     #^C177,(SP)     ;;MAKE IT  7-BIT ASCII
5515  021776  021627  000007              CMP     (SP),#7         ;;IS IT A CONTROL-G?
5516  022002  001300                      BNE     2$              ;;IF NOT, PUT IT IN THE TTY QUEUE
5517                                                              ;;AND EXIT
5518
5519                                ;;**********************************************************************
5520                                ;*CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
5521                                ;*ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
5522                                ;*CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
5523  022004  123727  001134  000001 6$:    CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
5524  022012  001674                      BEQ     2$              ;;BRANCH IF YES
5525  022014  005726                      TST     (SP)+           ;;CLEAR CONTROL-G OFF STACK
5526  022016  004737  021462              JSR     PC,$TKINT       ;;FLUSH THE TTY INPUT QUEUE
5527  022022  005077  157116              CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
5528  022026  112737  000001  001135      MOVB    #1,$INTAG       ;;SET INTERRUPT MODE INDICATOR
5529
5530  022034  104401  022674              TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
5531  022040  104401  022701        $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
5532  022044  013746  000176              MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
5533  022050  104402                      TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
5534  022052  104401  022712              TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
5535  022056  005046                19$:   CLR     -(SP)           ;;CLEAR COUNTER
5536  022060  005046                      CLR     -(SP)           ;;THE NEW SWR
5537  022062  105777  157056        7$:   TSTB    @$TKS           ;;CHAR THERE?
5538  022066  100375                      BPL     7$              ;;IF NOT TRY AGAIN
5539
5540  022070  117746  157052              MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
5541  022074  042716  177600              BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
5542
5543  022100  021627  000003              CMP     (SP),#3         ;;IS IT A CONTROL-C?
5544  022104  001015                      BNE     9$              ;;BRANCH IF NOT
5545  022106  104401  022662              TYPE    ,$CNTLC         ;;YES, ECHO CONTROL-C (^C)
5546  022112  062706  000006              ADD     #6,SP           ;;CLEAN UP STACK
5547  022116  123727  001135  000001      CMPB    $INTAG,#1       ;;REENABLE TTY KEYBOARD INTERRUPTS?
5548  022124  001003                      BNE     8$              ;;BRANCH IF NO
5549  022126  012777  000100  157010      MOV     #100,@$TKS      ;;ALLOW TTY KEYBOARD INTERRUPTS
```

```
5550  022134  000137  004730          8$:     JMP     START2          ;;CONTROL-C RESTART
5551
5552
5553  022140  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
5554  022144  001005                          BNE     10$             ;;BRANCH IF NOT
5555  022146  104401  022667                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
5556  022152  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
5557  022156  000737                          BR      19$             ;;LET'S TRY IT AGAIN
5558
5559
5560  022160  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
5561  022164  001022                          BNE     16$             ;;BRANCH IF NO
5562  022166  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
5563  022172  001403                          BEQ     11$             ;;BRANCH IF YES
5564  022174  016677  000002  156736          MOV     2(SP),@SWR      ;;SAVE NEW SWR
5565  022202  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
5566  022206  104401  001215          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
5567  022212  123727  001135  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
5568  022220  001003                          BNE     15$             ;;BRANCH IF NOT
5569  022222  012777  000100  156714          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
5570  022230  000002          15$:    RTI                             ;;RETURN
5571  022232  004737  020730          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
5572  022236  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
5573  022242  002420                          BLT     18$             ;;BRANCH IF YES
5574  022244  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
5575  022250  003015                          BGT     18$             ;;BRANCH IF YES
5576  022252  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
5577  022256  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
5578  022262  001403                          BEQ     17$             ;;BRANCH IF YES
5579  022264  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
5580  022266  006316                          ASL     (SP)            ;;   CHAR OVER TO MAKE
5581  022270  006316                          ASL     (SP)            ;;   ROOM FOR NEW ONE.
5582  022272  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
5583  022276  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
5584  022302  000667                          BR      7$              ;;GET THE NEXT ONE
5585  022304  104401  001214          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
5586  022310  000720                          BR      20$             ;;SIMULATE CONTROL-U
5587                                  .DSABL  LSB
5588
5589
5590                                  ;;**************************************************************************
5591                                  ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
5592                                  ;*CALL:
5593                                  ;*      RDCHR                   ;;GET A CHARACTER FROM THE QUEUE
5594                                  ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
5595                                  ;*                              ;;WITH PARITY BIT STRIPPED OFF
5596                                  ;
5597
5598  022312  011646          $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC AND
5599  022314  016666  000004  000002          MOV     4(SP),2(SP)     ;;THE PS
5600  022322  005066  000004                  CLR     4(SP)           ;;GET READY FOR A CHARACTER
5601  022326  005046                          CLR     -(SP)           ;;PUT NEW PS ON STACK
5602  022330  012746  022336                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
5603  022334  000002                          RTI                     ;;POP NEW PC AND PS
5604  022336                          64$:
5605  022336  005737  021452          1$:     TST     $TKCNT          ;;WAIT ON A CHARACTER
```

```
5606  022342  001775                          BEQ     1$
5607  022344  005337  021452                  DEC     $TKCNT          ;;DECREMENT THE COUNTER
5608  022350  117766  177102  000004          MOVB    @$TKQOUT,4(SP)  ;;GET ONE CHARACTER
5609  022356  005237  021456                  INC     $TKQOUT         ;;UPDATE THE POINTER
5610  022362  023727  021456  021462          CMP     $TKQOUT,#$TKQEND ;;DID IT GO OFF OF THE END?
5611  022370  001003                          BNE     2$              ;;BRANCH IF NO
5612  022372  012737  021460  021456          MOV     #$TKQSRT,$TKQOUT ;;RESET THE POINTER
5613  022400  000002                  2$:     RTI                     ;;RETURN
5614                                  ;;**************************************************************
5615                                  ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
5616                                  ;*CALL:
5617                                  ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
5618                                  ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
5619                                  ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
5620
5621  022402  010346                  $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
5622  022404  005046                          CLR     -(SP)           ;;CLEAR THE RUBOUT KEY
5623  022406  012703  022636          1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
5624  022412  022703  022662          2$:     CMP     #$TTYIN+20.,R3  ;;BUFFER FULL?
5625  022416  101456                          BLOS    4$              ;;BR IF YES
5626  022420  104410                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
5627  022422  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
5628  022424  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
5629  022430  001022                          BNE     5$              ;;BR IF NO
5630  022432  005716                          TST     (SP)            ;;IS THIS THE FIRST RUBOUT?
5631  022434  001007                          BNE     6$              ;;BR IF NO
5632  022436  112737  000134  022634          MOVB    #'\,9$          ;;TYPE A BACK SLASH
5633  022444  104401  022634                  TYPE    ,9$
5634  022450  012716  177777                  MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
5635  022454  005303                  6$:     DEC     R3              ;;BACKUP BY ONE
5636  022456  020327  022636                  CMP     R3,#$TTYIN      ;;STACK EMPTY?
5637  022462  103434                          BLO     4$              ;;BR IF YES
5638  022464  111337  022634                  MOVB    (R3),9$         ;;SETUP TO TYPEOUT THE DELETED CHAR.
5639  022470  104401  022634                  TYPE    ,9$             ;;GO TYPE
5640  022474  000746                          BR      2$              ;;GO READ ANOTHER CHAR.
5641  022476  005716                  5$:     TST     (SP)            ;;RUBOUT KEY SET?
5642  022500  001406                          BEQ     7$              ;;BR IF NO
5643  022502  112737  000134  022634          MOVB    #'\,9$          ;;TYPE A BACK SLASH
5644  022510  104401  022634                  TYPE    ,9$
5645  022514  005016                          CLR     (SP)            ;;CLEAR THE RUBOUT KEY
5646  022516  122713  000025          7$:     CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
5647  022522  001003                          BNE     8$              ;;BR IF NO
5648  022524  104401  022667                  TYPE    ,$CNTLU         ;;TYPE A CONTROL "U"
5649  022530  000726                          BR      1$              ;;GO START OVER
5650  022532  122713  000022          8$:     CMPB    #22,(R3)        ;;IS CHARACTER A "^R"?
5651  022536  001011                          BNE     3$              ;;BRANCH IF NO
5652  022540  105013                          CLRB    (R3)            ;;CLEAR THE CHARACTER
5653  022542  104401  001215                  TYPE    ,$CRLF          ;;TYPE A "CR" & "LF"
5654  022546  104401  022636                  TYPE    ,$TTYIN         ;;TYPE THE INPUT STRING
5655  022552  000717                          BR      2$              ;;GO PICKUP ANOTHER CHACTER
5656  022554  104401  001214          4$:     TYPE    ,$QUES          ;;TYPE A '?'
5657  022560  000712                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
5658  022562  111337  022634          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
5659  022566  104401  022634                  TYPE    ,9$
5660  022572  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
5661  022576  001305                          BNE     2$              ;;LOOP IF NOT RETURN
```

```
 5662  022600  105063  177777                    CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
 5663  022604  10440i  001216                    TYPE    ,$LF            ;;TYPE A LINE FEED
 5664  022610  005726                            TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
 5665  022612  012603                            MOV     (SP)+,R3        ;;RESTORE R3
 5666  022614  011646                            MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
 5667  022616  016666  000004  000002            MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
 5668  022624  012766  022636  000004            MOV     #$TTYIN,4(SP)
 5669  022632  000002                            RTI                     ;;RETURN
 5670  022634     000                  9$:       .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
 5671  022635     000                          .BYTE   0               ;;TERMINATOR
 5672  022636  000024                  $TTYIN: .BLKB   20.             ;;RESERVE 20. BYTES FOR TTY INPUT
 5673  022662  041536  005015     000  $CNTLC: .ASCIZ  /^C/<15><12>    ;;CONTROL "C"
 5674  022667     136  006525  000012  $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL "U"
 5675  022674  043536  005015     000  $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL "G"
 5676  022701     015  051412  051127  $MSWR:  .ASCIZ  <15><12>/SWR = /
 5677  022706  036440  000040
 5678  022712  020040  042516  020127  $MNEW:  .ASCIZ  /  NEW = /
 5679  022720  020075     000
 5680          022724                  .EVEN
 5681
 5682                                  .SBTTL  SCOPE HANDLER ROUTINE
 5683
 5684                                  ;;**********************************************************************
 5685                                  ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
 5686                                  ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
 5687                                  ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
 5688                                  ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
 5689                                  ;*SW14=1         LOOP ON TEST
 5690                                  ;*SW11=1         INHIBIT ITERATIONS
 5691                                  ;*SW09=1         LOOP ON ERROR
 5692                                  ;*CALL
 5693                                  ;*      SCOPE           ;;SCOPE=IOT
 5694
 5695  022724                  $SCOPE:
 5696  022724  104407                          CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
 5697  022726  032777  040000  156204  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
 5698  022734  001101                          BNE     $OVER           ;;YES IF SW14=1
 5699                                  ;#####START OF CODE FOR THE XOR TESTER#####
 5700  022736  000416                  $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
 5701                                                                  ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
 5702  022740  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
 5703  022744  012737  022764  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
 5704  022752  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
 5705  022756  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
 5706  022762  000453                          BR      $SVLAD          ;;GO TO THE NEXT TEST
 5707  022764  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
 5708  022766  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
 5709  022772  000413                          BR      7$              ;;LOOP ON THE PRESENT TEST
 5710  022774                  6$:;#####END OF CODE FOR THE XOR TESTER#####
 5711  022774  105737  001103  2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
 5712  023000  001421                  BEQ     3$              ;;BR IF NO
 5713  023002  123737  001115  001103          CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
 5714  023010  101015                          BHI     3$              ;;BR IF NO
 5715  023012  032777  001000  156120          BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
 5716  023020  001404                          BEQ     4$              ;;BR IF NO
 5717  023022  013737  001110  001106  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
```

```
5718  023030  000443                          BR      $OVER
5719  023032  105037  001103          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
5720  023036  005037  001204                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
5721  023042  000415                          BR      1$              ;;ESCAPE TO THE NEXT TEST
5722  023044  032777  004000  156066  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
5723  023052  001011                          BNE     1$              ;;BR IF YES
5724  023054  005737  001100                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
5725  023060  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
5726  023062  005237  001104                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
5727  023066  023737  001204  001104          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
5728  023074  002021                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
5729  023076  012737  000001  001104  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
5730  023104  013737  023154  001204          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
5731  023112  105237  001102          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
5732  023116  011637  001106                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
5733  023122  011637  001110                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
5734  023126  005037  001206                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
5735  023132  112737  000001  001115          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
5736  023140  013777  001102  155774  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
5737  023146  013716  001106                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
5738  023152  000002                          RTI                     ;;FIXES PS
5739  023154  000001                  $MXCNT: 1                       ;;MAX. NUMBER OF ITERATIONS
5740
5741                                  .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES
5742
5743                                  ;;**************************************************************
5744                                  ;*SAVE R0-R5
5745                                  ;*CALL:
5746                                  ;*      SAVREG
5747                                  ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
5748                                  ;*
5749                                  ;*TOP---(+16)
5750                                  ;* +2---(+18)
5751                                  ;* +4---R5
5752                                  ;* +6---R4
5753                                  ;* +8---R3
5754                                  ;*+10---R2
5755                                  ;*+12---R1
5756                                  ;*+14---R0
5757
5758  023156                          $SAVREG:
5759  023156  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
5760  023160  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
5761  023162  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
5762  023164  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
5763  023166  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
5764  023170  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
5765  023172  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
5766  023176  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
5767  023202  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
5768  023206  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
5769  023212  000002                          RTI
577C
5771                                  ;*RESTORE R0-R5
5772                                  ;*CALL:
5773                                  ;*      RESREG
```

```
 5774  023214
 5775  023214  012666  000022          $RESREG:
                                               MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
 5776  023220  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
 5777  023224  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
 5778  023230  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
 5779  023234  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
 5780  023236  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
 5781  023240  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
 5782  023242  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
 5783  023244  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
 5784  023246  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
 5785  023250  000002                          RTI
 5786
 5787                                  .SBTTL  TRAP DECODER
 5788
 5789                                  ;;**********************************************************************
 5790                                  ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
 5791                                  ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
 5792                                  ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
 5793                                  ;*GO TO THAT ROUTINE.
 5794
 5795  023252  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
 5796  023254  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
 5797  023260  005740                          TST     -(R0)           ;;BACKUP BY 2
 5798  023262  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
 5799  023264  006300                          ASL     R0              ;;POSITION FOR INDEXING
 5800  023266  016000  023306                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
 5801  023272  000200                          RTS     R0              ;;GO TO ROUTINE
 5802
 5803
 5804                                  ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
 5805
 5806  023274  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
 5807  023276  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
 5808  023304  000002                          RTI                     ;;RESTORE THE PSW
 5809
 5810                                  .SBTTL  TRAP TABLE
 5811
 5812                                  ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
 5813                                  ;*BY THE "TRAP" INSTRUCTION.
 5814
 5815                                  ;       ROUTINE
 5816                                  ;       -------
 5817  023306  023274                  $TRPAD: .WORD   $TRAP2
 5818  023310  020560                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
 5819  023312  021024                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
 5820  023314  021000                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
 5821  023316  021040                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
 5822  023320  021226                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
 5823
 5824  023322  022040                          $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING
 5825
 5826  023324  021750                          $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
 5827  023326  022312                          $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
 5828  023330  022402                          $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
 5829  023332  023156                          $SAVREG ;;CALL=SAVREG   TRAP+12(104412) SAVE R0-R5 ROUTINE
```

```
 5830  023334  023214                          $RESREG ;;CALL=RESREG   TRAP+13(104413) RESTORE R0-R5 ROUTINE
 5831
 5832                                  .SBTTL  SINGLE LENGTH BINARY TO DECIMAL ASCIZ ROUTINE
 5833
 5834                                  ;;*********************************************************************
 5835                                  ;*THIS ROUTINE WILL CONVERT A 16-BIT UNSIGNED BINARY NUMBER TO AN
 5836                                  ;*UNSIGNED DECIMAL ASCIZ NUMBER.
 5837                                  ;*CALL
 5838                                  ;*      MOV     NUMBER,-(SP)    ;;PUT BINARY NUMBER ON THE STACK
 5839                                  ;*      JSR     PC,@#$SB2D      ;;CALL
 5840                                  ;*      RETURN                  ;;ADDRESS OF THE 1ST ASCIZ CHAR.IS ON THE STACK
 5841
 5842
 5843  023336  016637  000002  023366  $SB2D:  MOV     2(SP),1$        ;;SAVE BINARY NUMBER
 5844  023344  012746  023366                  MOV     #1$,-(SP)       ;;SET POINTER
 5845  023350  004737  023372                  JSR     PC,@#$DB2D      ;;CALL DOUBLE LENGTH CONVERT
 5846  023354  062716  000005                  ADD     #5,(SP)         ;;ONLY ALLOW FIVE CHARACTERS
 5847  023360  012666  000002                  MOV     (SP)+,2(SP)     ;;PICKUP POINTER
 5848  023364  000207                          RTS     PC              ;;RETURN
 5849  023366  000000  000000          1$:     .WORD   0,0
 5850
 5851                                  .SBTTL  DOUBLE LENGTH BINARY TO DECIMAL ASCII CONVERT ROUTINE
 5852
 5853                                  ;;*********************************************************************
 5854                                  ;*THIS ROUTINE WILL CONVERT A 32-BIT BINARY NUMBER TO AN UNSIGNED
 5855                                  ;*DECIMAL (ASCII) NUMBER. THE SIGN OF THE BINARY NUMBER MUST BE
 5856                                  ;*POSITIVE.
 5857                                  ;*CALL
 5858                                  ;*      MOV     #PNTR,-(SP)     ;;POINTER TO LOW WORD OF BINARY NUMBER
 5859                                  ;*      JSR     PC,@#$DB2D
 5860                                  ;*      RETURN                  ;;THE FIRST ADDRESS OF ASCIZ
 5861                                                                  ;;IS ON THE STACK
 5862
 5863
 5864  023372  104412                  $DB2D:  SAVREG                  ;;SAVE REGISTERS
 5865  023374  016602  000002                  MOV     2(SP),R2        ;;PICKUP THE DATA POINTER
 5866  023400  012700  023552                  MOV     #$DECVL,R0      ;;GET ADDRESS OF "$DECVL" STRING
 5867  023404  010066  000002                  MOV     R0,2(SP)        ;;PUT ADDRESS OF ASCIZ STRING ON STACK
 5868  023410  012201                          MOV     (R2)+,R1        ;;PICKUP THE BINARY NUMBER
 5869  023412  012202                          MOV     (R2)+,R2
 5870  023414  012737  000012  023470          MOV     #10.,4$         ;;SET UP TO DO 10 CONVERSIONS
 5871  023422  012704  023502                  MOV     #$TNPWR,R4      ;;ADDRESS OF TEN POWER
 5872  023426  012705  023504                  MOV     #$TNPWR+2,R5
 5873  023432  005003                  1$:     CLR     R3              ;;CLEAR PARTIAL
 5874  023434  161401                  2$:     SUB     (R4),R1         ;;SUBTRACT TEN POWER
 5875  023436  005602                          SBC     R2
 5876  023440  161502                          SUB     (R5),R2
 5877  023442  002402                          BLT     3$              ;;BR IF TEN POWER TO LARGE
 5878  023444  005203                          INC     R3              ;;ADD 1 TO PARTIAL
 5879  023446  000772                          BR      2$              ;;LOOP
 5880  023450  062401                  3$:     ADD     (R4)+,R1        ;;RESTORE SUBTRACTED VALUE
 5881  023452  005502                          ADC     R2
 5882  023454  062402                          ADD     (R4)+,R2
 5883  023456  022525                          CMP     (R5)+,(R5)+     ;;MOVE TO NEXT TEN POWER
 5884  023460  052703  000060                  BIS     #'0,R3          ;;CHANGE PARTIAL TO ASCII
 5885  023464  110320                          MOVB    R3,(R0)+        ;;SAVE IT
```

```
 5886  023466  005327                        DEC     (PC)+           ;;DONE?
 5887  023470  000000                4$:     .WORD   0
 5888  023472  001357                        BNE     1$              ;;BR IF NO
 5889  023474  105020                        CLRB    (R0)+           ;;TERMINATOR
 5890  023476  104413                        RESREG                  ;;RESTORE REGISTERS
 5891  023500  000207                        RTS     PC              ;;RETURN
 5892  023502  145000                $TNPWR: 145000                  ;;1.0E09
 5893  023504  035632                        35632
 5894  023506  160400                        160400                  ;;1.0E08
 5895  023510  002765                        2765
 5896  023512  113200                        113200                  ;;1.0E07
 5897  023514  000230                        230
 5898  023516  041100                        041100                  ;;1.0E06
 5899  023520  000017                        17
 5900  023522  103240                        103240                  ;;1.0E05
 5901  023524  000001                        1
 5902  023526  023420                        23420                   ;;1.0E04
 5903  023530  000000                        0
 5904  023532  001750                        1750                    ;;1.0E03
 5905  023534  000000                        0
 5906  023536  000144                        144                     ;;1.0E02
 5907  023540  000000                        0
 5908  023542  000012                        12                      ;;1.0E01
 5909  023544  000000                        0
 5910  023546  000001                        1                       ;;1.0E00
 5911  023550  000000                        0
 5912  023552  000014                $DECVL: .BLKB   12.             ;;RESERVE STORAGE FOR ASCIZ STRING
 5913
 5914                                  .SBTTL  TYPE NUMERICAL ASCIZ STRING SUPPRESS LEADING ZEROS
 5915
 5916                                  ;;**************************************************************
 5917                                  ;*THIS ROUTINE IS USED TO TYPE AN ASCIZ NUMBER SUPPRESSING THE
 5918                                  ;*LEADING NUMBERS.
 5919                                  ;*CALL
 5920                                  ;*      MOV     #NUMADR,-(SP)   ;;FIRST ADDRESS OF ASCIZ STRING
 5921                                  ;*      JSR     PC,@#$SUPRS
 5922
 5923
 5924  023566  010046                $SUPRS: MOV     R0,-(SP)        ;;SAVE R0
 5925  023570  016600  000004                MOV     4(SP),R0        ;;PICKUP THE POINTER
 5926  023574  105710                1$:     TSTB    (R0)            ;;TERMINATEOR?
 5927  023576  001403                        BEQ     2$              ;;BR IF YES
 5928  023600  122720  000060                CMPB    #'0,(R0)+       ;;IS THIS AN ASCII "0" ?
 5929  023604  001773                        BEQ     1$              ;;BR IF YES
 5930  023606  005300                2$:     DEC     R0              ;;BACKUP BY "1"
 5931  023610  010037  023616                MOV     R0,3$           ;;SAVE FOR TYPING
 5932  023614  104401                        TYPE                    ;;GO TYPE
 5933  023616  000000                3$:     .WORD   0               ;;ASCIZ POINTER GOES HERE
 5934  023620  012600                        MOV     (SP)+,R0        ;;RESTORE R0
 5935  023622  012616                        MOV     (SP)+,(SP)      ;;RESTORE THE STACK
 5936  023624  000207                        RTS     PC              ;;RETURN
 5937
 5938                                  .SBTTL  RANDOM NUMBER GENERATOR ROUTINE
 5939
 5940                                  ;;**************************************************************
 5941                                  ;*THIS ROUTINE IS A DOUBLE PRECISION PSEUDO RANDOM NUMBER GENERATOR
```

```
                                ;*WITH A RANGE OF 0 TO 2(+33)-1.
                                ;*CALL:
                                ;*      JSR     PC,$RAND        ;;CALL THE ROUTINE
                                ;*      RETURN                  ;;RETURN HERE THE RANDOM
                                ;*                              ;;NUMBER WILL BE IN
                                ;*                              ;;$HINUM,$LONUM

023626                          $RAND:
023626  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
023630  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
023632  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
023634  013700  023726                  MOV     $LONUM,R0       ;;SET R0 WITH LOW
023640  013701  023724                  MOV     $HINUM,R1       ;;SET R1 WITH HIGH
023644  012702  177771                  MOV     #-7,R2          ;;SET SHIFT COUNT
023650  006300                  1$:     ASL     R0              ;;SHIFT R0 LEFT AND
023652  006101                          ROL     R1              ;;ROTATE CARRY INTO R1 AND
023654  005202                          INC     R2              ;;CHECK FOR DONE
023656  001374                          BNE     1$              ;;CONTINUE SHIFT LOOP
023660  063700  023726                  ADD     $LONUM,R0       ;;ADD NUMBER TO MAKE X 129
023664  005501                          ADC     R1              ;;PROPOGATE CARRY
023666  063701  023724                  ADD     $HINUM,R1       ;;ADD NUMBER TO MAKE X 129
023672  062700  001057                  ADD     #1057,R0        ;;ADD LOW CONSTANT
023676  005501                          ADC     R1              ;;PROPOGATE CARRY
023700  062701  047401                  ADD     #47401,R1       ;;ADD HIGH CONSTANT
023704  010037  023726                  MOV     R0,$LONUM       ;;SAVE R0
023710  010137  023724                  MOV     R1,$HINUM       ;;SAVE R1
023714  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
023716  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
023720  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
023722  000207                          RTS     PC              ;;RETURN
023724  176543                  $HINUM: .WORD   176543
023726  123456                  $LONUM: .WORD   123456

                                .SBTTL  INTEGER DIVIDE ROUTINE

                                ;;******************************************************************
                                ;*THIS ROUTINE WILL DIVIDE A 32-BIT TWO'S COMPLEMENT INTEGER
                                ;*DIVIDEND BY A 16-BIT TWO'S COMPLEMENT INTEGER DIVISOR GIVING
                                ;*A 16-BIT TWO'S COMPLEMENT INTEGER QUOTIENT AND A 16-BIT REMAINDER.
                                ;*DIVISION WILL BE PERFORMED SO THAT THE REMAINDER IS OF THE
                                ;*SAVE SIGN AS THE DIVIDEND.
                                ;*CALL:
                                ;*      MOV     LOW DIVIDEND,-(SP) ;;THE HIGH DIVIDEND MUST BE < 1/2
                                ;*      MOV     HIGH DIVIDEND,-(SP);    AS LARGE AS THE DIVISOR
                                ;*      MOV     DIVISOR,-(SP)
                                ;*      JSR     PC,$DIV
                                ;*      RETURN                  ;;QUOTIENT & REMAINDER ARE ON THE STACK
                                ;*      "V"=0   IMPLIES NO ERROR
                                ;*      "V"=1   IMPLIES ERROR OCCURRED
                                ;*              "C"=0   DIVIDE OVERFLOW OCCURRED
                                ;*              "C"=1   ATTEMPTED TO DIVIDE BY ZERO
                                ;*
                                ;*
                                ;*      STACK   NO ERROR        OVERFLOW        DIVIDE BY ZERO
                                ;*      -----   --------        --------        --------------
                                ;*      TOP     REMAINDER       ALL ZEROS       ALL ONES
```

```
                                        ;*      +2      QUOTIENT        ALL ZEROS       ALL ONES
                                        $DIV:
023730  104400                          TRAP                            ;;PUSH OLD PSW AND PC ON STACK
023732  042716  000017                  BIC     #17,(SP)                ;;STRIP AWAY CONDITION CODES
023736  010046                          MOV     R0,-(SP)                ;;PUSH R0 ON STACK
023740  010146                          MOV     R1,-(SP)                ;;PUSH R1 ON STACK
023742  010246                          MOV     R2,-(SP)                ;;PUSH R2 ON STACK
023744  010346                          MOV     R3,-(SP)                ;;PUSH R3 ON STACK
023746  005046                          CLR     -(SP)                   ;;SAVE A PLACE FOR SIGNS
023750  012746  000021                  MOV     #17.,-(SP)              ;;SETUP THE ITERATION COUNTER
023754  016601  000024                  MOV     24(SP),R1               ;;PICKUP THE DIVIDEND
023760  016600  000022                  MOV     22(SP),R0
023764  100005                          BPL     1$                      ;;CHECK THE SIGN
023766  105366  000003                  DECB    3(SP)                   ;;KEEP TRACK OF THE SIGN
023772  005400                          NEG     R0                      ;;AND NEGATE THE ORIGINAL
023774  005401                          NEG     R1                      ;;NUMBER
023776  005600                          SBC     R0
024000  016602  000020          1$:     MOV     20(SP),R2               ;;PICKUP THE DIVISOR
024004  002407                          BLT     2$                      ;;CHECK THE SIGN
024006  003011                          BGT     3$                      ;;DIVISOR OF 0 IS A NO-NO
024010  052766  000003  000014          BIS     #3,14(SP)               ;;SET "V" & "C"
024016  012700  177777                  MOV     #-1,R0                  ;;SET REMAINDER TO ALL ONES
024022  000424                          BR      7$                      ;;EXIT
024024  005266  000002          2$:     INC     2(SP)                   ;;KEEP TRACK OF DIVISORS SIGN
024030  000401                          BR      4$
024032  005402                  3$:     NEG     R2                      ;;NEGATE THE ORIGINAL NUMBER
024034  000241                  4$:     CLC                             ;;CLEAR "C"
024036  000405                          BR      6$                      ;;START FORMING QUOTIENT
024040  006100                  5$:     ROL     R0                      ;;POSITION MSB'S
024042  010003                          MOV     R0,R3                   ;;COPY
024044  060203                          ADD     R2,R3                   ;;COMPARE DIVIDEND & DIVISOR
024046  103001                          BCC     6$                      ;;BR IF DIVIDEND > DIVISOR
024050  010300                          MOV     R3,R0                   ;;REMAINDER AFTER THIS LOOP
024052  006101                  6$:     ROL     R1                      ;;QUOTIENT BIT ENTERS HERE
024054  005316                          DEC     (SP)                    ;;DONE?
024056  001370                          BNE     5$                      ;;BR IF NO
024060  005701                          TST     R1                      ;;OVERFLOW?
024062  100005                          BPL     8$                      ;;BR IF NO
024064  052766  000002  000014          BIS     #2,14(SP)               ;;SET "V" IN RETURN STATUS WORD
024072  005000                          CLR     R0                      ;;SET REMAINDER TO ALL ZEROS
024074  010001                  7$:     MOV     R0,R1                   ;;COPY REMAINDER INTO QUOTIENT
024076  005726                  8$:     TST     (SP)+                   ;;CLEAR COUNTER FROM STACK
024100  005716                          TST     (SP)                    ;;REMAINDER SIGN CORRECTION NEEDED?
024102  002004                          BGE     9$                      ;;BR IF NO
024104  005400                          NEG     R0                      ;;NEGATE REMAINDER
024106  105066  000001                  CLRB    1(SP)                   ;;CLEAR SIGN
024112  005316                          DEC     (SP)                    ;;BUT DON'T FORGET QUOTIENT
024114  005726                  9$:     TST     (SP)+                   ;;QUOTIENT SIGN CORRECTION NEEDED?
024116  001401                          BEQ     10$                     ;;BR IF NO
024120  005401                          NEG     R1                      ;;NEGATE QUOTIENT
024122  010166  000020          10$:    MOV     R1,20(SP)               ;;RETURN QUOTIENT AND
024126  010066  000016                  MOV     R0,16(SP)               ;;REMAINDER TO USER
024132  012603                          MOV     (SP)+,R3                ;;POP STACK INTO R3
024134  012602                          MOV     (SP)+,R2                ;;POP STACK INTO R2
024136  012601                          MOV     (SP)+,R1                ;;POP STACK INTO R1
```

```
6054  024140  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
6055  024142  012666  000002                  MOV     (SP)+,2(SP)     ;;SETUP TO RETURN CONDITION CODES
6056  024146  000002                          RTI                     ;;RETURN
6057
6058                                .SBTTL  *** PROGRAM SUBROUTINES ***
6059
6060                                ;SET "LPTAVL" TO THE PROPER STATE.
6061                                ; LPTAVL = 0 IF NO LINE PRINTER AVAILABLE
6062                                ; LPTAVL = 1 IF LINE PRINTER IS AVAILABLE
6063                                ;CALL
6064                                ;       JSR     PC,@#LP.AVL
6065                                ;       RETURN
6066
6067  024150  005037  001230        LP.AVL: CLR     @#LPTAVL        ;START WITH NO PRINTER AVAIABLE
6068  024154  012737  024200  000004        MOV     #1$,@#ERRVEC    ;SETUP THE TIMEOUT VECTOR
6069  024162  005037  000006                CLR     @#ERRVEC+2
6070  024166  005777  155226                TST     @LPS            ;IS THERE A LINE PRINTER?
6071  024172  005237  001230                INC     @#LPTAVL        ;YES--SET AVAILABLE SWITCH
6072  024176  000401                        BR      2$
6073  024200  022626                1$:     CMP     (SP)+,(SP)+     ;NO--POP STACK
6074  024202  012737  000006  000004  2$:     MOV     #ERRVEC+2,@#ERRVEC ;RESTORE TIMEOUT VECTOR
6075  024210  000207                        RTS     PC              ;RETURN
6076
6077                                ;THIS ROUTINE WILL DETERMINE IF THERE IS A CLOCK ON THE SYSTEM
6078                                ;AND IF THERE IS IT WILL SETUP THE VECTOR AND START THE CLOCK
6079                                ;"CLKSTA" WILL INDICATE THE CLOCK TYPE
6080                                ; 0= NO CLOCK
6081                                ;+1= KW11-P
6082                                ;-1= KW11-L
6083                                ;THIS ROUTINE WILL ALSO SETUP "TICKMS" (TIME
6084                                ;PER CLOCK TICK IN MILLISECONDS) AND "TICKUS"
6085                                ;(TIME PER CLOCK TICK IN MICROSECONDS) AS
6086                                ;PER SW00.
6087                                ;SW00=0 -- 60HZ
6088                                ;SW00=1 -- 50HZ
6089                                ;CALL
6090                                ;       JSR     PC,@#ST.CLK
6091                                ;       RETURN
6092
6093  024212  010146                ST.CLK: MOV     R1,-(SP)        ;SAVE R1
6094  024214  012701  000006                MOV     #ERRVEC+2,R1    ;SAVE AND SETUP TIMEOUT VECTOR
6095  024220  011146                        MOV     (R1),-(SP)
6096  024222  005011                        CLR     (R1)            ;LEVEL 0
6097  024224  014146                        MOV     -(R1),-(SP)
6098  024226  012711  024256                MOV     #1$,(R1)        ;GO TO 1$ ON TIMEOUT
6099  024232  005037  001244                CLR     CLKSTA          ;SET CLOCK STATUS TO NO CLOCK
6100  024236  005777  155136                TST     @PKCS           ;IS THERE A KW11-P?
6101  024242  012737  000001  001244        MOV     #1,CLKSTA       ;YES--SET STATUS TO KW11-P
6102  024250  004737  024360                JSR     PC,ST.PCLK      ;START THE KW11-P
6103  024254  000414                        BR      3$              ;GO TO EXIT
6104  024256  022626                1$:     CMP     (SP)+,(SP)+     ;CLEAN UP THE STACK
6105  024260  012711  024304                MOV     #2$,(R1)        ;IF TIMEOUT GO TO 2$
6106  024264  005777  155122                TST     @LKS            ;IS THERE A KW11-L?
6107  024270  012737  177777  001244        MOV     #-1,CLKSTA      ;YES-- SET STATUS TO KW11-L
6108  024276  004737  024422                JSR     PC,ST.LCLK      ;START THE KW11-L
6109  024302  000401                        BR      3$              ;EXIT
```

```
6110  024304  022626                  2$:     CMP     (SP)+,(SP)+     ;CLEAN UP THE STACK
6111  024306  012621                  3$:     MOV     (SP)+,(R1)+     ;RESTORE THE TIMEOUT VECTOR
6112  024310  012621                          MOV     (SP)+,(R1)+
6113  024312  012601                          MOV     (SP)+,R1        ;RESTORE R1
6114  024314  032737  000100  001220          BIT     #SW06,@#C.SWR   ;50HZ OR 60HZ?
6115  024322  001407                          BEQ     4$              ;BRANCH IF 60
6116  024324  012737  000020  001246          MOV     #20,@#TICKMS    ;SETUP TIME PER
6117  024332  012737  047040  001250          MOV     #20000.,@#TICKUS ;TICK FOR 50HZ
6118  024340  000406                          BR      5$
6119  024342  012737  000016  001246  4$:     MOV     #16,@#TICKMS    ;SETUP TIME PER
6120  024350  012737  040432  001250          MOV     #16666.,@#TICKUS ;TICK FOR 60HZ
6121  024356  000207                  5$:     RTS     PC              ;RETURN
6122
6123  024360                          ST.PCLK:
6124  024360  032737  000040  001220          BIT     #SW05,@#C.SWR   ;ALLOW SOFTWARE TIMEOUTS?
6125  024366  001014                          BNE     1$              ;NO--BRANCH
6126  024370  012777  024456  154776          MOV     #SRVCLK,@PKV    ;SETUP THE KW11-P VECTOR
6127  024376  012777  000300  154772          MOV     #300,@PKV+2
6128  024404  012777  000001  154770          MOV     #1,@PKB         ;COUNT ONE TICK
6129  024412  012777  000115  154760          MOV     #115,@PKCS      ;''INT.EN.'',COUNT DOWN'', ''MODE 1 (REPEAT)'',
6130                                                                  ;''LINE FREQ'', AND ''RUN''
6131  024420  000207                  1$:     RTS     PC              ;RETURN
6132
6133  024422                          ST.LCLK:
6134  024422  032737  000040  001220          BIT     #SW05,@#C.SWR   ;ALLOW SOFTWARE TIMEOUTS?
6135  024430  001011                          BNE     1$              ;NO--BRANCH
6136  024432  012777  024456  154746          MOV     #SRVCLK,@LKV    ;SETUP THE KW11-L VECTOR
6137  024440  012777  000300  154742          MOV     #300,@LKV+2
6138  024446  012777  000100  154736          MOV     #100,@LKS       ;START THE KW11-L
6139  024454  000207                  1$:     RTS     PC              ;RETURN
6140
6141  024456  013746  001246          SRVCLK: MOV     @#TICKMS,-(SP)  ;TIME PER TICK IN MILLISECONDS
6142  024462  004737  041102                  JSR     PC,@#RPTMR      ;COUNT THE ELASPED TIME
6143  024466  000002                          RTI                     ;RETURN AFTER INTERRUPT
6144
6145                                  ;THIS ROUTINE SETS UP DEFAULT PARAMETER VALUES WHEN THE PROGRAM IS
6146                                  ;STARTED OR WHEN THE VALUE OF BIT00 IN 'C.SWR' IS CHANGED.
6147                                  ;CALL
6148                                  ;       JSR     PC,LODFLT
6149                                  ;       RETURN
6150
6151  024470                          LODFLT:
6152  024470  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
6153  024472  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
6154  024474  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
6155  024476  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
6156  024500  012737  176777  001234          MOV     #176777,TSTNMS  ;SELECT TESTS 0-10, 12-17
6157  024506  012737  000001  001236          MOV     #1,TSTNMS+2     ;SET SELECT BIT FOR TEST 20
6158  024514  012700  001664                  MOV     #DFLT,R0        ;DEFAULT PARAMETERS POINTER
6159  024520  012701  002330                  MOV     #PRM0,R1        ;TABLE POINTER
6160  024524  010102                          MOV     R1,R2           ;STOP ADDRESS
6161  024526  012021                  1$:     MOV     (R0)+,(R1)+     ;MOVE DEFAULT PARAMETERS INTO
6162  024530  020002                          CMP     R0,R2           ;RUN TIME TABLES ** DONE?
6163  024532  103775                          BLO     1$              ;NO--BRANCH
6164  024534  012700  003504                  MOV     #PAT8,R0        ;PAT0 DEFAULTS TO PATTERN 8
6165  024540  012701  003104                  MOV     #PAT0,R1
```

```
6166  024544  012021                    2$:     MOV     (R0)+,(R1)+
6167  024546  020027  003544                    CMP     R0,#PAT9
6168  024552  103774                            BLO     2$
6169  024554  032737  000001  001220            BIT     #BIT00,C.SWR    ;16 BIT MODE ?
6170  024562  001012                            BNE     3$              ;BR IF 18
6171  024564  012737  000025  001630            MOV     #21.,PRMLMT+22  ;SET 'FS' LIMIT TO 21.
6172  024572  012737  000025  001632            MOV     #21.,PRMLMT+24  ;SET 'LS' LIMIT TO 21.
6173  024600  012737  165000  001352            MOV     #-<256.*22.>,TRCKWC  ;WORD COUNT FOR A 16 BIT TRACK
6174  024606  000411                            BR      4$              ;CONTINUE
6175  024610  012737  000023  001630  3$:       MOV     #19.,PRMLMT+22  ;SET 'FS' LIMIT TO 19.
6176  024616  012737  000023  001632            MOV     #19.,PRMLMT+24  ;SET 'LS' LIMIT TO 19.
6177  024624  012737  166000  001352            MOV     #-<256.*20.>,TRCKWC  ;WORD COUNT FOR COUNT FOR AN 18 BIT TRACK
6178  024632  012701  001536            4$:     MOV     #PRMPT,R1       ;ADDRESS OF PARAMETER POINTER TABLE
6179  024636  005711                    5$:     TST     (R1)            ;END OF THE TABLE ?
6180  024640  001425                            BEQ     8$              ;BR IF END
6181  024642  032731  002000                    BIT     #BIT10,@(R1)+   ;'LS' SELECTED ?
6182  024646  001773                            BEQ     5$              ;BR IF NOT
6183  024650  016102  177776                    MOV     -2(R1),R2       ;PARAMETER TABLE ADDRESS
6184  024654  011246                            MOV     (R2),-(SP)      ;PARAMETER ALLOCATION BITS
6185  024656  012703  000013                    MOV     #11.,R3         ;NUMBER OF PARAMETERS (MAXIMUM) BEFORE 'LS'
6186  024662  006216                    6$:     ASR     (SP)            ;COUNT THE PARAMETER
6187  024664  103002                            BCC     7$              ;BR IF NOT USED
6188  024666  062702  000002                    ADD     #2,R2           ;INCREMENT THE PARAMETER TABLE ADDRESS
6189  024672  005303                    7$:     DEC     R3              ;COUNT THE PARAMETER
6190  024674  001372                            BNE     6$              ;BR IF NOT THERE YET
6191  024676  005726                            TST     (SP)+           ;CORRECT THE STACK POINTER
6192  024700  021237  001630                    CMP     (R2),PRMLMT+22  ;IS 'LS' TOO LARGE FOR THE MODE SELECTED ?
6193  024704  101754                            BLOS    5$              ;BR IF NOT
6194  024706  013712  001630                    MOV     PRMLMT+22,(R2)  ;RESET VALUE FOR MODE USED
6195  024712  000751                            BR      5$              ;CONTINUE
6196  024714                            8$:
6197  024714  012603                            MOV     (SP)+,R3        ;;POP STACK INTO R3
6198  024716  012602                            MOV     (SP)+,R2        ;;POP STACK INTO R2
6199  024720  012601                            MOV     (SP)+,R1        ;;POP STACK INTO R1
6200  024722  012600                            MOV     (SP)+,R0        ;;POP STACK INTO R0
6201  024724  000207                            RTS     PC              ;RETURN
6202
6203                                    ;THIS ROUTINE FILLS THE PARAMETER TABLE THE CURRENT TEST.
6204                                    ;CALL
6205                                    ;       MOV     #TESTNUM,$TSTNM  ;LOAD THE TEST NUMBER
6206                                    ;       JSR     PC,LODPRM
6207                                    ;       RETURN
6208
6209  024726                            LODPRM:
6210  024726  010146                            MOV     R1,-(SP)        ;;PUSH R1 ON STACK
6211  024730  010246                            MOV     R2,-(SP)        ;;PUSH R2 ON STACK
6212  024732  010346                            MOV     R3,-(SP)        ;;PUSH R3 ON STACK
6213  024734  010446                            MOV     R4,-(SP)        ;;PUSH R4 ON STACK
6214  024736  005004                            CLR     R4              ;CLEAR R4
6215  024740  113704  001102                    MOVB    $TSTNM,R4       ;GET THE TEST NUMBER
6216  024744  006304                            ASL     R4              ;SETUP TO ADDRESS WORDS
6217  024746  016401  001536                    MOV     PRMPT(R4),R1    ;GET THE TEST'S PARAMETER TABLE ADDRESS
6218  024752  012702  001504                    MOV     #PRM,R2         ;PARAMETER EXECUTION TABLE
6219  024756  005003                            CLR     R3              ;R3 IS USED AS A COUNTER
6220  024760  013704  001254                    MOV     CHKDRV,R4       ;DRIVE'S ADDRESS
6221  024764  012122                            MOV     (R1)+,(R2)+     ;PARAMETER SPECIFIER
```

```
 6222  024766  006237  001504          1$:     ASR     PRM             ;THIS PARAMETER USED IN THE TEST ?
 6223  024772  103002                          BCC     2$              ;BR IF NOT
 6224  024774  012122                          MOV     (R1)+,(R2)+     ;LOAD THE VALUE
 6225  024776  000401                          BR      3$              ;CONTINUE
 6226  025000  005022                  2$:     CLR     (R2)+           ;CLEAR THE UNUSED PARAMETER LOCATION
 6227  025002  005203                  3$:     INC     R3              ;COUNT THE POSITION IN THE OUTPUT TABLE
 6228  025004  020327  000014                  CMP     R3,#12.         ;FINISHED ?
 6229  025010  001430                          BEQ     6$              ;BR IF YES
 6230  025012  020327  000003                  CMP     R3,#3           ;DOING THE CYLINDER ADDRESSES ?
 6231  025016  001363                          BNE     1$              ;BR IF NOT
 6232  025020  132764  000004  034514          BITB    #BIT02,DRVTYP(R4)  ;RP06 ?
 6233  025026  001016                          BNE     5$              ;BR IF IT IS
 6234  025030  062703  000002                  ADD     #2,R3           ;COUNT THE BYPASSED PARAMETERS (FC' & LC')
 6235  025034  006237  001504                  ASR     PRM             ;SHIFT THE COUNTER
 6236  025040  103002                          BCC     4$              ;BR IF FC' IS NOT USED
 6237  025042  062701  000002                  ADD     #2,R1           ;MOVE THE INPUT POINTER
 6238  025046  006237  001504          4$:     ASR     PRM             ;COUNT THE PARAMETER
 6239  025052  103345                          BCC     1$              ;BR IF LC' NOT USED
 6240  025054  062701  000002                  ADD     #2,R1           ;MOVE THE INPUT PINTER
 6241  025060  000742                          BR      1$              ;KEEP GOING
 6242  025062  000741                          BR      1$              ;KEEP GOING
 6243  025064  162702  000004          5$:     SUB     #4,R2           ;BACKUP THE OUTPUT POINTER
 6244  025070  000736                          BR      1$              ;KEEP GOING
 6245  025072                          6$:
 6246  025072  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
 6247  025074  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
 6248  025076  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
 6249  025100  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
 6250  025102  000207                          RTS     PC              ;RETURN
 6251
 6252                                  ;THIS ROUTINE LOADS A READ HEADER AND DATA COMMAND OR A SEEK COMMAND
 6253                                  ;INTO DPB.B+2 AND DPB.C+2, DEPENDING ON THE STATE OF "CONTROL SWITCH"
 6254                                  ;BIT07.
 6255                                  ;CALL
 6256                                  ;       JSR     PC,@#LDCMD
 6257                                  ;       RETURN
 6258
 6259  025104  032737  000200  001220  LDCMD:  BIT     #SW07,@#C.SWR   ;DO EXPLICIT SEEKS?
 6260  025112  001007                          BNE     1$              ;YES--BRANCH
 6261  025114  012737  000173  004126          MOV     #READHD,@#DPB.B+2 ;NO--SET UP FOR READ HEADER AND
 6262  025122  012737  000173  004146          MOV     #READHD,@#DPB.C+2 ;DATA COMMAND
 6263  025130  000406                          BR      2$
 6264  025132  012737  000105  004126  1$:     MOV     #SEEK,@#DPB.B+2 ;SETUP FOR SEEK COMMAND
 6265  025140  012737  000105  004146          MOV     #SEEK,@#DPB.C+2
 6266  025146  000207                  2$:     RTS     PC
 6267
 6268                                  ;THIS ROUTINE WILL CALL THE RP04/5/6 DRIVER AND THEN WAIT ON THE FUNCTION
 6269                                  ;TO COMPLETE. IF AN ERROR OCCURS IT IS REPORTED.
 6270                                  ;CALL
 6271                                  ;       FILL "DPB" WITH COMMAND INFORMATION
 6272                                  ;       JSR     R0,@#CALL.A
 6273                                  ;       RETURN
 6274
 6275  025150  005037  001206          CALL.A: CLR     @#$ESCAPE       ;NO ESCAPE ADDRESS
 6276  025154  004037  035450                  JSR     R0,@#RP04       ;CALL RP04 DRIVER
 6277  025160  004104                          DPB.A
```

```
6278  025162  000772                          BR      CALL.A
6279  025164  005737  004122          1$:     TST     @#DPB.A+16      ;DONE?
6280  025170  001775                          BEQ     1$              ;NO--LOOP
6281  025172  100032                          BPL     3$              ;BRANCH IF NO ERROR
6282  025174  012737  025250  001206          MOV     #2$,$ESCAPE     ;;ESCAPE TO 2$ ON ERROR
6283  025202  013737  004116  001270          MOV     @#DPB.A+12,@#CYL.DS ;CYLINDER
6284  025210  113737  004115  001274          MOVB    @#DPB.A+11,@#TRK.DS ;TRACK
6285  025216  113737  004114  001272          MOVB    @#DPB.A+10,@#SEC.DS ;SECTOR
6286  025224  012746  004122                  MOV     #DPB.A+16,-(SP) ;STATUS/ERROR INDICATOR ADDRESS
6287  025230  004737  026364                  JSR     PC,@#ERINDX     ;FORM DISPATCH INDEX
6288  025234  062607                          ADD     (SP)+,PC        ;REPORT PROPER ERROR
6289  025236  104041                          ERROR   41              ;
6290  025240  104042                          ERROR   42              ;PARITY ERROR
6291  025242  104043                          ERROR   43              ;UNSAFE ERROR
6292  025244  104044                          ERROR   44              ;NON-I/O ERROR
6293  025246  104045                          ERROR   45              ;I/O ERROR
6294  025250  013746  004122          2$:     MOV     DPB.A+16,-(SP)  ;STATUS WORD
6295  025254  004737  026324                  JSR     PC,LOP.CK       ;SEE IF LOOP, ABORT, OR CONTINUE
6296  025260  000200                  3$:     RTS     R0              ;RETURN
6297
6298                                  ;THIS ROUTINE IS THE SAME AS "CALL.A" EXCEPT FOR THE DPB USED AND IF
6299                                  ;THE COMMAND IS A READ HEADER AND DATA THE HEADER (CYLINDER, TRACK,
6300                                  ;AND SECTOR) READ IS CHECKED FOR VALIDITY.
6301                                  ;CALL
6302                                  ;       FILL DPB
6303                                  ;       JSR     R0,@#CALL.B
6304                                  ;       RETURN
6305
6306  025262  005037  001206          CALL.B: CLR     @#$ESCAPE       ;NO ESCAPE ADDRESS
6307  025266  004037  035450                  JSR     R0,@#RP04       ;CALL RP04 DRIVER
6308  025272  004124                          DPB.B
6309  025274  000772                          BR      CALL.B
6310  025276  005737  004142          1$:     TST     DPB.B+16        ;DONE?
6311  025302  001775                          BEQ     1$              ;NO--BRANCH
6312  025304  100042                          BPL     4$              ;BRANCH IF NO ERROR
6313  025306  012737  025400  001206          MOV     #3$,$ESCAPE     ;;ESCAPE TO 3$ ON ERROR
6314  025314  013737  004136  001270          MOV     @#DPB.B+12,@#CYL.DS ;CYLINDER
6315  025322  113737  004135  001274          MOVB    @#DPB.B+11,@#TRK.DS ;TRACK
6316  025330  113737  004134  001272          MOVB    @#DPB.B+10,@#SEC.DS ;SECTOR
6317  025336  012746  004142                  MOV     #DPB.B+16,-(SP) ;STATUS/ERROR INDICATOR ADDRESS
6318  025342  004737  026364                  JSR     PC,@#ERINDX     ;FORM DISPATCH INDEX
6319  025346  062607                          ADD     (SP)+,PC        ;REPORT PROPER ERROR
6320  025350  104041                          ERROR   41              ;
6321  025352  104042                          ERROR   42              ;PARITY ERROR
6322  025354  104043                          ERROR   43              ;UNSAFE ERROR
6323  025356  104044                          ERROR   44              ;NON-I/O ERROR
6324  025360  005737  004220                  TST     RP.REG+RPER1    ;DRIVE ERROR ?
6325  025364  001404                          BEQ     2$              ;BR IF NOT
6326  025366  032737  177677  004220          BIT     #^C100,RP.REG+RPER1  ;SEE IF ONLY 'HCE' SET
6327  025374  001406                          BEQ     4$              ;BR IF IT IS
6328  025376  104045                  2$:     ERROR   45              ;I/O ERROR
6329  025400  013746  004142          3$:     MOV     DPB.B+16,-(SP)  ;STATUS WORD
6330  025404  004737  026324                  JSR     PC,LOP.CK       ;SEE IF LOOP, ABORT, OR CONTINUE
6331  025410  000410                          BR      5$              ;CHECK FOR STALL
6332  025412  123727  004126  000173  4$:     CMPB    @#DPB.B+2,#READHD ;DOING IMPLIED SEEKS?
6333  025420  001004                          BNE     5$              ;NO--BRANCH
```

```
6334  025422  004037  026644                  JSR     R0,@#VERIFY     ;YES--GO CHECK THE DATA
6335  025426  004134                          DPB.B+10
6336  025430  000407                          BR      6$              ;ERROR DURING VERIFY
6337  025432  032737  040000  001220  5$:     BIT     #SW14,@#C.SWR   ;STALL?
6338  025440  001403                          BEQ     6$              ;NO--BRANCH
6339  025442  004037  026502                  JSR     R0,@#STALL      ;YES--CALL STALL ROUTINE
6340  025446  001354                          .WORD   STALL1          ;STALL TIME POINTER
6341  025450  000200                  6$:     RTS     R0              ;RETURN
6342
6343                                  ;THIS ROUTINE IS THE SAME AS "CALL.B" EXCEPT FOR THE DPB USED.
6344                                  ;CALL
6345                                  ;       FILL DPB
6346                                  ;       JSR     R0,@#CALL.C
6347                                  ;       RETURN
6348
6349  025452  005037  001206          CALL.C: CLR     @#$ESCAPE       ;NO ESCAPE ADDRESS
6350  025456  004037  035450                  JSR     R0,@#RP04       ;CALL RP04 DRIVER
6351  025462  004144                          DPB.C
6352  025464  000772                          BR      CALL.C
6353  025466  005737  004162          1$:     TST     @#DPB.C+16      ;DONE?
6354  025472  001775                          BEQ     1$              ;NO--LOOP
6355  025474  100042                          BPL     4$              ;YES--BRANCH IF NO ERROR
6356  025476  012737  025570  001206          MOV     #3$,$ESCAPE     ;;ESCAPE TO 3$ ON ERROR
6357  025504  013737  004156  001270          MOV     @#DPB.C+12,@#CYL.DS ;CYLINDER
6358  025512  113737  004155  001274          MOVB    @#DPB.C+11,@#TRK.DS ;TRACK
6359  025520  113737  004154  001272          MOVB    @#DPB.C+10,@#SEC.DS ;SECTOR
6360  025526  012746  004162                  MOV     #DPB.C+16,-(SP) ;STATUS/ERROR INDICATOR ADDRESS
6361  025532  004737  026364                  JSR     PC,@#ERINDX     ;FORM DISPATCH INDEX
6362  025536  062607                          ADD     (SP)+,PC        ;REPORT PROPER ERROR
6363  025540  104041                          ERROR   41              ;
6364  025542  104042                          ERROR   42              ;PARITY ERROR
6365  025544  104043                          ERROR   43              ;UNSAFE ERROR
6366  025546  104044                          ERROR   44              ;NON-I/O ERROR
6367  025550  005737  004220                  TST     RP.REG+RPER1    ;DRIVE ERROR ?
6368  025554  001404                          BEQ     2$              ;BR IF NOT
6369  025556  032737  177677  004220          BIT     #^C100,RP.REG+RPER1  ;SEE IF ONLY 'HCE' SET
6370  025564  001406                          BEQ     4$              ;BR IF IT IS
6371  025566  104045                  2$:     ERROR   45              ;I/O ERROR
6372  025570  013746  004162          3$:     MOV     DPB.C+16,-(SP)  ;STATUS WORD
6373  025574  004737  026324                  JSR     PC,LOP.CK       ;SEE IF LOOP, ABORT, OR CONTINUE
6374  025600  000410                          BR      5$
6375  025602  123727  004146  000173  4$:     CMPB    @#DPB.C+2,#READHD ;DOING IMPLIED SEEK?
6376  025610  001004                          BNE     5$              ;NO--EXIT
6377  025612  004037  026644                  JSR     R0,@#VERIFY     ;YES--CHECK THE DATA
6378  025616  004154                          DPB.C+10
6379  025620  000407                          BR      6$              ;ERROR DURING VERIFY
6380  025622  032737  040000  001220  5$:     BIT     #SW14,@#C.SWR   ;STALL?
6381  025630  001403                          BEQ     6$              ;NO--BRANCH
6382  025632  004037  026502                  JSR     R0,@#STALL      ;YES--CALL STALL ROUTINE
6383  025636  001354                          .WORD   STALL1          ;STALL TIME POINTER
6384  025640  000200                  6$:     RTS     R0
6385
6386
6387                                  ;THIS ROUTINE IS THE SAME AS "CALL.A" EXCEPT FOR THE DPB USED AND
6388                                  ;ON AN ERROR LOCATION "ERR.CT" IS EXAMINED. IF ERR.CT IS EQUAL TO
6389                                  ;$ERFLG EXIT IS TO THE NEXT TEST.
```

```
                                        ;CALL
                                        ;       FILL DPB
                                        ;       JSR     R0,@#DRVCAL
                                        ;       RETURN

025642  005037  001206          DRVCAL: CLR     @#$ESCAPE       ;NO ESCAPE ADDRESS
025646  005037  001334                  CLR     @#WCEFLG        ;CLEAR WRITE CHECK ERROR FLAG
025652  004037  035450                  JSR     R0,@#RP04       ;CALL RP04 DRIVER
025656  004164                          DTADPB
025660  000770                          BR      DRVCAL
025662  005737  004202          DRVCL1: TST     @#DTADPB+16     ;DONE
025666  001775                          BEQ     DRVCL1          ;NO--LOOP
025670  100402                          BMI     1$              ;BR IF ERRORS
025672  000137  026304                  JMP     10$             ;NO ERRORS
025676                          1$:
025676  012737  025752  001206          MOV     #2$,$ESCAPE     ;;ESCAPE TO 2$ ON ERROR
025704  013737  004176  001270          MOV     @#DTADPB+12,@#CYL.DS ;CYLINDER
025712  113737  004175  001274          MOVB    @#DTADPB+11,@#TRK.DS ;TRACK
025720  113737  004174  001272          MOVB    @#DTADPB+10,@#SEC.DS ;SECTOR
025726  012746  004202                  MOV     #DTADPB+16,-(SP) ;STATUS/ERROR INDICATOR ADDRESS
025732  004737  026364                  JSR     PC,@#ERINDX     ;FORM DISPATCH INDEX
025736  062607                          ADD     (SP)+,PC        ;REPORT PROPER ERROR
025740  104041                          ERROR   41              ;
025742  104042                          ERROR   42              ;PARITY ERROR
025744  104043                          ERROR   43              ;UNSAFE ERROR
025746  104044                          ERROR   44              ;NON-I/O ERROR
025750  104045                          ERROR   45              ;I/O ERROR
025752  122737  000020  001102  2$:     CMPB    #20,@#$TSTNM    ;TEST 20?
025760  001137                          BNE     8$              ;NO--BRANCH
025762  013746  004202                  MOV     DTADPB+16,-(SP) ;STATUS WORD
025766  004737  026324                  JSR     PC,LOP.CK       ;SEE IF LOOP, ABORT, OR CONTINUE
025772  122737  000151  004166          CMPB    #WRCKD,@#DTADPB+2 ;DOING A WRITE CHECK?
026000  001133                          BNE     12$             ;NO--BRANCH
026002  032737  040000  004214          BIT     #BIT14,@#RP.REG+10 ;IS "WCE"=1?
026010  001527                          BEQ     12$             ;NO--BRANCH
026012  032777  000020  153120          BIT     #SW04,@SWR      ;INHIBIT WRITES?
026020  001123                          BNE     12$             ;YES--BRANCH
026022  112737  000161  004166          MOVB    #WRITE,@#DTADPB+2 ;SETUP FOR A WRITE
026030  005037  001206                  CLR     @#$ESCAPE       ;NO ESCAPE ADDRESS
026034  004037  035450                  JSR     R0,@#RP04       ;DO THE WRITE
026040  004164                          DTADPB
026042  000240                          NOP
026044  005737  004202          3$:     TST     @#DTADPB+16     ;DONE?
026050  001775                          BEQ     3$              ;NO--LOOP
026052  100026                          BPL     4$              ;YES--BRANCH IF NO ERROR
026054  012737  026260  001206          MOV     #8$,$ESCAPE     ;;ESCAPE TO 8$ ON ERROR
026062  013737  004176  001270          MOV     @#DTADPB+12,@#CYL.DS ;CYLINDER
026070  113737  004175  001274          MOVB    @#DTADPB+11,@#TRK.DS ;TRACK
026076  113737  004174  001272          MOVB    @#DTADPB+10,@#SEC.DS ;SECTOR
026104  012746  004202                  MOV     #DTADPB+16,-(SP) ;STATUS/ERROR INDICATOR ADDRESS
026110  004737  026364                  JSR     PC,@#ERINDX     ;FORM DISPATCH INDEX
026114  062607                          ADD     (SP)+,PC        ;REPORT PROPER ERROR
026116  104041                          ERROR   41              ;
026120  104042                          ERROR   42              ;PARITY ERROR
026122  104043                          ERROR   43              ;UNSAFE ERROR
026124  104044                          ERROR   44              ;NON-I/O ERROR
```

```
 6446  026126  104045                          ERROR   45              ;I/O ERROR
 6447  026130  112737  000151  004166  4$:     MOVB    #WRCKD,@#DTADPB+2 ;COMMAND=WRITE CHECK DATA
 6448  026136  004037  035450                  JSR     R0,@#RP04       ;DO THE WRITE CHECK
 6449  026142  004164                          DTADPB
 6450  026144  000240                          NOP
 6451  026146  005737  004202          5$:     TST     @#DTADPB+16     ;DONE?
 6452  026152  001775                          BEQ     5$              ;NO--LOOP
 6453  026154  100410                          BMI     7$              ;YES--BRANCH IF ERROR
 6454  026156  004037  035450                  JSR     R0,@#RP04       ;DO A 2ND WRITE CHECK
 6455  026162  004164                          DTADPB
 6456  026164  000240                          NOP
 6457  026166  005737  004202          6$:     TST     @#DTADPB+16     ;DONE?
 6458  026172  001775                          BEQ     6$              ;NO--LOOP
 6459  026174  100043                          BPL     10$             ;YES--BRANCH IF NO ERROR
 6460  026176  012737  000001  001334  7$:     MOV     #1,@#WCEFLG     ;SET THE WRITE CHECK ERROR FLAG
 6461  026204  012737  026260  001206          MOV     #8$,$ESCAPE     ;;ESCAPE TO 8$ ON ERROR
 6462  026212  013737  004176  001270          MOV     @#DTADPB+12,@#CYL.DS ;CYLINDER
 6463  026220  113737  004175  001274          MOVB    @#DTADPB+11,@#TRK.DS ;TRACK
 6464  026226  113737  004174  001272          MOVB    @#DTADPB+10,@#SEC.DS ;SECTOR
 6465  026234  012746  004202                  MOV     #DTADPB+16,-(SP) ;STATUS/ERROR INDICATOR ADDRESS
 6466  026240  004737  026364                  JSR     PC,@#ERINDX     ;FORM DISPATCH INDEX
 6467  026244  062607                          ADD     (SP)+,PC        ;REPORT PROPER ERROR
 6468  026246  104041                          ERROR   41              ;
 6469  026250  104042                          ERROR   42              ;PARITY ERROR
 6470  026252  104043                          ERROR   43              ;UNSAFE ERROR
 6471  026254  104044                          ERROR   44              ;NON-I/O ERROR
 6472  026256  104046                          ERROR   46              ;FATAL WRITE CHECK
 6473  026260  013746  004202          8$:     MOV     DTADPB+16,-(SP) ;STATUS WORD
 6474  026264  004737  026324                  JSR     PC,LOP.CK       ;SEE IF LOOP, ABORT, OR CONTINUE
 6475  026270  123737  001364  001103  12$:    CMPB    @#ERR.CT,@#$ERFLG ;GO TO NEXT TEST?
 6476  026276  101002                          BHI     10$             ;NO--BRANCH
 6477  026300  013700  001252          9$:     MOV     @#BYPASS,R0     ;YES--GET EXIT ADDRESS
 6478  026304  032737  040000  001220  10$:    BIT     #SW14,@#C.SWR   ;STALL?
 6479  026312  001403                          BEQ     11$             ;NO--BRANCH
 6480  026314  004037  026502                  JSR     R0,@#STALL      ;YES--CALL STALL ROUTINE
 6481  026320  001356                          .WORD   STALL2          ;STALL TIME POINTER
 6482  026322  000200                  11$:    RTS     R0
 6483
 6484                                  ;THIS SUBROUTINE CHECK FOR LOOP, ABORT, OR CONTINUE SWITCHES AFTER
 6485                                  ;ERRORS 41, 42, 43, 44, 45, AND 46.
 6486                                  ;CALL
 6487                                  ;       MOV     DTA+16,-(SP)    ;STATUS WORD FROM DPB IN USE
 6488                                  ;       JSR     PC,LOP.CK
 6489                                  ;       RETURN
 6490
 6491  026324  032777  001000  152606  LOP.CK: BIT     #SW9,@SWR       ;LOOP ON ERROR
 6492  026332  001402                          BEQ     1$              ;BR IF NOT
 6493  026334  000177  152550                  JMP     @$LPERR         ;START AT THE LOOP ADDRESS
 6494  026340  005037  001206          1$:     CLR     $ESCAPE         ;CLEAR ERROR ESCAPE FLAG
 6495  026344  032766  072006  000002          BIT     #BIT14!BIT13!BIT12!BIT10!BIT02!BIT01,2(SP)  ;CHECK ERROR TYPE
 6496  026352  001402                          BEQ     2$              ;BR IF DRIVE NOT OFFLINE, UNLOADED, OR
 6497                                                                  ;PERSISTENT UNSAFE OR FATAL MASSBUS PARITY
 6498  026354  000137  017502                  JMP     $EOP            ;TERMINATE DRIVE
 6499  026360  012616                  2$:     MOV     (SP)+,(SP)      ;ADJUST RETURN ADDRESS
 6500  026362  000207                          RTS     PC
 6501
```

```
                                ;THIS ROUTINE FORMS AN INDEX THAT WILL BE USED TO DISPATCH
                                ;TO THE PROPER ERROR CALL.  THE INDEX IS FORMED BY EXAMINING
                                ;THE STATUS/ERROR INDICATOR OF THE APPLICABLE DPB.
                                ;INDEX           STATUS/ERROR
                                ;-----           ------------
                                ;     0 BIT14!BIT13!BIT08!BIT01
                                ;     2 BIT11!BIT10!BIT02
                                ;     4 BIT12!BIT04
                                ;     6 BIT05!BIT03!<BIT09 & COMMAND=NON-I/O>
                                ;    10 BIT06!<BIT09 & COMMAND=I/O>
                                ;CALL
                                ;       JSR     #DPB+16,-(SP)   ;ADDRESS OF STATUS/ERROR INDICATOR
                                ;       JSR     PC,@#ERINDX     ;FORM INDEX
                                ;       RETURN                  ;INDEX IS ON THE STACK

026364  010046                 ERINDX: MOV     R0,-(SP)        ;SAVE R0
026366  010146                         MOV     R1,-(SP)        ;SAVE R1
026370  016600  000006                 MOV     6(SP),R0        ;GET STATUS/ERROR INDICATOR POINTER
026374  011037  001260                 MOV     (R0),@#SVSTAT   ;SAVE THE STATUS/ERROR INDICATOR
026400  005001                         CLR     R1              ;START INDEX AT ZERO
026402  032710                         BIT     (PC)+,(R0)      ;FORM INDEX OF 0?
026404  020402                         .WORD   BIT13!BIT08!BIT01
026406  001027                         BNE     5$              ;YES--BRANCH
026410  032710                         BIT     (PC)+,(R0)      ;FORM PARITY ERROR OR PORT REQUEST INDEX (2)?
026412  006004                         .WORD   BIT11!BIT10!BIT02
026414  001023                         BNE     4$              ;YES--BRANCH
026416  032710                         BIT     (PC)+,(R0)      ;FORM UNSAFE INDEX (4)?
026420  050020                         .WORD   BIT14!BIT12!BIT04
026422  001017                         BNE     3$              ;YES--BRANCH
026424  032710                         BIT     (PC)+,(R0)      ;FORM NON-I/O ERROR INDEX (6)?
026426  000050                         .WORD   BIT05!BIT03
026430  001013                         BNE     2$              ;YES--BRANCH
026432  032710                         BIT     (PC)+,(R0)      ;FORM I/O ERROR INDEX (10)?
026434  000100                         .WORD   BIT06
026436  001007                         BNE     1$              ;YES--BRANCH
026440  032710                         BIT     (PC)+,(R0)      ;SOFTWARE TIMEOUT?
026442  001000                         .WORD   BIT09
026444  001410                         BEQ     5$              ;NO--FORM INDEX OF 0
026446  122760  000150  177762         CMPB    #150,-16(R0)    ;YES--I/O?
026454  003001                         BGT     2$              ;NO--BRANCH
026456  005201                 1$:     INC     R1              ;INDEX=10---ERROR=45 OR 46
026460  005201                 2$:     INC     R1              ;INDEX=6---ERROR=44
026462  005201                 3$:     INC     R1              ;INDEX=4---ERROR=43
026464  005201                 4$:     INC     R1              ;INDEX=2---ERROR=42
026466  006301                 5$:     ASL     R1              ;INDEX=0---ERROR=41
026470  010166  000006                 MOV     R1,6(SP)        ;RETURN INDEX TO USER
026474  012601                         MOV     (SP)+,R1        ;RESTORE R1
026476  012600                         MOV     (SP)+,R0        ;RESTORE R0
026500  000207                         RTS     PC              ;RETURN FROM CALL

                                ;THIS ROUTINE WILL PROVIDE A STALL IN MILLISECONDS FOR A SPECIFIC
                                ;AMOUNT OF TIME IF BIT13 OF C.SWR = 0 OR A RANDOM AMOUNT OF TIME
                                ;IF BIT 13 OF C.SWR = 1.
                                ;STALL1 CONTAINS SPECIFIED TIME FOR TESTS 0 - 7, AND STALL2
                                ;CONTAINS THE TIME FOR TESTS 16-21.
                                ;CALL
```

```
6558                                    ;       JSR     R0,@#STALL
6559                                    ;       TIME POINTER            ;WHERE TO FIND THE STALL TIME
6560
6561  026502  013046                    STALL:  MOV     @(R0)+,-(SP)    ;PICKUP STALL TIME
6562  026504  032737  020000  001220            BIT     #SW13,@#C.SWR   ;USE A RANDOM TIME?
6563  026512  001406                            BEQ     1$              ;NO--BRANCH
6564  026514  004737  023626                    JSR     PC,@#$RAND      ;YES--FORM RANDOM NUMBER
6565  026520  013716  023726                    MOV     @#$LONUM,(SP)   ;AND USE IT FOR THE STALL TIME
6566  026524  042716  177700                    BIC     #^C77,(SP)      ;BUT NEVER > 64 MILLISECONDS
6567  026530  005046                    1$:     CLR     -(SP)           ;CLEAR TEMP. LOCATION
6568  026532  162766  000001  000002    2$:     SUB     #1,2(SP)        ;MORE STALL REQUIRED?
6569  026540  103407                            BLO     4$              ;NO--BRANCH
6570  026542  012716  000144                    MOV     #100.,(SP)      ;STALL FOR ABOUT 1 MILLISECOND
6571  026546  005700                    3$:     TST     R0              ;NOP TO KILL TIME
6572  026550  005366  000000                    DEC     0(SP)           ;COUNT
6573  026554  001374                            BNE     3$              ;LOOP IF MORE COUNTS NEEDED
6574  026556  000765                            BR      2$
6575  026560  022626                    4$:     CMP     (SP)+,(SP)+     ;CLEAN OFF THE STACK
6576  026562  000200                            RTS     R0              ;EXIT
6577
6578
6579                                    ;ROUTINE TO PROVIDE A 2 MS STALL AFTER A SEEK OPERATION IN THE SEEK TIMING
6580                                    ;TESTS.  THIS STALL IS REQUIRED TO COMPENSATE FOR THE 'ACCESS READY' DELAY
6581                                    ;IN THE RP04.  THIS STALL TIME IS NOT INCLUDED IN THE CALCULATED SEEK TIMES.
6582                                    ;CALL
6583                                    ;       JSR     PC,@#TWOMS
6584                                    ;       RETURN
6585
6586  026564  013746  177776    TWOMS:  MOV     @#PS,-(SP)      ;SAVE THE PRESENT PROCESSOR STATUS
6587  026570  012737  000240  177776            MOV     #<5*32.>,@#PS   ;SET THE PROCESSOR PRIORITY TO 5
6588  026576  017746  152572                    MOV     @PKV,-(SP)      ;SAVE THE OLD CLOCK VECTOR ADDRESS
6589  026602  012777  026626  152564            MOV     #1$,@PKV        ;SETUP NEW VECTOR ADDRESS
6590  026610  012777  000310  152564            MOV     #200.,@PKB      ;LOAD THE CLOCK BUFFER
6591  026616  012777  000101  152554            MOV     #101,@PKCS      ;START THE CLOCK
6592  026624  000001                            WAIT                    ;WAIT FOR 2 MS
6593  026626  062706  000004            1$:     ADD     #4,SP           ;INCREMENT STACK FOR RETURN
6594  026632  012677  152536                    MOV     (SP)+,@PKV      ;RESTORE OLD CLOCK VECTOR
6595  026636  012637  177776                    MOV     (SP)+,@#PS      ;RESTORE THE OLD PROCESSOR STATUS
6596  026642  000207                            RTS     PC              ;RETURN
6597
6598                                    ;ROUTINE TO SOFTWARE COMPARE HEADER ON IMPLIED SEEKS
6599                                    ;CALL
6600                                    ;       JSR     R0,@#VERIFY
6601                                    ;       ADR POINTER             ;ADDRESS OF DPB+10 (SECTOR NUMBER)
6602                                    ;       RETURN
6603
6604  026644  010146                    VERIFY: MOV     R1,-(SP)        ;SAVE R1
6605  026646  012001                            MOV     (R0)+,R1        ;GET ADDRESS OF DPB+10
6606  026650  042737  010000  050202            BIC     #FMT22,@#BUFFER ;STRIP FORMAT BIT FROM CYLINDER NUMBER
6607  026656  023761  050202  000002            CMP     @#BUFFER,2(R1)  ;CYLINDER NUMBER OK?
6608  026664  001003                            BNE     1$              ;NO--BRANCH
6609  026666  023711  050204                    CMP     @#BUFFER+2,(R1) ;YES--HOW ABOUT TRACK/SECTOR?
6610  026672  001441                            BEQ     3$              ;BRANCH IF GOOD
6611  026674  013737  050202  001262    1$:     MOV     @#BUFFER,@#CYL.RD ;SAVE THE EXPECTED AND THE
6612  026702  113737  050205  001264            MOVB    @#BUFFER+3,@#TRK.RD ;RECIEVED CYLINDER, TRACK,
6613  026710  113737  050204  001266            MOVB    @#BUFFER+2,@#SEC.RD ;AND SECTOR
```

```
 6614  026716  112137  001272                MOVB    (R1)+,@#SEC.DS
 6615  026722  112137  001274                MOVB    (R1)+,@#TRK.DS
 6616  026726  011137  001270                MOV     (R1),@#CYL.DS
 6617  026732  012737  026744  001206        MOV     #2$,$ESCAPE     ;;ESCAPE TO 2$ ON ERROR
 6618  026740  005740                        TST     -(R0)           ;MAKE IT TEST PC+4
 6619  026742  104012                        ERROR   12              ;REPORT THE ERROR
 6620  026744  012737  000107  004106  2$:   MOV     #RECAL,DPB.A+2  ;LOAD RECALIBRATE ORDER CODE
 6621  026752  004037  025150                JSR     R0,@#CALL.A     ;GO EXECUTE THE COMMAND
 6622  026756  005037  001206                CLR     $ESCAPE         ;CLEAR ERROR ESCAPE FLAG
 6623  026762  032777  001000  152150        BIT     #SW9,@SWR       ;LOOP ON ERROR ?
 6624  026770  001404                        BEQ     4$              ;BR IF NOT
 6625  026772  000177  152112                JMP     @$LPERR         ;RETURN TO ERROR LOOP ADDRESS
 6626  026776  062700  000002          3$:   ADD     #2,R0           ;INCREMENT RETURN ADDRESS
 6627  027002  012601                  4$:   MOV     (SP)+,R1        ;RESTORE R1
 6628  027004  000200                        RTS     R0              ;EXIT
 6629
 6630                                  ;THIS ROUTINE WILL PERFORM A "MASSBUS" INIT. FOLLOWED BY
 6631                                  ;A "RECALIBRATE" ON THE DRIVE UNDER TEST.
 6632                                  ;NOTE: THIS ROUTINE DESTROYS R1 AND R4
 6633                                  ;CALL
 6634                                  ;       JSR     R0,SRCH00       ;DO A MASSBUS INIT. AND RECAL
 6635                                  ;       RETURN1                 ;RETURN HERE IF NO ERROR
 6636                                  ;       RETURN2                 ;RETURN HERE ON ERROR
 6637
 6638  027006  005001                  SRCH00: CLR     R1              ;INCASE OF ERROR (TYPTIM)
 6639  027010  005037  177776                  CLR     @#PS
 6640  027014  012777  037544  005612          MOV     #ISR,@RPVEC     ;SETUP INTERRUPT VECTOR
 6641  027022  013704  034632                  MOV     @#RPADR,R4      ;PICKUP ADDRESS OF RPCS1
 6642  027026  012764  000040  000010          MOV     #BIT05,RPCS2(R4) ;MASSBUS INIT.
 6643  027034  005037  004174                  CLR     @#DTADPB+10     ;TRACK=0; SECTOR=0
 6644  027040  005037  004176                  CLR     @#DTADPB+12     ;CYLINDER =0
 6645  027044  012737  000107  004166          MOV     #RECAL,@#DTADPB+2 ;COMMAND = RECALIBRATE
 6646  027052  005037  001206                  CLR     @#$ESCAPE       ;NO ESCAPE ADDRESS
 6647  027056  004037  035450                  JSR     R0,@#RP04       ;CALL THE DRIVER
 6648  027062  004164                          DTADPB                  ;DPB POINTER
 6649  027064  000440                          BR      4$              ;QUEUE IS FULL
 6650  027066  005737  004202          1$:     TST     DTADPB+16       ;WAIT ON DONE
 6651  027072  001775                          BEQ     1$
 6652  027074  100030                          BPL     3$              ;TAKE NORMAL EXIT IF NO ERROR
 6653  027076  012737  027152  001206          MOV     #2$,$ESCAPE     ;;ESCAPE TO 2$ ON ERROR
 6654  027104  013737  004176  001270          MOV     @#DTADPB+12,@#CYL.DS ;CYLINDER
 6655  027112  113737  004175  001274          MOVB    @#DTADPB+11,@#TRK.DS ;TRACK
 6656  027120  113737  004174  001272          MOVB    @#DTADPB+10,@#SEC.DS ;SECTOR
 6657  027126  012746  004202                  MOV     #DTADPB+16,-(SP) ;STATUS/ERROR INDICATOR ADDRESS
 6658  027132  004737  026364                  JSR     PC,@#ERINDX     ;FORM DISPATCH INDEX
 6659  027136  062607                          ADD     (SP)+,PC        ;REPORT PROPER ERROR
 6660  027140  104041                          ERROR   41              ;
 6661  027142  104042                          ERROR   42              ;PARITY ERROR
 6662  027144  104043                          ERROR   43              ;UNSAFE ERROR
 6663  027146  104044                          ERROR   44              ;NON-I/O ERROR
 6664  027150  104045                          ERROR   45              ;I/O ERROR
 6665  027152  005720                  2$:     TST     (R0)+           ;ADJUST FOR ERROR EXIT
 6666  027154  000404                          BR      4$              ;GO TO THE EXIT
 6667  027156  005064  000006          3$:     CLR     RPDA(R4)        ;TRACK AND SECTOR = 0
 6668  027162  005064  000034                  CLR     RPCA(R4)        ;CYLINDER = 0
 6669  027166  000200                  4$:     RTS     R0              ;RETURN
```

```
6670
6671                                    ;THIS IS AN RTI WHICH IS USED BY THE TIMING TESTS & THE SERVO SETTLE DOWN TEST
6672
6673  027170  000002                    DORTI:  RTI                     ;RETURN FROM INTERRUPT
6674
6675                                    ;THIS ROUTINE WILL INITIALIZE THE TIMERS USED BY THE "TIMING ROUTI ES
6676                                    ;CALL
6677                                    ;       JSR     PC,@#STRTMR
6678                                    ;       RETURN
6679
6680  027172  104412                    STRTMR: SAVREG                  ;SAVE R0-R5
6681  027174  012700  001276                    MOV     #TIM.UP,R0      ;START AT TIM.UP (MINIMUM)
6682  027200  012701  001332                    MOV     #TIM.PT,R1      ;STOP AT TIM.PT
6683  027204  005020                    1$:     CLR     (R0)+           ;CLEAR
6684  027206  020001                            CMP     R0,R1           ;DONE?
6685  027210  103775                            BLO     1$              ;NO--BRANCH
6686  027212  012710  050202                    MOV     #BUFFER,(R0)    ;SETUP POINTER
6687  027216  012737  077777  001276            MOV     #^CBIT15,@#TIM.UP ;SET MINIMUM TIME TO MAXIMUM
6688  027224  012737  077777  001314            MOV     #^CBIT15,@#TIM.DN ;POSITIVE NUMBER
6689  027232  104413                            RESREG                  ;RESTORE R0-R5
6690  027234  000207                            RTS     PC              ;RETURN
6691
6692                                    ;THIS ROUTINE WILL ADD THE ELAPSED TIME TO THE AVERAGE COUNTER AND
6693                                    ;MAINTAIN THE MINIMUM AND MAXIMUM TIMES.
6694                                    ;NOTE: THIS ROUTINE DESTROYS R2
6695                                    ;CALL
6696                                    ;       MOV     #TP,R3          ;PARAMETER POINTER
6697                                    ;       MOV     FLAG,R5         ;FLAG=0=COUNT UP
6698                                    ;                               ;FLAG=-1=COUNT DOWN
6699                                    ;       JSR     PC,@#COUNT
6700                                    ;       RETURN
6701
6702  027236  012702  001276            COUNT:  MOV     #TIM.UP,R2      ;PICKUP THE "UP" POINTER
6703  027242  005705                            TST     R5              ;USE IT?
6704  027244  001402                            BEQ     1$              ;YES--BRANCH
6705  027246  012702  001314                    MOV     #TIM.DN,R2      ;NO--PICKUP "DOWN" POINTER
6706  027252  027722  152126            1$:     CMP     @PKC,(R2)+      ;LESS THAN PREVIOUS LOW?
6707  027256  002003                            BGE     2$              ;NO--BRANCH
6708  027260  017762  152120  177776            MOV     @PKC,-2(R2)     ;YES--SAVE IT
6709  027266  027763  152112  000004    2$:     CMP     @PKC,4(R3)      ;LESS THAN THE LOW LIMIT?
6710  027274  002001                            BGE     3$              ;NO--BRANCH
6711  027276  005212                            INC     (R2)            ;YES--COUNT IT
6712  027300  005722                    3$:     TST     (R2)+           ;ADVANCE THE POINTER
6713  027302  027722  152076                    CMP     @PKC,(R2)+      ;GREATER THAN PREVIOUS HIGH?
6714  027306  003403                            BLE     4$              ;NO--BRANCH
6715  027310  017762  152070  177776            MOV     @PKC,-2(R2)     ;YES--SAVE IT
6716  027316  027763  152062  000006    4$:     CMP     @PKC,6(R3)      ;GREATER THAN THE HIGH LIMIT?
6717  027324  003401                            BLE     5$              ;NO--BRANCH
6718  027326  005212                            INC     (R2)            ;YES--COUNT IT
6719  027330  005722                    5$:     TST     (R2)+           ;ADVANCE THE POINTER
6720  027332  067722  152046                    ADD     @PKC,(R2)+      ;ADD THIS COUNT TO THE TOTAL
6721  027336  005522                            ADC     (R2)+
6722  027340  005212                            INC     (R2)            ;COUNT THIS READING
6723  027342  022737  056472  001332            CMP     #BUFFER+<4*814.>,@#TIM.PT ;SAVE THIS COUNT?
6724  027350  101406                            BLOS    6$              ;NO--BRANCH
6725  027352  017777  152026  151752            MOV     @PKC,@TIM.PT    ;YES--WELL SAVE IT THEN
```

```
 6726  027360  062737  000002  001332          ADD     #2,@#TIM.PT     ;ADVANCE THE POINTER
 6727  027366  000207                  6$:     RTS     PC              ;RETURN
 6728
 6729                                  ;THIS ROUTINE IS USED TO TYPE THE MINIMUM,
 6730                                  ;MAXIMUM, AND AVERAGE TIMES FOR THE TIMING TESTS
 6731                                  ;IT WILL ALSO CHECK THE TIMES TO ENSURE
 6732                                  ;THEY ARE WITHIN TOLERANCE AND IF NOT FLAG THE BAD TIMES.
 6733                                  ;NOTE: THIS ROUTINE DESTROYS R2-R5
 6734                                  ;CALL
 6735                                  ;       JSR     R0,@#TYPTIM     ;GO REPORT THE TIMES
 6736                                  ;       TABLE                   ;POINT TO THE PROPER TABLE
 6737                                  ;       RETURN
 6738                                  ;
 6739                                  ;TABLE:MSGADR1                  ;ADDRESS OF ASCIZ MESSAGE NUMBER 1
 6740                                  ;       MSGADR2                 ;ADDRESS OF ASCIZ MESSAGE NUMBER 2
 6741                                  ;       MIN.ALLOWED             ;MINIMUM TIME ALLOWED
 6742                                  ;       MAX.ALLOWED             ;MAXIMUM TIME ALLOWED
 6743
 6744  027370  012002                  TYPTIM: MOV     (R0)+,R2        ;PICKUP THE TABLE POINTER
 6745  027372  032777  000100  151540          BIT     #SW06,@SWR      ;INHIBIT TIME REPORTS?
 6746  027400  001145                          BNE     7$              ;YES--BRANCH
 6747  027402  012237  027422                  MOV     (R2)+,2$        ;ADDRESS OF MESSAGE NUMBER 1
 6748  027406  012205                          MOV     (R2)+,R5        ;ADDRESS OF MESSAGE NUMBER 2
 6749  027410  012203                          MOV     (R2)+,R3        ;PICKUP THE LOW LIMIT
 6750  027412  011202                          MOV     (R2),R2         ;AND THE HIGH LIMIT
 6751  027414  012704  001276                  MOV     #TIM.UP,R4      ;PARAMETER POINTER
 6752  027420  104401                  1$:     TYPE                    ;TYPE THE MESSAGE
 6753  027422  000000                  2$:     .WORD   0               ;ASCIZ MESSAGE POINTER GOES HERE
 6754  027424  005764  000014                  TST     14(R4)          ;DID ANY COUNTS OCCUR?
 6755  027430  001527                          BEQ     6$              ;NO--BRANCH
 6756  027432  104401  044170                  TYPE    ,MSGMIN         ;"MIN="
 6757  027436  012446                          MOV     (R4)+,-(SP)             ;PUT (R4)+ ON THE STACK
 6758  027440  004737  023336                  JSR     PC,@#$SB2D      ;CHANGE (R4)+ TO DECIMAL ASCIZ
 6759  027444  004737  023566                  JSR     PC,@#$SUPRS     ;TYPE WITHOUT LEADING ZEROS
 6760  027450  104401  044215                  TYPE    ,MSGOUS         ;"0 US"
 6761  027454  005724                          TST     (R4)+           ;ANY SEEKS BELOW THE LOW LIMIT
 6762  027456  001421                          BEQ     3$              ;NO--BRANCH
 6763  027460  104401  044330                  TYPE    ,MSG.SP         ;" "
 6764  027464  016446  177776                  MOV     -2(R4),-(SP)            ;PUT -2(R4) ON THE STACK
 6765  027470  004737  023336                  JSR     PC,@#$SB2D      ;CHANGE -2(R4) TO DECIMAL ASCIZ
 6766  027474  004737  023566                  JSR     PC,@#$SUPRS     ;TYPE WITHOUT LEADING ZEROS
 6767  027500  104401  044222                  TYPE    ,MBELOW         ;"BELOW THE MINIMUM OF"
 6768  027504  010346                          MOV     R3,-(SP)                ;PUT R3 ON THE STACK
 6769  027506  004737  023336                  JSR     PC,@#$SB2D      ;CHANGE R3 TO DECIMAL ASCIZ
 6770  027512  004737  023566                  JSR     PC,@#$SUPRS     ;TYPE WITHOUT LEADING ZEROS
 6771  027516  104401  044215                  TYPE    ,MSGOUS
 6772  027522  104401  044177          3$:     TYPE    ,MSGMAX         ;"MAX="
 6773  027526  012446                          MOV     (R4)+,-(SP)             ;PUT (R4)+ ON THE STACK
 6774  027530  004737  023336                  JSR     PC,@#$SB2D      ;CHANGE (R4)+ TO DECIMAL ASCIZ
 6775  027534  004737  023566                  JSR     PC,@#$SUPRS     ;TYPE WITHOUT LEADING ZEROS
 6776  027540  104401  044215                  TYPE    ,MSGOUS
 6777  027544  005724                          TST     (R4)+           ;ANY SEEKS ABOVE THE HIGH LIMIT
 6778  027546  001421                          BEQ     4$              ;NO--BRANCH
 6779  027550  104401  044330                  TYPE    ,MSG.SP         ;YES--REPORT HOW MANY
 6780  027554  016446  177776                  MOV     -2(R4),-(SP)            ;PUT -2(R4) ON THE STACK
 6781  027560  004737  023336                  JSR     PC,@#$SB2D      ;CHANGE -2(R4) TO DECIMAL ASCIZ
```

```
6782  027564  004737  023566          JSR     PC,@#$SUPRS     ;TYPE WITHOUT LEADING ZEROS
6783  027570  104401  044251          TYPE    ,MABOVE         ;"ABOVE THE MAXIMUM OF"
6784  027574  010246                  MOV     R2,-(SP)                ;PUT R2 ON THE STACK
6785  027576  004737  023336          JSR     PC,@#$SB2D      ;CHANGE R2 TO DECIMAL ASCIZ
6786  027602  004737  023566          JSR     PC,@#$SUPRS     ;TYPE WITHOUT LEADING ZEROS
6787  027606  104401  044215          TYPE    ,MSGOUS
6788  027612  104401  044206  4$:     TYPE    ,MSGAVG         ;"AVG="
6789  027616  012446                  MOV     (R4)+,-(SP)     ;FORM THE AVERAGE
6790  027620  012446                  MOV     (R4)+,-(SP)
6791  027622  012446                  MOV     (R4)+,-(SP)
6792  027624  004737  023730          JSR     PC,@#$DIV
6793  027630  006126                  ROL     (SP)+           ;IS THE REMAINDER OVER HALF?
6794  027632  100001                  BPL     5$              ;NO--BRANCH
6795  027634  005216                  INC     (SP)            ;YES--ROUND UP
6796  027636                  5$:
6797  027636  004737  023336          JSR     PC,@#$SB2D      ;CHANGE  TO DECIMAL ASCIZ
6798  027642  004737  023566          JSR     PC,@#$SUPRS     ;TYPE WITHOUT LEADING ZEROS
6799  027646  104401  044215          TYPE    ,MSGOUS
6800  027652  104401  044330          TYPE    ,MSG.SP
6801  027656  016446  177776          MOV     -2(R4),-(SP)            ;PUT -2(R4) ON THE STACK
6802  027662  004737  023336          JSR     PC,@#$SB2D      ;CHANGE -2(R4) TO DECIMAL ASCIZ
6803  027666  004737  023566          JSR     PC,@#$SUPRS     ;TYPE WITHOUT LEADING ZEROS
6804  027672  104401  044300          TYPE    ,MSGNUM         ;"SEEKS TIMED"
6805  027676  010537  027422          MOV     R5,2$           ;NEXT MESSAGE POINTER
6806  027702  001404                  BEQ     7$              ;IF NONE EXIT
6807  027704* 005005                  CLR     R5              ;NO MORE THAN 2
6808  027706  000644                  BR      1$
6809  027710  104401  044315  6$:     TYPE    ,MSGNON
6810  027714  000200          7$:     RTS     R0              ;EXIT
6811
6812                          ;THIS SUBROUTINE WILL INCREMENT THE TRACK
6813                          ;NUMBER (R2) BY THE AMOUNT SPECIFIED BY 'IT'.
6814                          ;CALL
6815                          ;       JSR     R0,@#INCTRK
6816                          ;       RETURN1                 ;TRACK NUMBER GREATER THAN LT15
6817                          ;       RETURN2                 ;TRACK NUMBER INCREMENTED
6818
6819  027716  020237  001520  INCTRK: CMP     R2,@#LT         ;LAST TRACK COMPLETED?
6820  027722  001410                  BEQ     2$              ;YES--EXIT
6821  027724  063702  001522          ADD     @#IT,R2         ;NO--UPDATE TRACK
6822  027730  020237  001520          CMP     R2,@#LT         ;TRACK TO BIG?
6823  027734  003402                  BLE     1$              ;NO--EXIT
6824  027736  013702  001520          MOV     @#LT,R2         ;YES--SET TRACK TO LAST TRACK
6825  027742  005720          1$:     TST     (R0)+           ;ADJUST FOR RETURN 2
6826  027744  000200          2$:     RTS     R0              ;RETURN
6827
6828
6829                          ;THIS SUBROUTINE WILL INCREMENT THE CYLINDER
6830                          ;NUMBER (R1) BY THE AMOUNT SPECIFIED BY 'IC'.
6831                          ;CALL
6832                          ;       JSR     R0,@#INCCYL
6833                          ;       RETURN1                 ;CYLINDER NUMBER GREATER THAN LC15
6834                          ;       RETURN2                 ;CYLINDER NUMBER INCREMENTED
6835
6836  027746  020137  001512  INCCYL: CMP     R1,@#LC         ;LAST CYLINDER COMPLETED?
6837  027752  001410                  BEQ     2$              ;YES--EXIT
```

```
 6838  027754  063701  001514                ADD     @#IC,R1          ;NO--UPDATE CYLINDER
 6839  027760  020137  001512                CMP     R1,@#LC          ;CYLINDER TO BIG?
 6840  027764  003402                        BLE     1$               ;NO--EXIT
 6841  027766  013701  001512                MOV     @#LC,R1          ;YES--SET CYLINDER TO LAST CYLINDER
 6842  027772  005720              1$:       TST     (R0)+            ;ADJUST FOR RETURN 2
 6843  027774  000200              2$:       RTS     R0               ;RETURN
 6844
 6845                              ;THIS ROUTINE DECREMENTS THE SECTOR ADDRESS.
 6846                              ;CALL
 6847                              ;         CLR     -(SP)            ;CLEAR THE STACK
 6848                              ;         JSR     PC,DECSEC        ;SUBROUTINE ENTRY
 6849                              ;         RETURN
 6850
 6851  027776  113766  004212  000002  DECSEC: MOVB    RP.REG+RPDA,2(SP)  ;PUT THE SECTOR ADDRESS ON THE STACK
 6852  030004  005366  000002                DEC     2(SP)            ;DECREMENT THE ADDRESS
 6853  030010  100003                        BPL     1$               ;BR IF NOT CORRECTION NEEDED
 6854  030012  013766  001634  000002        MOV     PRMLMT+22.,2(SP)  ;OVERFLOW OCCURED, FORCE TO MAXIMUM ADDRESS
 6855  030020  000207              1$:       RTS     PC               ;RETURN
 6856
 6857                              ;THIS SUBROUTINE IS USED TO FILL THE DATA BUFFER
 6858                              ;WITH ADDRESSES FROM 0 TO 21 WITH EACH ADDRESS
 6859                              ;BEING STORED IN 256 CONSECUTIVE LOCATIONS
 6860                              ;CALL
 6861                              ;         JSR     PC,@#FILBUF
 6862                              ;         RETURN
 6863
 6864  030022  104412              FILBUF:  SAVREG                   ;SAVE R0 - R5
 6865  030024  005000                       CLR     R0               ;FIRST DISK ADDRESS
 6866  030026  012701  050202               MOV     #BUFFER,R1       ;START FILLING HERE
 6867  030032  012702  000400       1$:     MOV     #256.,R2         ;DO 256 WORDS
 6868  030036  010021              2$:      MOV     R0,(R1)+         ;STORE
 6869  030040  005302                       DEC     R2               ;MORE?
 6870  030042  003375                       BGT     2$               ;YES--BRANCH
 6871  030044  005200                       INC     R0               ;NO--UPDATE DISK ADDRESS
 6872  030046  023700  001630               CMP     PRMLMT+22,R0     ;DONE?
 6873  030052  103367                       BHIS    1$               ;NO--BRANCH
 6874  030054  104413                       RESREG                   ;RESTORE R0 - R5
 6875  030056  000207                       RTS     PC               ;RETURN
 6876
 6877                              ;THIS ROUTINE WILL CLEAR THE BUFFER BY
 6878                              ;SETTING EACH WORD TO "177400".
 6879                              ;CALL
 6880                              ;         JSR     R0,@#CLRBUF
 6881                              ;         RETURN
 6882
 6883  030060  104412              CLRBUF:  SAVREG                   ;SAVE R0 - R5
 6884  030062  012701  177400               MOV     #177400,R1       ;WORD TO FILL BUFFER WITH
 6885  030066  012702  050202               MOV     #BUFFER,R2       ;FIRST ADDRESS OF BUFFER
 6886  030072  012703  076202               MOV     #BUFFER+<512.*22.>,R3 ;LAST ADDRESS+2 OF BUFFER
 6887  030076  010122              1$:      MOV     R1,(R2)+         ;FILL WORDS 1, 9,...249,....5625
 6888  030100  010122                       MOV     R1,(R2)+         ;FILL WORDS 2,10,...250,....5626
 6889  030102  010122                       MOV     R1,(R2)+         ;FILL WORDS 3,11,...251,....5627
 6890  030104  010122                       MOV     R1,(R2)+         ;FILL WORDS 4,12,...252,....5628
 6891  030106  010122                       MOV     R1,(R2)+         ;FILL WORDS 5,13,...253,....5629
 6892  030110  010122                       MOV     R1,(R2)+         ;FILL WORDS 6,14,...254,....5630
 6893  030112  010122                       MOV     R1,(R2)+         ;FILL WORDS 7,15,...255,....5631
```

```
6894  030114  010122                  MOV     R1,(R2)+        ;FILL WORDS 8,16,...256,...5632
6895  030116  020203                  CMP     R2,R3           ;DONE?
6896  030120  103766                  BLO     1$              ;NO--BRANCH
6897  030122  104413                  RESREG                  ;RESTORE R0 - R5
6898  030124  000200                  RTS     R0              ;RETURN FROM CALL
6899
6900                          ;THIS ROUTINE IS USED TO CHECK THE DATA BUFFER
6901                          ;FOR ADDRESSES 0 THROUGH 21 WITH EACH ADDRESS
6902                          ;BEING STORED IN 256 CONSECUTIVE LOCATIONS
6903                          ;CALL
6904                          ;       JSR     R0,@#CKSCTR
6905                          ;       RETURN
6906
6907  030126  104412          CKSCTR: SAVREG                  ;SAVE R0 - R5
6908  030130  162706  000004          SUB     #4,SP           ;RESERVE TEMP. STORAGE AREA
6909  030134  005001                  CLR     R1              ;FIRST SECTOR
6910  030136  012716  050202          MOV     #BUFFER,(SP)    ;FIRST ADDRESS OF DATA BUFFER
6911  030142  005066  000002          CLR     2(SP)           ;NO ERRORS
6912  030146  012702  000020  1$:     MOV     #16.,R2 ;LOOP COUNT (16*16=256)
6913  030152  011603                  MOV     (SP),R3         ;GET 1ST ADDRESS OF THIS SECTORS DATA
6914  030154                  2$:
6915  030154  020123                  CMP     R1,(R3)+        ;WORD 1
6916  030156  001063                  BNE     7$              ;BRANCH IF BAD
6917  030160  020123                  CMP     R1,(R3)+        ;WORD 2
6918  030162  001061                  BNE     7$              ;BRANCH IF BAD
6919  030164  020123                  CMP     R1,(R3)+        ;WORD 3
6920  030166  001057                  BNE     7$              ;BRANCH IF BAD
6921  030170  020123                  CMP     R1,(R3)+        ;WORD 4
6922  030172  001055                  BNE     7$              ;BRANCH IF BAD
6923  030174  020123                  CMP     R1,(R3)+        ;WORD 5
6924  030176  001053                  BNE     7$              ;BRANCH IF BAD
6925  030200  020123                  CMP     R1,(R3)+        ;WORD 6
6926  030202  001051                  BNE     7$              ;BRANCH IF BAD
6927  030204  020123                  CMP     R1,(R3)+        ;WORD 7
6928  030206  001047                  BNE     7$              ;BRANCH IF BAD
6929  030210  020123                  CMP     R1,(R3)+        ;WORD 8
6930  030212  001045                  BNE     7$              ;BRANCH IF BAD
6931  030214  020123                  CMP     R1,(R3)+        ;WORD 9
6932  030216  001043                  BNE     7$              ;BRANCH IF BAD
6933  030220  020123                  CMP     R1,(R3)+        ;WORD 10
6934  030222  001041                  BNE     7$              ;BRANCH IF BAD
6935  030224  020123                  CMP     R1,(R3)+        ;WORD 11
6936  030226  001037                  BNE     7$              ;BRANCH IF BAD
6937  030230  020123                  CMP     R1,(R3)+        ;WORD 12
6938  030232  001035                  BNE     7$              ;BRANCH IF BAD
6939  030234  020123                  CMP     R1,(R3)+        ;WORD 13
6940  030236  001033                  BNE     7$              ;BRANCH IF BAD
6941  030240  020123                  CMP     R1,(R3)+        ;WORD 14
6942  030242  001031                  BNE     7$              ;BRANCH IF BAD
6943  030244  020123                  CMP     R1,(R3)+        ;WORD 15
6944  030246  001027                  BNE     7$              ;BRANCH IF BAD
6945  030250  020123                  CMP     R1,(R3)+        ;WORD 16
6946  030252  001025                  BNE     7$              ;BRANCH IF BAD
6947  030254  005302                  DEC     R2              ;FINISHED WITH THIS SECTORS DATA?
6948  030256  001336                  BNE     2$              ;NO--BRANCH
6949  030260  062716  001000  3$:     ADD     #512.,(SP)      ;YES--FIRST ADDRESS OF NEXT SECTOR
```

```
6950  030264  005201                          INC     R1                      ;MOVE TO NEXT SECTOR
6951  030266  023701  001630                  CMP     PRMLMT+22,R1            ;DONE?
6952  030272  103325                          BHIS    1$                      ;NO--BRANCH
6953  030274  005766  000002          4$:     TST     2(SP)                   ;ERROR OCCUR?
6954  030300  001406                          BEQ     6$                      ;NO--BRANCH
6955  030302  123737  001364  001103          CMPB    @#ERR.CT,@#$ERFLG ;MAX. ERROR OCCURRED?
6956  030310  101002                          BHI     6$                      ;NO--BRANCH
6957  030312  013700  001252          5$:     MOV     @#BYPASS,R0             ;TAKE ERROR EXIT
6958  030316  062706  000004          6$:     ADD     #4,SP                   ;FREE TEMP. AREA
6959  030322  104413                          RESREG                          ;RESTORE R0 - R5
6960  030324  000200                          RTS     R0                      ;RETURN FROM CALL
6961  030326  010304          7$:     MOV     R3,R4                   ;FORM WORD NUMBER AND
6962  030330  161604                          SUB     (SP),R4                 ;ADDRESS TO CONTINUE FROM
6963  030332  010405                          MOV     R4,R5
6964  030334  006204                          ASR     R4                      ;WORD NUMBER
6965  030336  042705  177740                  BIC     #^C37,R5
6966  030342  001002                          BNE     8$                      ;BRANCH IF NOT A MULTIPLE OF 16
6967  030344  012705  000040                  MOV     #40,R5                  ;SET TO WORD 16
6968  030350  006305          8$:     ASL     R5
6969  030352  062705  030154                  ADD     #2$,R5                  ;ADDRESS
6970  030356  016337  177776  001126          MOV     -2(R3),@#$BDDAT ;SAVE BAD DATA
6971  030364  005766  000002                  TST     2(SP)                   ;FIRST ERROR?
6972  030370  001015                          BNE     10$                     ;NO--BRANCH
6973  030372  013737  004176  001270          MOV     @#DTADPB+12,@#CYL.DS ;CYLINDER NUMBER
6974  030400  113737  004175  001274          MOVB    @#DTADPB+11,@#TRK.DS ;TRACK NUMBER
6975  030406  012737  030416  001206          MOV     #9$,$ESCAPE             ;;ESCAPE TO 9$ ON ERROR
6976  030414  104021                          ERROR   21                      ;REPORT THE ERROR
6977  030416  105166  000002          9$:     COMB    2(SP)                   ;SET ERROR SWITCH
6978  030422  000404                          BR      11$
6979  030424                          10$:
6980  030424  012737  030434  001206          MOV     #11$,$ESCAPE    ;;ESCAPE TO 11$ ON ERROR
6981  030432  104022                          ERROR   22                      ;REPORT THE ERROR
6982  030434  032777  001000  150476  11$:    BIT     #SW09,@SWR              ;LOOP ON ERROR?
6983  030442  001323                          BNE     5$                      ;YES
6984  030444  032777  000002  150466          BIT     #SW01,@SWR              ;STOP DATA COMPARE?
6985  030452  001310                          BNE     4$                      ;YES--BRANCH
6986  030454  123737  001364  001103          CMPB    @#ERR.CT,@#$ERFLG ;MAX. ERRORS?
6987  030462  101713                          BLOS    5$                      ;YES--BRANCH
6988  030464  032777  000040  150446          BIT     #SW05,@SWR              ;REPORT ONLY 1ST ERROR PER SECTOR?
6989  030472  001272                          BNE     3$                      ;YES--BRANCH
6990  030474  000115                          JMP     (R5)
6991
6992                                  ;THIS ROUTINE WILL MOVE THE 16 WORDS OF THE
6993                                  ;DESIRED PATTERN INTO THE DATA BUFFER.
6994                                  ;CALL
6995                                  ;       MOV     #NX,R0                  ;PATTERN NUMBER INDEX TO R0
6996                                  ;       JSR     PC,@#SETBUF
6997
6998  030476  104412                  SETBUF: SAVREG                          ;SAVE R0 - R5
6999  030500  012701  050202                  MOV     #BUFFER,R1              ;FIRST ADDRESS
7000  030504  013702  004170                  MOV     @#DTADPB+4,R2   ;WORD COUNT
7001  030510  016003  003044          1$:     MOV     PAT.PT(R0),R3   ;PICKUP PATTERN POINTER
7002  030514  012321                          MOV     (R3)+,(R1)+             ;MOVE WORD 1 INTO DATA BUFFER
7003  030516  012321                          MOV     (R3)+,(R1)+             ;MOVE WORD 2 INTO DATA BUFFER
7004  030520  012321                          MOV     (R3)+,(R1)+             ;MOVE WORD 3 INTO DATA BUFFER
7005  030522  012321                          MOV     (R3)+,(R1)+             ;MOVE WORD 4 INTO DATA BUFFER
```

```
 7006  030524  012321                    MOV     (R3)+,(R1)+     ;MOVE WORD 5 INTO DATA BUFFER
 7007  030526  012321                    MOV     (R3)+,(R1)+     ;MOVE WORD 6 INTO DATA BUFFER
 7008  030530  012321                    MOV     (R3)+,(R1)+     ;MOVE WORD 7 INTO DATA BUFFER
 7009  030532  012321                    MOV     (R3)+,(R1)+     ;MOVE WORD 8 INTO DATA BUFFER
 7010  030534  012321                    MOV     (R3)+,(R1)+     ;MOVE WORD 9 INTO DATA BUFFER
 7011  030536  012321                    MOV     (R3)+,(R1)+     ;MOVE WORD 10 INTO DATA BUFFER
 7012  030540  012321                    MOV     (R3)+,(R1)+     ;MOVE WORD 11 INTO DATA BUFFER
 7013  030542  012321                    MOV     (R3)+,(R1)+     ;MOVE WORD 12 INTO DATA BUFFER
 7014  030544  012321                    MOV     (R3)+,(R1)+     ;MOVE WORD 13 INTO DATA BUFFER
 7015  030546  012321                    MOV     (R3)+,(R1)+     ;MOVE WORD 14 INTO DATA BUFFER
 7016  030550  012321                    MOV     (R3)+,(R1)+     ;MOVE WORD 15 INTO DATA BUFFER
 7017  030552  012321                    MOV     (R3)+,(R1)+     ;MOVE WORD 16 INTO DATA BUFFER
```

```
7018  030554  062702  000020          ADD     #16.,R2 ;DONE?
7019  030560  001353                  BNE     1$              ;NO--BRANCH
7020  030562  104413                  RESREG                  ;RESTORE R0 - R5
7021  030564  000207                  RTS     PC              ;RETURN
7022
7023                                  ;THIS ROUTINE COMPARES A 16 WORD DATA PATTERN
7024                                  ;AGAINST THE DATA BUFFER
7025                                  ;CALL
7026                                  ;       MOV     #NX,R0          ;PATTERN NUMBER INDEX TO R0
7027                                  ;       JSR     PC,@#DATCMP
7028                                  ;       RETURN
7029
7030  030566  104412                  DATCMP: SAVREG                  ;SAVE R0 - R5
7031  030570  012701  050202                  MOV     #BUFFER,R1      ;FIRST ADDRESS OF BUFFER
7032  030574  013702  004170                  MOV     @#DTADPB+4,R2   ;WORD COUNT
7033  030600  005046                  CLR     -(SP)           ;NO ERROR
7034  030602  016003  003044          1$:     MOV     PAT.PT(R0),R3   ;PATTERN POINTER
7035  030606                          2$:
7036  030606  162321                  SUB     (R3)+,(R1)+     ;CHECK WORD 1
7037  030610  001044                  BNE     4$              ;BRANCH IF DIFFERENT
7038  030612  162321                  SUB     (R3)+,(R1)+     ;CHECK WORD 2
7039  030614  001042                  BNE     4$              ;BRANCH IF DIFFERENT
7040  030616  162321                  SUB     (R3)+,(R1)+     ;CHECK WORD 3
7041  030620  001040                  BNE     4$              ;BRANCH IF DIFFERENT
7042  030622  162321                  SUB     (R3)+,(R1)+     ;CHECK WORD 4
7043  030624  001036                  BNE     4$              ;BRANCH IF DIFFERENT
7044  030626  162321                  SUB     (R3)+,(R1)+     ;CHECK WORD 5
7045  030630  001034                  BNE     4$              ;BRANCH IF DIFFERENT
7046  030632  162321                  SUB     (R3)+,(R1)+     ;CHECK WORD 6
7047  030634  001032                  BNE     4$              ;BRANCH IF DIFFERENT
7048  030636  162321                  SUB     (R3)+,(R1)+     ;CHECK WORD 7
7049  030640  001030                  BNE     4$              ;BRANCH IF DIFFERENT
7050  030642  162321                  SUB     (R3)+,(R1)+     ;CHECK WORD 8
7051  030644  001026                  BNE     4$              ;BRANCH IF DIFFERENT
7052  030646  162321                  SUB     (R3)+,(R1)+     ;CHECK WORD 9
7053  030650  001024                  BNE     4$              ;BRANCH IF DIFFERENT
7054  030652  162321                  SUB     (R3)+,(R1)+     ;CHECK WORD 10
7055  030654  001022                  BNE     4$              ;BRANCH IF DIFFERENT
7056  030656  162321                  SUB     (R3)+,(R1)+     ;CHECK WORD 11
7057  030660  001020                  BNE     4$              ;BRANCH IF DIFFERENT
7058  030662  162321                  SUB     (R3)+,(R1)+     ;CHECK WORD 12
7059  030664  001016                  BNE     4$              ;BRANCH IF DIFFERENT
7060  030666  162321                  SUB     (R3)+,(R1)+     ;CHECK WORD 13
7061  030670  001014                  BNE     4$              ;BRANCH IF DIFFERENT
7062  030672  162321                  SUB     (R3)+,(R1)+     ;CHECK WORD 14
7063  030674  001012                  BNE     4$              ;BRANCH IF DIFFERENT
7064  030676  162321                  SUB     (R3)+,(R1)+     ;CHECK WORD 15
7065  030700  001010                  BNE     4$              ;BRANCH IF DIFFERENT
7066  030702  162321                  SUB     (R3)+,(R1)+     ;CHECK WORD 16
7067  030704  001006                  BNE     4$              ;BRANCH IF DIFFERENT
7068  030706  062702  000020          ADD     #16.,R2         ;DONE ?
7069  030712  001333                  BNE     1$              ;NO--BRANCH
7070  030714  005726          3$:     TST     (SP)+           ;YES -- CLEAN UP STACK
7071  030716  104413                  RESREG                  ;RESTORE R0 - R5
7072  030720  000207                  RTS     PC
7073  030722  010104          4$:     MOV     R1,R4           ;FORM THE WORD NUMBER
```

```
7074  030724  162704  050202            SUB     #BUFFER,R4
7075  030730  006204                    ASR     R4              ;WORD NUMBER
7076  030732  010305                    MOV     R3,R5           ;FORM ADDRESS TO CONTINUE FROM
7077  030734  166005  003044            SUB     PAT.PT(R0),R5
7078  030740  006305                    ASL     R5
7079  030742  062705  030606            ADD     #2$,R5          ;ADDRESS
7080  030746  064341                    ADD     -(R3),-(R1)     ;RECONSTRUCT THE BAD WORD
7081  030750  010137  001122            MOV     R1,@#$BDADR     ;SAVE THE ERROR INFORMATION
7082  030754  010337  001120            MOV     R3,@#$GDADR     ;
7083  030760  012137  001126            MOV     (R1)+,@#$BDDAT  ;
7084  030764  012337  001124            MOV     (R3)+,@#$GDDAT  ;
7085  030770  005716                    TST     (SP)            ;1ST DATA COMPARE ERROR?
7086  030772  001023                    BNE     6$              ;NO--BRANCH
7087  030774  013737  004176  001270    MOV     @#DTADPB+12,@#CYL.DS ;CYLINDER
7088  031002  113737  004175  001274    MOVB    @#DTADPB+11,@#TRK.DS ;TRACK
7089  031010  113737  004174  001272    MOVB    @#DTADPB+10,@#SEC.DS ;SECTOR
7090  031016  016600  000026            MOV     26(SP),R0       ;GET TEST PC+4
7091  031022  012737  031032  001206    MOV     #5$,$ESCAPE     ;;ESCAPE TO 5$ ON ERROR
7092  031030  104013                    ERROR   13              ;REPORT THE ERROR
7093  031032  016600  000020          5$: MOV     20(SP),R0       ;PATTERN NUMBER INDEX
7094  031036  105116                    COMB    (SP)            ;SET THE ERROR SWITCH
7095  031040  000404                    BR      7$
7096  031042                          6$:
7097  031042  012737  031052  001206    MOV     #7$,$ESCAPE     ;;ESCAPE TO 7$ ON ERROR
7098  031050  104014                    ERROR   14              ;REPORT THE ERROR
7099  031052  032777  000002  150060  7$: BIT     #SW01,@SWR      ;STOP DATA COMPARE?
7100  031060  001315                    BNE     3$              ;YES--EXIT
7101  031062  123737  001364  001103    CMPB    @#ERR.CT,@#$ERFLG ;MAX. ERRORS?
7102  031070  101004                    BHI     8$              ;NO--BRANCH
7103  031072  013766  001252  000016    MOV     @#BYPASS,16(SP) ;YES--ERROR EXIT
7104  031100  000705                    BR      3$
7105  031102  000115                  8$: JMP     (R5)            ;NO--CONTINUE AT NEXT WORD
7106
7107                                    ;THIS ROUTINE WILL FILL THE DATA BUFFER (256*22 WORDS) WITH
7108                                    ;A RANDOM PATTERN.  THE FIRST TWO WORDS OF EVERY 256 WILL
7109                                    ;BE THE BASE OF THE RANDOM NUMBER GENERATOR FOR THE
7110                                    ;NEXT 254 WORDS.
7111                                    ;NOTE: THIS ROUTINE DESTROYS R1 AND R2
7112                                    ;CALL
7113                                    ;       JSR     R0,@#FILRAN
7114                                    ;       RETURN
7115
7116  031104  012701  050202          FILRAN: MOV     #BUFFER,R1
7117  031110  013702  001630                  MOV     PRMLMT+22,R2    ;MAXIMUM NUMBER OF SECTORS
7118  031114  004037  031324          1$:     JSR     R0,@#RANPAT
7119  031120  005302                          DEC     R2
7120  031122  100374                          BPL     1$
7121  031124  000200                          RTS     R0
7122
7123                                    ;THIS ROUTINE USES THE FIRST TWO WORDS OF THE
7124                                    ;READ BUFFER TO GENERATED A RANDOM PATTERN.  THEN
7125                                    ;THE READ BUFFER IS COMPARED TO THE PATTERN GENERATED.
7126                                    ;NOTE: THIS ROUTINE DESTROYS R1-R4
7127                                    ;CALL
7128                                    ;       JSR     R0,@#RANCK
7129                                    ;       RETURN
```

```
7130
7131  031126  013746  023724          RANCK:  MOV     @#$HINUM,-(SP)  ;SAVE THE PRESENT RANDOM NUMBER
7132  031132  013746  023726                  MOV     @#$LONUM,-(SP)
7133  031136  012702  051202                  MOV     #BUFFER+512.,R2 ;READ BUFFER ADDRESS
7134  031142  012701  052202                  MOV     #BUFFER+1024.,R1 ;RANDOM PATTERN ADDRESS
7135  031146  010103                          MOV     R1,R3           ;COPY IT INTO R3 FOR LATER USE
7136  031150  011237  023726                  MOV     (R2),@#$LONUM   ;PRIME THE RANDOM NUMBER GENERATOR
7137  031154  016237  000002  023724          MOV     2(R2),@#$HINUM
7138  031162  004037  031324                  JSR     R0,@#RANPAT     ;GENERATE A RANDOM PATTERN
7139  031166  012637  023726                  MOV     (SP)+,@#$LONUM  ;RESTORE PRESENT RANDOM NUMBER
7140  031172  012637  023724                  MOV     (SP)+,@#$HINUM
7141  031176  005046                          CLR     -(SP)           ;NO ERRORS
7142  031200  162322                  1$:     SUB     (R3)+,(R2)+     ;ARE THESE TWO WORDS DIFFERENT?
7143  031202  001441                          BEQ     4$              ;NO--BRANCH
7144  031204  012737  031256  001206          MOV     #3$,$ESCAPE     ;;ESCAPE TO 3$ ON ERROR
7145  031212  064342                          ADD     -(R3),-(R2)     ;RECREATE THE BAD WORD
7146  031214  010237  001122                  MOV     R2,@#$BDADR     ;ADDRESS OF BAD DATA
7147  031220  010337  001120                  MOV     R3,@#$GDADR     ;ADDRESS OF GOOD DATA
7148  031224  012237  001126                  MOV     (R2)+,@#$BDDAT  ;BAD DATA
7149  031230  012337  001124                  MOV     (R3)+,@#$GDDAT  ;GOOD DATA
7150  031234  010204                          MOV     R2,R4           ;FORM WORD NUMBER (1 TO 256)
7151  031236  162704  051202                  SUB     #BUFFER+512.,R4
7152  031242  006204                          ASR     R4
7153  031244  005716                          TST     (SP)            ;FIRST ERROR
7154  031246  001002                          BNE     2$              ;NO--BRANCH
7155  031250  105116                          COMB    (SP)            ;YES--SET ERROR SWITCH
7156  031252  104015                          ERROR   15              ;REPORT THE ERROR
7157  031254  104016                  2$:     ERROR   16              ;REPORT THE ERROR
7158  031256  032777  001000  147654  3$:     BIT     #SW09,@SWR      ;LOOP ON ERROR?
7159  031264  001012                          BNE     5$              ;YES--BRANCH
7160  031266  123737  001364  001103          CMPB    @#ERR.CT,@#$ERFLG ;MAX. ERRORS OCCURRED?
7161  031274  101406                          BLOS    5$              ;YES--BRANCH
7162  031276  032777  000002  147634          BIT     #SW01,@SWR      ;STOP COMPARING?
7163  031304  001002                          BNE     5$              ;YES--BRANCH
7164  031306  020103                  4$:     CMP     R1,R3           ;ALL DATA BEEN COMPARED?
7165  031310  101333                          BHI     1$              ;NO--BRANCH
7166  031312  005726                  5$:     TST     (SP)+           ;ERROR OCCUR?
7167  031314  001402                          BEQ     6$              ;NO--BRANCH
7168  031316  013700  001252                  MOV     @#BYPASS,R0     ;TAKE ERROR EXIT
7169  031322  000200                  6$:     RTS     R0              ;EXIT
7170
7171                                  ;THIS ROUTINE FILLS A 256 WORD BUFFER WITH A RANDOM
7172                                  ;PATTERN OF WHICH THE FIRST TWO WORDS ARE THE BASE
7173                                  ;OF THE PATTERN.
7174                                  ;CALL
7175                                  ;       MOV     #ADR,R1         ;ADDRESS OF THE BUFFER
7176                                  ;       JSR     R0,@#RANPAT
7177                                  ;       RETURN
7178
7179  031324  010246                  RANPAT: MOV     R2,-(SP)        ;SAVE R2
7180  031326  012702  000200                  MOV     #256./2.,R2     ;GENERATE 256 WORDS
7181  031332  000402                          BR      2$
7182  031334  004737  023626          1$:     JSR     PC,@#$RAND      ;GENERATE A RANDOM NUMBER
7183  031340  013721  023726          2$:     MOV     @#$LONUM,(R1)+  ;PUT LOW WORD IN BUFFER
7184  031344  013721  023724                  MOV     @#$HINUM,(R1)+  ;PUT HIGH WORD IN BUFFER
7185  031350  005302                          DEC     R2              ;DONE?
```

```
 7186  031352  003370                        BGT     1$              ;NO--BRANCH
 7187  031354  012602                        MOV     (SP)+,R2        ;RESTORE R2
 7188  031356  000200                        RTS     R0              ;EXIT
 7189
 7190                                ;THIS ROUTINE GENERATES RANDOM CYLINDER, TRACK, AND SECTOR
 7191                                ;ADDRESSES AND SAVES THEN IN THE DPB (DTADPB+10 AND DTADPB+12).
 7192                                ;NOTE: THIS ROUTINE DESTROYS R1-R3
 7193                                ;CALL
 7194                                ;       JSR     R0,@#RANADR
 7195                                ;       RETURN
 7196
 7197  031360  004737  023626        RANADR: JSR     PC,@#$RAND      ;GENERATE A RANDOM NUMBER
 7198  031364  113701  023726                MOVB    @#$LONUM,R1     ;FORM SECTOR IN R1
 7199  031370  113702  023727                MOVB    @#$LONUM+1,R2   ;FORM TRACK IN R2
 7200  031374  013703  023724                MOV     @#$HINUM,R3     ;FORM CYLINDER IN R3
 7201  031400  105701                        TSTB    R1              ;ENSURE THE SECTOR IS BETWEEN 0 AND 21
 7202  031402  002403                        BLT     2$
 7203  031404  123701  001630        1$:     CMPB    PRMLMT+22,R1    ;CHECK MAXIMUM SECTOR ADDRESS
 7204  031410  103003                        BHIS    3$
 7205  031412  000241                2$:     CLC
 7206  031414  106001                        RORB    R1
 7207  031416  000772                        BR      1$
 7208  031420  105702                3$:     TSTB    R2              ;ENSURE THE TRACK IS BETWEEN 0 AND 18
 7209  031422  002403                        BLT     5$
 7210  031424  122702  000023        4$:     CMPB    #19.,R2
 7211  031430  003003                        BGT     6$
 7212  031432  000241                5$:     CLC
 7213  031434  106002                        RORB    R2
 7214  031436  000772                        BR      4$
 7215  031440  023703  001510        6$:     CMP     @#FC,R3         ;ENSURE THE CYLINDER IS BETWEEN FC AND LC
 7216  031444  003413                        BLE     7$
 7217  031446  000241                        CLC
 7218  031450  006003                        ROR     R3
 7219  031452  005503                        ADC     R3
 7220  031454  001371                        BNE     6$
 7221  031456  010103                        MOV     R1,R3
 7222  031460  000303                        SWAB    R3
 7223  031462  060203                        ADD     R2,R3
 7224  031464  005203                        INC     R3
 7225  031466  003364                        BGT     6$
 7226  031470  005403                        NEG     R3
 7227  031472  000762                        BR      6$
 7228  031474  023703  001512        7$:     CMP     @#LC,R3
 7229  031500  002003                        BGE     8$
 7230  031502  000241                        CLC
 7231  031504  006003                        ROR     R3
 7232  031506  000772                        BR      7$
 7233  031510  023703  001510        8$:     CMP     @#FC,R3
 7234  031514  003403                        BLE     9$
 7235  031516  005203                        INC     R3
 7236  031520  000303                        SWAB    R3
 7237  031522  000764                        BR      7$
 7238  031524  110137  004174        9$:     MOVB    R1,@#DTADPB+10  ;SAVE SECTOR ADDRESS
 7239  031530  110237  004175                MOVB    R2,@#DTADPB+11  ;SAVE TRACK ADDRESS
 7240  031534  010337  004176                MOV     R3,@#DTADPB+12  ;SAVE CYLINDER ADDRESS
 7241  031540  000200                        RTS     R0              ;RETURN
```

```
7242
7243                                    ;THIS ROUTINE IS USED TO INPUT THE "CONTROL SWITCHES".
7244                                    ;IF SWR<07>=1 THE PRESENT SETTING WILL BE TYPED AND THE NEW
7245                                    ;SETTING IS READ AND STORED.
7246                                    ;NOTE: THIS ROUTINE DESTROYS R3 AND R4
7247                                    ;CALL
7248                                    ;       JSR     PC,@#GETSWR
7249                                    ;       RETURN                  ;(C.SWR)=DESIRED CONTROL SWITCHES
7250
7251  031542  032777  000200  147370   GETSWR: BIT     #SW07,@SWR      ;READ CONTROL SWITCHES?
7252  031550  001430                           BEQ     2$              ;NO--BRANCH
7253  031552  104401  031560                   TYPE    ,65$            ;;TYPE ASCIZ STRING
7254  031556  000410                           BR      64$             ;;GET OVER THE ASCIZ
7255                                  ;;65$:   .ASCIZ  <CR><LF>/SET SWR<07>=0/
7256  031600                          64$:
7257  031600  012703  043216           1$:     MOV     #MSG.CS,R3      ;"CONTROL SWITCHES="
7258  031604  013704  001220                   MOV     @#C.SWR,R4      ;PRESENT CONTROL SWITCH SETTINGS
7259  031610  004037  031634                   JSR     R0,@#GETNUM     ;GET THE NEW SWITCH SETTINGS
7260  031614  000771                           BR      1$              ; COMMA
7261  031616  000240                           NOP                     ;PERIOD
7262  031620  013737  001220  001222           MOV     C.SWR,SAVCSW    ;SAVE PREVIOUS VALUE
7263  031626  010437  001220                   MOV     R4,@#C.SWR      ; DOUBLE PERIOD-SAVE NEW SWITCH SETTING
7264  031632  000207                   2$:     RTS     PC              ;RETURN FROM CALL
7265
7266                                    ;THIS ROUTINE WILL TYPE AN ASCIZ MESSAGE AND THEN
7267                                    ;INPUT AN ASCIZ STRING AND CHANGE THE STRING TO OCTAL
7268                                    ;IF REQUIRED.
7269                                    ;NOTE: THIS ROUTINE DESTROYS R1
7270                                    ;CALL
7271                                    ;       MOV     #ADR,R3         ;ADDRESS OF ASCIZ MESSAGE
7272                                    ;       MOV     #NUM,R4         ;OCTAL NUMBER
7273                                    ;       JSR     R0,@#GETNUM
7274                                    ;       RETURN1                 ;INPUT TERMINATED WITH A COMMA
7275                                    ;       RETURN2                 ;WITH A PERIOD
7276                                    ;       RETURN3                 ;WITH A DOUBLE PERIOD
7277                                    ;                               ;R4=INPUT NUMBER AND
7278                                    ;                               ;R2=R4*32 FOR ALL
7279                                    ;                               ;THREE RETURNS
7280
7281  031634  010337  031642   GETNUM: MOV     R3,2$           ;SAVE MESSAGE POINTER
7282  031640  104401           1$:     TYPE                    ;TYPE THE MESSAGE
7283  031642  000000           2$:     .WORD   0               ;MESSAGE POINTER GOES HERE
7284  031644  010446                   MOV     R4,-(SP)        ;;SAVE R4 FOR TYPEOUT
7285  031646  104402                   TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
7286  031650  104401  043364           TYPE    .SLASH          ;' / '
7287  031654  104411                   RDLIN                   ;READ AN ASCIZ STRING
7288  031656  012601                   MOV     (SP)+,R1        ;ADDRESS OF FIRST CHARACTER
7289  031660  004037  034104           JSR     R0,@#CK.CHR     ;CHECK ONE CHARACTER
7290  031664  031640                   1$                      ;ILLEGAL CHARACTER
7291  031666  031640                   1$                      ;CARRIAGE RETURN
7292  031670  031702                   3$                      ;"/"
7293  031672  031726                   7$                      ;"."
7294  031674  031734                   8$                      ;"-"
7295  031676  031700                   11$                     ;DIGIT 0-9
7296  031700  005301           11$:    DEC     R1              ;DECREMENT THE INPUT POINTER
7297  031702                   3$:
```

```
7298  031702  004037  034344          JSR     R0,@#CK.NUM     ;CHECK THE NUMBER
7299  031706  031640                  1$                      ;ILLEGAL INPUT
7300  031710  031722                  6$                      ;TERMINATED WITH A ",'' OR "CR"
7301  031712  031720                  5$                      ;TERMINATED WITH A "."
7302  031714  031716                  4$                      ;TERMINATED WITH A ".."
7303  031716  005720          4$:     TST     (R0)+           ;DOUBLE PERIOD
7304  031720  005720          5$:     TST     (R0)+           ;SINGLE PERIOD
7305  031722  010204          6$:     MOV     R2,R4           ;COMMA--SAVE INPUT NUMBER
7306  031724  000414                  BR      10$             ;GO TO EXIT
7307  031726  105711          7$:     TSTB    (R1)            ;TERMINATOR AFTER A COMMA?
7308  031730  001343                  BNE     1$              ;NO--LOOP
7309  031732  000411                  BR      10$             ;YES--EXIT
7310  031734  105711          8$:     TSTB    (R1)            ;TERMINATOR AFTER A PERIOD?
7311  031736  001406                  BEQ     9$              ;YES--EXIT
7312  031740  122721  000056          CMPB    #'.,(R1)+       ;NO--DOUBLE PERIOD?
7313  031744  001335                  BNE     1$              ;NO--LOOP
7314  031746  105711                  TSTB    (R1)            ;YES--TERMINATOR?
7315  031750  001333                  BNE     1$              ;NO--LOOP
7316  031752  005720                  TST     (R0)+           ;DOUBLE PERIOD
7317  031754  005720          9$:     TST     (R0)+           ;PERIOD
7318  031756  010402          10$:    MOV     R4,R2           ;COMMA--POSITION THE
7319  031760  000302                  SWAB    R2              ;NUMBER IN CASE IT
7320  031762  006202                  ASR     R2              ;IS THE PRIORITY LEVEL
7321  031764  006202                  ASR     R2
7322  031766  006202                  ASR     R2
7323  031770  000200                  RTS     R0              ;EXIT
7324
7325                          ;THIS ROUTINE IS USED TO CHANGE OR MODIFY
7326                          ;THE TEST PARAMETERS.  IT GIVES THE OPERATOR
7327                          ;THE CAPABILITY OF SPECIFYING WHICH DRIVES TO TEST, WHICH
7328                          ;TESTS TO RUN AND HOW MANY TIMES TO
7329                          ;REPEAT EACH TEST
7330
7331  031772  104412          GT.PRM: SAVREG                  ;SAVE R0 - R5
7332  031774  005037  001252  GT.PR1: CLR     DRVSEL          ;NO DRIVE SELECTED
7333  032000  104401  032006          TYPE    ,65$            ;;TYPE ASCIZ STRING
7334  032004  000406                  BR      64$             ;;GET OVER THE ASCIZ
7335                          ;;65$:  .ASCIZ  <CR><LF>/DRIVE(S)=/
7336  032022                  64$:
7337  032022  104411                  RDLIN                   ;READ TTY
7338  032024  012601                  MOV     (SP)+,R1        ;ADDRESS OF ASCIZ STRING
7339  032026  004037  034104          JSR     R0,@#CK.CHR     ;CHECK ONE CHARACTER
7340  032032  031774                  GT.PR1                  ;ILLEGAL CHARACTER
7341  032034  031774                  GT.PR1                  ;CARRIAGE RETURN
7342  032036  031774                  GT.PR1                  ;"/"
7343  032040  031774                  GT.PR1                  ;","
7344  032042  031774                  GT.PR1                  ;"."
7345  032044  032046                  1$                      ;DIGIT 0-9
7346  032046  005301          1$:     DEC     R1
7347  032050                  2$:
7348  032050  012702  000007          MOV     #7,R2   ;UPPER LIMIT OF INPUT
7349  032054  004037  034160          JSR     R0,@#CK.DIG     ;CHECK THE DIGIT(S)
7350  032060  031774                  GT.PR1                  ;ILLEGAL INPUT
7351  032062  031774                  GT.PR1                  ;INPUT TO LARGE
7352  032064  032072                  3$                      ;TERMINATED WITH A "," OR "CR"
7353  032066  032116                  4$                      ;TERMINATED WITH A "."
```

```
7354  032070  032116                        4$                      ;TERMINATED WITH A ".."
7355  032072  156237  034620  001232  3$:   BISB    ATABIT(R2),DRVSEL ;SET THE DRIVE SELECTED BIT
7356  032100  105741                        TSTB    -(R1)           ;WAS THE LINE TERMINATED?
7357  032102  001362                        BNE     2$              ;NO-GET THE NEXT DRIVE
7358  032104  005037  001234                CLR     @#TSTNMS        ;DESELECT ALL TESTS
7359  032110  005037  001236                CLR     TSTNMS+2
7360  032114  000405                        BR      GTTST1          ;YES--SELECT TEST
7361  032116  156237  034620  001232  4$:   BISB    ATABIT(R2),DRVSEL ;SET THE SELECTED DRIVE BITS
7362  032124  104413                GT.PR2: RESREG                  ;RESTORE R0 - R5
7363  032126  000207                        RTS     PC              ;EXIT
7364
7365  032130                        GTTST1:
7366  032130  104401  032136                TYPE    ,65$            ;;TYPE ASCIZ STRING
7367  032134  000403                        BR      64$             ;;GET OVER THE ASCIZ
7368                                ;;65$:  .ASCIZ  /TEST=/
7369  032144                        64$:
7370  032144  104411                        RDLIN                   ;READ AN ASCIZ STRING
7371  032146  012601                        MOV     (SP)+,R1        ;POINTER TO R1
7372  032150  122711  000056                CMPB    #'.,(R1)        ;DOUBLE PERIOD?
7373  032154  001007                        BNE     1$              ;NO--BRANCH
7374  032156  122761  000056  000001        CMPB    #'.,1(R1)
7375  032164  001003                        BNE     1$
7376  032166  105761  000002                TSTB    2(R1)           ;"CR"?
7377  032172  001754                        BEQ     GT.PR2          ;YES--EXIT
7378  032174  005037  001234          1$:   CLR     TSTNMS          ;NO TEST SELECTED
7379  032200  005037  001236                CLR     TSTNMS+2
7380  032204  005037  001240                CLR     OPNFLG          ;NO TESTS TO BE OPENED
7381  032210  005037  001242                CLR     OPNFLG+2
7382  032214  121127  000123        GTTST2: CMPB    (R1),#'S        ;ALL SEEK TESTS?
7383  032220  001004                        BNE     1$              ;NO--BRANCH
7384  032222  052737  000777  001234        BIS     #777,TSTNMS     ;YES--SELECT TESTS 0-10
7385  032230  000552                        BR      GTTST3
7386  032232  121127  000124          1$:   CMPB    (R1),#'T        ;ALL TIMING TESTS?
7387  032236  001004                        BNE     2$              ;NO--BRANCH
7388  032240  052737  036000  001234        BIS     #36000,TSTNMS   ;YES--SELECT TESTS 12-15
7389  032246  000543                        BR      GTTST3
7390  032250  121127  000101          2$:   CMPB    (R1),#'A        ;ALL ADDRESSING TESTS?
7391  032254  001004                        BNE     3$              ;NO--BRANCH
7392  032256  052737  140000  001234        BIS     #140000,TSTNMS  ;YES--SELECT TESTS 16 & 17
7393  032264  000534                        BR      GTTST3
7394  032266  121127  000104          3$:   CMPB    (R1),#'D        ;DATA TEST?
7395  032272  001004                        BNE     4$              ;NO--BRANCH
7396  032274  052737  000001  001236        BIS     #1,TSTNMS+2     ;YES--SELECT TEST 20
7397  032302  000525                        BR      GTTST3
7398  032304  121127  000105          4$:   CMPB    (R1),#'E        ;EXERCISER TEST?
7399  032310  001004                        BNE     5$              ;NO--BRANCH
7400  032312  052737  000002  001236        BIS     #2,TSTNMS+2     ;YES--SELECT TEST 21
7401  032320  000516                        BR      GTTST3
7402  032322  004037  034030          5$:   JSR     R0,@#CK.OCT     ;OCTAL DIGIT?
7403  032326  000514                        BR      GTTST4          ;NO--BRANCH
7404  032330  010205                        MOV     R2,R5           ;YES--SAVE IT
7405  032332  005201                        INC     R1              ;MOVE TO NEXT CHARACTER
7406  032334  004037  034030                JSR     R0,@#CK.OCT     ;OCTAL DIGIT
7407  032340  000405                        BR      6$              ;NO--BRANCH
7408  032342  005201                        INC     R1              ;MOVE TO NEXT CHARACTER
7409  032344  006305                        ASL     R5              ;SCALE HIGH DIGIT
```

```
7410  032346  006305                        ASL     R5
7411  032350  006305                        ASL     R5
7412  032352  060502                        ADD     R5,R2           ;COMBINE HIGH & LOW DIGITS
7413  032354  020227  000022        6$:     CMP     R2,#$TN-1       ;VALID TEST NUMBER?
7414  032360  003263                        BGT     GTTST1          ;NO--BRANCH
7415  032362  010237  032554                MOV     R2,9$           ;SAVE THE TEST NUMBER
7416  032366  010204                        MOV     R2,R4           ;CONVERT TEST NUMBER INTO AN INDEX
7417  032370  042704  000017                BIC     #17,R4          ;CLEAR UNWANTED BITS
7418  032374  006204                        ASR     R4              ;SHIFT THE BITS
7419  032376  006204                        ASR     R4              ;SHIFT THE BITS
7420  032400  006204                        ASR     R4              ;SHIFT THE BITS
7421  032402  006302                        ASL     R2
7422  032404  056264  001424  001234        BIS     BITS(R2),TSTNMS(R4)  ;SELECT TEST
7423  032412  121127  000055                CMPB    (R1),#'-        ;TEST STRING?
7424  032416  001060                        BNE     GTTST4          ;NO--BRANCH
7425  032420  005201                        INC     R1              ;YES--MOVE TO NEXT CHARACTER
7426  032422  004037  034030                JSR     R0,@#CK.OCT     ;OCTAL DIGIT?
7427  032426  000640                        BR      GTTST1          ;NO--BRANCH
7428  032430  010205                        MOV     R2,R5           ;YES--SAVE IT
7429  032432  005201                        INC     R1              ;MOVE TO NEXT CHARACTER
7430  032434  004037  034030                JSR     R0,@#CK.OCT     ;OCTAL DIGIT?
7431  032440  000405                        BR      7$              ;NO--BRANCH
7432  032442  005201                        INC     R1              ;YES--MOVE TO NEXT CHARACTER
7433  032444  006305                        ASL     R5              ;SCALE HIGH DIGIT
7434  032446  006305                        ASL     R5
7435  032450  006305                        ASL     R5
7436  032452  060502                        ADD     R5,R2           ;COMBINE HIGH & LOW DIGIT
7437  032454  020227  000022        7$:     CMP     R2,#$TN-1       ;VALID TEST NUMBER?
7438  032460  003223                        BGT     GTTST1          ;NO--BRANCH
7439  032462  023702  032554                CMP     9$,R2           ;IS THE FIRST NUMBER OF THE
7440                                                                ;STRING SMALLER THAN THE LAST?
7441  032466  002220                        BGE     GTTST1          ;NO--BRANCH
7442  032470  010246                        MOV     R2,-(SP)        ;SAVE ENDING TEST NUMBER
7443  032472  013702  032554                MOV     9$,R2           ;GET STARTING TEST NUMBER
7444  032476  012637  032554                MOV     (SP)+,9$        ;STORE ENDING TEST NUMBER
7445  032502  006337  032554                ASL     9$              ;SHIFT ENDING TEST NUMBER
7446  032506  006302                        ASL     R2              ;SHIFT TEST NUMBER
7447  032510  010204        8$:     MOV     R2,R4           ;COPY TEST NUMBER INTO R4
7448  032512  042704  000037                BIC     #37,R4          ;CLEAR LOWER BITS
7449  032516  006204                        ASR     R4              ;SHIFT THE TEST NUMBER
7450  032520  006204                        ASR     R4              ;SHIFT THE TEST NUMBER
7451  032522  006204                        ASR     R4              ;SHIFT THE TEST NUMBER
7452  032524  006204                        ASR     R4              ;SHIFT THE TEST NUMBER
7453  032526  056264  001424  001234        BIS     BITS(R2),TSTNMS(R4)  ;SELECT THE TEST
7454  032534  062702  000002                ADD     #2,R2           ;INECREMENT THE TEST NUMBER
7455  032540  020237  032554                CMP     R2,9$           ;SEE IF FINISHED
7456  032544  101761                        BLOS    8$              ;BR IF NOT
7457  032546  162702  000002                SUB     #2,R2           ;CORRECT TEST NUMBER
7458  032552  000402                        BR      GTTST4          ;CONTINUE
7459  032554  000000        9$:     .WORD   0               ;STORE TEST NUMBER HERE
7460  032556  005201        GTTST3: INC     R1              ;MOVE TO NEXT CHARACTER
7461  032560  121127  000056        GTTST4: CMPB    (R1),#'.        ;"PERIOD"?
7462  032564  001441                        BEQ     GTTST5          ;YES--BRANCH
7463  032566  005737  001234                TST     TSTNMS          ;ANY TEST SELECTED THIS CYCLE?
7464  032572  001005                        BNE     1$              ;BR IF YES
7465  032574  005737  001236                TST     TSTNMS+2        ;ANY TEST SELECTED THIS CYCLE ?
```

```
7466  032600  001002                        BNE     1$              ;BR IF YES
7467  032602  000137  032130                JMP     GTTST1          ;NO
7468  032606  121127  000057        1$:     CMPB    (R1),#'/        ;"OPEN"?
7469  032612  001004                        BNE     2$              ;NO--BRANCH
7470  032614  056264  001424  001240        BIS     BITS(R2), OPNFLG(R4)  ;YES--SET BITS FOR TEST TO OPEN
7471  032622  000405                        BR      3$
7472  032624  121127  000054        2$:     CMPB    (R1),#',        ;"COMMA"?
7473  032630  001402                        BEQ     3$              ;BR IF YES
7474  032632  000137  032130                JMP     GTTST1          ;NO
7475  032636  005201                3$:     INC     R1              ;MOVE TO NEXT CHARACTER
7476  032640  105711                        TSTB    (R1)            ;"CR"?
7477  032642  001402                        BEQ     4$              ;BR IF 'CR'
7478  032644  000137  032214                JMP     GTTST2          ;NO--GO GET NEXT CHARACTER
7479  032650  005737  001240        4$:     TST     OPNFLG          ;ANY TESTS TO OPEN ?
7480  032654  001042                        BNE     OPNTST          ;BR IF YES
7481  032656  005737  001242                TST     OPNFLG+2        ;ANY TESTS TO OPEN ?
7482  032662  001037                        BNE     OPNTST          ;BR IF YES
7483  032664  000137  032130                JMP     GTTST1          ;NO--START AGAIN
7484  032670  005201        GTTST5: INC     R1              ;MOVE TO NEXT CHARACTER
7485  032672  121127  000056                CMPB    (R1),#'.        ;"PERIOD"?
7486  032676  001414                        BEQ     GTTST6          ;YES--BRANCH
7487  032700  105711                        TSTB    (R1)            ;"CR"?
7488  032702  001402                        BEQ     1$              ;YES--BRANCH
7489  032704  000137  032130                JMP     GTTST1
7490  032710  005737  001240        1$:     TST     OPNFLG          ;ANY TESTS TO OPEN ?
7491  032714  001022                        BNE     OPNTST          ;BR IF YES
7492  032716  005737  001242                TST     OPNFLG+2        ;ANY TESTS TO OPEN ?
7493  032722  001017                        BNE     OPNTST          ;BR IF YES
7494  032724  000137  032124                JMP     GT.PR2          ;NO--GO START TESTING
7495  032730  005201        GTTST6: INC     R1              ;MOVE TO NEXT CHARACTER
7496  032732  105711                        TSTB    (R1)            ;"CR"?
7497  032734  001402                        BEQ     1$              ;YES--BRANCH
7498  032736  000137  032130                JMP     GTTST1          ;NO--GO ASK FOR TEST
7499  032742  005737  001240        1$:     TST     OPNFLG          ;ANY TESTS TO OPEN ?
7500  032746  001005                        BNE     OPNTST          ;BR IF YES
7501  032750  005737  001242                TST     OPNFLG+2        ;ANY TESTS TO OPEN ?
7502  032754  001002                        BNE     OPNTST          ;BR IF YES
7503  032756  000137  032124                JMP     GT.PR2          ;NO--GO START TESTING
7504
7505                                ;OPEN THE SELECTED TEST FOR CHANGES
7506
7507  032762  104412        OPNTST: SAVREG                          ;SAVE R0 - R5
7508  032764  005027                        CLR     (PC)+           ;START WITH TEST 0
7509  032766  000000        OPN.CT: .WORD   0               ;COUNT STORED HERE
7510  032770  000411                        BR      OPN.2           ;SKIP THE INCREMENT
7511  032772  005237  032766        OPN.1:  INC     OPN.CT          ;MOVE TO THE NEXT TEST
7512  032776  022737  000022  032766        CMP     #$TN-1,OPN.CT   ;TEST NUMBER TOO BIG?
7513  033004  002003                        BGE     OPN.2           ;NO--OPEN THE NEXT TEST
7514  033006  104413                        RESREG                  ;RESTORE R0 - R5
7515  033010  000137  032130                JMP     GTTST1          ;YES--GO ASK FOR MORE TESTS
7516  033014  013705  032766        OPN.2:  MOV     OPN.CT,R5       ;SETUP TO USE THE
7517  033020  006305                        ASL     R5              ;TEST NUMBER AS AN INDEX
7518  033022  013703  032766                MOV     OPN.CT,R3       ;GET INDEX
7519  033026  042703  000017                BIC     #17,R3          ;CLEAR LOWER TEST BITS
7520  033032  006203                        ASR     R3              ;SHIFT TEST NUMBER
7521  033034  006203                        ASR     R3              ;SHIFT TEST NUMBER
```

```
7522  033036  006203                        ASR     R3                      ;SHIFT TEST NUMBER
7523  033040  036563  001424  001240        BIT     BITS(R5),OPNFLG(R3)  ;OPEN THIS TEST?
7524  033046  001751                        BEQ     OPN.1                   ;NO--MOVE TO NEXT TEST
7525  033050  104401  033056                TYPE    ,65$                    ;;TYPE ASCIZ STRING
7526  033054  000404                        BR      64$                     ;;GET OVER THE ASCIZ
7527                                ;;65$:  .ASCIZ  / TEST /
7528  033066                            64$:
7529  033066  013746  032766                MOV     OPN.CT,-(SP)            ;;SAVE OPN.CT FOR TYPEOUT
7530                                                                        ;;TEST NUMBER
7531  033072  104403                        TYPOS                           ;;GO TYPE--OCTAL ASCII
7532  033074     002                        .BYTE   2                       ;;TYPE 2 DIGIT(S)
7533  033075     000                        .BYTE   0                       ;;SUPPRESS LEADING ZEROS
7534  033076  104401  001215                TYPE    ,$CRLF                  ;TYPE "CR" & "LF"
7535  033102  016500  001536                MOV     PRMPT(R5),R0            ;PICKUP PARAMETER POINTER
7536  033106  011046                        MOV     (R0),-(SP)              ;SAVE THE VARIABLE INDICATOR
7537  033110  012702  001504                MOV     #PRM,R2                 ;FIRST ADDRESS OF TABLE
7538  033114  000405                        BR      2$
7539  033116  006216                  1$:   ASR     (SP)                    ;CHECK FOR A VARIABLE
7540  033120  103403                        BCS     2$                      ;GO MOVE THIS ONE
7541  033122  001404                        BEQ     OPNPRM                  ;DONE
7542  033124  005722                        TST     (R2)+                   ;BUMP THE POINTER
7543  033126  000773                        BR      1$
7544  033130  012022                  2$:   MOV     (R0)+,(R2)+             ;MOVE THIS VARIABLE INTO THE
7545  033132  000771                        BR      1$                      ;COMMON AREA
7546  033134  013716  001504          OPNPRM: MOV     @#PRM,(SP)              ;GET THE VARIABLE INDICATOR
7547  033140  005004                        CLR     R4                      ;ZERO THE INDEX
7548  033142  006216                  1$:   ASR     (SP)                    ;CHECK FOR A VARIABLE
7549  033144  103403                        BCS     3$                      ;GO GET IT
7550  033146  001772                        BEQ     OPNPRM                  ;OUT OF VARIABLES
7551  033150  005724                  2$:   TST     (R4)+                   ;UPDATE THE INDEX
7552  033152  000773                        BR      1$
7553  033154  005764  001606          3$:   TST     PRMLMT(R4)              ;IS THE MAX. MAGNITUDE NEG?
7554  033160  100466                        BMI     OPNPAT                  ;YES--THEN IT IS THE PATTERN
7555  033162  104401  044330                TYPE    ,MSG.SP                 ;TYPE SPACES
7556  033166  016437  001636  033176        MOV     PRMMSG(R4),4$           ;TYPE THE NAME OF THIS VARIABLE
7557  033174  104401                        TYPE
7558  033176  000000                  4$:   .WORD   0
7559  033200  104401  043214                TYPE    ,MSG.EQ                 ;TYPE "="
7560  033204  016446  001506                MOV     RPT(R4),-(SP)           ;PUT RPT(R4) ON THE STACK
7561  033210  004737  023336                JSR     PC,@#$SB2D              ;CHANGE RPT(R4) TO DECIMAL ASCIZ
7562  033214  004737  023566                JSR     PC,@#$SUPRS             ;TYPE WITHOUT LEADING ZEROS
7563  033220  104401  043364                TYPE    ,SLASH                  ;' / '
7564  033224  104411                        RDLIN
7565  033226  012601                        MOV     (SP)+,R1                ;READ AN ASCIZ STRING
7566  033230  004037  034104                JSR     R0,@#CK.CHR             ;CHECK ONE CHARACTER
7567  033234  033154                        3$                              ;ILLEGAL CHARACTER
7568  033236  033150                        2$                              ;CARRIAGE RETURN
7569  033240  033306                        8$                              ;"/"
7570  033242  033250                        5$                              ;"."
7571  033244  033256                        6$                              ;","
7572  033246  033304                        7$                              ;DIGIT 0-9
7573  033250  105711                  5$:   TSTB    (R1)                    ;"CR"?
7574  033252  001340                        BNE     3$                      ;NO--STAY ON THIS VARIABLE
7575  033254  000735                        BR      2$                      ;YES--MOVE TO NEXT VARIABLE
7576  033256  105711                  6$:   TSTB    (R1)                    ;IS THERE A "CR" AFTER THE PERIOD?
7577  033260  001002                        BNE     64$                     ;NO
```

```
7578  033262  000137  033676            JMP     OPN.N2          ;YES--GO CLOSE THIS TEST
7579  033266  122721  000056  64$:      CMPB    #'.,(R1)+       ;DOUBLE PERIOD?
7580  033272  001330                    BNE     3$              ;NO--GO ASK FOR THIS VARIABLE
7581  033274  105711                    TSTB    (R1)            ;YES--IS A "CR" AFTER THE DOUBLE PERIOD?
7582  033276  001326                    BNE     3$              ;NO--ASK FOR THIS VARIABLE AGAIN
7583  033300  000137  033714            JMP     OPN.X2          ;YES--CLOSE ALL TEST
7584  033304  005301            7$:     DEC     R1              ;BACK THE POINTER UP BY ONE
7585  033306                    8$:
7586  033306  016402  001606            MOV     PRMLMT(R4),R2   ;UPPER LIMIT OF INPUT
7587  033312  004037  034160            JSR     R0,@#CK.DIG     ;CHECK THE DIGIT(S)
7588  033316  033154                    3$                      ;ILLEGAL INPUT
7589  033320  033154                    3$                      ;INPUT TO LARGE
7590  033322  033330                    9$                      ;TERMINATED WITH A "," OR "CR"
7591  033324  033672                    OPN.N1                  ;TERMINATED WITH A "."
7592  033326  033710                    OPN.X1                  ;TERMINATED WITH A ".."
7593  033330  010264  001506    9$:     MOV     R2,RPT(R4)      ;SAVE THIS VARIABLE
7594  033334  000705                    BR      2$              ;MOVE TO NEXT VARIABLE
7595  033336  104401  044330    OPNPAT: TYPE    ,MSG.SP         ;TYPE SPACES
7596  033342  104401  043210            TYPE    ,MSG.PAT        ;TYPE "PAT"
7597  033346  104401  043214            TYPE    ,MSG.EQ         ;TYPE "="
7598  033352  016446  001506            MOV     RPT(R4),-(SP)   ;;SAVE RPT(R4) FOR TYPEOUT
7599  033356  104402                    TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
7600  033360  104401  044331            TYPE    ,MSG.SP+1       ;TYPE ONE SPACE
7601  033364  104411                    RDLIN                   ;READ ASCIZ STRING
7602  033366  012601                    MOV     (SP)+,R1        ;PICKUP POINTER
7603  033370  004037  034104            JSR     R0,@#CK.CHR     ;CHECK ONE CHARACTER
7604  033374  033336                    OPNPAT                  ;ILLEGAL CHARACTER
7605  033376  033134                    OPNPRM                  ;CARRIAGE RETURN
7606  033400  033432                    3$                      ;"/"
7607  033402  033134                    OPNPRM                  ;":"
7608  033404  033410                    1$                      ;"."
7609  033406  033430                    2$                      ;DIGIT 0-9
7610  033410  105711            1$:     TSTB    (R1)            ;"CR" AFTER THE PERIOD?
7611  033412  001531                    BEQ     OPN.N2          ;YES--GO CLOSE THIS TEST
7612  033414  122721  000056            CMPB    #'.,(R1)+       ;NO--PERIOD?
7613  033420  001346                    BNE     OPNPAT          ;NO--LOOP
7614  033422  105711                    TSTB    (R1)            ;"CR" AFTER A DOUBLE PERIOD?
7615  033424  001533                    BEQ     OPN.X2          ;YES--GO START TESTING
7616  033426  000743                    BR      OPNPAT          ;NO--LOOP
7617  033430  005301            2$:     DEC     R1              ;BACKUP THE ASCII POINTER
7618  033432                    3$:
7619  033432  004037  034344            JSR     R0,@#CK.NUM     ;CHECK THE NUMBER
7620  033436  033336                    OPNPAT                  ;ILLEGAL INPUT
7621  033440  033446                    4$                      ;TERMINATED WITH A "," OR "CR"
7622  033442  033672                    OPN.N1                  ;TERMINATED WITH A "."
7623  033444  033710                    OPN.X1                  ;TERMINATED WITH A ".."
7624  033446  010264  001506    4$:     MOV     R2,RPT(R4)      ;SAVE THE INPUT NUMBER
7625  033452  006002                    ROR     R2              ;OPEN PATTERN 0?
7626  033454  103227                    BCC     OPNPRM          ;NO--START AT BEGINNING OF PARAMETER TABLE
7627  033456  104412                    SAVREG                  ;SAVE R0 - R5
7628  033460  005000    OPNWDS: CLR     R0                      ;START WITH WORD 0
7629  033462  012704  003104            MOV     #PAT0,R4
7630  033466                    1$:
7631  033466  104401  033474            TYPE    ,65$            ;;TYPE ASCIZ STRING
7632  033472  000403                    BR      64$             ;;GET OVER THE ASCIZ
7633                    ;;65$:  .ASCIZ  /   WD/
```

```
7634  033502                          64$:
7635  033502  010046                        MOV     R0,-(SP)                ;PUT R0 ON THE STACK
7636  033504  004737  023336                JSR     PC,@#$SB2D      ;CHANGE R0 TO DECIMAL ASCIZ
7637  033510  004737  023566                JSR     PC,@#$SUPRS     ;TYPE WITHOUT LEADING ZEROS
7638  033514  104401  043214                TYPE    ,MSG.EQ         ;TYPE "="
7639  033520  011446                        MOV     (R4),-(SP)      ;;SAVE (R4) FOR TYPEOUT
7640  033522  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
7641  033524  104411                        RDLIN                   ;READ ASCIZ STRING
7642  033526  012601                        MOV     (SP)+,R1        ;PICKUP THE POINTER
7643  033530  004037  034104                JSR     R0,@#CK.CHR     ;CHECK ONE CHARACTER
7644  033534  033466                        1$                      ;ILLEGAL CHARACTER
7645  033536  033570                        4$                      ;CARRIAGE RETURN
7646  033540  033552                        2$                      ;"/"
7647  033542  033570                        4$                      ;","
7648  033544  033604                        5$                      ;"."
7649  033546  033550                        10$                     ;DIGIT 0-9
7650  033550  005301                10$:    DEC     R1              ;BACKUP THE ASCII POINTER
7651  033552                        2$:
7652  033552  004037  034344                JSR     R0,@#CK.NUM     ;CHECK THE NUMBER
7653  033556  033466                        1$                      ;ILLEGAL INPUT
7654  033560  033566                        3$                      ;TERMINATED WITH A "," OR "CR"
7655  033562  033624                        6$                      ;TERMINATED WITH A "."
7656  033564  033636                        8$                      ;TERMINATED WITH A ".."
7657  033566  010214                3$:     MOV     R2,(R4)         ;SAVE THE INPUT
7658  033570  005724                4$:     TST     (R4)+           ;MOVE TO NEXT WORD
7659  033572  005200                        INC     R0              ;INCREMENT THE COUNT
7660  033574  022700  000020                CMP     #16.,R0         ;COUNT TO LARGE?
7661  033600  003332                        BGT     1$              ;NO--BRANCH
7662  033602  000726                        BR      OPNWDS          ;YES--BRANCH
7663  033604  105711                5$:     TSTB    (R1)            ;"CR" AFTER THE PERIOD?
7664  033606  001407                        BEQ     7$              ;YES--GO CLOSE THIS TEST
7665  033610  122721  000056                CMPB    #'.,(R1)+       ;NO--PERIOD?
7666  033614  001324                        BNE     1$              ;NO--BRANCH ILLEGAL INPUT STRING
7667  033616  105711                        TSTB    (R1)            ;"CR" AFTER THE "PERIOD-PERIOD"?
7668  033620  001407                        BEQ     9$              ;YES--GO START TESTING
7669  033622  000721                        BR      1$              ;NO--LOOP
7670  033624  010224                6$:     MOV     R2,(R4)+        ;SAVE THE INPUT
7671  033626  004737  033650        7$:     JSR     PC,@#CLSWDS     ;CLOSE THE DATA PATTERN
7672  033632  104413                        RESREG                  ;RESTORE R0 - R5
7673  033634  000420                        BR      OPN.N2          ;MOVE TO NEXT TEST
7674  033636  010224                8$:     MOV     R2,(R4)+        ;SAVE THE INPUT
7675  033640  004737  033650        9$:     JSR     PC,@#CLSWDS     ;CLOSE THE DATA PATTERN
7676  033644  104413                        RESREG                  ;RESTORE R0 - R5
7677  033646  000422                        BR      OPN.X2          ;START TESTING
7678  033650  012701  003104        CLSWDS: MOV     #PAT0,R1        ;FIRST ADDRESS OF DATA PATTERN
7679  033654  005200                1$:     INC     R0              ;COUNT THE LAST WORD THAT WAS STORED
7680  033656  022700  000017                CMP     #15.,R0         ;END OF TABLE
7681  033662  002402                        BLT     2$              ;YES--EXIT
7682  033664  012124                        MOV     (R1)+,(R4)+     ;COPY
7683  033666  000772                        BR      1$              ;LOOP
7684  033670  000207                2$:     RTS     PC              ;RETURN
7685  033672  010264  001506        OPN.N1: MOV     R2,RPT(R4)      ;SAVE THIS VARIABLE
7686  033676  005726                OPN.N2: TST     (SP)+           ;CLEAN OFF THE STACK
7687  033700  004737  033750                JSR     PC,CLOSE        ;CLOSE THIS TEST
7688  033704  000137  032772                JMP     OPN.1           ;GO OPEN THE NEXT TEST
7689  033710  010264  001506        OPN.X1: MOV     R2,RPT(R4)      ;SAVE THIS VARIABLE
```

```
7690  033714  005726                OPN.X2: TST     (SP)+           ;CLEAN OFF THE STACK
7691  033716  004737  033750        1$:     JSR     PC,CLOSE        ;CLOSE THIS TEST
7692  033722  005725                2$:     TST     (R5)+           ;UPDATE THE INDEX
7693  033724  020527  000034                CMP     R5,#16*2        ;INDEX TO BIG?
7694  033730  002403                        BLT     3$              ;NO--BRANCH
7695  033732  104413                        RESREG                  ;RESTORE R0 - R5
7696  033734  000137  032124                JMP     GT.PR2          ;GO TO EXIT
7697  033740  036503  001424        3$:     BIT     BITS(R5),R3     ;IS THIS TEST OPEN FOR CHANGE?
7698  033744  001364                        BNE     1$              ;YES--GO CLOSE IT
7699  033746  000765                        BR      2$              ;NO--MOVE TO NEXT TEST
7700  033750  104412                CLOSE:  SAVREG                  ;SAVE R0 - R5
7701  033752  012700  001504                MOV     #PRM,R0         ;"FROM" ADDRESS
7702  033756  016501  001536                MOV     PRMPT(R5),R1    ;"TO" ADDRESS
7703  033762  012002                        MOV     (R0)+,R2        ;"FROM" INDICATOR
7704  033764  012103                        MOV     (R1)+,R3        ;"TO" INDICATOR
7705  033766  012704  000001                MOV     #1,R4           ;TEST BIT START A "RPT"
7706  033772  030402                1$:     BIT     R4,R2           ;PARAMETER TO BE MOVED?
7707  033774  001403                        BEQ     2$              ;NO--BRANCH
7708  033776  030403                        BIT     R4,R3           ;A PLACE TO PUT IT?
7709  034000  001404                        BEQ     3$              ;NO--BRANCH
7710  034002  011011                        MOV     (R0),(R1)       ;YES--MOVE "FROM" TO "TO"
7711  034004  030403                2$:     BIT     R4,R3           ;"TO" PARAMETER?
7712  034006  001401                        BEQ     3$              ;NO--BRANCH
7713  034010  005721                        TST     (R1)+           ;YES--UPDATE THE POINTER
7714  034012  005720                3$:     TST     (R0)+           ;UPDATE FROM POINTER
7715  034014  006304                        ASL     R4              ;POSITION THE TEST BIT
7716  034016  032704  002000                BIT     #BIT10,R4       ;DONE?
7717  034022  001763                        BEQ     1$              ;NO--BRANCH
7718  034024  104413                        RESREG                  ;RESTORE R0 - R5
7719  034026  000207                        RTS     PC              ;RETURN
7720                                ;THIS ROUTINE IS USED TO CHECK IF AN
7721                                ;ASCII CHARACTER IS A DIGIT BETWEEN 0 AND 7.
7722                                ;CALL
7723                                ;       MOV     #ADR,R1         ;ADDRESS OF ASCII CHARACTER
7724                                ;       JSR     R0,@#CK.OCT     ;CHECK THE CHARACTER
7725                                ;       RETURN1                 ;CHARACTER IS NOT BETWEEN 0-7
7726                                ;       RETURN2                 ;CHARACTER IS IN R2 AS A
7727                                ;                               ;OCTAL DIGIT
7728
7729  034030  121127  000060        CK.OCT: CMPB    (R1),#'0        ;LESS THAN ZERO?
7730  034034  103407                        BLO     1$              ;YES -- BRANCH
7731  034036  121127  000067                CMPB    (R1),#'7        ;GREATER THAN SEVEN?
7732  034042  101004                        BHI     1$              ;YES -- BRANCH
7733  034044  111102                        MOVB    (R1),R2         ;GET THE CHARACTER
7734  034046  042702  177770                BIC     #^C7,R2         ;STRIP AWAY THE ASCII
7735  034052  005720                        TST     (R0)+           ;ADJUST FOR RETURN
7736  034054  000200                1$:     RTS     R0              ;RETURN
7737
7738                                ;THIS ROUTINE IS USED TO CHECK AN ASCII CHARACTER
7739                                ;AND DETERMINE IF IT IS A DIGIT BETWEEN 0 AND 9.
7740                                ;CALL
7741                                ;       MOV     #ADR,R1         ;ADDRESS OF ASCII CHARACTER
7742                                ;       JSR     R0,@#CK.DEC     ;CHECK THE CHARACTER
7743                                ;       RETURN1                 ;NOT BETWEEN 0 AND 9
7744                                ;       RETURN2                 ;BETWEEN 0 AND 9
7745                                ;                               ;R2 = DIGIT
```

```
7746
7747  034056  121127  000060          CK.DEC: CMPB    (R1),#'0         ;LESS THAN ZERO?
7748  034062  103407                          BLO     1$               ;YES -- BRANCH
7749  034064  121127  000071                  CMPB    (R1),#'9                 ;GREATER THAN NINE?
7750  034070  101004                          BHI     1$               ;YES -- BRANCH
7751  034072  111102                          MOVB    (R1),R2          ;GET THE CHARACTER
7752  034074  042702  000060                  BIC     #'0,R2           ;STRIP AWAY THE ASCII
7753  034100  005720                          TST     (R0)+            ;ADJUST FOR RETURN
7754  034102  000200                  1$:     RTS     R0               ;RETURN
7755
7756                                  ;THIS ROUTINE WILL CHECK AN ASCII CHARACTER TO
7757                                  ;DETERMINE WHAT IT IS.
7758                                  ;CALL
7759                                  ;       MOV     #ADR,R1          ;ADDRESS OF ASCII CHARACTER
7760                                  ;       JSR     R0,@#CK.CHR      ;CHECK CHARACTER
7761                                  ;       RETURN  ADR1             ;UNKNOWN CHARACTER
7762                                  ;       RETURN  ADR2             ;CARRIAGE RETURN * (R1)=ADR+1
7763                                  ;       RETURN  ADR3             ;SLASH * (R1)=ADR+1
7764                                  ;       RETURN  ADR4             ;COMMA * (R1)=ADR+1
7765                                  ;       RETURN  ADR5             ;PERIOD * (R1)=ADR+1
7766                                  ;       RETURN  ADR6             ;DIGIT BETWEEN 0 AND 9.
7767                                  ;                                ;R2 = DIGIT * (R1)=ADR+1
7768
7769  034104  105711                  CK.CHR: TSTB    (R1)             ;"CARRIAGE RETURN"?
7770  034106  001420                          BEQ     4$               ;YES -- BRANCH
7771  034110  121127  000057                  CMPB    (R1),#'/         ;"SLASH"?
7772  034114  001414                          BEQ     3$               ;YES -- BRANCH
7773  034116  121127  000054                  CMPB    (R1),#',         ;"COMMA"?
7774  034122  001410                          BEQ     2$               ;YES -- BRANCH
7775  034124  121127  000056                  CMPB    (R1),#'.         ;"PERIOD"?
7776  034130  001404                          BEQ     1$               ;YES -- BRANCH
7777  034132  004037  034056                  JSR     R0,@#CK.DEC      ;"DIGIT"?
7778  034136  000406                          BR      5$               ;NO -- BRANCH
7779  034140  005720                          TST     (R0)+            ;DIGIT BETWEEN 0-9
7780  034142  005720                  1$:     TST     (R0)+            ;PERIOD
7781  034144  005720                  2$:     TST     (R0)+            ;COMMA
7782  034146  005720                  3$:     TST     (R0)+            ;SLASH
7783  034150  005720                  4$:     TST     (R0)+            ;CARRIAGE RETURN
7784  034152  005201                          INC     R1               ;MOVE POINTER TO NEXT CHARACTER
7785  034154  011000                  5$:     MOV     (R0),R0          ;UNKNOWN CHARACTER
7786  034156  000200                          RTS     R0               ;RETURN
7787
7788                                  ;THIS ROUTINE CHECKS AN ASCII STRING FOR LEGAL
7789                                  ;CHARACTERS AND FORMS A DECIMAL VALUE BINARY NUMBER IN R2.
7790                                  ;CALL
7791                                  ;       MOV     #ADR,R1          ;ADDRESS OF ASCIZ STRING
7792                                  ;       MOV     #NUM,R2          ;MAX. MAGNITUDE OF INPUT NUMBER
7793                                  ;       JSR     R0,@#CK.DIG      ;CHECK DIGITS
7794                                  ;       RETURN  ADR1             ;ILLEGAL CHARACTER -- R2=?
7795                                  ;       RETURN  ADR2             ;INPUT NUMBER TO LARGE -- R2=?
7796                                  ;       RETURN  ADR3             ;"COMMA" -- R2 = NUMBER
7797                                  ;       RETURN  ADR4             ;"PERIOD" -- R2 = NUMBER
7798                                  ;       RETURN  ADR5             ;"PERIOD-PERIOD" -- R2 = NUMBER
7799
7800  034160  010446                  CK.DIG: MOV     R4,-(SP)         ;SAVE R4
7801  034162  010346                          MOV     R3,-(SP)         ;SAVE R3
```

```
 7802  034164  010246                        MOV     R2,-(SP)        ;SAVE THE MAX. SIZE ON THE STACK
 7803  034166  005^02                        CLR     R2              ;START WITH 0
 7804  034170  005003                        CLR     R3
 7805  034172  005004                        CLR     R4
 7806  034174  004037  034104                JSR     R0,@#CK.CHR     ;CHECK ONE CHARACTER
 7807  034200  034330                        8$                      ;ILLEGAL CHARACTER
 7808  034202  034330                        8$                      ;CARRIAGE RETURN
 7809  034204  034330                        8$                      ;"/"
 7810  034206  034330                        8$                      ;":"
 7811  034210  034330                        8$                      ;","
 7812  034212  034214                        1$                      ;DIGIT 0-9
 7813  034214  006303                1$:     ASL     R3              ;2
 7814  034216  010346                        MOV     R3,-(SP)        ;SAVE *2
 7815  034220  006303                        ASL     R3              ;4
 7816  034222  006303                        ASL     R3              ;8
 7817  034224  062603                        ADD     (SP)+,R3        ;(*8)+(*2)=*10.
 7818  034226  060203                        ADD     R2,R3           ;UPDATE THE INPUT NUMBER
 7819  034230  004037  034104                JSR     R0,@#CK.CHR     ;CHECK ONE CHARACTER
 7820  034234  034330                        8$                      ;ILLEGAL CHARACTER
 7821  034236  034250                        9$                      ;CARRIAGE RETURN
 7822  034240  034330                        8$                      ;"/"
 7823  034242  034256                        3$                      ;":"
 7824  034244  034254                        2$                      ;","
 7825  034246  034214                        1$                      ;DIGIT 0-9
 7826  034250  005301                9$:     DEC     R1              ;BACKUP THE CHARACTER POINTER
 7827  034252  000401                        BR      3$              ;CONTINE
 7828  034254  005724                2$:     TST     (R4)+           ;"PERIOD"
 7829  034256  005724                3$:     TST     (R4)+           ;"COMMA" OR "CR"
 7830  034260  004037  034104                JSR     R0,@#CK.CHR     ;CHECK ONE CHARACTER
 7831  034264  034330                        8$                      ;ILLEGAL CHARACTER
 7832  034266  034320                        6$                      ;CARRIAGE RETURN
 7833  034270  034330                        8$                      ;"/"
 7834  034272  034330                        8$                      ;":"
 7835  034274  034300                        4$                      ;","
 7836  034276  034310                        5$                      ;DIGIT 0-9
 7837  034300  005724                4$:     TST     (R4)+           ;"PERIOD-PERIOD"
 7838  034302  105711                        TSTB    (R1)            ;"CR"?
 7839  034304  001405                        BEQ     6$              ;YES--BRANCH
 7840  034306  000410                        BR      8$
 7841  034310  126127  177776  000054  5$:   CMPB    -2(R1),#',      ;WAS CHARACTER BEFORE THE DIGIT A COMMA?
 7842  034316  001004                        BNE     8$              ;NO--EXIT
 7843  034320  020316                6$:     CMP     R3,(SP)         ;INPUT TO LARGE?
 7844  034322  101001                        BHI     7$              ;YES -- BRANCH
 7845  034324  060400                        ADD     R4,R0           ;ADJUST RETURN ADDRESS
 7846  034326  005720                7$:     TST     (R0)+
 7847  034330  010302                8$:     MOV     R3,R2           ;NUMBER TO R2
 7848  034332  005726                        TST     (SP)+           ;CLEAN MAX. SIZE OFF OF STACK
 7849  034334  012603                        MOV     (SP)+,R3        ;RESTORE R3
 7850  034336  012604                        MOV     (SP)+,R4        ;RESTORE R4
 7851  034340  011000                        MOV     (R0),R0         ;GET RETURN ADDRESS
 7852  034342  000200                        RTS     R0              ;RETURN
 7853
 7854                          ;THIS ROUTINE CHECKS AN ASCIZ STRING FOR LEGAL CHARACTERS
 7855                          ;AND FORMS AN OCTAL NUMBER IN R2
 7856                          ;CALL:
 7857                          ;       MOV     #ADR,R1         ;ADDRESS OF ASCIZ STRING
```

```
7858                                    ;       JSR     R0,@#CK.NUM     ;GO FORM THE NUMBER
7859                                    ;       RETURN  ADR1            ;ILLEGAL CHARACTER IN THE INPUT STRING
7860                                    ;       RETURN  ADR2            ;"COMMA" OR "CR"--R2=NUMBER
7861                                    ;       RETURN  ADR3            ;"PERIOD"--R2=NUMBER
7862                                    ;       RETURN  ADR4            ;"PERIOD-PERIOD"--R2=NUMBER
7863
7864  034344  010346                    CK.NUM: MOV     R3,-(SP)        ;SAVE R3
7865  034346  005003                            CLR     R3              ;START NUMBER AT ZERO
7866  034350  004037  034030                    JSR     R0,@#CK.OCT     ;OCTAL DIGIT?
7867  034354  000440                            BR      6$              ;NO--BRANCH
7868  034356  005201                    1$:     INC     R1              ;MOVE TO NEXT CHARACTER
7869  034360  006303                            ASL     R3              ;FOR THE OCTAL NUMBER IN R3
7870  034362  103435                            BCS     6$              ;DON'T LET IT GET TO BIG
7871  034364  006303                            ASL     R3
7872  034366  103433                            BCS     6$
7873  034370  006303                            ASL     R3
7874  034372  103431                            BCS     6$
7875  034374  060203                            ADD     R2,R3
7876  034376  004037  034030                    JSR     R0,@#CK.OCT     ;IS THIS AN OCTAL DIGIT?
7877  034402  000401                            BR      2$              ;NO--FIND OUT WHAT IT IS
7878  034404  000764                            BR      1$              ;YES--MAKE IT PART OF THE NUMBER
7879  034406  010302                    2$:     MOV     R3,R2           ;SAVE THE OCTAL NUMBER
7880  034410  005003                            CLR     R3              ;START WITH ZERO INDEX
7881  034412  004037  034104                    JSR     R0,@#CK.CHR     ;CHECK ONE CHARACTER
7882  034416  034456                            6$                      ;ILLEGAL CHARACTER
7883  034420  034446                            5$                      ;CARRIAGE RETURN
7884  034422  034456                            6$                      ;"/"
7885  034424  034446                            5$                      ;","
7886  034426  034432                            3$                      ;"."
7887  034430  034456                            6$                      ;DIGIT 0-9
7888  034432  005723                    3$:     TST     (R3)+           ;"PERIOD"
7889  034434  121127  000056                    CMPB    (R1),#'.        ;"PERIOD-PERIOD"?
7890  034440  001002                            BNE     5$              ;NO--BRANCH
7891  034442  005201                            INC     R1              ;YES--ADVANCE THE POINTER
7892  034444  005723                    4$:     TST     (R3)+           ;"PERIOD-PERIOD"
7893  034446  005723                    5$:     TST     (R3)+           ;"COMMA"
7894  034450  105711                            TSTB    (R1)            ;"CR"?
7895  034452  001001                            BNE     6$              ;NO--BRANCH
7896  034454  060300                            ADD     R3,R0           ;YES--SAVE THE OCTAL NUMBER
7897  034456  012603                    6$:     MOV     (SP)+,R3        ;RESTORE R3
7898  034460  011000                            MOV     (R0),R0         ;PICKUP EXIT ADDRESS
7899  034462  000200                            RTS     R0              ;RETURN
```

```
                                    ;;*****************************************************************

                                    .SBTTL  SINGLE/DUAL PORT RH11/RP04/5/6 DRIVER (REV 1.0)

                                    ;COPYRIGHT (C) 1976
                                    ;DIGITAL EQUIPMENT CORP.
                                    ;MAYNARD, MA 01754
                                    ;AUTHOR(S): JIM LACEY/CHUCK HESS

                                    ;;*****************************************************************

                                    ;STORAGE FOR RPDS1, RPER1, RPER2, AND RPER3 ON AN ERROR "2"
                                            ;RPERRS   = RPDS1
                                            ;RPERRS+2 = RPER1
                                            ;RPERRS+4 = RPER2
                                            ;RPERRS+6 = RPER3

034464  000000  000000  000000  RPERRS: .WORD   0,0,0,0
034472  000000

                                    ;TABLE OF DRIVE ACTIVE INDICATORS (DRVACT=8 BYTES)
                                            ;DRVACT=0 IF DRIVE IS IDLE
                                            ;DRVACT>0 IF DRIVE IS ACTIVE WITH A COMMAND
                                            ;DRVACT<0 IF DRIVE IS ACTIVE WITH AN ERROR RECOVERY OPERATION

034474      000                 DRVACT: .BYTE   0               ;DRIVE 0
034475      000                         .BYTE   0               ;DRIVE 1
034476      000                         .BYTE   0               ;DRIVE 2
034477      000                         .BYTE   0               ;DRIVE 3
034500      000                         .BYTE   0               ;DRIVE 4
034501      000                         .BYTE   0               ;DRIVE 5
034502      000                         .BYTE   0               ;DRIVE 6
034503      000                         .BYTE   0               ;DRIVE 7

                                ;TABLE OF DRIVE STATUS INDICATORS (DRVSTA=8 BYTES)
                                        ;DRVSTA=0 IF DRIVE IS OFFLINE OR NONEXSITENT
                                        ;DRVSTA>0 IF DRIVE IS ONLINE
                                        ;DRVSTA<0 IF DRIVE IS UNSAFE

034504      000                 DRVSTA: .BYTE   0               ;DRIVE 0
034505      000                         .BYTE   0               ;DRIVE 1
034506      000                         .BYTE   0               ;DRIVE 2
034507      000                         .BYTE   0               ;DRIVE 3
034510      000                         .BYTE   0               ;DRIVE 4
034511      000                         .BYTE   0               ;DRIVE 5
034512      000                         .BYTE   0               ;DRIVE 6
034513      000                         .BYTE   0               ;DRIVE 7

                                ;TABLE OF DRIVE TYPES (DRVTYP=8 BYTES)
                                        ;DRVTYP=0 IF DRIVE IS NONEXISTENT (DRVSTA=0, ALSO)
                                        ;DRVTYP=1 IF DRIVE IS RP04
                                        ;DRVTYP=2 IF DRIVE IS RP05
                                        ;DRVTYP=4 IF DRIVE IS RP06
                                        ;DRVTYP=-1 IF NOT RP04/5/6
```

```
7956  034514  000                 DRVTYP: .BYTE   0               ;DRIVE 0
7957  034515  000                         .BYTE   0               ;DRIVE 1
7958  034516  000                         .BYTE   0               ;DRIVE 2
7959  034517  000                         .BYTE   0               ;DRIVE 3
7960  034520  000                         .BYTE   0               ;DRIVE 4
7961  034521  000                         .BYTE   0               ;DRIVE 5
7962  034522  000                         .BYTE   0               ;DRIVE 6
7963  034523  000                         .BYTE   0               ;DRIVE 7
7964
7965                              ;TABLE OF DUAL PORT INITIALIZATION INDICATORS
7966                                      ;DPINT=0 IF INITIALIZATION IS NOT ACTIVE ON THE DRIVE
7967                                      ;DPINT<0 IF INITIALIZATION IS IN PROGRESS
7968
7969  034524  000                 DPINT:  .BYTE   0               ;DRIVE 0
7970  034525  000                         .BYTE   0               ;DRIVE 1
7971  034526  000                         .BYTE   0               ;DRIVE 2
7972  034527  000                         .BYTE   0               ;DRIVE 3
7973  034530  000                         .BYTE   0               ;DRIVE 4
7974  034531  000                         .BYTE   0               ;DRIVE 5
7975  034532  000                         .BYTE   0               ;DRIVE 6
7976  034533  000                         .BYTE   0               ;DRIVE 7
7977
7978                              ;TABLE OF PENDING DUAL PORT REQUESTS
7979                                      ;DPRQS=0 IF THAT A DUAL PORT REQUEST IS NOT PENDING FOR THAT DRIVE
7980                                      ;DPRQS<0 IF THAT A DUAL PORT REQUEST IS PENDING FOR THAT DRIVE
7981
7982  034534  000                 DPRQS:  .BYTE   0               ;DRIVE 0
7983  034535  000                         .BYTE   0               ;DRIVE 1
7984  034536  000                         .BYTE   0               ;DRIVE 2
7985  034537  000                         .BYTE   0               ;DRIVE 3
7986  034540  000                         .BYTE   0               ;DRIVE 4
7987  034541  000                         .BYTE   0               ;DRIVE 5
7988  034542  000                         .BYTE   0               ;DRIVE 6
7989  034543  000                         .BYTE   0               ;DRIVE 7
7990
7991                              ;TRANSFER WAIT FLAG (TRNSWT=1 WORD)
7992                                      ;THIS IS A ONE WORD QUEUE. IT WILL CONTAIN THE ADDRESS OF
7993                                      ;"DPB" OF THE I/O OPERATION.
7994
7995  034544  000000              TRNSWT: .WORD   0
7996
7997                              ;SEARCH WAIT KEYS (SRCHWT=1 WORD)
7998                                      ;THIS IS A ONE WORD QUEUE THAT WILL CONTAIN A KEY FOR EACH OF
7999                                      ;THE DRIVES THAT ARE PERFORMING A SEARCH COMMAND FOR THE I/O
8000                                      ;REQUEST THAT IS AT THE TOP OF THEIR REQUEST QUEUE.
8001                                      ;EACH DRIVE IS ASSIGNED ONE BIT, STARTING AT BIT00 FOR DRIVE 0.
8002
8003  034546  000000              SRCHWT: .WORD   0
8004
8005                              ;RP04/5/6 DRIVER ACTIVE FLAG (ACTDRV=1 BYTE)
8006                                      ;ACTDRV=0 IF DRIVER IS INACTIVE
8007                                      ;ACTDRV>0 IF DRIVER IS ACTIVE
8008
8009  034550  000                 ACTDRV: .BYTE   0
8010
8011                              ;SOFTWARE TIMER ROUTINE ACTIVE FLAG (ACTSTR=1 BYTE)
```

```
                                        ;ACTSTR=0 IF SOFTWARE TIMER ROUTINE IS INACTIVE
                                        ;ACTSTR>0 IF SOFTWARE TIMER ROUTINE IS ACTIVE

034551  000                     ACTSTR: .BYTE   0

                                ;UNLOAD FLAG (ULDFLG=8 BYTES)
                                        ;ULDFLG=0 IF NO UNLOAD COMMAND
                                        ;ULDFLG>0 IF UNLOAD COMMAND IN PROGRESS
                                        ;ULDFLG<0 IF UNLOAD COMMAND IN WAIT QUEUE

034552  000                     ULDFLG: .BYTE   0               ;DRIVE 0
034553  000                             .BYTE   0               ;DRIVE 1
034554  000                             .BYTE   0               ;DRIVE 2
034555  000                             .BYTE   0               ;DRIVE 3
034556  000                             .BYTE   0               ;DRIVE 4
034557  000                             .BYTE   0               ;DRIVE 5
034560  000                             .BYTE   0               ;DRIVE 6
034561  000                             .BYTE   0               ;DRIVE 7

                                ;LOOK AHEAD COUNT (LACNT=8 BYTES)
                                        ;LACNT WILL INDICATE THE NUMBER OF LOOK AHEADS PERFORMED

034562  000                     LACNT:  .BYTE   0               ;DRIVE 0
034563  000                             .BYTE   0               ;DRIVE 1
034564  000                             .BYTE   0               ;DRIVE 2
034565  000                             .BYTE   0               ;DRIVE 3
034566  000                             .BYTE   0               ;DRIVE 4
034567  000                             .BYTE   0               ;DRIVE 5
034570  000                             .BYTE   0               ;DRIVE 6
034571  000                             .BYTE   0               ;DRIVE 7

                                ;SAVE REGISTERS FLAG (SAVEFG =1 WORD)
                                        ;SAVEFG <0 IF SAVE THE RH11/RP04/5/6 REGISTERS WHEN THE
                                        ;OPERATION IS COMPLETED AS PER (DPB+14).
                                        ;SAVEFG=0 IF SAVE THE RH11/RP04/5/6 REGISTERS, AS PER
                                        ;(DPB+14), AFTER AN ERROR.

034572  000000                  SAVEFG: .WORD   0

                                ;SEEK FLAG (SEEKFG=1 WORD)
                                        ;SEEKFG=0 IF WHEN THE DISK ADDRESS ISN'T IN THE WINDOW
                                        ;FOR A DATA TRANSFER START A SEARCH COMMAND
                                        ;SEEKFG<0 IF DATA TRANSFER WILL DO IMPLIED SEEKS,
                                        ;DISREGARD THE WINDOW

034574  000000                  SEEKFG: .WORD   0

                                ;TIMEOUT TABLE (TIMER=8 WORDS)
                                        ;THIS TABLE CONTAINS THE TIME ALLOWED FOR AN OPERATION

034576  177777                  TIMER:  .WORD   -1       ;DRIVE 0
034600  177777                          .WORD   -1       ;DRIVE 1
034602  177777                          .WORD   -1       ;DRIVE 2
034604  177777                          .WORD   -1       ;DRIVE 3
034606  177777                          .WORD   -1       ;DRIVE 4
034610  177777                          .WORD   -1       ;DRIVE 5
```

```
 8068  034612  177777                          .WORD   -1      ;DRIVE 6
 8069  034614  177777                          .WORD   -1      ;DRIVE 7
 8070
 8071                                  ;DATA TRANSFER UNDERWAY INDICATOR (DTUW=1 WORD)
 8072                                          ;DTUW<0 IF NO DATA TRANSFER UNDERWAY
 8073                                          ;DTUW=+N (WHERE N=0 TO 7) IMPLIES DATA TRANSFER UNDERWAY ON DRIVE N
 8074
 8075  034616  177777                  DTUW:   .WORD   -1
 8076
 8077                                  ;ATTENTION BITS TABLE (ATABIT=8 BYTES)
 8078                                          ;THIS TABLE CONTAINS THE CORRESPONDING BIT TO EACH DRIVES
 8079                                          ;ATTENTION BIT
 8080
 8081  034620     001                  ATABIT: .BYTE   1       ;DRIVE 0
 8082  034621     002                          .BYTE   2       ;DRIVE 1
 8083  034622     004                          .BYTE   4       ;DRIVE 2
 8084  034623     010                          .BYTE   10      ;DRIVE 3
 8085  034624     020                          .BYTE   20      ;DRIVE 4
 8086  034625     040                          .BYTE   40      ;DRIVE 5
 8087  034626     100                          .BYTE   100     ;DRIVE 6
 8088  034627     200                          .BYTE   200     ;DRIVE 7
 8089
 8090                                  ;RP04/5/6 TO RH11 "MASSBUS CONTROL BUS PARITY ERRORS" (MCPE) ALLOWED BEFORE
 8091                                  ;CALLING IT FATAL (MCPEMX=1 WORD)
 8092
 8093  034630  000003                  MCPEMX: .WORD   3
 8094
 8095                                  ;STORAGE FOR RPADR (THE FIRST ADDRESS (776700) OF THE RH11/RP04/5/6),
 8096                                  ;RPVEC (THE VECTOR ADDRESS (254)), AND RPVEC+2 (THE BR LEVEL (5)).
 8097
 8098  034632  176700                  RPADR:  .WORD   176700
 8099  034634  000254  000240          RPVEC:  .WORD   254,5*32.
 8100
 8101                                  ;MAXIMUM NUMBER OF LOOK AHEADS ALLOWED IS 4 (MXLACT=1 WORD)
 8102
 8103  034640  000004                  MXLACT: .WORD   4
 8104                                  ;MAXIMUM DELTA DELAY IS 8 SECTORS (MXDLTA=1 WORD)
 8105
 8106  034642  001000                  MXDLTA: .WORD   8.*64.
 8107                                  ;MINIMUM DELTA DELAY IS 2 SECTORS (MNDLTA=1 WORD)
 8108
 8109  034644  000200                  MNDLTA: .WORD   2*64.
 8110                                  ;MAXIMUM SEARCH FOR I/O WINDOW IS 5 SECTORS (MXWNDW=1 WORD)
 8111
 8112  034646  000005                  MXWNDW: .WORD   5
 8113
 8114                                  ;DEFINITIONS OF THE RH11/RP04/5/6 ADDRESS INDEXES
 8115
 8116          000000                  RPCS1=0                         ;CONTROL AND STATUS REGISTER #1 (DRIVE REG. 00)
 8117          000002                  RPWC=2                          ;WORD COUNT REGISTER (NOT A DRIVE REG)
 8118          000004                  RPBA=4                          ;UNIBUS ADDRESS REGISTER (NOT A DRIVE REG)
 8119          000006                  RPDA=6                          ;DESIRED SECTOR/TRACK ADDRESS REGISTER (DRIVE REG. 05)
 8120          000010                  RPCS2=10                        ;CONTROL AND STATUS REGISTER #2 (NOT A DRIVE REG)
 8121          000012                  RPDS1=12                        ;DRIVE STATUS REGISTER (DRIVE REG 01)
 8122          000014                  RPER1=14                        ;ERROR REGISTER #1 (DRIVE REG. 02)
 8123          000016                  RPAS=16                         ;ATTENTION SUMMARY PSEUDO REGISTER (DRIVE REG. 04)
```

```
 8124            000020                  RPLA=20                         ;LOOK AHEAD REGISTER (DRIVE REG. 07)
 8125            000022                  RPDB=22                         ;DATA BUFFER REGISTER (NOT A DRIVE REG.)
 8126            000024                  RPMR=24                         ;MAINTAINABILITY REGISTER (DRIVE REG. 03)
 8127            000026                  RPDT=26                         ;DRIVE TYPE REGISTER (DRIVE REG. 06)
 8128            000030                  RPSN=30                         ;SERIAL NUMBER REGISTER (DRIVE REG. 10)
 8129            000032                  RPOF=32                         ;OFFSET REGISTER (DRIVE REG. 11)
 8130            000034                  RPCA=34                         ;DESIRED CYLINDER ADDRESS REGISTER (DRIVE REG. 12)
 8131            000036                  RPCC=36                         ;CURRENT CYLINDER ADDRESS REGISTER (DRIVE REG. 13)
 8132            000040                  RPER2=40                        ;ERROR REGISTER #2 (DRIVE REG. 14)
 8133            000042                  RPER3=42                        ;ERROR REGISTER #3 (DRIVE REG. 15)
 8134            000044                  RPEC1=44                        ;ECC POSITION REGISTER (DRIVE REG. 16)
 8135            000046                  RPEC2=46                        ;ECC PATTERN REGISTER (DRIVE REG.  17)
 8136
 8137                                    ;RH11/RP04/5/6 DRIVER INITIALIZATION CODE
 8138                                            ;THIS ROUTINE WILL DETERMINE WHICH RP04/5/6 DRIVES ARE
 8139                                            ;AVAILABLE FOR TESTING AND SET THE DRVSTA INDICATOR
 8140                                            ;TO THE PROPER STATE FOR EACH DRIVE.
 8141                                            ;NOTE: THIS ROUTINE CALLS DRVINT
 8142                                    ;
 8143                                    ;CALL
 8144                                    ;
 8145                                    ;       JSR     PC,RPINIT
 8146                                    ;       RETURN
 8147                                    ;
 8148                                    ;NOTE: THE 'P' OR 'L' CLOCK MUST BE STARTED
 8149                                    ;
 8150  034650  104412                    RPINIT: SAVREG                          ;SAVE R0 - R5
 8151  034652  013746  177776                    MOV     @#PS,-(SP)              ;SAVE THE PRESENT PROCESSOR STATUS
 8152  034656  012737  000240  177776            MOV     #<5*32.>,@#PS           ;CHANGE THE PRIORITY TO 5
 8153  034664  004737  042664                    JSR     PC,CLRQUE               ;CLEAR ALL REQUEST QUEUES
 8154  034670  012701  034464                    MOV     #RPERRS,R1              ;FIRST ADDRESS TO BE CLEARED
 8155  034674  012702  034574                    MOV     #SEEKFG,R2              ;LAST ADDRESS TO BE CLEARED
 8156  034700  005021                    1$:     CLR     (R1)+                   ;CLEAR
 8157  034702  020102                            CMP     R1,R2                   ;ARE WE DONE?
 8158  034704  101775                            BLOS    1$                      ;BRANCH IF NO
 8159  034706  012702  034616                    MOV     #DTUW,R2                ;LAST ADDRESS
 8160  034712  012721  177777            2$:     MOV     #-1,(R1)+               ;INITIALIZE
 8161  034716  020102                            CMP     R1,R2                   ;DONE?
 8162  034720  101774                            BLOS    2$                      ;LOOP IF NO
 8163  034722  005037  034504                    CLR     DRVSTA                  ;SET ALL DRIVES TO OFFLINE
 8164  034726  005037  034506                    CLR     DRVSTA+2
 8165  034732  005037  034510                    CLR     DRVSTA+4
 8166  034736  005037  034512                    CLR     DRVSTA+6
 8167  034742  013703  034634                    MOV     RPVEC,R3                ;SETUP THE RH11/RP04/5/6 VECTOR
 8168  034746  012723  037544                    MOV     #ISR,(R3)+
 8169  034752  013713  034636                    MOV     RPVEC+2,(R3)
 8170  034756  013704  034632                    MOV     RPADR,R4                ;FIRST ADDRESS OF RH11/RP04
 8171  034762  012764  000040  000010            MOV     #BIT05,RPCS2(R4)  ;MASSBUS INIT
 8172  034770  005001                            CLR     R1                      ;START WITH DRIVE 0
 8173  034772  004037  035062            3$:     JSR     R0,DRVINT               ;INIT THE DRIVE
 8174  034776  000401                            BR      4$                      ;'DVA' NOT SET OR PARITY ERROR
 8175  035000  000402                            BR      5$                      ;NORMAL RETURN
 8176  035002  105061  034504            4$:     CLRB    DRVSTA(R1)              ;SET DRIVE STATUS TO OFFLINE
 8177  035006  005201                    5$:     INC     R1                      ;GO TO NEXT DRIVE
 8178  035010  042701  177770                    BIC     #^C7,R1                 ;MASK OUT UNUSED BITS
 8179  035014  001366                            BNE     3$                      ;BR IF MORE DRIVES TO GO
```

```
 8180  035016  012701  000007                MOV     #7,R1           ;START WITH DRIVE 7
 8181  035022  005037  177776                CLR     @#PS            ;CLEAR THE PROCESSOR STATUS
 8182  035026  105761  034524        6$:     TSTB    DPINT(R1)       ;WAITING FOR DRIVE TO SWITCH PORTS ?
 8183  035032  001405                        BEQ     8$              ;BR NOT WAITING
 8184  035034  004737  042320                JSR     PC,SET.IE       ;SET INTERRUPT
 8185  035040  105761  034524        7$:     TSTB    DPINT(R1)       ;DRIVE SWITCHED PORTS ?
 8186  035044  001375                        BNE     7$              ;BR IF NOT
 8187  035046  005301                8$:     DEC     R1              ;GO TO THE NEXT DRIVE
 8188  035050  100366                        BPL     6$              ;CHECK NEXT DRIVE
 8189  035052  012637  177776                MOV     (SP)+,@#PS      ;RESTORE THE PROCESSOR STATUS
 8190  035056  104413                        RESREG                  ;RESTORE R0 - R5
 8191  035060  000207                        RTS     PC              ;BYE-BYE
 8192
 8193                                ;DRIVE INITILIZATION ROUTINE
 8194                                        ;THIS ROUTINE DETERMINES IF A DRIVE EXIST AND IF IT IS
 8195                                        ;AN RP04/5/6. IF IT IS, A "READ-IN PRESET" IS ISSUED AND FMT22
 8196                                        ;IS SET TO A "1". THEN MOL, DPR, DRY, AND VV ARE CHECKED TO
 8197                                        ;INSURE THEY ARE ALL ON A "1". AND DEPENDING ON THEIR STATE,
 8198                                        ;DRVSTA IS SET TO THE PROPER CONDITION.
 8199                                ;CALL
 8200                                ;       MOV     #DRVNUM,R1      ;DRIVE NUMBER TO R1
 8201                                ;       MOV     RPADR,R4        ;UNIBUS ADDRESS OF RH11/RP04/5/6 (RPCS1)
 8202                                ;       JSR     R0,DRVINT       ;CALLED BY A JSR
 8203                                ;       RETURN1                 ;ERROR OCCURRED (PARITY)
 8204                                ;       RETURN2                 ;NORMAL RETURN
 8205                                ;
 8206
 8207  035062  010546                DRVINT: MOV     R5,-(SP)        ;SAVE R5
 8208  035064  105061  034504                CLRB    DRVSTA(R1)      ;START DRIVE STATUS AS OFFLINE
 8209  035070  105061  034514                CLRB    DRVTYP(R1)      ;CLEAR THE DRIVE TYPE INDICATOR
 8210  035074  105061  034552                CLRB    ULDFLG(R1)      ;CLEAR THE UNLOAD FLAG
 8211  035100  010164  000010                MOV     R1,RPCS2(R4)    ;SELECT A DRIVE
 8212  035104  112764  000111  000000        MOVB    #111,RPCS1(R4)  ;DO A DRIVE CLEAR COMMAND (& SEIZE DRIVE)
 8213  035112  032764  010000  000010        BIT     #BIT12,RPCS2(R4) ;NONEXISTENT DRIVE?
 8214  035120  001403                        BEQ     1$              ;NO---BRANCH
 8215  035122  004737  042320                JSR     PC,SET.IE       ;GO SET "IE" WITHOUT A "TRE"
 8216  035126  000533                        BR      6$              ;LEAVE THIS ROUTINE
 8217  035130  105061  034504        1$:     CLRB    DRVSTA(R1)      ;SET DRIVE STATUS TO OFFLINE
 8218  035134  032764  004000  000000        BIT     #BIT11,RPCS1(R4)  ;SEE IF DRIVE AVAILABLE
 8219  035142  001527                        BEQ     7$              ;BR IF DRIVE NOT AVAILABLE
 8220  035144  004037  041630                JSR     R0,RD.RP        ;READ THE DRIVE TYPE REG.
 8221  035150  000026                        RPDT
 8222  035152  035442                        8$                      ;ERROR RETURN ADDRESS
 8223  035154  012605                        MOV     (SP)+,R5        ;PUT DRIVE TYPE IN R5
 8224  035156  112761  000001  034514        MOVB    #1,DRVTYP(R1)   ;SET RP04 INDICATOR
 8225  035164  022705  020020                CMP     #20020,R5       ;IS IT A SINGLE PORT RP04?
 8226  035170  001431                        BEQ     2$              ;BRANCH IF YES
 8227  035172  022705  024020                CMP     #24020,R5       ;IS IT A DUAL PORT RP04?
 8228  035176  001426                        BEQ     2$              ;BR IF YES
 8229  035200  112761  000002  034514        MOVB    #2,DRVTYP(R1)   ;SET RP05 INDICATOR
 8230  035206  022705  020021                CMP     #20021,R5       ;SINGLE PORT RP05 ?
 8231  035212  001420                        BEQ     2$              ;BR IF YES
 8232  035214  022705  024021                CMP     #24021,R5       ;DUAL PORT RP05 ?
 8233  035220  001415                        BEQ     2$              ;BR IF YES
 8234  035222  112761  000004  034514        MOVB    #4,DRVTYP(R1)   ;SET RP06 INDICATOR
 8235  035230  022705  020022                CMP     #20022,R5       ;SINGLE PORT RP06 ?
```

```
8236  035234  001407                          BEQ     2$                ;BR IF YES
8237  035236  022705  024022                  CMP     #24022,R5         ;DUAL PORT RP06 ?
8238  035242  001404                          BEQ     2$                ;BR IF YES
8239  035244  112761  177777  034514          MOVB    #-1,DRVTYP(R1)    ;SET INDICATOR TO 'OTHER'
8240  035252  000461                          BR      6$                ;EXIT
8241  035254  005737  035446          2$:     TST     TSTPGM            ;INHIBIT PROGRAMMABLE DRIVE?
8242  035260  001010                          BNE     9$                ;BRANCH IF NO
8243  035262  032764  001000  000012          BIT     #BIT09, RPDS1(R4);IS DRIVE PROGRAMMABLE?
8244  035270  001404                          BEQ     9$                ;BRANCH IF NO
8245  035272  152761  000010  034514          BISB    #BIT03, DRVTYP(R1);SET INDICATOR
8246  035300  000446                          BR      6$                ;EXIT
8247  035302  012746  000121          9$:     MOV     #121,-(SP)        ;DO A "READ-IN PRESET"
8248  035306  004037  042010                  JSR     R0,WRT.RP
8249  035312  000000                          RPCS1
8250  035314  035442                          8$
8251  035316  012746  010000                  MOV     #BIT12,-(SP)      ;SET FMT22=1
8252  035322  004037  042010                  JSR     R0,WRT.RP
8253  035326  000032                          RPOF
8254  035330  035442                          8$
8255  035332  004037  041630                  JSR     R0,RD.RP          ;READ RPDS1
8256  035336  000012                          RPDS1
8257  035340  035442                          8$
8258  035342  012605                          MOV     (SP)+,R5          ;AND SAVE IT IN R5
8259  035344  100015                          BPL     4$                ;BRANCH IF ATA=0
8260  035346  116164  034620  000016          MOVB    ATABIT(R1),RPAS(R4) ;CLEAR ATTENTION BIT
8261  035354  004037  041630                  JSR     R0,RD.RP          ;FIND OUT WHY ATA=1
8262  035360  000014                          RPER1
8263  035362  035442                          8$
8264  035364  006126                          ROL     (SP)+             ;IS IT UNSAFE?
8265  035366  100004                          BPL     4$                ;BR IF NOT
8266  035370  112761  177777  034504          MOVB    #-1,DRVSTA(R1)    ;SET UNSAFE INDICATOR
8267  035376  000407                          BR      6$                ;EXIT
8268  035400  005105                  4$:     COM     R5                ;CHECK MOL, DPR, DRY, AND VV
8269  035402  042705  167077                  BIC     #^C<BIT12!BIT08!BIT07!BIT06>,R5
8270  035406  001003                          BNE     6$                ;BRANCH IF MOL, DPR, DRY, OR VV IS CLEAR
8271  035410  112761  000001  034504          MOVB    #1,DRVSTA(R1)     ;SET DRIVE STATUS TO ONLINE
8272  035416  005720                  6$:     TST     (R0)+             ;STEP OVER THE ERROR RETURN
8273  035420  000410                          BR      8$                ;EXIT
8274  035422  006301                  7$:     ASL     R1                ;CHANGE INDEX TO ADDRESS WORDS
8275  035424  012761  003720  034576          MOV     #2000.,TIMER(R1)  ;START 2 SEC TIMER
8276  035432  006201                          ASR     R1                ;RESTORE R1
8277  035434  105161  034524                  COMB    DPINT(R1)
8278  035440  005720                          TST     (R0)+
8279  035442  012605                  8$:     MOV     (SP)+,R5          ;RESTORE R5
8280  035444  000200                          RTS     R0                ;EXIT
8281
8282                                  ;TEST PROGRAMMABLE DRIVE FLAG (TSTPGM=1 WORD)
8283                                          ;THE FLAG WILL BE SET BY THE PROGRAM UNDER
8284                                          ;MANUFACTURING CONDITIONS (ACT,APT) AND
8285                                          ;CLEARED UNDER FIELD CONDITIONS(XXDP CHAIN,
8286                                          ;STANDALONE) WITH STARTING ADDRESS 200.
8287                                          ;THE FLAG WILL BE SET UNDER ALL CONDITIONS
8288                                          ;WITH STARTING ADDRESS 220.
8289
8290  035446  000000                  TSTPGM: .WORD   0         ;=1 TEST PROGRAMMABLE DRIVE
8291
```

```
                                        ;REQUEST PRE-PROCESSOR-HANDLES SUBSYSTEM REQUEST
                                        ;
                                        ;CALL
                                        ;
                                        ;       JSR     R0,@#RP04       ;CALL THE RP04/5/6 DRIVER
                                        ;       PNTADR                  ;ADDRESS OF POINTER OF DRIVES PARAMETER BLOCK
                                        ;       RETURN1                 ;RETURN HERE IF QUEUE IS FULL
                                        ;       RETURN2                 ;RETURN HERE IF REQUEST IS IN QUEUE OR THERE
                                                                        ;IS AN ERROR CONDITION

035450  013746  177776          RP04:   MOV     @#PS,-(SP)      ;SAVE THE CALLING STATUS
035454  013737  034636  177776          MOV     RPVEC+2,@#PS    ;DON'T ALLOW ANY RP04/5/6 INTERRUPTS
035462  112737  000001  034550          MOVB    #1,ACTDRV       ;SET "ACTIVE DRIVER" FLAG
035470  104412                        SAVREG                  ;SAVE R0 - R5
035472  011002                        MOV     (R0),R2         ;PICKUP THE DRIVE PARAMETER BLOCK POINTER
035474  005062  000016                CLR     16(R2)          ;CLEAR THE STATUS/ERROR INDICATOR
035500  111201                        MOVB    (R2),R1         ;PICKUP THE DRIVE NUMBER
035502  013704  034632                MOV     RPADR,R4        ;UNIBUS ADDRESS OF RPCS1
035506  105761  034504                TSTB    DRVSTA(R1)      ;CHECK DRIVES STATUS
035512  003014                        BGT     1$              ;BRANCH IF ONLINE
035514  105761  034552                TSTB    ULDFLG(R1)      ;UNLOAD COMMAND IN QUEUE?
035520  001036                        BNE     3$              ;BRANCH IF YES
035522  105761  034524                TSTB    DPINT(R1)       ;TRYING TO INIT THE DRIVE
035526  001042                        BNE     5$              ;BR IF YES
035530  004037  035062                JSR     R0,DRVINT       ;GO INIT. THE DRIVE
035534  000434                        BR      4$              ;ERROR RETURN
035536  105761  034504        1$:     TSTB    DRVSTA(R1)      ;IS DRIVE STATUS ONLINE?
035542  003445                        BLE     6$              ;BR IF NOT
035544  105761  034534                TSTB    DPRQS(R1)       ;OUTSTANDING PORT REQUEST FOR THE DRIVE ?
035550  001031                        BNE     5$              ;BR IF YES
035552  010164  000010                MOV     R1,RPCS2(R4)    ;SELECT THE DRIVE
035556  004037  042762                JSR     R0,DRVQUE       ;PUT THIS REQUEST IN QUEUE
035562  000460                        BR      9$              ;QUEUE IS FULL
035564  122762  000103  000002        CMPB    #103,2(R2)      ;IS THIS REQ. FOR AN UNLOAD?
035572  001003                        BNE     2$              ;BR IF NO
035574  112761  177777  034552        MOVB    #-1,ULDFLG(R1)  ;SET THE "UNLOAD IN QUEUE" FLAG
035602  105761  034474        2$:     TSTB    DRVACT(R1)      ;IS THIS DRIVE ACTIVE?
035606  001043                        BNE     8$              ;BR IF YES
035610  004737  035742                JSR     PC,OPT          ;CALL THE OPTIMIZER
035614  000440                        BR      8$
035616  012762  120000  000016  3$:   MOV     #BIT15!BIT13,16(R2) ;SET THE "UNLOAD IN QUEUE" ERROR FLAG
035624  000434                        BR      8$              ;EXIT
035626  004737  037052        4$:     JSR     PC,C17          ;GO HANDLE THE PARITY ERROR
035632  000431                        BR      8$
035634  004037  042762        5$:     JSR     R0,DRVQUE       ;PUT REQUEST IN QUEUE
035640  000431                        BR      9$              ;QUEUE IS FULL
035642  032714  000100                BIT     #BIT06,(R4)     ;IS 'IE' SET ALREADY ?
035646  001023                        BNE     8$              ;BR IF IT IS
035650  004737  042320                JSR     PC,SET.IE       ;SET INTERRUPT
035654  000420                        BR      8$              ;RETURN, REQUEST IN QUEUE
035656  105761  034504        6$:     TSTB    DRVSTA(R1)      ;SEE IF DRIVE OFFLINE OR UNSAFE
035662  002412                        BLT     7$              ;BR IF UNSAFE
035664  012762  140000  000016        MOV     #BIT15!BIT14,16(R2)  ;SET OFFLINE ERROR INDICATOR
035672  105761  034514                TSTB    DRVTYP(R1)      ;SEE IF OFFLINE OR NONEXISTENT
035676  001007                        BNE     8$              ;BR IF OFFLINE
035700  012762  100002  000016        MOV     #BIT15!BIT01,16(R2)  ;REPORT DRIVE NONEXISTENT
```

```
 8348  035706  000403                          BR      8$                  ;GO TO EXIT
 8349  035710  012762  110000  000016  7$:     MOV     #BIT15!BIT12,16(R2)  ;DRIVE IS UNSAFE
 8350  035716  104413                  8$:     RESREG                      ;RESTORE R0 - R5
 8351  035720  005720                          TST     (R0)+               ;SETUP FOR NORMAL RETURN
 8352  035722  000401                          BR      10$                 ;FINISH UP, THEN EXIT
 8353  035724  104413                  9$:     RESREG                      ;RESTORE R0 - R5
 8354  035726  005720                  10$:    TST     (R0)+               ;CORRECT THE RETURN ADDRESS
 8355  035730  105037  034550                  CLRB    ACTDRV              ;CLEAR "ACTIVE DRIVER" FLAG
 8356  035734  012637  177776                  MOV     (SP)+,@#PS          ;RETURN "PS" TO USER LEVEL
 8357  035740  000200                          RTS     R0                  ;RETURN TO CALLER
 8358
 8359                                  ;OPTIMIZER-CALLED FOR A PARTICULAR DRIVE
 8360                                  ;
 8361                                  ;CALL
 8362                                  ;       MOV     #DRVNUM,R1          ;DRIVE NUMBER TO R1
 8363                                  ;       JSR     PC,OPT              ;SETUP A COMMAND
 8364                                  ;
 8365  035742  104412                  OPT:    SAVREG                      ;SAVE R0 - R5
 8366  035744  013746  177776                  MOV     @#PS,-(SP)          ;SAVE PROC. STATUS
 8367  035750  146137  034620  034546          BICB    ATABIT(R1),SRCHWT       ;CLEAR "SEARCH WAIT" KEY
 8368  035756  004737  043036                  JSR     PC,GETREQ           ;GET "DPB" POINTER OF REQUEST
 8369  035762  005702                          TST     R2                  ;IS THERE A REQUEST IN QUEUE?
 8370  035764  001505                          BEQ     7$                  ;NO--BRANCH TO EXIT
 8371  035766  032764  004000  000000          BIT     #BIT11,RPCS1(R4)  ;IS DVA STILL SET ?
 8372  035774  001407                          BEQ     10$                 ;BR IF NOT
 8373  035776  032764  000100  000012          BIT     #BIT6,RPDS1(R4)  ;IS VV SET ?
 8374  036004  001003                          BNE     10$                 ;BR IF IT IS
 8375  036006  004037  035062          9$:     JSR     R0,DRVINT           ;SEE IF DRIVE STILL ONLINE ?
 8376  036012  000470                          BR      6$                  ;PARITY OR 'DVA' NOT SET
 8377  036014  105761  034504          10$:    TSTB    DRVSTA(R1)          ;IS DRIVE ONLINE?
 8378  036020  003014                          BGT     1$                  ;YES--BRANCH
 8379  036022  004737  043060                  JSR     PC,POPQUE           ;NO--REMOVE REQUEST FROM QUEUE
 8380  036026  012762  140000  000016          MOV     #BIT15!BIT14,16(R2)  ;SET OFFLINE STATUS/ERROR INDICATOR
 8381  036034  105761  034504                  TSTB    DRVSTA(R1)          ;IS DRIVE UNSAFE ?
 8382  036040  100064                          BPL     8$                  ;BR TO EXIT IF NOT
 8383  036042  012762  110000  000016          MOV     #BIT15!BIT12,16(R2)  ;SET UNSAFE STATUS/ERROR INDICATOR
 8384  036050  000460                          BR      8$                  ;BRANCH TO EXIT
 8385  036052  012746  000111          1$:     MOV     #111,-(SP)          ;LOAD COMMAND ONTO THE STACK
 8386  036056  004037  042010                  JSR     R0,WRT.RP           ;LOAD THE REGISTER
 8387  036062  000000                          RPCS1                       ;REGISTER INCREMENT
 8388  036064  036174                          6$                          ;ERROR RETURN ADDRESS
 8389  036066  032714  004000                  BIT     #BIT11,(R4)         ;DRIVE AVAILABLE ?
 8390  036072  001427                          BEQ     5$                  ;BR IF NOT
 8391  036074  122762  000150  000002          CMPB    #150,2(R2)          ;IS THE REQUEST FOR I/O?
 8392  036102  002403                          BLT     2$                  ;YES--BRANCH
 8393  036104  004737  036436                  JSR     PC,CI4              ;CALL THE COMMAND INITIATOR
 8394  036110  000440                          BR      8$                  ;BRANCH TO EXIT
 8395  036112  005737  034616          2$:     TST     DTUW                ;DATA TRANSFER UNDERWAY?
 8396  036116  002012                          BGE     4$                  ;YES--GO START A SEARCH
 8397  036120  005737  034574                  TST     SEEKFG              ;DO IMPLIED SEEKS?
 8398  036124  100404                          BMI     3$                  ;YES---BRANCH
 8399  036126  004037  037406                  JSR     R0,LA               ;NO--DO LOOK AHEAD
 8400  036132  000427                          BR      8$                  ;RETURN HERE ON A PARITY ERROR
 8401  036134  000403                          BR      4$                  ;GO START A SEARCH
 8402  036136  004737  036222          3$:     JSR     PC,CI1              ;START A DATA TRANSFER
 8403  036142  000423                          BR      8$
```

```
 8404  036144  004737  036330          4$:     JSR     PC,C13          ;START A SEARCH
 8405  036150  000420                  BR      8$              ;GO TO THE EXIT
 8406  036152  112761  177777  034534  5$:     MOVB    #-1,DPRQS(R1)   ;SET PORT REQUEST INDICATOR
 8407  036160  010103                  MOV     R1,R3           ;SET UP TO ADDRESS WORDS
 8408  036162  006303                  ASL     R3              ;CONVERT TO WORD INDEX
 8409  036164  012763  023420  034576          MOV     #10000.,TIMER(R3)  ;START 10 SEC TIMER
 8410  036172  000402                  BR      7$              ;EXIT
 8411  036174  004737  037052          6$:     JSR     PC,C17          ;PROCESS THE PARITY ERROR
 8412  036200  032714  000100          7$:     BIT     #BIT06,(R4)     ;SEE IF 'IE' ALREADY SET
 8413  036204  001002                  BNE     8$              ;BR IF SET
 8414  036206  004737  042320          JSR     PC,SET.IE       ;SET "IE" WITHOUT A "TRE"
 8415  036212  012637  177776          8$:     MOV     (SP)+,@#PS      ;RESTORE PROC. STATUS
 8416  036216  104413                  RESREG                  ;RESTORE R0 - R5
 8417  036220  000207                  RTS     PC
 8418
 8419                                  ;COMMAND INITIATOR
 8420                                  ;
 8421                                  ;CALL
 8422                                  ;       MOV     #DRVNUM,R1      ;DRIVE NUMBER
 8423                                  ;       MOV     #DPB,R2         ;ADDRESS OF DPB
 8424                                  ;       JSR     PC,C1?          ;C1?= C11,C13, OR C14
 8425                                  ;                               ;WHERE:
 8426                                  ;                               ;C11=DATA TRANSFER
 8427                                  ;                               ;C12=SEARCH REQUESTED BY DATA XFER
 8428                                  ;                               ;C14=NOT DATA TRANSFER
 8429
 8430  036222  004737  043060          C11:    JSR     PC,POPQUE       ;REMOVE REQUEST FROM "DRIVES WAIT" QUEUE
 8431  036226  010237  034544                  MOV     R2,TRNSWT       ;PUT REQ. IN TRANSFER WAIT QUEUE
 8432  036232  010203                  MOV     R2,R3           ;DPB ADDRESS TO R3
 8433  036234  013704  034632          MOV     RPADR,R4        ;RPCS1 ADDRESS
 8434  036240  010164  000010          MOV     R1,RPCS2(R4)    ;SELECT DRIVE
 8435  036244  062703  000004          ADD     #4,R3           ;DESIRED WORD COUNT
 8436  036250  062704  000002          ADD     #2,R4           ;RPWC ADDRESS
 8437  036254  012324                  MOV     (R3)+,(R4)+     ;LOAD WORD COUNT
 8438  036256  012324                  MOV     (R3)+,(R4)+     ;LOAD BUFFER ADDRESS
 8439  036260  012346                  MOV     (R3)+,-(SP)     ;LOAD SECTOR AND TRACK
 8440  036262  004037  042010          JSR     R0,WRT.RP       ;CALL THE LOAD(WRITE) ROUTINE
 8441  036266  000006                  RPDA                    ;INDEX OF REGISTER TO LOAD
 8442  036270  037052                  C17                     ;ERROR RETURN ADDRESS
 8443  036272  012346                  MOV     (R3)+,-(SP)     ;LOAD CYLINDER ADDRESS
 8444  036274  004037  042010          JSR     R0,WRT.RP
 8445  036300  000034                  RPCA
 8446  036302  037052                  C17
 8447  036304  016246  000002          MOV     2(R2),-(SP)     ;LOAD "COMMAND+GO", "A17&A16", AND "PSEL"
 8448  036310  004037  042010          JSR     R0,WRT.RP
 8449  036314  000000                  RPCS1
 8450  036316  037052                  C17
 8451  036320  010137  034616          MOV     R1,DTUW         ;SET "DATA TRANSFER UNDERWAY"
 8452  036324  000137  037014          JMP     C15
 8453  036330  013704  034632          C13:    MOV     RPADR,R4        ;RPCS1 ADDRESS
 8454  036334  010164  000010          MOV     R1,RPCS2(R4)    ;SELECT DRIVE
 8455  036340  016246  000012          MOV     12(R2),-(SP)    ;DESIRED CYLINDER ADDRESS
 8456  036344  004037  042010          JSR     R0,WRT.RP
 8457  036350  000034                  RPCA
 8458  036352  037052                  C17
 8459  036354  116203  000010          MOVB    10(R2),R3       ;PICKUP SECTOR ADDRESS
```

```
 8460  036360  163703  034646              SUB     MXWNDW,R3       ;BACKUP BY MAX. SEARCH FOR I/O WINDOW
 8461  036364  002002                      BGE     1$
 8462  036366  062703  000026              ADD     #22.,R3
 8463  036372  010346                1$:   MOV     R3,-(SP)        ;COMBINE THE ADJUSTED SECTOR WITH
 8464  036374  116266  000011  000001      MOVB    11(R2),1(SP)    ;THE DESIRED TRACK
 8465  036402  004037  042010              JSR     R0,WRT.RP       ;LOAD DESIRED TRACK & SECTOR
 8466  036406  000006                      RPDA
 8467  036410  037052                      C17
 8468  036412  012746  000131              MOV     #131,-(SP)      ;START A SEARCH
 8469  036416  004037  042010              JSR     R0,WRT.RP
 8470  036422  000000                      RPCS1
 8471  036424  037052                      C17
 8472  036426  156137  034620  034546      BISB    ATABIT(R1),SRCHWT       ;SET "SEARCH WAIT" KEY
 8473  036434  000567                      BR      C15
 8474  036436  013704  034632        C14:  MOV     RPADR,R4        ;RPCS1 ADDRESS
 8475  036442  010164  000010              MOV     R1,RPCS2(R4)    ;SELECT DRIVE
 8476  036446  116203  000002              MOVB    2(R2),R3        ;PICKUP THE REQUESTED COMMAND
 8477  036452  122703  000131              CMPB    #131,R3         ;IS IT A SEARCH COMMAND?
 8478  036456  001007                      BNE     1$              ;BRANCH IF NO
 8479  036460  016246  000010              MOV     10(R2),-(SP)    ;LOAD DESIRED TRACK & SECTOR
 8480  036464  004037  042010              JSR     R0,WRT.RP
 8481  036470  000006                      RPDA
 8482  036472  037052                      C17
 8483  036474  000403                      BR      2$              ;GO LOAD CYLINDER
 8484  036476  122703  000105        1$:   CMPB    #105,R3         ;IS IT A SEEK COMMAND
 8485  036502  001007                      BNE     3$              ;BRANCH IF NO
 8486  036504  016246  000012        2$:   MOV     12(R2),-(SP)    ;LOAD DESIRED CYLINDER
 8487  036510  004037  042010              JSR     R0,WRT.RP
 8488  036514  000034                      RPCA
 8489  036516  037052                      C17
 8490  036520  000546                      BR      C16
 8491  036522  122703  000115        3$:   CMPB    #115,R3         ;IS IT AN "OFFSET" COMMAND?
 8492  036526  001013                      BNE     4$              ;BR IF NO
 8493  036530  004037  041630              JSR     R0,RD.RP        ;MERGE THE OFFSET VALUE INTO RPOF
 8494  036534  000032                      RPOF                    ;BUT DON'T CHANGE THE UPPER
 8495  036536  037052                      C17
 8496  036540  116216  000001              MOVB    1(R2),(SP)      ;BYTE WHEN LOADING THE
 8497  036544  004037  042010              JSR     R0,WRT.RP       ;REGISTER (RPOF)
 8498  036550  000032                      RPOF
 8499  036552  037052                      C17
 8500  036554  000530                      BR      C16             ;GO START THE COMMAND
 8501  036556  122703  000107        4$:   CMPB    #107,R3         ;IS IT A "RECALIBRATE" COMMAND?
 8502  036562  001525                      BEQ     C16             ;BRANCH IF YES
 8503  036564  122703  000117              CMPB    #117,R3         ;IS IT A RETURN TO CENTER?
 8504  036570  001522                      BEQ     C16             ;BRANCH IF YES
 8505  036572  122703  000103              CMPB    #103,R3         ;IS IT AN "UNLOAD" COMMAND?
 8506  036576  001016                      BNE     5$              ;BRANCH IF NO
 8507  036600  112761  000001  034474      MOVB    #1,DRVACT(R1)   ;SET THE DRIVE ACTIVE INDICATOR
 8508  036606  105061  034504              CLRB    DRVSTA(R1)      ;PUT DRIVE STATUS TO OFFLINE
 8509  036612  112761  000001  034552      MOVB    #1,ULDFLG(R1)   ;SET "UNLOAD IN PROGRESS" FLAG
 8510  036620  010346                      MOV     R3,-(SP)        ;START THE "UNLOAD" COMMAND
 8511  036622  004037  042010              JSR     R0,WRT.RP
 8512  036626  000000                      RPCS1
 8513  036630  037052                      C17
 8514  036632  000207                      RTS     PC              ;RETURN TO USER
 8515  036634  122703  000143        5$:   CMPB    #143,R3         ;IS IT A "SET FORMAT" COMMAND?
```

```
8516  036640  001014                        BNE     6$              ;BRANCH IF NO
8517  036642  004037  041630                JSR     R0,RD.RP        ;READ THE OFFSET REGISTER
8518  036646  000032                        RPOF
8519  036650  037052                        C17
8520  036652  116266  000001  000001        MOVB    1(R2),1(SP)     ;COMBINE "FMT22","ECI", AND "HCI"
8521  036660  004037  042010                JSR     R0,WRT.RP       ;LOAD "FMT22", "ECI", AND/OR "HCI".
8522  036664  000032                        RPOF
8523  036666  037052                        C17
8524  036670  000436                        BR      12$
8525  036672  122703  000141          6$:   CMPB    #141,R3         ;IS IT A "GET REGISTER" COMMAND?
8526  036676  001023                        BNE     10$             ;BRANCH IF NO
8527  036700  016203  000006          7$:   MOV     6(R2),R3        ;POINTS TO 1ST ADDRESS OF WHERE
8528                                                                ;TO PUT THE REGISTER(S)
8529  036704  116237  000010  036722        MOVB    10(R2),9$       ;INIT. THE INDEX FOR THE FIRST REG.
8530  036712  116205  000011                MOVB    11(R2),R5       ;INDEX OF LAST REG. TO MOVE
8531  036716  004037  041630          8$:   JSR     R0,RD.RP        ;READ RP04/5/6 REGISTER
8532  036722  000000          9$:   RPCS1                           ;INDEX OF REG. TO READ
8533  036724  037052                        C17
8534  036726  012623                        MOV     (SP)+,(R3)+     ;GET THE CONTENTS OF RH11/RP04/5/6 REG.
8535  036730  023705  036722                CMP     9$,R5           ;LAST REG. BEEN READ?
8536  036734  001414                        BEQ     12$             ;GET OUT IF YES
8537  036736  062737  000002  036722        ADD     #2,9$           ;INCREASE THE INDEX BY 2
8538  036744  000764                        BR      8$              ;LOOP--MORE TO READ
8539  036746  122703  000145         10$:   CMPB    #145,R3         ;IS IT A "SELECT DRIVE" COMMAND?
8540  036752  001405                        BEQ     12$             ;BRANCH IF YES
8541  036754  010346          11$:  MOV     R3,-(SP)        ;LOAD THE COMMAND
8542  036756  004037  042010                JSR     R0,WRT.RP
8543  036762  000000                        RPCS1
8544  036764  037052                        C17
8545  036766  004737  043060         12$:   JSR     PC,POPQUE       ;REMOVE REQ. FROM QUEUE
8546  036772  052762  000200  000016        BIS     #BIT07,16(R2)   ;SET THE "DONE" BIT
8547  037000  005737  034572                TST     SAVEFG          ;SAVE THE RH11/RP04/5/6 REGISTERS?
8548  037004  100002                        BPL     13$             ;BRANCH IF NO
8549  037006  004737  042202                JSR     PC,SVRH11       ;YES--GO SAVE THE REGISTERS
8550  037012  000207          13$:  RTS     PC              ;RETURN TO USER
8551  037014  006301          C15:  ASL     R1
8552  037016  012761  001750  034576        MOV     #1000.,TIMER(R1)        ;SET A ONE SECOND TIMER
8553  037024  006201                        ASR     R1
8554  037026  112761  000001  034474        MOVB    #1,DRVACT(R1)   ;SET THE DRIVE ACTIVE
8555  037034  000207                        RTS     PC              ;RETURN TO THE USER
8556  037036  010346          C16:  MOV     R3,-(SP)        ;LOAD THE COMMAND
8557  037040  004037  042010                JSR     R0,WRT.RP
8558  037044  000000                        RPCS1
8559  037046  037052                        C17
8560  037050  000761                        BR      C15
8561  037052  032764  010000  000010  C17:   BIT     #BIT12,RPCS2(R4)  ;DRIVE NON-EXISTENT ?
8562  037060  001034                        BNE     C18             ;BR IF YES
8563  037062  005702                1$:     TST     R2              ;ANYTHING IN QUEUE ?
8564  037064  001405                        BEQ     C17B            ;BR IF NOT
8565  037066  012762  104000  000016        MOV     #BIT15!BIT11,16(R2)     ;SET "PARITY" ERROR INDICATOR
8566  037074  004737  042202                JSR     PC,SVRH11       ;GO SAVE THE RH11/RP04/5/6 REGISTERS
8567  037100  012746  000111          C17B:  MOV     #111,-(SP)      ;DO A "DRIVE CLEAR"
8568  037104  004037  042010                JSR     R0,WRT.RP
8569  037110  000000                        RPCS1
8570  037112  037152                        C18
8571  037114  004737  042742                JSR     PC,EMPTYQ       ;EMPTY THE QUEUE
```

```
 8572  037120  105061  034552                CLRB    ULDFLG(R1)        ;CLEAR THE UNLOAD IN QUEUE FLAG
 8573  037124  105061  034474                CLRB    DRVACT(R1)        ;DRIVE IS IDLE
 8574  037130  020137  034616                CMP     R1,DTUW           ;IF THIS DRIVE HAD AN I/O REQUEST
 8575  037134  001005                        BNE     1$                ;IN PROGRESS CLEAR ALL OF THE FLAGS
 8576  037136  005037  034544                CLR     TRNSWT
 8577  037142  012737  177777  034616        MOV     #-1,DTUW
 8578  037150  000207                1$:     RTS     PC
 8579  037152  104412                C18:    SAVREG                    ;SAVE R0 - R5
 8580  037154  032764  010000  000010        BIT     #BIT12,RPCS2(R4)  ;IS 'NED' SET ?
 8581  037162  001002                        BNE     1$                ;BR IF YES
 8582  037164  005001                        CLR     R1
 8583  037166  005003                        CLR     R3
 8584  037170  105761  034474          1$:   TSTB    DRVACT(R1)        ;DRIVE ACTIVE?
 8585  037174  001443                        BEQ     5$                ;BRANCH IF NO
 8586  037176  013702  034544                MOV     TRNSWT,R2         ;GET THE "TRANSFER WAIT" QUEUE
 8587  037202  020137  034616                CMP     R1,DTUW           ;DID THIS DRIVE HAVE AN I/O IN PROGRESS?
 8588  037206  001402                        BEQ     2$                ;BRANCH IF YES
 8589  037210  004737  043036                JSR     PC,GETREQ         ;GET THE DPB POINTER
 8590  037214  005702                2$:     TST     R2                ;QUEUE ENTRY FOR DRIVE ?
 8591  037216  001415                        BEQ     4$                ;BR IF NOT
 8592  037220  032764  010000  000010        BIT     #BIT12,RPCS2(R4)  ;'NED' SET ?
 8593  037226  001404                        BEQ     3$                ;BR IF NOT
 8594  037230  012762  100002  000016        MOV     #BIT15!BIT01,16(R2)  ;SET 'DRIVE NON-EXISTENT' INDICATOR
 8595  037236  000405                        BR      4$                ;CONTINUE
 8596  037240  012762  102000  000016  3$:   MOV     #BIT15!BIT10,16(R2) ;SET "NON-CLEARABLE PARITY" ERROR INDICATOR
 8597  037246  004737  042202                JSR     PC,SVRH11         ;SAVE RH11/RP04/5/6 REGISTERS
 8598  037252  012763  177777  034576  4$:   MOV     #-1,TIMER(R3)     ;STOP THE TIMER
 8599  037260  105061  034474                CLRB    DRVACT(R1)        ;SET "DRIVE ACTIVE" TO IDLE
 8600  037264  020137  034616                CMP     R1,DTUW           ;IS THIS DRIVE SETUP FOR A TRANSFER
 8601  037270  001005                        BNE     5$                ;BR IF NOT
 8602  037272  012737  177777  034616        MOV     #-1,DTUW          ;RESET THE INDICATOR
 8603  037300  005037  034544                CLR     TRNSWT            ;CLEAR THE TRANSFER QUEUE
 8604  037304  105061  034552          5$:   CLRB    ULDFLG(R1)        ;CLEAR UNLOAD FLAG
 8605  037310  032764  010000  000010        BIT     #BIT12,RPCS2(R4)  ;'NED' SET ?
 8606  037316  001021                        BNE     6$                ;BR IF YES
 8607  037320  005201                        INC     R1                ;MOVE TO THE NEXT DRIVE
 8608  037322  062703  000002                ADD     #2,R3
 8609  037326  042701  177770                BIC     #^C7,R1
 8610  037332  001316                        BNE     1$                ;BRANCH IF MORE DRIVES
 8611  037334  012737  177777  034616        MOV     #-1,DTUW          ;NO DATA TRANSFERS UNDERWAY
 8612  037342  005037  034544                CLR     TRNSWT            ;CLEAR THE 'TRANSFER WAIT' QUEUE
 8613  037346  004737  042664                JSR     PC,CLRQUE         ;CLEAR ALL OF THE REQUEST QUEUES
 8614  037352  012764  000040  000010        MOV     #BIT05,RPCS2(R4)  ;DO A MASSBUS INIT.
 8615  037360  000406                        BR      7$                ;CONTINUE
 8616  037362  004737  042742          6$:   JSR     PC,EMPTYQ         ;CLEAR THE DRIVE'S QUEUE
 8617  037366  105061  034504                CLRB    DRVSTA(R1)        ;SET DRIVE TO OFFLINE
 8618  037372  105061  034514                CLRB    DRVTYP(R1)        ;CLEAR THE DRIVE TYPE INDICATOR
 8619  037376  004737  042320          7$:   JSR     PC,SET.IE         ;SET "IE" WITHOUT "TRE"
 8620  037402  104413                        RESREG                    ;RESTORE R0 - R5
 8621  037404  000207                        RTS     PC                ;RETURN
 8622
 8623                                ;LOOK AHEAD ROUTINE
 8624                                ;
 8625                                ;CALL
 8626                                ;       MOV     #DRVNUM,R1        ;DRIVE NUMBER
 8627                                ;       MOV     #DPB,R2           ;POINT TO DPB
```

```
8628                                    ;       JSR     R0,LA           ;GO CHECK THE WINDOW
8629                                    ;       RETURN1                 ;ERROR RETURN
8630                                    ;       RETURN2                 ;START A SEARCH
8631                                    ;       RETURN3                 ;START A DATA TRANSFER
8632
8633  037406  013704  034632            LA:     MOV     RPADR,R4        ;GET RPCS1'S ADDRESS
8634  037412  010164  000010                    MOV     R1,RPCS2(R4)    ;SELECT DRIVE
8635  037416  004037  041630                    JSR     R0,RD.RP        ;READ CURRENT CYLINDER
8636  037422  000036                            RPCC
8637  037424  037536                            4$                      ;ERROR RETURN ADDRESS
8638  037426  022662  000012                    CMP     (SP)+,12(R2)    ;IS CURRENT CYLINDER=DESIRED
8639                                                                    ;CYLINDER?
8640  037432  001037                            BNE     3$              ;EXIT IF NO
8641  037434  105261  034562                    INCB    LACNT(R1)       ;INCREMENT THE LOOK AHEAD COUNT
8642  037440  126137  034562  034640           CMPB    LACNT(R1),MXLACT ;EXCEED MAX?
8643  037446  003026                            BGT     2$              ;BRANCH IF YES
8644  037450  116203  000010                    MOVB    10(R2),R3       ;GET DESIRED SECTOR ADDRESS AND
8645  037454  000303                            SWAB    R3              ;MULT. BY 64--ALIGN WITH
8646  037456  006203                            ASR     R3              ;LOOK AHEAD REGISTER
8647  037460  006203                            ASR     R3
8648  037462  012737  000340  177776           MOV     #340,@#PS       ;PRIORITY LEVEL "7"
8649  037470  004037  041630                    JSR     R0,RD.RP        ;READ LOOK AHEAD REGISTER
8650  037474  000020                            RPLA
8651  037476  037536                            4$
8652  037500  162603                            SUB     (SP)+,R3        ;CALCULATE THE DELTA
8653  037502  002002                            BGE     1$
8654  037504  062703  002600                    ADD     #<22.*64.>,R3   ;MAKE THE DELTA POSITIVE
8655  037510  023703  034642            1$:     CMP     MXDLTA,R3       ;CHECK THE DELTA TO SEE
8656  037514  002406                            BLT     3$              ;IF IT IS WITHIN THE
8657  037516  023703  034644                    CMP     MNDLTA,R3       ;WINDOW---IF YES, ZERO
8658  037522  002003                            BGE     3$              ;THE LOOK AHEAD COUNT
8659  037524  105061  034562            2$:     CLRB    LACNT(R1)       ;AND TAKE THE I/O EXIT
8660  037530  005720                            TST     (R0)+
8661  037532  005720                    3$:     TST     (R0)+           ;ADJUST THE RETURN ADDRESS
8662  037534  000402                            BR      5$              ;EXIT
8663  037536  004737  037052            4$:     JSR     PC,CI7          ;PROCESS THE ERROR
8664  037542  000200                    5$:     RTS     R0              ;RETURN
8665
8666                                    ;INTERRUPT SERVICE ROUTINE
8667
8668  037544  112737  000001  034550    ISR:    MOVB    #1,ACTDRV       ;SET "ACTIVE DRIVER" FLAG
8669  037552  104412                            SAVREG                  ;SAVE R0 - R5
8670  037554  013704  034632                    MOV     RPADR,R4        ;ADDRESS OF RHSCS1
8671  037560  013701  034616                    MOV     DTUW,R1         ;GET "DATA TRANSFER UNDERWAY" INDICATOR
8672  037564  002403                            BLT     1$              ;BRANCH IF NO DATA TRANSFER UNDERWAY
8673  037566  004737  037610                    JSR     PC,TD           ;CALL TRANSFER DONE
8674  037572  000402                            BR      2$              ;EXIT
8675  037574  004737  037750            1$:     JSR     PC,SC           ;CALL SPECIAL CONDITIONS
8676  037600  104413                    2$:     RESREG                  ;RESTORE R0 - R5
8677  037602  105037  034550                    CLRB    ACTDRV          ;CLEAR "ACTIVE DRIVER" FLAG
8678  037606  000002                            RTI                     ;RETURN
8679
8680                                    ;TRANSFER DONE ROUTINE
8681
8682  037610  105061  034474            TD:     CLRB    DRVACT(R1)      ;SET DRIVE ACTIVE INDICATOR TO IDLE
8683  037614  012737  177777  034616            MOV     #-1,DTUW        ;NO DATA TRANSFERS UNDERWAY
```

```
8684  037622  006301                        ASL     R1
8685  037624  012761  177777  034576        MOV     #-1,TIMER(R1)   ;CANCEL TIMEOUT
8686  037632  006201                        ASR     R1
8687  037634  013702  034544                MOV     TRNSWT,R2       ;GET "DPB" ADDRESS FROM THE
8688  037640  005037  034544                CLR     TRNSWT          ;TRANSFER WAIT QUEUE--CLEAR QUEUE
8689  037644  052762  000200  000016        BIS     #BIT07,16(R2)   ;SET DONE
8690  037652  010164  000010                MOV     R1,RPCS2(R4)    ;SELECT THE DRIVE
8691  037656  004037  041630                JSR     R0,RD.RP        ;TRANSFER ERROR(TRE=1)?
8692  037662  000000                        RPCS1
8693  037664  037052                        C17
8694  037666  006126                        ROL     (SP)+
8695  037670  100413                        BMI     3$              ;BR IF YES
8696  037672  005737  034572                TST     SAVEFG          ;SAVE THE RH11/RP04/5/6 REGISTERS?
8697  037676  100002                        BPL     1$              ;BRANCH IF NO
8698  037700  004737  042202                JSR     PC,SVRH11       ;YES--SAVE THE REGISTERS
8699  037704  004737  035742        1$:     JSR     PC,OPT          ;CALL OPTIMIZER
8700  037710  000417                        BR      SC              ;CHECK OTHER DRIVES
8701  037712  012714  000113        2$:     MOV     #113,(R4)       ;RELEASE THE DRIVE
8702  037716  000414                        BR      SC              ;CHECK FOR OTHER DRIVES
8703  037720  052762  100100  000016  3$:   BIS     #BIT15!BIT06,16(R2) ;SET DATA ERROR FLAG
8704  037726  004737  042742                JSR     PC,EMPTYQ       ;EMPTY THE "DRIVE'S WAIT" QUEUE
8705  037732  004737  042202                JSR     PC,SVRH11       ;SAVE THE RH11/RP04/5/6 REGISTERS
8706  037736  012714  040111                MOV     #40111,(R4)     ;ISSUE A "DRIVE CLEAR"
8707  037742  012714  000113                MOV     #113,(R4)       ;ISSUE A RELEASE TO THE DRIVE
8708  037746  000400                        BR      SC              ;CHECK FOR OTHER DRIVES
8709
8710                                  ;SPECIAL CONDITION ROUTINE
8711
8712  037750  116403  000016        SC:     MOVB    RPAS(R4),R3     ;READ "RPAS"
8713  037754  001014                        BNE     2$              ;BRANCH IF ANY 'ATA' BITS SET
8714  037756  004037  041630                JSR     R0,RD.RP        ;READ CONTROL AND STATUS REGISTER
8715  037762  000000                        RPCS1
8716  037764  037152                        C18
8717  037766  106126                        ROLB    (SP)+           ;IS "IE"=1?
8718  037770  100405                        BMI     1$              ;YES, NO DRIVES TO CHECK
8719  037772  004037  043122                JSR     R0,ES.SAV       ;SAVE THE ADDRESS IN '$ESCAPE'
8720  037776  104001                        ERROR   1               ;REPORT AN ILLEGAL INTERRUPT
8721  040000  004737  042320                JSR     PC,SET.IE       ;SET INTERRUPT ENABLE
8722  040004  000207                1$:     RTS     PC              ;RETURN
8723  040006  005046                2$:     CLR     -(SP)           ;PROCESS ALL DRIVES THAT HAVE
8724  040010  110316                        MOVB    R3,(SP)         ;AN "ATA"=1
8725  040012  012703  000001                MOV     #1,R3
8726  040016  005001                        CLR     R1
8727  040020  030316                SC3:    BIT     R3,(SP)         ;ATA=1?
8728  040022  001005                        BNE     SC5             ;YES--BRANCH
8729  040024  005201                SC4:    INC     R1              ;MOVE TO THE NEXT DRIVE
8730  040026  106303                        ASLB    R3
8731  040030  001373                        BNE     SC3             ;BRANCH IF MORE TO CHECK?
8732  040032  005726                        TST     (SP)+           ;CLEAN OFF THE STACK
8733  040034  000207                        RTS     PC              ;RETURN TO USER
8734  040036  105761  034524        SC5:    TSTB    DPINT(R1)       ;INITIALIZING THE DRIVE ?
8735  040042  001402                        BEQ     1$              ;BR IF NOT
8736  040044  000137  040742                JMP     SC13            ;PROCESS THE DRIVE
8737  040050  105761  034534        1$:     TSTB    DPRQS(R1)       ;PORT REQUEST OUTSTANDING ?
8738  040054  001402                        BEQ     2$              ;BR IF NOT
8739  040056  000137  040742                JMP     SC13            ;START THE OUTSTANDING COMMAND
```

```
 8740  040062  105761  034504          2$:     TSTB    DRVSTA(R1)      ;CHECK THE DRIVE STATUS
 8741  040066  003025                          BGT     5$              ;BRANCH IF ONLINE
 8742  040070  105761  034552                  TSTB    ULDFLG(R1)      ;UNLOAD IN PROGRESS?
 8743  040074  003422                          BLE     5$              ;BRANCH IF NOT
 8744  040076  004737  043036                  JSR     PC,GETREQ       ;GET DPB POINTER
 8745  040102  004737  042202                  JSR     PC,SVRH11       ;SAVE THE RH11/RP04/5/6 REGISTERS
 8746  040106  004737  040672                  JSR     PC,SC12         ;SAVE RPDS1, RPER1, RPER2, AND RPER3
 8747                                                                  ;ALSO DO A DRIVE INIT (DRVINT)
 8748  040112  105761  034504                  TSTB    DRVSTA(R1)      ;DID DRIVE COME ONLINE?
 8749  040116  003416                          BLE     6$              ;NO---BRANCH
 8750  040120  032737  040000  034464          BIT     #BIT14,RPERRS   ;WAS THERE AN ERROR?
 8751  040126  001002                          BNE     3$              ;BR IF ERROR
 8752  040130  000137  040602                  JMP     SC11            ;NO ERROR
 8753  040134  013705  034466          3$:     MOV     RPERRS+2,R5     ;YES -- PICKUP RPER1 AND
 8754  040140  000502                          BR      SC6A            ;GO PROCESS THE ERROR
 8755  040142  105761  034474          5$:     TSTB    DRVACT(R1)      ;DRIVE ACTIVE WITH COMMAND OR ERROR RECOVERY ?
 8756  040146  001033                          BNE     SC6             ;BR IF EITHER
 8757  040150  004737  040672                  JSR     PC,SC12         ;SAVE RPDS1, RPER1, RPER2, AND RPER3
 8758                                                                  ;ALSO DO A DRVINT
 8759  040154  105761  034524          6$:     TSTB    DPINT(R1)       ;TRYING TO INIT THE DRIVE ?
 8760  040160  001321                          BNE     SC4             ;BR IF YES, CHECK ON MORE DRIVES
 8761  040162  105761  034504                  TSTB    DRVSTA(R1)      ;CHECK ON DRIVE'S STATUS
 8762  040166  100412                          BMI     7$              ;BR IF UNSAFE
 8763  040170  032737  020000  034472          BIT     #BIT13,RPERRS+6 ;ADDRESS PLUG CHANGED ?
 8764  040176  001013                          BNE     8$              ;BR IF YES
 8765  040200  012746  000113                  MOV     #113,-(SP)      ;RELEASE COMMAND
 8766  040204  004037  042010                  JSR     R0,WRT.RP       ;WRITE THE COMMAND INTO RPCS1
 8767  040210  000000                          RPCS1                   ;REGISTER INDEX
 8768  040212  040552                          SC8                     ;PARITY EXIT ADDRESS
 8769  040214  011605          7$:     MOV     (SP),R5         ;PICKUP (RPAS) BEFORE THE ERROR CALL
 8770  040216  004037  043122                  JSR     R0,ES.SAV       ;SAVE THE ADDRESS IN '$ESCAPE'
 8771  040222  104002                          ERROR   2               ;REPORT THE UNEXPECTED ATTENTION
 8772  040224  000677                          BR      SC4             ;GO CHECK FOR MORE ATA'S
 8773  040226                          8$:
 8774  040226  004037  043122                  JSR     R0,ES.SAV       ;SAVE THE ADDRESS IN '$ESCAPE'
 8775  040232  104005                          ERROR   5               ;REPORT THE ADDRESS PLUG CHANGE
 8776  040234  000673                          BR      SC4             ;CHECK FOR MORE DRIVES
 8777  040236  006301          SC6:    ASL     R1              ;SETUP TO ADDRESS WORDS
 8778  040240  012761  177777  034576          MOV     #-1,TIMER(R1)   ;STOP THE TIMER
 8779  040246  006201                          ASR     R1              ;RESTORE THE DRIVE ADDRESS
 8780  040250  004737  043036                  JSR     PC,GETREQ       ;GET THE DPB POINTER FROM THE QUEUE
 8781  040254  010164  000010                  MOV     R1,RPCS2(R4)    ;SELECT DRIVE
 8782  040260  004037  041630                  JSR     R0,RD.RP        ;READ THE RP04'S STATUS REG.
 8783  040264  000012                          RPDS1
 8784  040266  040552                          SC8
 8785  040270  011605                          MOV     (SP),R5         ;AND PUT IT IN R5
 8786  040272  006126                          ROL     (SP)+           ;WAS THERE AN ERROR?
 8787  040274  100407                          BMI     1$              ;BR IF ERROR
 8788  040276  105761  034474                  TSTB    DRVACT(R1)      ;CHECK DRIVE'S STATE
 8789  040302  003137                          BGT     SC11            ;BR IF DRIVE ACTIVE WITH ORDER
 8790  040304  052762  100210  000016          BIS     #BIT15!BIT07!BIT03,16(R2)  ;INFORM USER OF ERROR RECOVER COMPLETION
 8791  040312  000470                          BR      SC7
 8792  040314  004037  041630          1$:     JSR     R0,RD.RP        ;READ ERROR REGISTER #1
 8793  040320  000014                          RPER1
 8794  040322  040552                          SC8
 8795  040324  012605                          MOV     (SP)+,R5        ;AND SAVE IT IN R5
```

```
 8796  040326  004737  042202                  JSR     PC,SVRH11       ;SAVE RH11/RP04/5/6 REGISTERS
 8797  040332  012746  000111                  MOV     #111,-(SP)      ;ISSUE A DRIVE CLEAR
 8798  040336  004037  042010                  JSR     R0,WRT.RP
 8799  040342  000000                          RPCS1
 8800  040344  040552                          SC8
 8801  040346  006105                  SC6A:   ROL     R5              ;WAS "UNSAFE" CONDITION =1?
 8802  040350  100406                          BMI     1$              ;BRANCH IF YES
 8803  040352  005702                          TST     R2              ;ANYTHING IN QUEUE ?
 8804  040354  001447                          BEQ     SC7             ;BR IF NOT
 8805  040356  052762  100240  000016          BIS     #BIT15!BIT07!BIT05,16(R2)       ;INFORM USER OF ERROR
 8806  040364  000443                          BR      SC7
 8807  040366  004037  041630          1$:     JSR     R0,RD.RP        ;READ DRIVE STATUS REG. #1
 8808  040372  000012                          RPDS1
 8809  040374  040552                          SC8
 8810  040376  011605                          MOV     (SP),R5         ;SAVE RPDS1 IN R5
 8811  040400  006126                          ROL     (SP)+           ;"ERR"=1?
 8812  040402  100011                          BPL     2$              ;BR IF NO--UNSAFE CLEARED
 8813  040404  112761  177777  034504          MOVB    #-1,DRVSTA(R1)  ;DRIVE IS UNSAFE
 8814  040412  004737  042202                  JSR     PC,SVRH11       ;SAVE RH11/RP04/5/6 REGISTERS
 8815  040416  052762  110000  000016          BIS     #BIT15!BIT12,16(R2)     ;INFORM USER OF UNSAFE ERROR
 8816  040424  000423                          BR      SC7
 8817  040426  032705  010000          2$:     BIT     #BIT12,R5       ;"MOL" = 1 ?
 8818  040432  001015                          BNE     3$              ;BR IF YES
 8819  040434  112761  177777  034474          MOVB    #-1,DRVACT(R1)  ;ACTIVE ERROR RECOVER
 8820  040442  112761  000001  034504          MOVB    #1,DRVSTA(R1)   ;ONLINE
 8821  040450  006301                          ASL     R1
 8822  040452  012761  072460  034576          MOV     #30000.,TIMER(R1) ;START 30 SECOND TIMER
 8823  040460  006201                          ASR     R1
 8824  040462  000137  040024                  JMP     SC4
 8825  040466  052762  100220  000016  3$:     BIS     #BIT15!BIT07!BIT04,16(R2)       ;INFORM USER OF ERROR
 8826  040474  105061  034474          SC7:    CLRB    DRVACT(R1)      ;DRIVE IS IDLE
 8827  040500  004737  042742                  JSR     PC,EMPTYQ       ;DUMP THE QUEUE
 8828  040504  105761  034552                  TSTB    ULDFLG(R1)      ;UNLOAD IN PROGRESS OR QUEUE?
 8829  040510  003002                          BGT     1$              ;BR IF NOT
 8830  040512  105061  034552                  CLRB    ULDFLG(R1)      ;CLEAR UNLOAD FLAG
 8831  040516  116164  034620  000016  1$:     MOVB    ATABIT(R1),RPAS(R4)  ;CLEAR ATTENTION BIT
 8832  040524  105761  034504                  TSTB    DRVSTA(R1)      ;IS THE DRIVE UNSAFE ?
 8833  040530  100406                          BMI     2$              ;BR IF IT IS
 8834  040532  012746  000113                  MOV     #113,-(SP)      ;RELEASE COMMAND
 8835  040536  004037  042010                  JSR     R0,WRT.RP       ;WRITE THE COMMAND INTO RPCS1
 8836  040542  000000                          RPCS1                   ;REGISTER INDEX
 8837  040544  040552                          SC8                     ;PARITY EXIT ADDRESS
 8838  040546  000137  040024          2$:     JMP     SC4             ;CHECK FOR MORE DRIVES
 8839  040552  105761  034474          SC8:    TSTB    DRVACT(R1)      ;IS DRIVE IDLE?
 8840  040556  001405                          BEQ     1$              ;YES--BRANCH
 8841  040560  004737  043036                  JSR     PC,GETREQ       ;GET DPB POINTER
 8842  040564  004737  037052                  JSR     PC,C17          ;PROCESS THE PARITY ERROR
 8843  040570  000402                          BR      2$              ;CONTINUE
 8844  040572  004737  037100          1$:     JSR     PC,C17B         ;PROCESS THE UNCORRECTABLE PARITY ERROR
 8845  040576  000137  040024          2$:     JMP     SC4             ;CHECK MORE DRIVES
 8846  040602  105761  034552          SC11:   TSTB    ULDFLG(R1)      ;"UNLOAD IN PROGRESS"?
 8847  040606  003402                          BLE     1$              ;BRANCH IF NO
 8848  040610  105061  034552                  CLRB    ULDFLG(R1)      ;CLEAR UNLOAD FLAG
 8849  040614  105061  034474          1$:     CLRB    DRVACT(R1)      ;SET DRIVE IDLE
 8850  040620  136137  034620  034546          BITB    ATABIT(R1),SRCHWT  ;DOING A SEARCH OPERATION FOR
 8851                                                                  ;AN I/O COMMAND?
```

```
8852  040626  001012                        BNE     2$              ;BRANCH IF YES
8853  040630  004737  043060                JSR     PC,POPQUE       ;REMOVE REQUEST FROM QUEUE
8854  040634  052762  000200  000016        BIS     #BIT07,16(R2)   ;SET "DONE" BIT
8855  040642  005737  034572                TST     SAVEFG          ;SAVE THE REGISTERS?
8856  040646  100002                        BPL     2$              ;BRANCH IF NO
8857  040650  004737  042202                JSR     PC,SVRH11       ;YES--SAVE ALL OF THE RH11/RP04/5/6 REG'S
8858  040654  116164  034620  000016  2$:   MOVB    ATABIT(R1),RPAS(R4)  ;CLEAR ATTENTION BIT
8859  040662  004737  035742                JSR     PC,OPT          ;START A REQUEST
8860  040666  000137  040024                JMP     SC4             ;CHECK FOR MORE DRIVES
8861  040672  010164  000010        SC12:   MOV     R1,RPCS2(R4)    ;SELECT DRIVE
8862  040676  016437  000012  034464        MOV     RPDS1(R4),RPERRS        ;SAVE THE FOUR REGISTERS THAT
8863  040704  016437  000014  034466        MOV     RPER1(R4),RPERRS+2      ;WILL TELL US SOMETHING
8864  040712  016437  000040  034470        MOV     RPER2(R4),RPERRS+4
8865
8866  040720  016437  000042  034472        MOV     RPER3(R4),RPERRS+6
8867  040726  004037  035062                JSR     R0,DRVINT       ;INIT. THE STATE OF THE DRIVE
8868  040732  000401                        BR      1$              ;TAKE ERROR EXIT
8869  040734  000207                        RTS     PC              ;RETURN
8870  040736  005726                1$:     TST     (SP)+           ;POP PC OFF OF THE STACK
8871  040740  000704                        BR      SC8             ;PROCESS THE PARITY ERROR
8872  040742  006301                SC13:   ASL     R1              ;SETUP TO ADDRESS WORDS
8873  040744  012761  177777  034576        MOV     #-1,TIMER(R1)   ;STOP THE TIMER
8874  040752  006201                        ASR     R1              ;
8875  040754  010164  000010                MOV     R1,RPCS2(R4)    ;SELECT THE DRIVE
8876  040760  116164  034620  000016        MOVB    ATABIT(R1),RPAS(R4)  ;CLEAR THE ATTENTION BIT
8877  040766  032714  004000                BIT     #BIT11,(R4)     ;DRIVE AVAILABLE ?
8878  040772  001006                        BNE     1$              ;BR IF AVAILABLE
8879  040774  006301                        ASL     R1
8880  040776  012761  023420  034576        MOV     #10000.,TIMER(R1)  ;START 10 SEC TIMER AGAIN
8881
8882  041004  006201                        ASR     R1
8883  041006  000433                        BR      3$              ;EXIT
8884  041010  105761  034524          1$:   TSTB    DPINT(R1)       ;INITIALIZING THE DRIVE ?
8885  041014  001424                        BEQ     2$              ;BR IF NOT
8886  041016  105061  034524                CLRB    DPINT(R1)       ;CLEAR THE INIT INDICATOR
8887  041022  004037  035062                JSR     R0,DRVINT       ;GO INIT THE DRIVE
8888  041026  000240                        NOP                     ;DUMMY PARITY ERROR RETURN
8889  041030  105761  034504                TSTB    DRVSTA(R1)      ;DRIVE ONLINE ?
8890  041034  003014                        BGT     2$              ;BR IF YES -- START ORDER
8891  041036  005702                        TST     R2              ;QUEUE ENTRY FOR THE DRIVE
8892  041040  001416                        BEQ     3$              ;BR IF NOT
8893  041042  004737  043036                JSR     PC,GETREQ       ;GET DPB ADDRESS
8894  041046  052762  140000  000016        BIS     #BIT15!BIT14,16(R2)  ;INFORM USER THAT DRIVE OFFLINE
8895  041054  004737  042202                JSR     PC,SVRH11       ;SAVE THE REGISTERS
8896  041060  004737  042742                JSR     PC,EMPTYQ       ;EMPTY THE REQUEST QUEUE
8897  041064  000404                        BR      3$
8898  041066  105061  034534          2$:   CLRB    DPRQS(R1)       ;CLEAR THE PORT REQUEST INDICATOR
8899  041072  004737  035742                JSR     PC,OPT          ;START THE PENDING REQUEST
8900  041076  000137  040024          3$:   JMP     SC4             ;PROCESS OTHER DRIVES
8901
8902                                  ;RP04/5/6 TIMER ROUTINE
8903                                  ;CALL
8904                                  ;       MOV     #TIME,-(SP)     ;ELASPED TIME IN MILLISECONDS ON THE STACK
8905                                  ;       JSR     PC,RPTMR        ;CALL RP04/5/6 TIME ROUTINE
8906
8907  041102  005737  034550          RPTMR:  TST     ACTDRV          ;CHECK "ACTDRV & ACTSTR"
```

```
8908  041106  001030                        BNE     4$                ;IF NON ZERO EXIT
8909  041110  112737  000001  034551        MOVB    #1,ACTSTR         ;SET "ACTSTR"
8910  041116  104412                        SAVREG                    ;SAVE R0 - R5
8911  041120  005001                        CLR     R1                ;START WITH DRIVE 0
8912  041122  005003                        CLR     R3
8913  041124  005763  034576        1$:     TST     TIMER(R3)         ;IS THE TIMER RUNNING?
8914  041130  002407                        BLT     2$                ;BRANCH IF NO
8915  041132  166663  000002  034576        SUB     2(SP),TIMER(R3)   ;COUNT THE INTERVAL
8916  041140  003003                        BGT     2$                ;BR IF NO SOFTWARE TIMEOUT
8917  041142  004737  041174                JSR     PC,STO            ;CALL SOFTWARE TIMEOUT ROUTINE
8918  041146  000405                        BR      3$                ;GO TO THE EXIT
8919  041150  005201                2$:     INC     R1                ;MOVE TO NEXT DRIVE
8920  041152  005723                        TST     (R3)+
8921  041154  022701  000010                CMP     #8.,R1            ;OUT OF DRIVES?
8922  041160  003361                        BGT     1$                ;BRANCH IF NO
8923  041162  104413                3$:     RESREG                    ;RESTORE R0 - R5
8924  041164  105037  034551                CLRB    ACTSTR            ;ZERO ACTIVE SOFTWARE TIMEOUT ROUTINE FLAG
8925  041170  012616                4$:     MOV     (SP)+,(SP)        ;ADJUST THE STACK
8926  041172  000207                        RTS     PC                ;RETURN
8927
8928                                  ;SOFTWARE TIMEOUT ROUTINE
8929                                  ;
8930                                  ;NOTE: THIS ROUTINE MUST BE ENTERED AT PRIORITY 6
8931                                  ;       OR GREATER
8932                                  ;
8933                                  ;CALL:  STO
8934                                  ;       MOV     #DRVNUM,R1        ;DRIVE NUMBER
8935                                  ;       JSR     PC,STO            ;CALL
8936                                  ;       RETURN
8937
8938  041174  010146                STO:    MOV     R1,-(SP)          ;SAVE R1
8939  041176  010346                        MOV     R3,-(SP)          ;SAVE R3
8940  041200  013704  034632                MOV     RPADR,R4          ;GET ADDRESS OF "RPCS1"
8941  041204  010164  000010                MOV     R1,RPCS2(R4)      ;SELECT THE DRIVE
8942  041210  004037  041630                JSR     R0,RD.RP          ;READ "DRIVE STATUS REG"
8943  041214  000012                        RPDS1
8944  041216  041516                        STO5
8945  041220  105726                        TSTB    (SP)+             ;IS "DRY"=1?
8946  041222  100477                        BMI     STO2              ;BR IF YES
8947  041224  105761  034524        STO1:   TSTB    DPINT(R1)         ;TRYING TO INTIALIZE THE DRIVE ?
8948  041230  001074                        BNE     STO2              ;BR IF YES
8949  041232  105761  034534                TSTB    DPRQS(R1)         ;OUTSTANDING PORT REQUEST FOR THE DRIVE ?
8950  041236  001071                        BNE     STO2              ;BR IF YES
8951  041240  013702  034544                MOV     TRNSWT,R2         ;PICKUP TRANSFER WAIT QUEUE
8952  041244  020137  034616                CMP     R1,DTUW           ;TRANSFER UNDERWAY ON THIS DRIVE?
8953  041250  001402                        BEQ     1$                ;BRANCH IF YES
8954  041252  004737  043036                JSR     PC,GETREQ         ;GET DPB ADDRESS
8955  041256  052762  101000  000016  1$:    BIS     #BIT15!BIT09,16(R2) ;SET THE ERROR FLAGS
8956  041264  004737  042202                JSR     PC,SVRH11         ;SAVE RH11/RP04/5/6 REGISTERS
8957  041270  012764  000040  000010        MOV     #BIT05,RPCS2(R4)  ;"INIT" THE MASS BUS
8958  041276  105061  034474                CLRB    DRVACT(R1)        ;DRIVE IS IDLE
8959  041302  105061  034552                CLRB    ULDFLG(R1)        ;CLEAR THE UNLOAD FLAG
8960  041306  005001                        CLR     R1                ;START WITH DRIVE 0
8961  041310  005003                        CLR     R3
8962  041312  004037  035062        2$:     JSR     R0,DRVINT         ;INIT. THIS DRIVE
8963  041316  000477                        BR      STO5              ;PARITY ERROR RETURN
```

```
 8964  041320  105761  034474                      TSTB    DRVACT(R1)          ;DRIVE IDLE BEFORE THE INIT.?
 8965  041324  001414                              BEQ     4$                  ;YES--BRANCH
 8966  041326  013702  034544                      MOV     TRNSWT,R2           ;GET TRANSFER WAIT QUEUE
 8967  041332  023701  034616                      CMP     DTUW,R1             ;WAS THERE I/O ON THIS DRIVE?
 8968  041336  001402                              BEQ     3$                  ;YES--BRANCH
 8969  041340  004737  043036                      JSR     PC,GETREQ           ;GET THE DPB POINTER FROM QUEUE
 8970  041344  052762  100400  000016  3$:         BIS     #BIT15!BIT08,16(R2) ;INFORM USER OF INIT.
 8971  041352  105061  034474                      CLRB    DRVACT(R1)          ;SET DRIVE ACTIVE TO IDLE
 8972  041356  105061  034552              4$:     CLRB    ULDFLG(R1)          ;NO UNLOAD
 8973  041362  012763  177777  034576              MOV     #-1,TIMER(R3)       ;STOP THE TIMER
 8974  041370  005723                              TST     (R3)+               ;UPDATE THE INDEX
 8975  041372  005201                              INC     R1                  ;INCREMENT THE DRIVE NUMBER
 8976  041374  022701  000010                      CMP     #8.,R1              ;LAST DRIVE BEEN CHECKED?
 8977  041400  003344                              BGT     2$                  ;NO--LOOP
 8978  041402  012737  177777  034616              MOV     #-1,DTUW            ;NO DATA TRANSFERS UNDERWAY
 8979  041410  005037  034544                      CLR     TRNSWT              ;CLEAR TRANSFER WAIT QUEUE
 8980  041414  004737  042664                      JSR     PC,CLRQUE           ;CLEAR ALL REQUEST QUEUES
 8981  041420  000500                              BR      ST09                ;EXIT
 8982  041422  116405  000016          ST02:       MOVB    RPAS(R4),R5         ;READ ATTENTION REG
 8983  041426  136105  034620                      BITB    ATABIT(R1),R5       ;IS ATTENTION FOR THIS DRIVE UP ?
 8984  041432  001017                              BNE     ST03                ;YES--BRANCH
 8985  041434  105761  034524                      TSTB    DPINT(R1)           ;TRYING TO INTIALIZE THE DRIVE ?
 8986  041440  001031                              BNE     ST06                ;BR IF YES - DRIVE NOT ONLINE
 8987  041442  105761  034534                      TSTB.   DPRQS(R1)           ;OUTSTANDING PORT REQUEST FOR THE DRIVE ?
 8988  041446  001045                              BNE     ST07                ;BR IF YES - NO RESPONSE TO REQUEST
 8989  041450  020137  034616                      CMP     R1,DTUW             ;DATA TRANSFER UNDERWAY FOR THIS DRIVE
 8990  041454  001263                              BNE     ST01                ;BR IF NO
 8991  041456  004037  041630                      JSR     R0,RD.RP            ;YES--CHECK "RDY"
 8992  041462  000000                              RPCS1
 8993  041464  041516                              ST05
 8994  041466  105726                              TSTB    (SP)+
 8995  041470  100255                              BPL     ST01                ;BR IF "RDY"=0
 8996  041472  105761  034524          ST03:       TSTB    DPINT(R1)           ;INITIALIZING THE DRIVE ?
 8997  041476  001003                              BNE     1$                  ;BR IF INIT PENDING
 8998  041500  105761  034534                      TSTB    DPRQS(R1)           ;PORT REQUEST PENDING ?
 8999  041504  001446                              BEQ     ST09                ;BR IF NOT
 9000  041506  012763  177777  034576  1$:         MOV     #-1,TIMER(R3)       ;STOP THE TIMER
 9001  041514  000442                              BR      ST09                ;EXIT
 9002  041516  004737  037152          ST05:       JSR     PC,C18              ;GO HANDLE THE PARITY ERROR
 9003  041522  000437                              BR      ST09
 9004  041524  105061  034524          ST06:       CLRB    DPINT(R1)           ;CLEAR THE INITIALIZE INDICATOR
 9005  041530  105061  034504                      CLRB    DRVSTA(R1)          ;SET UNIT OFFLINE
 9006  041534  012763  177777  034576              MOV     #-1,TIMER(R3)  ;STOP THE TIMER
 9007  041542  004737  043036                      JSR     PC,GETREQ           ;GET THE DPB ADDRESS
 9008  041546  005702                              TST     R2                  ;REQUEST IN QUEUE ?
 9009  041550  001424                              BEQ     ST09                ;BR IF NOT
 9010  041552  052762  140000  000016              BIS     #BIT15!BIT14,16(R2)  ;INFORM THE USER DRIVE NOT AVAILABLE
 9011  041560  000414                              BR      ST08                ;FINISH
 9012  041562  012763  177777  034576  ST07:       MOV     #-1,TIMER(R3)       ;STOP THE TIMER
 9013  041570  105061  034534                      CLRB    DPRQS(R1)           ;CLEAR PORT REQUEST INDICATOR
 9014  041574  004737  043036                      JSR     PC,GETREQ           ;GET DPB ADDRESS
 9015  041600  005702                              TST     R2                  ;QUEUE ENTRY FOR DRIVE ?
 9016  041602  001407                              BEQ     ST09                ;BR IF NONE
 9017  041604  012762  100004  000016              MOV     #BIT15!BIT2,16(R2)  ;INFORM USER OF PORT REQUEST ERROR
 9018  041612  004737  042742          ST08:       JSR     PC,EMPTYQ           ;CLEAR THE QUEUE FOR THE DRIVE
 9019  041616  004737  042202                      JSR     PC,SVRH11           ;SAVE THE REGISTERS
```

```
9020  041622  012603                  ST09:   MOV     (SP)+,R3        ;RESTORE R3
9021  041624  012601                          MOV     (SP)+,R1        ;RESTORE R1
9022  041626  000207                          RTS     PC              ;RETURN
9023
9024                                  ;ROUTINE TO READ A RH11/RP04/5/6 REGISTER
9025                                  ;
9026                                  ;CALL
9027                                  ;       JSR     R0,RD.RP        ;GO READ A REGISTER
9028                                  ;       INDEX                   ;REG. INDEX FROM BASE
9029                                  ;       ERRADR                  ;ERROR ADDRESS--PROCESS ERROR STARTING
9030                                  ;                               ;AT THIS ADDRESS
9031                                  ;       RETURN                  ;CONTENTS OF REG. IS ON THE STACK
9032                                  ;
9033  041630  013737  034630  041776  RD.RP:  MOV     MCPEMX,RD.RP2   ;MAX. RETRYS ALLOWED
9034  041636  011646                          MOV     (SP),-(SP)      ;SAVE R0 FOR RETURN
9035  041640  013737  034632  041654          MOV     RPADR,RD.ADR    ;FORM THE DESIRED ADDRESS
9036  041646  062037  041654                  ADD     (R0)+,RD.ADR    ;USING THE BASE AND THE INDEX
9037  041652  013727                  RD.RP1: MOV     @(PC)+,(PC)+    ;READ THE DESIRED REGISTER OF THE RP04
9038  041654  000000                  RD.ADR: .WORD   0               ;ADDRESS IS FORMED HERE
9039  041656  000000                  RD.WRD: .WORD   0               ;REG. CONTENTS PUT HERE
9040  041660  013766  041656  000002          MOV     RD.WRD,2(SP)    ;RETURN IT TO THE USER
9041  041666  013746  034632                  MOV     RPADR,-(SP)     ;PUT THE ADDRESS ON THE STACK
9042  041672  062716  000010                  ADD     #RPCS2,(SP)     ;FORM THE ADDRESS OF RPCS2
9043  041676  032736  010000                  BIT     #BIT12,@(SP)+   ;CHECK THE 'NED' BIT
9044  041702  001037                          BNE     RD.RP3          ;BR IF DRIVE NON-EXISTENT
9045  041704  017746  172722                  MOV     @RPADR,-(SP)    ;READ RPCS1
9046  041710  032716  020000                  BIT     #BIT13,(SP)     ;DID MCPE SET?
9047  041714  001002                          BNE     1$              ;BRANCH IF YES
9048  041716  022620                          CMP     (SP)+,(R0)+     ;ADJUST FOR RETURN
9049  041720  000432                          BR      RD.RP4          ;EXIT
9050  041722                          1$:
9051  041722  004037  043122                  JSR     R0,ES.SAV       ;SAVE THE ADDRESS IN '$ESCAPE'
9052  041726  104003                          ERROR   3               ;REPORT "MCPE" ERROR
9053  041730  005737  034616                  TST     DTUW            ;DATA TRANSFER UNDERWAY?
9054  041734  100405                          BMI     2$              ;NO--BRANCH
9055  041736  032716  040000                  BIT     #BIT14,(SP)     ;NO--"TRE"=1?
9056  041742  001402                          BEQ     2$              ;NO--BRANCH
9057  041744  005726                          TST     (SP)+           ;YES--CLEAN OFF THE STACK AND
9058  041746  000415                          BR      RD.RP3          ;TAKE THE FATAL ERROR EXIT
9059  041750  052716  040000          2$:     BIS     #BIT14,(SP)     ;CLEAR "MCPE" BY SENDING A "1" TO "TRE"
9060  041754  000316                          SWAB    (SP)            ;POSITION BEFORE WRITING
9061  041756  013737  034632  041772          MOV     RPADR,3$        ;FORM ADDRESS OF HIGH BYTE
9062  041764  005237  041772                  INC     3$
9063  041770  112637                          MOVB    (SP)+,@(PC)+    ;WRITE THE HIGH BYTE OF RPCS1
9064  041772  000000                  3$:     .WORD   0               ;ADDRESS STORAGE
9065  041774  005327                          DEC     (PC)+           ;EXCEEDED MAX. RETRYS
9066  041776  000003                  RD.RP2: .WORD   3
9067  042000  002324                          BGE     RD.RP1          ;BRANCH IF NO
9068  042002  011000                  RD.RP3: MOV     (R0),R0         ;FATAL ERROR EXIT
9069  042004  012616                          MOV     (SP)+,(SP)
9070  042006  000200                  RD.RP4: RTS     R0
9071
9072                                  ;ROUTINE TO WRITE A REGISTER
9073                                  ;
9074                                  ;CALL
9075                                  ;       MOV     DATA,-(SP)      ;DATA TO BE LOADED ON THE STACK
```

```
 9076                                   ;       JSR     R0,WRT.RP       ;CALL THE ROUTINE TO LOAD(WRITE) THE REG.
 9077                                   ;       INDEX                   ;INDEX OF THE REGISTER TO BE LOADED
 9078                                   ;       ERRADR                  ;ADDRESS TO RETURN TO ON AN ERROR
 9079                                   ;       RETURN                  ;ERROR FREE RETURN
 9080
 9081  042010  013737  034630  042164  WRT.RP: MOV     MCPEMX,WRT.R2   ;MAX RETRYS ALLOWED
 9082  042016  016637  000002  042076          MOV     2(SP),WRT.WD    ;SAVE THE WORD TO WRITE
 9083  042024  012616                          MOV     (SP)+,(SP)      ;ADJUST THE STACK
 9084  042026  012037  042100                  MOV     (R0)+,WRT.AD    ;GET INDEX OF REGISTER TO BE WRITTEN
 9085  042032  001015                          BNE     1$              ;BRANCH IF NOT RPCS1
 9086  042034  122737  000150  042076          CMPB    #150,WRT.WD     ;IS THE COMMAND FOR DATA TRANSFERS?
 9087  042042  002411                          BLT     1$              ;YES--DON'T GET THE OLD A16 & A17, & PSEL
 9088  042044  004037  041630                  JSR     R0,RD.RP        ;NO---COMBINE A16&A17, & PSEL WITH
 9089  042050  000000                          RPCS1                   ;THE COMMAND BEFORE SENDING IT TO
 9090  042052  042172                          WRT.R3                  ;THE RH11/RP04
 9091  042054  000316                          SWAB    (SP)
 9092  042056  042716  177770                  BIC     #^C7,(SP)
 9093  042062  112637  042077                  MOVB    (SP)+,WRT.WD+1
 9094  042066  063737  034632  042100  1$:     ADD     RPADR,WRT.AD    ;FORM THE ADDRESS OF THE DISK REG.
 9095  042074  012737                  WRT.R1: MOV     (PC)+,@(PC)+    ;LOAD THE DESIRED REG.
 9096  042076  000000                  WRT.WD: .WORD   0               ;WORD TO WRITE GOES HERE
 9097  042100  000000                  WRT.AD: .WORD   0               ;ADDRESS IS FORMED HERE
 9098  042102  013746  034632                  MOV     RPADR,-(SP)     ;PUT THE ADDRESS ON THE STACK
 9099  042106  062716  000010                  ADD     #RPCS2,(SP)     ;FORM THE ADDRESS OF RPCS2
 9100  042112  032736  010000                  BIT     #BIT12,@(SP)+   ;CHECK THE 'NED' BIT
 9101  042116  001025                          BNE     WRT.R3          ;BR IF DRIVE NON-EXISTENT
 9102  042120  004037  041630                  JSR     R0,RD.RP        ;CHECK FOR PARITY ERROR ON WRITE
 9103  042124  000014                          RPER1
 9104  042126  042172                          WRT.R3
 9105  042130  032726  000010                  BIT     #BIT03,(SP)+
 9106  042134  001420                          BEQ     WRT.R4          ;BRANCH IF "PAR=0"
 9107  042136  016037  177776  042150          MOV     -2(R0),1$       ;PICKUP THE INDEX
 9108  042144  004037  041630                  JSR     R0,RD.RP        ;READ THE REG.
 9109  042150  000000                  1$:     .WORD   0               ;REG. INDEX
 9110  042152  042172                          WRT.R3                  ;RETURN TO THIS ADDRESS ON ERROR
 9111  042154  004037  043122                  JSR     R0,ES.SAV       ;SAVE THE ADDRESS IN '$ESCAPE'
 9112  042160  104004                          ERROR   4               ;REPORT THE PARITY ON WRITE ERROR
 9113  042162  005327                          DEC     (PC)+           ;DECREMENT THE ERROR COUNT
 9114  042164  000003                  WRT.R2: .WORD   3               ;RETRY COUNTER
 9115  042166  002342                          BGE     WRT.R1          ;TRY AGAIN IF NOT FINISHED
 9116  042170  005726                          TST     (SP)+           ;CLEAN OFF THE STACK
 9117  042172  011000                  WRT.R3: MOV     (R0),R0         ;TAKE THE "PARITY ON WRITE" ERROR EXIT
 9118  042174  000401                          BR      WRT.R5          ;EXIT
 9119  042176  005720                  WRT.R4: TST     (R0)+           ;ADJUST FOR ERROR FREE EXIT
 9120  042200  000200                  WRT.R5: RTS     R0
 9121
 9122                                   ;ROUTINE TO SAVE THE RH11/RP04/5/6 REGISTERS AS PER DPB+14
 9123                                   ;
 9124                                   ;CALL
 9125                                   ;       MOV     #DPBNUM,R2      ;DPB POINTER TO R2
 9126                                   ;       JSR     PC,SVRH11       ;SAVE THE DRIVES REG'S
 9127
 9128  042202  104412                  SVRH11: SAVREG                  ;SAVE R0 - R5
 9129  042204  005702                          TST     R2              ;QUEUE ENTRY FOR THE DRIVE ?
 9130  042206  001430                          BEQ     4$              ;BR IF NONE
 9131  042210  013704  034632                  MOV     RPADR,R4
```

```
9132  042214  111264  000010                  MOVB    (R2),RPCS2(R4)  ;SELECT DRIVE
9133  042220  016203  000014                  MOV     14(R2),R3       ;GET THE ERROR TABLE POINTER
9134  042224  001433                          BEQ     6$              ;EXIT IF NO ADDRESS
9135  042226  005037  042262                  CLR     3$              ;COUNTER & POINTER
9136  042232  023727  042262  000022  1$:     CMP     3$,#RPDB        ;REACHED THE BUFFER REGISTER ?
9137  042240  001006                          BNE     2$              ;BR IF NOT
9138  042242  032764  000200  000010          BIT     #BIT07,RPCS2(R4)  ;'OR' SET ?
9139  042250  001002                          BNE     2$              ;BR IF SET
9140  042252  005023                          CLR     (R3)+           ;STORE RPDB AS ZEROES
9141  042254  000405                          BR      4$              ;CONTINUE
9142  042256  004037  041630          2$:     JSR     R0,RD.RP        ;READ THE SELECTED REGISTER
9143  042262  000000                  3$:     .WORD   0               ;REGISTER INDEX
9144  042264  042310                          5$                      ;ERROR RETURN ADDRESS
9145  042266  012623                          MOV     (SP)+,(R3)+     ;STORE THE REGISTER CONTENTS
9146  042270  023727  042262  000046  4$:     CMP     3$,#RPEC2       ;REACHED THE END ?
9147  042276  001406                          BEQ     6$              ;BR IF YES
9148  042300  062737  000002  042262          ADD     #2,3$           ;INCREMENT THE REGISTER INDEX
9149  042306  000751                          BR      1$              ;CONTINUE READING THE REGISTERS
9150  042310  004737  037052          5$:     JSR     PC,C17          ;PROCESS THE UNCORRECTABLE PARITY ERROR
9151  042314  104413                  6$:     RESREG                  ;RESTORE R0 - R5
9152  042316  000207                          RTS     PC              ;RETURN
9153
9154                                  ;ROUTINE TO SET THE INTERRUPT WITHOUT GETTING A "TRE"
9155                                  ;CALL
9156                                  ;       MOV     #DRVNUM,R1      ;DRIVE NUMBER TO R1
9157                                  ;       JSR     PC,SET.IE       ;SET "IE"
9158                                  ;       RETURN
9159
9160  042320  010446                  SET.IE: MOV     R4,-(SP)        ;SAVE R4
9161  042322  013704  034632                  MOV     RPADR,R4        ;PICKUP ADDRESS OF RPCS1
9162  042326  010164  000010                  MOV     R1,RPCS2(R4)    ;SELECT DRIVE
9163  042332  011446                          MOV     (R4),-(SP)      ;READ RPCS1
9164  042334  052716  040000                  BIS     #BIT14,(SP)     ;SET THE "TRE" BIT OF THE WORD READ
9165  042340  000316                          SWAB    (SP)            ;ADJUST FOR DATO
9166  042342  112714  000100                  MOVB    #BIT06,(R4)     ;SET "IE"
9167  042346  032764  010000  000010          BIT     #BIT12,RPCS2(R4) ;IS "NED"=1?
9168  042354  001002                          BNE     1$              ;YES--CLEAR "TRE"
9169  042356  005726                          TST     (SP)+           ;CLEAN OFF THE STACK
9170  042360  000402                          BR      2$
9171  042362  112664  000001          1$:     MOVB    (SP)+,1(R4)     ;CLEAR "TRE"
9172  042366  012604                  2$:     MOV     (SP)+,R4        ;RESTORE R4
9173  042370  000207                          RTS     PC              ;RETURN TO CALLER
9174
9175                                  ;QUEUE COUNT
9176  042372      000                 QCNT:   .BYTE   0               ;DRIVE 0
9177  042373      000                         .BYTE   0               ;DRIVE 1
9178  042374      000                         .BYTE   0               ;DRIVE 2
9179  042375      000                         .BYTE   0               ;DRIVE 3
9180  042376      000                         .BYTE   0               ;DRIVE 4
9181  042377      000                         .BYTE   0               ;DRIVE 5
9182  042400      000                         .BYTE   0               ;DRIVE 6
9183  042401      000                         .BYTE   0               ;DRIVE 7
9184
9185                                  ;QUEUE INPUT POINTERS
9186
9187  042402  042464                  QINPT:  .WORD   QDRV0            ;DRIVE 0
```

```
 9188  042404  042504                        .WORD   QDRV1           ;DRIVE 1
 9189  042406  042524                        .WORD   QDRV2           ;DRIVE 2
 9190  042410  042544                        .WORD   QDRV3           ;DRIVE 3
 9191  042412  042564                        .WORD   QDRV4           ;DRIVE 4
 9192  042414  042604                        .WORD   QDRV5           ;DRIVE 5
 9193  042416  042624                        .WORD   QDRV6           ;DRIVE 6
 9194  042420  042644                        .WORD   QDRV7           ;DRIVE 7
 9195
 9196                                ;QUEUE OUTPUT POINTERS
 9197
 9198  042422  042464                QOUTPT: .WORD   QDRV0           ;DRIVE 0
 9199  042424  042504                        .WORD   QDRV1           ;DRIVE 1
 9200  042426  042524                        .WORD   QDRV2           ;DRIVE 2
 9201  042430  042544                        .WORD   QDRV3           ;DRIVE 3
 9202  042432  042564                        .WORD   QDRV4           ;DRIVE 4
 9203  042434  042604                        .WORD   QDRV5           ;DRIVE 5
 9204  042436  042624                        .WORD   QDRV6           ;DRIVE 6
 9205  042440  042644                        .WORD   QDRV7           ;DRIVE 7
 9206
 9207  042442  042464                QSTART: .WORD   QDRV0           ;DRIVE 0 START ADDRESS
 9208  042444  042504                QSTOP:  .WORD   QDRV1           ;DRIVE 0 STOP ADDRESS & DRIVE 1 START ADDRESS
 9209  042446  042524                        .WORD   QDRV2           ;STOP DRIVE 1--START DRIVE 2
 9210  042450  042544                        .WORD   QDRV3           ;STOP DRIVE 2--START DRIVE 3
 9211  042452  042564                        .WORD   QDRV4           ;STOP DRIVE 3--START DRIVE 4
 9212  042454  042604                        .WORD   QDRV5           ;STOP DRIVE 4--START DRIVE 5
 9213  042456  042624                        .WORD   QDRV6           ;STOP DRIVE 5--START DRIVE 6
 9214  042460  042644                        .WORD   QDRV7           ;STOP DRIVE 6--START DRIVE 7
 9215  042462  042664                        .WORD   QTERM           ;STOP DRIVE 7
 9216
 9217                                ;DRIVE REQUEST QUEUES
 9218
 9219  042464  000010                QDRV0:  .BLKW   10
 9220  042504  000010                QDRV1:  .BLKW   10
 9221  042524  000010                QDRV2:  .BLKW   10
 9222  042544  000010                QDRV3:  .BLKW   10
 9223  042564  000010                QDRV4:  .BLKW   10
 9224  042604  000010                QDRV5:  .BLKW   10
 9225  042624  000010                QDRV6:  .BLKW   10
 9226  042644  000010                QDRV7:  .BLKW   10
 9227          042664                QTERM=.
 9228
 9229                                ;ROUTINE TO CLEAR ALL OF THE REQUEST QUEUES
 9230                                ;
 9231                                ;CALL
 9232                                ;       JSR     PC,CLRQUE
 9233
 9234  042664  104412                CLRQUE: SAVREG                  ;SAVE R0 - R5
 9235  042666  012702  042372               MOV     #QCNT,R2        ;ZERO THE QUEUE COUNTS
 9236  042672  005022                        CLR     (R2)+           ;DRIVES 0 & 1
 9237  042674  005022                        CLR     (R2)+           ;DRIVES 2 & 3
 9238  042676  005022                        CLR     (R2)+           ;DRIVES 4 & 5
 9239  042700  005022                        CLR     (R2)+           ;DRIVES 6 & 7
 9240  042702  012703  000010               MOV     #8.,R3          ;MOVE THE STARTING
 9241  042706  012701  042442               MOV     #QSTART,R1      ;ADDRESS OF THE QUEUE INTO
 9242  042712  012122                1$:     MOV     (R1)+,(R2)+     ;THE QUEUE INPUT POINTER
 9243  042714  005303                        DEC     R3
```

```
 9244  042716  001375                          BNE     1$
 9245  042720  012703  000010                  MOV     #8.,R3          ;MOVE THE STARTING ADDRESS
 9246  042724  012701  042442                  MOV     #QSTART,R1      ;OF THE QUEUE INTO THE
 9247  042730  012122                  2$:     MOV     (R1)+,(R2)+     ;QUEUE OUTPUT POINTER
 9248  042732  005303                          DEC     R3
 9249  042734  001375                          BNE     2$
 9250  042736  104413                          RESREG                  ;RESTORE R0 - R5
 9251  042740  000207                          RTS     PC
 9252
 9253                                  ;EMPTY THE QUEUE SPECIFIED BY R1
 9254                                  ;
 9255                                  ;CALL
 9256                                  ;       MOV     DRVNUM,R1       ;DRIVE NUMBER TO R1
 9257                                  ;       JSR     PC,EMPTYQ
 9258
 9259  042742  105061  042372          EMPTYQ: CLRB    QCNT(R1)        ;CLEAR NUMBER OF ITEMS IN QUEUE
 9260  042746  006301                          ASL     R1
 9261  042750  016161  042402  042422          MOV     QINPT(R1),QOUTPT(R1) ;SET OUTPUT QUEUE POINTER=INPUT POINTER
 9262  042756  006201                          ASR     R1
 9263  042760  000207                          RTS     PC
 9264
 9265                                  ;ROUTINE TO PUT A REQUEST IN QUEUE
 9266                                  ;
 9267                                  ;CALL
 9268                                  ;       MOV     #DRVNUM,R1      ;DRIVE NUMBER
 9269                                  ;       MOV     #DPB,R2         ;ADDRESS OF PARAMETER BLOCK
 9270                                  ;       JSR     R0,DRVQUE       ;GO PUT REQUEST IN QUEUE
 9271                                  ;       RETURN1                 ;RETURN HERE IF QUEUE IS FULL
 9272                                  ;       RETURN2                 ;RETURN HERE IF REQUEST IS IN QUEUE
 9273
 9274  042762  122761  000010  042372  DRVQUE: CMPB    #10,QCNT(R1)    ;IS QUEUE FULL?
 9275  042770  001421                          BEQ     2$              ;BR IF YES-TAKE RETURN1
 9276  042772  105261  042372                  INCB    QCNT(R1)        ;INCREMENT QUEUE COUNT
 9277  042776  006301                          ASL     R1
 9278  043000  010271  042402                  MOV     R2,@QINPT(R1)   ;PUT THIS REQUEST IN QUEUE
 9279  043004  062761  000002  042402          ADD     #2,QINPT(R1)    ;UPDATE THE QUEUE POINTER
 9280  043012  026161  042402  042444          CMP     QINPT(R1),QSTOP(R1) ;TIME TO RESET THE POINTER
 9281  043020  001003                          BNE     1$              ;BRANCH IF NO
 9282  043022  016161  042442  042402          MOV     QSTART(R1),QINPT(R1) ;YES--RESET POINTER
 9283  043030  006201                  1$:     ASR     R1
 9284  043032  005720                          TST     (R0)+           ;TAKE RETURN 2
 9285  043034  000200                  2$:     RTS     R0              ;RETURN TO USER
 9286
 9287                                  ;ROUTINE TO GET THE "DPB" ADDRESS OF NEXT REQUEST IN QUEUE
 9288                                  ;
 9289                                  ;CALL
 9290                                  ;       MOV     #DRVNUM,R1      ;DRIVE NUMBER TO R1
 9291                                  ;       JSR     PC,GETREQ       ;GO GET THE REQUEST
 9292                                  ;       RETURN                  ;R2="DPB" ADDRESS OF THE REQUEST
 9293                                  ;                               ;R2=0 IF NO REQUEST IN QUEUE
 9294
 9295  043036  005002                  GETREQ: CLR     R2
 9296  043040  105761  042372                  TSTB    QCNT(R1)        ;IS THERE ANY REQUEST IN QUEUE?
 9297  043044  001404                          BEQ     2$              ;NO---BRANCH
 9298  043046  006301                  1$:     ASL     R1
 9299  043050  017102  042422                  MOV     @QOUTPT(R1),R2  ;PICKUP "DPB" POINTER FOR THIS DRIVE
```

```
 9300  043054  006201                        ASR     R1
 9301  043056  000207                2$:     RTS     PC              ;RETURN TO USER
 9302
 9303                                ;ROUTINE TO "POP" THE REQUEST FROM QUEUE
 9304                                ;
 9305                                ;CALL
 9306                                ;       MOV     #DRVNUM,R1      ;DRIVE NUMBER TO R1
 9307                                ;       JSR     PC,POPQUE       ;CALL TO REMOVE REQUEST
 9308                                ;       RETURN                  ;R2=ADDRESS OF DPB REMOVED
 9309
 9310  043060  105361  042372      POPQUE: DECB    QCNT(R1)        ;DECREMENT QUEUE COUNT
 9311  043064  006301                        ASL     R1
 9312  043066  017102  042422                MOV     @QOUTPT(R1),R2  ;GET THE "DPB" POINTER
 9313  043072  062761  000002  042422        ADD     #2,QOUTPT(R1)   ;UPDATE THE QUEUE POINTER
 9314  043100  026161  042422  042444        CMP     QOUTPT(R1),QSTOP(R1) ;TIME TO RESET THE POINTER?
 9315  043106  001003                        BNE     1$              ;NO--BRANCH TO EXIT
 9316  043110  016161  042442  042422        MOV     QSTART(R1),QOUTPT(R1) ;YES--RESET THE POINTER
 9317  043116  006201                1$:     ASR     R1
 9318  043120  000207                        RTS     PC              ;RETURN TO USER
 9319
 9320                                ;ROUTINE TO SAVE THE CONTENTS OF '$ESCAPE' WHEN THE DRIVER
 9321                                ;REPORTS AN ERROR DIRECTLY.
 9322                                ;
 9323                                ;CALL
 9324                                ;       JSR     R0,ES.SAV
 9325                                ;       ERROR   N               ;;THE ERROR CALL
 9326                                ;       RETURN                  ;THE RETURN IS PAST THE ERROR CALL
 9327
 9328  043122  012037  043136      ES.SAV: MOV     (R0)+,1$        ;GET THE ERROR CALL
 9329  043126  013746  001206              MOV     $ESCAPE,-(SP)   ;SAVE THE ADDRESS IN '$ESCAPE'
 9330  043132  005037  001206              CLR     $ESCAPE         ;CLEAR THE ESCAPE RETURN
 9331  043136  000000                1$:     .WORD   0               ;THE ERROR CALL IS MOVED HERE
 9332  043140  012637  001206              MOV     (SP)+,$ESCAPE   ;RESTORE THE ESCAPE ADDRESS
 9333  043144  000200                        RTS     R0              ;RETURN
 9334
 9335
 9336                                .SBTTL  ASCIZ MESSAGES
 9337
 9338  043146  000122                MSG.R:  .ASCIZ  /R/
 9339  043150  041506     000        MSG.FC: .ASCIZ  /FC/
 9340  043153     114  000103        MSG.LC: .ASCIZ  /LC/
 9341  043156  041506  000047        MSGFCP: .ASCIZ  /FC'/
 9342  043162  041514  000047        MSGLCP: .ASCIZ  /LC'/
 9343  043166  041511     000        MSG.IC: .ASCIZ  /IC/
 9344  043171     106  000124        MSG.FT: .ASCIZ  /FT/
 9345  043174  052114     000        MSG.LT: .ASCIZ  /LT/
 9346  043177     111  000124        MSG.IT: .ASCIZ  /IT/
 9347  043202  051506     000        MSG.FS: .ASCIZ  /FS/
 9348  043205     114  000123        MSG.LS: .ASCIZ  /LS/
 9349  043210  040520  000124        MSG.PAT:        .ASCIZ  /PAT/
 9350  043214  000075                MSG.EQ: .ASCIZ  /=/
 9351  043216  005015  047503  052116 MSG.CS: .ASCIZ  <CR><LF>/CONTROL SWITCHES=/
 9352  043224  047522  020114  053523
 9353  043232  052111  044103  051505
 9354  043240  000075
 9355  043242  005015  040527  047122 USE:    .ASCIZ  <CR><LF>/WARNING: PROGRAMMABLE DRIVES MAY BE USED/<CR><LF>
```

```
 9356  043250  047111  035107  050040
 9357  043256  047522  051107  046501
 9358  043264  040515  046102  020105
 9359  043272  051104  053111  051505
 9360  043300  046440  054501  041040
 9361  043306  020105  051525  042105
 9362  043314  005015     000
 9363  043317     040  051120  043517  NOUSE:  .ASCIZ  / PROGRAMMABLE-DRIVE WILL NOT BE USED/
 9364  043324  040522  046515  041101
 9365  043332  042514  042055  044522
 9366  043340  042526  053440  046111
 9367  043346  020114  047516  020124
 9368  043354  042502  052440  042523
 9369  043362  000104
 9370
 9371  043364  027440  000040          SLASH:  .ASCIZ  @ / @
 9372  043370  047125  052111  051440  SYSTAT: .ASCIZ  /UNIT STATUS:/<CR><LF><LF>
 9373  043376  040524  052524  035123
 9374  043404  005015  000012
 9375  043410  051104  053111  000105  UNTMSG: .ASCIZ  /DRIVE/
 9376  043416  047440  043106  044514  UNTOFF: .ASCIZ  / OFFLINE/
 9377  043424  042516     000
 9378  043427     040  047117  044514  UNTON:  .ASCIZ  / ONLINE/
 9379  043434  042516     000
 9380  043437     040  047516  020124  NOTPRS: .ASCIZ  / NOT PRESENT/
 9381  043444  051120  051505  047105
 9382  043452  000124
 9383  043454  052440  051516  043101  NOTSAF: .ASCIZ  / UNSAFE/
 9384  043462  000105
 9385  043464  047040  052117  051040  NOTRP:  .ASCIZ  @ NOT RP04/5/6@
 9386  043472  030120  027464  027465
 9387  043500  000066
 9388  043502  050122  032060     000  RP04B:  .ASCIZ  /RP04/
 9389  043507     122  030120  000065  RP05:   .ASCIZ  /RP05/
 9390  043514  050122  033060     000  RP06:   .ASCIZ  /RP06/
 9391  043521     015  042012  044522  DRIVES: .ASCIZ  <CR><LF>/DRIVE(S) TO BE TESTED /
 9392  043526  042526  051450  020051
 9393  043534  047524  041040  020105
 9394  043542  042524  052123  042105
 9395  043550  000040
 9396  043552  047516  042516     000  NONE:   .ASCIZ  /NONE/
 9397  043557     054     000          COMMA:  .ASCIZ  /,/
 9398  043561     015  047012  020117  NOCLOK: .ASCIZ  <CR><LF>/NO KW11-P CLOCK, TIMING TESTS WILL NOT BE PERFORMED/
 9399  043566  053513  030461  050055
 9400  043574  041440  047514  045503
 9401  043602  020054  044524  044515
 9402  043610  043516  052040  051505
 9403  043616  051524  053440  046111
 9404  043624  020114  047516  020124
 9405  043632  042502  050040  051105
 9406  043640  047506  046522  042105
 9407  043646     000
 9408  043647     015  005012  042524  TESTNG: .ASCIZ  <CR><LF><LF>/TESTING DRIVE /
 9409  043654  052123  047111  020107
 9410  043662  051104  053111  020105
 9411  043670     000
```

```
 9412  043671     123  051105  040511  SERIAL: .ASCIZ  /SERIAL NUMBER /
 9413  043676  020114  052516  041115
 9414  043704  051105  000040
 9415
 9416  043710  005015  051012  052117  MSG7:   .ASCIZ  <CR><LF><LF>/ROTATIONAL SPEED TIMES/
 9417  043716  052101  047511  040516
 9418  043724  020114  050123  042505
 9419  043732  020104  044524  042515
 9420  043740  000123
 9421  043742  005015  047412  042516  MSG10A: .ASCIZ  <CR><LF><LF>/ONE CYLINDER SEEK TIMES/<CR><LF>/ * FORWARD/
 9422  043750  041440  046131  047111
 9423  043756  042504  020122  042523
 9424  043764  045505  052040  046511
 9425  043772  051505  005015  025040
 9426  044000  043040  051117  040527
 9427  044006  042122     000
 9428  044011     015  020012  020052  MSG10B: .ASCIZ  <CR><LF>/ * REVERSE/
 9429  044016  042522  042526  051522
 9430  044024  000105
 9431  044026  005015  040412  041503  MSG11A: .ASCIZ  <CR><LF><LF>/ACCESS TIME MEASURMENT/<CR><LF>/ * FORWARD/
 9432  044034  051505  020123  044524
 9433  044042  042515  046440  040505
 9434  044050  052523  046522  047105
 9435  044056  006524  020012  020052
 9436  044064  047506  053522  051101
 9437  044072  000104
 9438  044074  005015  025040  051040  MSG11B: .ASCIZ  <CR><LF>/ * REVERSE/
 9439  044102  053105  051105  042523
 9440  044110     000
 9441  044111     015  005012  040515  MSG12A: .ASCIZ  <CR><LF><LF>/MAXIMUM SEEK TIMES/<CR><LF>/ * FORWARD/
 9442  044116  044530  052515  020115
 9443  044124  042523  045505  052040
 9444  044132  046511  051505  005015
 9445  044140  025040  043040  051117
 9446  044146  040527  042122     000
 9447  044153     015  020012  020052  MSG12B: .ASCIZ  <CR><LF>/ * REVERSE/
 9448  044160  042522  042526  051522
 9449  044166  000105
 9450
 9451  044170  005015  044515  036516  MSGMIN: .ASCIZ  <CR><LF>/MIN=/
 9452  044176     000
 9453  044177     015  046412  054101  MSGMAX: .ASCIZ  <CR><LF>/MAX=/
 9454  044204  000075
 9455  044206  005015  053101  036507  MSGAVG: .ASCIZ  <CR><LF>/AVG=/
 9456  044214     000
 9457  044215     060  052440  000123  MSGOUS: .ASCIZ  /0 US/
 9458  044222  041040  046105  053517  MBELOW: .ASCIZ  / BELOW THE MINIMUM OF /
 9459  044230  052040  042510  046440
 9460  044236  047111  046511  046525
 9461  044244  047440  020106     000
 9462  044251     040  041101  053117  MABOVE: .ASCIZ  / ABOVE THE MAXIMUM OF /
 9463  044256  020105  044124  020105
 9464  044264  040515  044530  052515
 9465  044272  020115  043117  000040
 9466  044300  051440  042505  051513  MSGNUM: .ASCIZ  / SEEKS TIMED/
 9467  044306  052040  046511  042105
```

```
 9468  044314     000
 9469  044315     040  047516  020124  MSGNON: .ASCIZ  / NOT TIMED/
 9470  044322  044524  042515  000104
 9471  044330  020040     000          MSG.SP: .ASCIZ  /  /            ;TWO (2) SPACES
 9472
 9473                                  .SBTTL  ERROR HEADER (EM) MESSAGES
 9474
 9475  044333     122  030510  020061  EM1:    .ASCIZ  /RH11 INTERRUPT OCCURRED (RPAS = 0)/
 9476  044340  047111  042524  051122
 9477  044346  050125  020124  041517
 9478  044354  052503  051122  042105
 9479  044362  024040  050122  051501
 9480  044370  036440  030040  000051
 9481  044376  047125  054105  042520  EM2:    .ASCIZ  /UNEXPECTED ATTENTION OCCURRED/
 9482  044404  052103  042105  040440
 9483  044412  052124  047105  044524
 9484  044420  047117  047440  041503
 9485  044426  051125  042522  000104
 9486  044434  040515  051523  052502  EM3:    .ASCIZ  /MASSBUS PARITY ERROR(MCPE=1)/
 9487  044442  020123  040520  044522
 9488  044450  054524  042440  051122
 9489  044456  051117  046450  050103
 9490  044464  036505  024461     000
 9491  044471     115  051501  041123  EM4:    .ASCIZ  /MASSBUS PARITY ERROR(PAR=1)/
 9492  044476  051525  050040  051101
 9493  044504  052111  020131  051105
 9494  044512  047522  024122  040520
 9495  044520  036522  024461     000
 9496  044525     101  042104  042522  EM5:    .ASCIZ  /ADDRESS PLUG CHANGE BIT SET/
 9497  044532  051523  050040  052514
 9498  044540  020107  044103  047101
 9499  044546  042507  041040  052111
 9500  044554  051440  052105     000
 9501  044561     122  030510  027461  EM10:   .ASCIZ  "RH11/RP04/5/6 FAILED TO RESPOND TO ADDRESSING"
 9502  044566  050122  032060  032457
 9503  044574  033057  043040  044501
 9504  044602  042514  020104  047524
 9505  044610  051040  051505  047520
 9506  044616  042116  052040  020117
 9507  044624  042101  051104  051505
 9508  044632  044523  043516     000
 9509  044637     104  044522  042526  EM11:   .ASCIZ  /DRIVE SELECTED IS NOT ONLINE/
 9510  044644  051440  046105  041505
 9511  044652  042524  020104  051511
 9512  044660  047040  052117  047440
 9513  044666  046116  047111  000105
 9514  044674  046511  051120  050117  EM12:   .ASCIZ  /IMPROPER HEADER DATA/
 9515  044702  051105  044040  040505
 9516  044710  042504  020122  040504
 9517  044716  040524     000
 9518  044721     104  052101  020101  EM13:   .ASCIZ  /DATA COMPARE FAILURE/
 9519  044726  047503  050115  051101
 9520  044734  020105  040506  046111
 9521  044742  051125  000105
 9522  044746  044504  045523  042440  EM17:   .ASCIZ  /DISK ERROR IN TIMING TEST/
 9523  044754  051122  051117  044440
```

```
 9524  044762  020116  044524  044515
 9525  044770  043516  052040  051505
 9526  044776  000124
 9527  045000  046103  041517  020113  EM20:   .ASCIZ  /CLOCK (KW11-P) OVERFLOW IN TIMING TEST/
 9528  045006  045450  030527  026461
 9529  045014  024520  047440  042526
 9530  045022  043122  047514  020127
 9531  045030  047111  052040  046511
 9532  045036  047111  020107  042524
 9533  045044  052123     000
 9534  045047     104  051511  020113  EM23:   .ASCIZ  /DISK ERROR DURING SEEK/
 9535  045054  051105  047522  020122
 9536  045062  052504  044522  043516
 9537  045070  051440  042505  000113
 9538  045076  042523  045505  047040  EM24:   .ASCIZ  /SEEK NOT COMPLETE WITHIN 120 MS/
 9539  045104  052117  041440  046517
 9540  045112  046120  052105  020105
 9541  045120  044527  044124  047111
 9542  045126  030440  030062  046440
 9543  045134  000123
 9544  045136  044122  030461  051057  EM41:   .ASCIZ  "RH11/RP04/5/6 ERROR"
 9545  045144  030120  027464  027465
 9546  045152  020066  051105  047522
 9547  045160  000122
 9548  045162  040506  040524  020114  EM46:   .ASCIZ  /FATAL WRITE CHECK ERROR/
 9549  045170  051127  052111  020105
 9550  045176  044103  041505  020113
 9551  045204  051105  047522  000122
 9552
 9553                                   .SBTTL  STATUS/ERROR INDICATOR MESSAGES
 9554
 9555  045212  043117  046106  047111  MSGB14: .ASCIZ  /OFFLINE OR UNSAFE DRIVE REQUESTED/
 9556  045220  020105  051117  052440
 9557  045226  051516  043101  020105
 9558  045234  051104  053111  020105
 9559  045242  042522  052521  051505
 9560  045250  042524  000104
 9561  045254  047125  047514  042101  MSGB13: .ASCIZ  /UNLOADED DRIVE REQUESTED/
 9562  045262  042105  042040  044522
 9563  045270  042526  051040  050505
 9564  045276  042525  052123  042105
 9565  045304     000
 9566  045305     120  051105  044523  MSGB12: .ASCIZ  /PERSISTENT UNSAFE/
 9567  045312  052123  047105  020124
 9568  045320  047125  040523  042506
 9569  045326     000
 9570  045327     120  051101  052111  MSGB11: .ASCIZ  /PARITY ERROR OCCURRED/
 9571  045334  020131  051105  047522
 9572  045342  020122  041517  052503
 9573  045350  051122  042105     000
 9574  045355     106  052101  046101  MSGB10: .ASCIZ  /FATAL PARITY ERROR/
 9575  045362  050040  051101  052111
 9576  045370  020131  051105  047522
 9577  045376  000122
 9578  045400  047523  052106  040527  MSGB09: .ASCIZ  /SOFTWARE TIMEOUT ON THIS DRIVE/
 9579  045406  042522  052040  046511
```

```
 9580  045414  047505  052125  047440
 9581  045422  020116  044124  051511
 9582  045430  042040  044522  042526
 9583  045436     000
 9584  045437     123  043117  053524  MSGB08: .ASCIZ  /SOFTWARE TIMEOUT ON ANOTHER DRIVE/
 9585  045444  051101  020105  044524
 9586  045452  042515  052517  020124
 9587  045460  047117  040440  047516
 9588  045466  044124  051105  042040
 9589  045474  044522  042526     000
 9590  045501     105  051122  051117  MSGB06: .ASCIZ  "ERROR OCCURRED DURING I/O OPERATION"
 9591  045506  047440  041503  051125
 9592  045514  042522  020104  052504
 9593  045522  044522  043516  044440
 9594  045530  047457  047440  042520
 9595  045536  040522  044524  047117
 9596  045544     000
 9597  045545     105  051122  051117  MSGB05: .ASCIZ  "ERROR OCCURRED DURING NON-I/O OPERATION"
 9598  045552  047440  041503  051125
 9599  045560  042522  020104  052504
 9600  045566  044522  043516  047040
 9601  045574  047117  044455  047457
 9602  045602  047440  042520  040522
 9603  045610  044524  047117     000
 9604  045615     125  051516  043101  MSGB04: .ASCIZ  /UNSAFE OCCURRED/
 9605  045622  020105  041517  052503
 9606  045630  051122  042105     000
 9607  045635     101  052125  046517  MSGB03: .ASCIZ  /AUTOMATIC RECALIBRATE SEQUENCE OCCURRED/
 9608  045642  052101  041511  051040
 9609  045650  041505  046101  041111
 9610  045656  040522  042524  051440
 9611  045664  050505  042525  041516
 9612  045672  020105  041517  052503
 9613  045700  051122  042105     000
 9614  045705     104  044522  042526  MSGB02: .ASCIZ  /DRIVE HAS NOT RESPONDED TO PORT REQUEST/
 9615  045712  044040  051501  047040
 9616  045720  052117  051040  051505
 9617  045726  047520  042116  042105
 9618  045734  052040  020117  047520
 9619  045742  052122  051040  050505
 9620  045750  042525  052123     000
 9621  045755     104  044522  042526  MSGB01: .ASCIZ  /DRIVE HAS BECOME NON-EXISTENT/
 9622  045762  044040  051501  041040
 9623  045770  041505  046517  020105
 9624  045776  047516  026516  054105
 9625  046004  051511  042524  052116
 9626  046012     000
 9627
 9628                                  .SBTTL  DATA HEADER (DT) MESSAGES
 9629
 9630  046013     105  051122  050040  DH1:    .ASCIZ  /ERR PC  RPAS/
 9631  046020  020103  051040  040520
 9632  046026  000123
 9633  046030  051105  020122  041520  DH2:    .ASCIZ  /ERR PC  DRIVE   RPAS    RPDS1   RPER1   RPER2   RPER3/
 9634  046036  020040  051104  053111
 9635  046044  020105  020040  050122
```

```
 9636  046052  051501  020040  020040
 9637  046060  050122  051504  020061
 9638  046066  020040  050122  051105
 9639  046074  020061  020040  050122
 9640  046102  051105  020062  020040
 9641  046110  050122  051105  000063
 9642  046116  042524  052123  020040  DH3:    .ASCIZ  /TEST    ERR PC  ADDRESS DATA/
 9643  046124  020040  051105  020122
 9644  046132  041520  020040  042101
 9645  046140  051104  051505  020123
 9646  046146  040504  040524     000
 9647  046153     124  051505  020124  DH4:    .ASCIZ  /TEST    ERR PC  ADDRESS GDDATA  BDDATA/
 9648  046160  020040  042440  051122
 9649  046166  050040  020103  040440
 9650  046174  042104  042522  051523
 9651  046202  043440  042104  052101
 9652  046210  020101  041040  042104
 9653  046216  052101  000101
 9654  046222  050122  051503  020061  DH10:   .ASCIZ  /RPCS1   ERR PC/
 9655  046230  020040  051105  020122
 9656  046236  041520     000
 9657  046241     104  044522  042526  DH11:   .ASCIZ  /DRIVE   ERR PC/
 9658  046246  020040  042440  051122
 9659  046254  050040  000103
 9660  046260  042524  052123  020040  DH12:   .ASCIZ  /TEST    ERR PC  TST PC  DRIVE   CYLNDR  TRACK   SECTOR/
 9661  046266  020040  051105  020122
 9662  046274  041520  020040  051524
 9663  046302  020124  041520  020040
 9664  046310  051104  053111  020105
 9665  046316  020040  054503  047114
 9666  046324  051104  020040  051124
 9667  046332  041501  020113  020040
 9668  046340  042523  052103  051117
 9669  046346     000
 9670  046347     107  041504  046131  DH12A:  .ASCIZ  /GDCYL   GDTRK   GDSCTR  BDCYL   BDTRK   BDSCTR/
 9671  046354  020040  043440  052104
 9672  046362  045522  020040  043440
 9673  046370  051504  052103  020122
 9674  046376  041040  041504  046131
 9675  046404  020040  041040  052104
 9676  046412  045522  020040  041040
 9677  046420  051504  052103  000122
 9678  046426  042107  040504  020124  DH13A:  .ASCIZ  /GDDAT   BDDAT   WRDCNT  GDADR   BDADR/
 9679  046434  020040  042102  040504
 9680  046442  020124  020040  051127
 9681  046450  041504  052116  020040
 9682  046456  042107  042101  020122
 9683  046464  020040  042102  042101
 9684  046472  000122
 9685  046474  042524  052123  020040  DH17:   .ASCIZ  /TEST    ERR PC  DRIVE   RPCS1   RPDS1   RPER1   RPER2   RPER3/
 9686  046502  020040  051105  020122
 9687  046510  041520  020040  051104
 9688  046516  053111  020105  020040
 9689  046524  050122  051503  020061
 9690  046532  020040  050122  051504
 9691  046540  020061  020040  050122
```

```
 9692  046546  051105  020061  020040
 9693  046554  050122  051105  020062
 9694  046562  020040  050122  051105
 9695  046570  000063
 9696  046572  042524  052123  020040  DH21:   .ASCIZ  /TEST    ERR PC  TST PC  DRIVE   CYLNDR  TRACK/
 9697  046600  020040  051105  020122
 9698  046606  041520  020040  051524
 9699  046614  020124  041520  020040
 9700  046622  051104  053111  020105
 9701  046630  020040  054503  047114
 9702  046636  051104  020040  051124
 9703  046644  041501  000113
 9704  046650  042107  040504  020124  DH21A:  .ASCIZ  /GDDAT   BDDAT   WRDCNT  SECTOR/
 9705  046656  020040  042102  040504
 9706  046664  020124  020040  051127
 9707  046672  041504  052116  020040
 9708  046700  042523  052103  051117
 9709  046706     000
 9710  046707     124  051505  020124  DH23:   .ASCIZ  /TEST    ERR PC  DRIVE   CYLNDR  RPCS1   RPCS2   RPDS1/
 9711  046714  020040  042440  051122
 9712  046722  050040  020103  042040
 9713  046730  044522  042526  020040
 9714  046736  041440  046131  042116
 9715  046744  020122  051040  041520
 9716  046752  030523  020040  051040
 9717  046760  041520  031123  020040
 9718  046766  051040  042120  030523
 9719  046774     000
 9720  046775     122  042520  030522  DH23A:  .ASCIZ  /RPER1   RPER2   RPER3   RPCA    RPCC/
 9721  047002  020040  051040  042520
 9722  047010  031122  020040  051040
 9723  047016  042520  031522  020040
 9724  047024  051040  041520  020101
 9725  047032  020040  051040  041520
 9726  047040  000103
 9727  047042  042524  052123  020040  DH11:   .ASCIZ  /TEST    ERR PC  TST PC  DRIVE/
 9728  047050  020040  051105  020122
 9729  047056  041520  020040  051524
 9730  047064  020124  041520  020040
 9731  047072  051104  053111  000105
 9732  047100  042524  052123  020040  DH42:   .ASCIZ  /TEST    ERR PC  TST PC  DRIVE   RPCS1   RPCS2   RPDS1/
 9733  047106  020040  051105  020122
 9734  047114  041520  020040  051524
 9735  047122  020124  041520  020040
 9736  047130  051104  053111  020105
 9737  047136  020040  050122  051503
 9738  047144  020061  020040  050122
 9739  047152  051503  020062  020040
 9740  047160  050122  051504  000061
 9741  047166  050122  051105  020061  DH43A:  .ASCIZ  /RPER1   RPER2   RPER3/
 9742  047174  020040  050122  051105
 9743  047202  020062  020040  050122
 9744  047210  051105  000063
 9745  047214  050122  051503  020061  DH44A:  .ASCIZ  /RPCS1   RPCS2   RPDS1   RPCC    RPCA    RPDA/
 9746  047222  020040  050122  051503
 9747  047230  020062  020040  050122
```

```
 9748  047236  051504  020061  020040
 9749  047244  050122  041503  020040
 9750  047252  020040  050122  040503
 9751  047260  020040  020040  050122
 9752  047266  040504     000
 9753  047271     122  042520  030522  DH44B:  .ASCIZ  /RPER1   RPER2   RPER3/
 9754  047276  020040  051040  042520
 9755  047304  031122  020040  051040
 9756  047312  042520  031522     000
 9757  047317     122  042520  030522  DH45A:  .ASCIZ  /RPER1   RPER2   RPER3   RPWC    RPBA    RPDB/
 9758  047324  020040  051040  042520
 9759  047332  031122  020040  051040
 9760  047340  042520  031522  020040
 9761  047346  051040  053520  020103
 9762  047354  020040  051040  041120
 9763  047362  020101  020040  051040
 9764  047370  042120  000102
 9765
 9766
 9767                                  .EVEN
 9768
 9769                                  .SBTTL  DATA TABLE (DT)
 9770
 9771  047374  001116  001170          DT1:    .WORD   $ERRPC,$REG3
 9772  047400  001116  001164  001170  DT2:    .WORD   $ERRPC,$REG1,$REG3,RPERRS,RPERRS+2,RPERRS+4,RPERRS+6
 9773  047406  034464  034466  034470
 9774  047414  034472
 9775  047416  001176  001116  041654  DT3:    .WORD   $TMPO,$ERRPC,RD.ADR,RD.WRD
 9776  047424  041656
 9777  047426  001176  001116  042100  DT4:    .WORD   $TMPO,$ERRPC,WRT.ADR,WRT.WD,RD.WRD
 9778  047434  042076  041656
 9779  047440  001176  001116  001164  DT5:    .WORD   $TMPO,$ERRPC,$REG1,$REG5,RPERRS,RPERRS+2,RPERRS+4,RPERRS+6
 9780  047446  001174  034464  034466
 9781  047454  034470  034472
 9782  047460  001366  001116          DT10:   .WORD   RH.ADR,$ERRPC
 9783  047464  001166  001116          DT11:   .WORD   $REG2,$ERRPC
 9784  047470  001176  001116  001162  DT12:   .WORD   $TMPO,$ERRPC,$REC0,CHKDRV,CYL.DS,TRK.DS,SEC.DS
 9785  047476  001254  001270  001274
 9786  047504  001272
 9787  047506  001270  001274  001272  DT12A:  .WORD   CYL.DS,TRK.DS,SEC.DS,CYL.RD,TRK.RD,SEC.RD
 9788  047514  001262  001264  001266
 9789  047522  001176  001116  001162  DT13:   .WORD   $TMPO,$ERRPC,$REG0,CHKDRV,CYL.DS,TRK.DS,SEC.DS
 9790  047530  001254  001270  001274
 9791  047536  001272
 9792  047540  001124  001126  001172  DT13A:  .WORD   $GDDAT,$BDDAT,$REG4,$GDADR,$BDADR
 9793  047546  001120  001122
 9794  047552  001176  001116  001254  DT17:   .WORD   $TMPO,$ERRPC,CHKDRV,RP.REG,RP.REG+12,RP.REG+14,RP.REG+40,RP.REG+42
 9795  047560  004204  004216  004220
 9796  047566  004244  004246
 9797  047572  001176  001116  001162  DT21:   .WORD   $TMPO,$ERRPC,$REG0,CHKDRV,CYL.DS,TRK.DS
 9798  047600  001254  001270  001274
 9799  047606  001164  001126  001172  DT21A:  .WORD   $REG1,$BDDAT,$REG4,$REG1
 9800  047614  001164
 9801  047616  001176  001116  001254  DT23:   .WORD   $TMPO,$ERRPC,CHKDRV,CYL.DS,RP.REG,RP.REG+10,RP.REG+12
 9802  047624  001270  004204  004214
 9803  047632  004216
```

```
9804  047634  004220  004244  004246  DT23A:  .WORD   RP.REG+14,RP.REG+40,RP.REG+42,RP.REG+34,RP.REG+36
9805  047642  004240  004242
9806  047646  001176  001116  001162  DT41:   .WORD   $TMPO,$ERRPC,$REGO,CHKDRV
9807  047654  001254
9808  047656  001176  001116  001162  DT42:   .WORD   $TMPO,$ERRPC,$REGO,CHKDRV,RP.REG,RP.REG+10,RP.REG+12
9809  047664  001254  004204  004214
9810  047672  004216
9811  047674  001176  001116  001162  DT43:   .WORD   $TMPO,$ERRPC,$REGO,CHKDRV,RP.REG,RP.REG+10,RP.REG+12
9812  047702  001254  004204  004214
9813  047710  004216
9814  047712  004220  004244  004246  DT43A:  .WORD   RP.REG+14,RP.REG+40,RP.REG+42
9815  047720  001176  001116  001162  DT44:   .WORD   $TMPO,$ERRPC,$REGO,CHKDRV,CYL.DS,TRK.DS,SEC.DS
9816  047726  001254  001270  001274
9817  047734  001272
9818  047736  004204  004214  004216  DT44A:  .WORD   RP.REG,RP.REG+10,RP.REG+12,RP.REG+36,RP.REG+34,RP.REG+06
9819  047744  004242  004240  004212
9820  047752  004220  004244  004246  DT44B:  .WORD   RP.REG+14,RP.REG+40,RP.REG+42
9821  047760  001176  001116  001162  DT45:   .WORD   $TMPO,$ERRPC,$REGO,CHKDRV,CYL.DS,TRK.DS,SEC.DS
9822  047766  001254  001270  001274
9823  047774  001272
9824  047776  004204  004214  004216  DT45A:  .WORD   RP.REG,RP.REG+10,RP.REG+12,RP.REG+36,RP.REG+34,RP.REG+06
9825  050004  004242  004240  004212
9826  050012  004220  004244  004246  DT45B:  .WORD   RP.REG+14,RP.REG+40,RP.REG+42,RP.REG+2,RP.REG+4,RP.REG+22
9827  050020  004206  004210  004226
9828
9829                                  .SBTTL  DATA FORMAT (DF) TABLE
9830
9831  050026  000001                  DF1:    .WORD   1               ;NUMBER OF DATA HEADERS
9832  050030     002                          .BYTE   2               ;NUMBER OF WORDS IN DATA TABLE
9833  050031     000                          .BYTE   0               ;ALL 3 NUMBERS ARE OCTAL
9834
9835  050032  000001                  DF2:    .WORD   1
9836  050034     007                          .BYTE   7
9837  050035     000                          .BYTE   0
9838
9839  050036  000001                  DF3:    .WORD   1
9840  050040     004                          .BYTE   4
9841  050041     000                          .BYTE   0
9842
9843  050042  000001                  DF4:    .WORD   1
9844  050044     005                          .BYTE   5
9845  050045     000                          .BYTE   0
9846
9847  050046  000001                  DF10:   .WORD   1
9848  050050     002                          .BYTE   2
9849  050051     000                          .BYTE   0
9850
9851  050052  000001                  DF11:   .WORD   1
9852  050054     002                          .BYTE   2
9853  050055     000                          .BYTE   0
9854
9855  050056  000002                  DF12:   .WORD   2               ;2 DH'S TO BE TYPED
9856  050060     007                          .BYTE   7               ;7 DATA WORDS FOLLOW THE 1ST DH
9857  050061     160                          .BYTE   160             ;WORDS 1-4 ARE OCTAL 5-7 ARE DECIMAL
9858  050062  046347                          .WORD   DH12A           ;ADDRESS OF 2ND DH
9859  050064     006                          .BYTE   6               ;6 DATA WORDS FOLLOW THE 2ND DH
```

```
 9860  050065     000                        .BYTE   0               ;ALL WORDS ARE OCTAL
 9861
 9862  050066  000002             DF13:      .WORD   2
 9863  050070     007                        .BYTE   7
 9864  050071     160                        .BYTE   160
 9865  050072  046426                        .WORD   DH13A
 9866  050074     005                        .BYTE   5
 9867  050075     004                        .BYTE   4               ;WORD 3 IS DECIMAL
 9868
 9869  050076  000000             DF14:      .WORD   0
 9870  050100     005                        .BYTE   5
 9871  050101     004                        .BYTE   4               ;WORD 3 IS DECIMAL
 9872
 9873  050102  000001             DF17:      .WORD   1
 9874  050104     010                        .BYTE   ^D8
 9875  050105     000                        .BYTE   0
 9876
 9877  050106  000002             DF21:      .WORD   2
 9878  050110     006                        .BYTE   6
 9879  050111     060                        .BYTE   60
 9880  050112  046650                        .WORD   DH21A
 9881  050114     004                        .BYTE   4
 9882  050115     014                        .BYTE   14
 9883
 9884  050116  000000             DF22:      .WORD   0
 9885  050120     004                        .BYTE   4
 9886  050121     014                        .BYTE   14
 9887
 9888  050122  000002             DF23:      .WORD   2
 9889  050124     007                        .BYTE   7
 9890  050125     010                        .BYTE   10              ;WORD 4 IS DECIMAL
 9891  050126  046775                        .WORD   DH23A
 9892  050130     005                        .BYTE   5
 9893  050131     000                        .BYTE   0
 9894
 9895
 9896  050132  000001             DF41:      .WORD   1
 9897  050134     004                        .BYTE   4
 9898  050135     000                        .BYTE   0
 9899
 9900  050136  000001             DF42:      .WORD   1
 9901  050140     007                        .BYTE   7
 9902  050141     000                        .BYTE   0
 9903
 9904  050142  000002             DF43:      .WORD   2
 9905  050144     007                        .BYTE   7
 9906  050145     000                        .BYTE   0
 9907  050146  047166                        .WORD   DH43A
 9908  050150     003                        .BYTE   3
 9909  050151     000                        .BYTE   0
 9910
 9911  050152  000003             DF44:      .WORD   3
 9912  050154     007                        .BYTE   7
 9913  050155     160                        .BYTE   160
 9914  050156  047214                        .WORD   DH44A
 9915  050160     006                        .BYTE   6
```

```
 9916  050161    000                          .BYTE   0
 9917  050162  04727i                         .WORD   DH44B
 9918  050164    003                          .BYTE   3
 9919  050165    000                          .BYTE   0
 9920
 9921  050166  000003                 DF45:   .WORD   3
 9922  050170    007                          .BYTE   7
 9923  050171    160                          .BYTE   160
 9924  050172  047214                         .WORD   DH44A
 9925  050174    006                          .BYTE   6
 9926  050175    000                          .BYTE   0
 9927  050176  047317                         .WORD   DH45A
 9928  050200    006                          .BYTE   6
 9929  050201    000                          .BYTE   0
 9930
 9931                                 .EVEN
 9932          050202                 BUFFER=.
 9933
 9934  050202  005015  041412  051132 TITLE:  .ASCII  <CR><LF><LF>/CZRJAD/<CR><LF>
 9935  050210  040512  006504    012
 9936  050215    122  030120  027464         .ASCIZ  @RP04/5/6 MECHANICAL & READ-WRITE TEST@<CR><LF><LF>
 9937  050222  027465  020066  042515
 9938  050230  044103  047101  041511
 9939  050236  046101  023040  051040
 9940  050244  040505  026504  051127
 9941  050252  052111  020105  042524
 9942  050260  052123  005015  000012
 9943  050266  005015  047524  052040 LOADRV: .ASCII  <CR><LF>/TO TEST DRIVE 0 REPLACE THE 'XXDP' PACK ON DRIVE 0/<CR><LF>
 9944  050274  051505  020124  051104
 9945  050302  053111  020105  020060
 9946  050310  042522  046120  041501
 9947  050316  020105  044124  020105
 9948  050324  054047  042130  023520
 9949  050332  050040  041501  020113
 9950  050340  047117  042040  044522
 9951  050346  042526  030040  005015
 9952  050354  044527  044124  040440         .ASCII  /WITH ANOTHER PACK, CLEAR MEMORY LOCATION 40, AND RESTART/<CR><LF>
 9953  050362  047516  044124  051105
 9954  050370  050040  041501  026113
 9955  050376  041440  042514  051101
 9956  050404  046440  046505  051117
 9957  050412  020131  047514  040503
 9958  050420  044524  047117  032040
 9959  050426  026060  040440  042116
 9960  050434  051040  051505  040524
 9961  050442  052122  005015
 9962  050446  044124  020105  051120         .ASCIZ  /THE PROGRAM/<CR><LF>
 9963  050454  043517  040522  006515
 9964  050462  000012
 9965  050464  005015  054523  052123 NOLOAD: .ASCIZ  <CR><LF>/SYSTEM HAS 16K MEMORY, 'XXDP' LOADER WILL BE OVERWRITTEN/<CR><L
 9966  050472  046505  044040  051501
 9967  050500  030440  045466  046440
 9968  050506  046505  051117  026131
 9969  050514  023440  054130  050104
 9970  050522  020047  047514  042101
 9971  050530  051105  053440  046111
```

```
 9972  050536  020114  042502  047440
 9973  050544  042526  053522  044522
 9974  050552  052124  047105  005015
 9975  050560  000012
```

```
 9976
 9977                                .EVEN
```

```
 9978
 9979                                   .SBTTL  ROUTINE TO SIZE MEMORY
 9980
 9981                                   ;;*****************************************************************
 9982                                   ;*CALL:
 9983                                   ;*      JSR     PC,$SIZE
 9984                                   ;*      RETURN
 9985                                   ;*$LSTAD WILL CONTAIN THE LAST AVAILABLE MEMORY LOCATION
 9986
 9987  050562  010046                   $SIZE:  MOV     R0,-(SP)        ;;SAVE R0 ON THE STACK
 9988  050564  010146                           MOV     R1,-(SP)        ;;SAVE R1 ON THE STACK
 9989  050566  013746  000004                   MOV     @#ERRVEC,-(SP)  ;;SAVE PRESENT ERROR VECTOR PS & PC
 9990  050572  013746  000006                   MOV     @#ERRVEC+2,-(SP)
 9991  050576  010600                           MOV     SP,R0           ;;SAVE THE STACK POINTER
 9992                                   ;;SET THE ERRVEC PS TO THE PRESENT PS
 9993  050600  104400                           TRAP                    ;;PUSH OLD PSW AND PC ON STACK
 9994  050602  012637  000006                   MOV     (SP)+,@#ERRVEC+2        ;;SAVE THE PSW IN @#ERRVEC+2
 9995  050606  012737  050626  000004           MOV     #2$,@#ERRVEC    ;;SET FOR TIMEOUT
 9996  050614  012701  020000                   MOV     #20000,R1       ;;FIRST ADDRESS
 9997  050620  005711                   1$:     TST     (R1)            ;;TEST THIS ADDRESS
 9998  050622  005721                           TST     (R1)+           ;;STEP TO NEXT ADDRESS
 9999  050624  000775                           BR      1$              ;;TRY ANOTHER
10000  050626  162701  000002           2$:     SUB     #2,R1           ;;DROP BACK
10001  050632  010006                           MOV     R0,SP           ;;RESTORE THE STACK
10002  050634  012637  000006                   MOV     (SP)+,@#ERRVEC+2 ;;RESTORE ERROR VECTOR
10003  050640  012637  000004                   MOV     (SP)+,@#ERRVEC
10004  050644  010137  050656                   MOV     R1,$LSTAD       ;;LAST ADDRESS
10005  050650  012601                           MOV     (SP)+,R1        ;;RESTORE R1
10006  050652  012600                           MOV     (SP)+,R0        ;;RESTORE R0
10007  050654  000207                           RTS     PC
10008  050656  000000                   $LSTAD: .WORD   0               ;;CONTAINS THE LAST ADDRESS
10009
10010                                   ;;*****************************************************************
10011                                   .SBTTL  GETADR - GET BUS ADDRESS AND VECTOR ADDRESS
10012                                   ;THIS ROUTINE IS USED TO ENSURE THE BUS ADDRESS
10013                                   ;OF THE RH11/RP04 IS SETUP TO READ THE PROPER VALUE.
10014                                   ;IT WILL ALSO READ THE ADDRESS FROM THE TTY IF
10015                                   ;REQUIRED.
10016                                   ;NOTE: THIS ROUTINE DESTROYS R0-R4
10017                                   ;CALL
10018                                   ;
10019                                   ;       JSR     PC,@#GETADR
10020                                   ;       RETURN
10021
10022  050660  005737  001226           GETADR: TST     @#BUSADR        ;INPUT FROM TTY REQUESTED?
10023  050664  001433                           BEQ     7$              ;NO--BRANCH
10024  050666  005037  001226                   CLR     @#BUSADR        ;YES--CLEAR THE REQUEST FLAG
10025  050672  012700  001366           1$:     MOV     #RH.ADR,R0      ;FIRST ADDRESS
10026  050676  012703  051046                   MOV     #MRPCS1,R3      ;"RPCS1="
10027  050702  011004                           MOV     (R0),R4         ;PRESENT RPCS1 ADDRESS
10028  050704  004737  051070                   JSR     PC,CLRBF        ;CLEAR INPUT BUFFER
10029  050710  004037  031634                   JSR     R0,@#GETNUM     ;GET NEW RPCS1
10030  050714  000402                           BR      2$              ;COMMA
10031  050716  000765                           BR      1$              ;PERIOD
10032  050720  000414                           BR      5$              ;DOUBLE PERIOD
10033  050722  010420                   2$:     MOV     R4,(R0)+        ;SAVE NEW RPCS1
```

```
10034  050724  012703  051057                    MOV     #MRHVEC,R3      ;"RHVEC="
10035  050730  011004                            MOV     (R0),R4         ;PRESENT RH11 VECTOR ADDRESS
10036  050732  004737  051070                    JSR     PC,CLRBF        ;CLEAR INPUT BUFFER
10037  050736  004037  031634                    JSR     R0,@#GETNUM     ;GET NEW RHVEC
10038  050742  000402                            BR      3$              ;COMMA
10039  050744  000752                            BR      1$              ;PERIOD
10040  050746  000401                            BR      5$              ;DOUBLE PERIOD
10041  050750  010420                    3$:     MOV     R4,(R0)+        ;SAVE NEW RHVEC
10042  050752  010410                    5$:     MOV     R4,(R0)         ;SAVE INPUT
10043  050754  013701  000004            7$:     MOV     @#ERRVEC,R1     ;SAVE THE ERROR VECTOR
10044  050760  012737  051014  000004            MOV     #8$,@#ERRVEC    ;SETUP FOR TRAP
10045  050766  005777  130374                    TST     @RH.ADR         ;CHECK FOR RH11/RP04
10046  050772  010137  000004                    MOV     R1,@#ERRVEC     ;RESTORE ERROR VECTOR
10047  050776  012700  001366                    MOV     #RH.ADR,R0      ;FIRST ADDRESS OF NEW PARAMETERS
10048  051002  012701  034632                    MOV     #RPADR,R1       ;FIRST ADDRESS OF WHERE TO PUT THEM
10049  051006  012021                            MOV     (R0)+,(R1)+     ;BUS ADDRESS
10050  051010  012021                            MOV     (R0)+,(R1)+     ;VECTOR ADDRESS
10051  051012  000207                            RTS     PC              ;RETURN
10052  051014  010137  000004            8$:     MOV     R1,@#ERRVEC     ;RESTORE ERROR VECTOR
10053  051020  022626                            CMP     (SP)+,(SP)+     ;CLEAN OFF THE STACK
10054  051022  104010                            ERROR   10              ;REPORT THE ERROR
10055  051024  005737  000042                    TST     @#42            ;IS THERE A MONITOR?
10056  051030  001720                            BEQ     1$              ;NO--GO ASK FOR ADDRESS
10057  051032  005037  001232                    CLR     @#DRVSEL        ;YES--NO DRIVES SELECTED
10058  051036  005037  017656                    CLR     @#$EOPCT        ;NO PASSES
10059  051042  000137  017502                    JMP     @#$EOP          ;GO TO END OF PROGRAM
10060
10061  051046  005015  050122  051503  MRPCS1: .ASCIZ   <CR><LF>/RPCS1=/
10062  051054  036461     000
10063  051057     015  051012  053110  MRHVEC: .ASCIZ   <CR><LF>/RHVEC=/
10064  051064  041505  000075
10065                                   ;*****************************************
10066                                   ;ROUTINE TO CLEAR INPUT BUFFER FOR NEW CS1 AND VEC
10067                                   ;
10068                                   ;
10069  051070  010046                    CLRBF:  MOV     R0,-(SP)        ;SAVE R0
10070  051072  010146                            MOV     R1,-(SP)
10071  051074  012701  000010                    MOV     #10,R1          ;COUNT 10 LOCATIONS
10072  051100  012700  022636                    MOV     #$TTYIN,R0      ;START ADR OF INPUT BUF
10073  051104  005020                       1$:  CLR     (R0)+
10074  051106  005301                            DEC     R1
10075  051110  001375                            BNE     1$
10076  051112  012601                            MOV     (SP)+,R1
10077  051114  012600                            MOV     (SP)+,R0        ;RESTORE R0
10078  051116  000207                            RTS     PC
10079          000001                    .END
```

```
ABS    = 000200          1834#
ACL    = 000040          1868#   1877#
ACTDRV  034550           8009#   8304*   8355*   8668*   8677*   8907
ACTSTR  034551           8015#   8909*   8924*
ACU    = 100000          1823#
AOE    = 001000          1740#
ATA    = 100000          1727#
ATABIT  034620           3335    7355    7361    8081#   8260    8367    8472    8831    8850    8858    8876    8983
AT0    = 000001          1760#
AT1    = 000002          1761#
AT2    = 000004          1762#
AT3    = 000010          1763#
AT4    = 000020          1764#
AT5    = 000040          1765#
AT6    = 000100          1766#
AT7    = 000200          1767#
A16    = 000400          1658#
A17    = 001000          1659#
BAI    = 000010          1676#
BITS     001424          2041#   3496    3530    3568    3611    3652    3732    3775    3825    3925    4082    4148    4252
                         4342    4440    4554    4625    4754    4800    4874    4963    7422    7453    7470    7523    7697
BIT0   = 000001          1633#
BIT00  = 000001          1623#   1633    2041    2057    3306    3323    3345    3398    6169
BIT01  = 000002          1622#   1632    2042    2058    3309    6495    6523    8347    8594
BIT02  = 000004          1621#   1631    2043    2059    6232    6495    6526
BIT03  = 000010          1620#   1630    2044    2060    3295    6532    8245    8790    9105
BIT04  = 000020          1619#   1629    2045    2061    6529    8825
BIT05  = 000040          1618#   1628    2046    2062    3968    3984    4060    4190    4204    4219    4224    4289    4313
                         4318    4377    4395    4412    4417    4475    4493    4510    4515    6532    6642    8171    8614
                         8805    8957
BIT06  = 000100          1617#   1627    2047    2063    6535    8269    8338    8412    8703    9166
BIT07  = 000200          1616#   1626    2048    2064    8269    8546    8689    8790    8805    8825    8854    9138
BIT08  = 000400          1615#   1625    2049    6523    8269    8970
BIT09  = 001000          1614#   1624    2050    5090    5715    6538    8243    8955
BIT1   = 000002          1632#
BIT10  = 002000          1613#   2051    5074    6181    6495    6526    7716    8596
BIT11  = 004000          1612#   2052    5722    6526    8218    8371    8389    8565    8877
BIT12  = 010000          1611#   2053    6495    6529    8213    8251    8269    8349    8383    8561    8580    8592    8605
                         8815    8817    9043    9100    9167
BIT13  = 020000          1610#   2054    5081    6495    6523    8332    8763    9046
BIT14  = 040000          1609#   2055    3979    4185    4199    4284    4372    4390    4470    4488    5697    6423    6495
                         6529    8344    8380    8750    8894    9010    9055    9059    9164
BIT15  = 100000          1608#   2056    6687    6688    8332    8344    8347    8349    8380    8383    8565    8594    8596
                         8703    8790    8805    8815    8825    8894    8955    8970    9010    9017
BIT2   = 000004          1631#   9017
BIT3   = 000010          1630#
BIT4   = 000020          1629#
BIT5   = 000040          1628#
BIT6   = 000100          1627#   8373
BIT7   = 000200          1626#
BIT8   = 000400          1625#
BIT9   = 001000          1624#
BPTVEC= 000014           1640#
BUFFER= 050202           2700    2724    2748    2772    3945    4572    4587    4600    4643    4653    4667    4788    4890
                         4914    4922    4930    4936    4944    6606*   6607    6609    6611    6612    6613    6686    6723
                         6866    6885    6886    6910    6999    7031    7074    7116    7133    7134    7151    9932#
```

```
BUSADR  001226          1978#  3161*  3164*  3167*  3170*  3174*  3177*  10022  10024*
BYPASS  001252          1987#  3396*  3510*  3546*  3581*  3624*  3665*  3745*  3788*  3839*  3952*  4099*  4564*
                        4635*  4789*  4885*  6477   6957   7103   7168
CALL.A  025150          3402   3404   3513   3858   4977   4982   6275#  6278   6621
CALL.B  025262          3514   3550   3586   3594   3628   3633   3675   3681   3697   3703   3751   3753   3797
                        3799   3801   3803   3805   3807   3880   4103   6306#  6309
CALL.C  025452          3549   4102   6349#  6352
CHKDRV  001254          1988#  3375*  3380*  3385   5008   6220   9784   9789   9794   9797   9801   9806   9808
                        9811   9815   9821
C11     036222          8402   8430#
C13     036330          8404   8453#
C14     036436          8393   8474#
C15     037014          8452   8473   8551#  8560
C16     037036          8490   8500   8502   8504   8556#
C17     037052          8334   8411   8442   8446   8450   8458   8467   8471   8482   8489   8495   8499   8513
                        8519   8523   8533   8544   8559   8561#  8663   8693   8842   9150
C17B    037100          8564   8567#  8844
C18     037152          8562   8570   8579#  8716   9002
CKSCTR  030126          4586   6907#
CKSWR = 104407          5064   5089   5696   5826#
CK.CHR  034104          7289   7339   7566   7603   7643   7769#  7806   7819   7830   7881
CK.DEC  034056          7747#  7777
CK.DIG  034160          7349   7587   7800#
CK.NUM  034344          7298   7619   7652   7864#
CK.OCT  034030          7402   7406   7426   7430   7729#  7866   7876
CLKSTA  001244          1983#  3367   3957   3987   4158   4262   4352   4450   6099*  6101*  6107*
CLOSE   033750          7687   7691   7700#
CLR   = 000040          1678#
CLRBF   051070          10028  10036  10069#
CLRBUF  030060          4583   6883#
CLRQUE  042664          3951   8153   8613   8980   9234#
CLSWDS  033650          7671   7675   7678#
CNTRLC  001224          1977#  3172*  3179*  3271   3319
COMMA   043557          3363   9397#
COUNT   027236          4208   4293   4382   4400   4480   4498   6702#
CR    = 000015          1548#  1973#  5001   5032   5269   5279   7256   7336   9351   9355   9372   9391   9398
                        9408   9416   9421   9428   9431   9438   9441   9447   9451   9453   9455   9934   9936
                        9943   9952   9962   9965   10061  10063
CRLF  = 000200          1549#  5240   5279
CSF   = 000002          1810#  1828#
CSU   = 000010          1812#  1830#
CYL.DS  001270          1995#  3971*  6283*  6314*  6357*  6406*  6436*  6462*  6616*  6654*  6973*  7087*  9784
                        9787   9789   9797   9801   9815   9821
CYL.RD  001262          1992#  6611*  9787
C.SWR   001220          1975#  3322   3325   3340   3398   3667   4172   4228   4892   6114   6124   6134   6169
                        6259   6337   6380   6478   6562   7258   7262   7263*
DATCMP  030566          4840   7030#
DCK   = 100000          1746#
DCL   = 000100          1869#  1878#
DCU   = 000001          1875#
DDISP = 177570          1555#  1946   3207
DECSEC  027776          6851#
DECSK   007202          3593#  3597
DELTA   001350          2016#  3943*  4033*  4034*  4035   4037*  4046
DE1   = 000040          1717#
DFF20 = 000002          1713#
```

```
DFLT    001664          2133#   6158
DF1     050026          2861    9831#
DF10    050046          2925    9847#
DF11    050052          2935    9851#
DF12    050056          2948    9855#
DF13    050066          2961    2982    9862#
DF14    050076          2969    2990    9869#
DF17    050102          3000    3010    9873#
DF2     050032          2871    2901    9835#
DF21    050106          3023    9877#
DF22    050116          3031    9884#
DF23    050122          3043    3055    9888#
DF3     050036          2881    9839#
DF4     050042          2891    9843#
DF41    050132          3088    9896#
DF42    050136          3098    9900#
DF43    050142          3110    9904#
DF44    050152          3125    9911#
DF45    050166          3140    3155    9921#
DH1     046013          2859    9630#
DH10    046222          2923    9654#
DH11    046241          2933    9657#
DH12    046260          2946    2959    2980    3123    3138    3153    9660#
DH12A   046347          9670#   9858
DH13A   046426          9678#   9865
DH17    046474          2998    3008    9685#
DH2     046030          2869    2899    9633#
DH21    046572          3021    9696#
DH21A   046650          9704#   9880
DH23    046707          3041    3053    9710#
DH23A   046775          9720#   9891
DH3     046116          2879    9642#
DH4     046153          2889    9647#
DH41    047042          3086    9727#
DH42    047100          3096    3108    9732#
DH43A   047166          9741#   9907
DH44A   047214          9745#   9914    9924
DH44B   047271          9753#   9917
DH45A   047317          9757#   9927
DIGB  = 000004          1714#
DISPLA  001142          1946#   3207*   3215*   3503*   3537*   3575*   3618*   3659*   3739*   3782*   3832*   3932*   4089*
                        4155*   4259*   4349*   4447*   4561*   4632*   4761*   4881*   4970*   5073*   5736*
DISPRE  000174          1509#   3215
DLT   = 100000          1688#
DL64  = 000020          1716#
DMD   = 000001          1750#
DORTI   027170          3966    4179    4277    4365    4463    6673#
DPB.A   004104          2695#   3391*   3397*   3400*   3401*   3403*   3506*   3836*   3854*   3861    3937*   3940*   3949
                        3970    3971    3981    4042*   4049*   4051    4053    4057    4975*   4976*   4980*   4981*   6277
                        6279    6283    6284    6285    6286    6294    6620*
DPB.B   004124          2719#   3392*   3507*   3508*   3509*   3540*   3542*   3544*   3579*   3580*   3582*   3587*   3588
                        3590*   3595*   3596    3622*   3623*   3625*   3632*   3663*   3664*   3674*   3680*   3696*   3702*
                        3743*   3744*   3750*   3752*   3786*   3787*   3796*   3798*   3800*   3802*   3804*   3806*   3835*
                        3837*   3842*   3852*   3854    3881*   3882    3884*   4093*   4095*   4097*   4108*   4109    6261*
                        6264*   6308    6310    6314    6315    6316    6317    6329    6332    6335
DPB.C   004144          2743#   3393*   3541*   3543*   3545*   4094*   4096*   4098*   4104*   4105    4107*   6262*   6265*
```

```
                    6351  6353  6357  6358  6359  6360  6372  6375  6378
DPINT   034524      7969# 8182  8185  8277* 8314  8734  8759  8884  8886* 8947  8985  8996  9004*
DPR   = 000400      1720#
DPRQS   034534      7982# 8320  8406* 8737  8898* 8949  8987  8998  9013*
DRIVES  043521      3346  9391#
DRQ   = 004000      1783#
DRVACT  034474      4020* 7926# 8328  8507* 8554* 8573* 8584  8599* 8682* 8755  8788  8819* 8826*
                    8839  8849* 8958* 8964  8971*
DRVCAL  025642      4576  4580  4585  4595  4602  4648  4659  4675  4681  4821  4827  4837  4899
                    4919  4925  4933  4939  4947  6395# 6399
DRVCLR= 000111      1896# 4013
DRVCL1  025662      4026  6400# 6401
DRVINT  035062      8173  8207# 8316  8375  8867  8887  8962
DRVMSK  001256      1989# 3376* 3377  3381* 3389
DRVOK   006236      3378  3385#
DRVQUE  042762      8323  8336  9274#
DRVSEL  001232      1980# 3330* 3335* 3345* 3349  3372  3377  3389* 5002  7332* 7355* 7361* 10057*
DRVSTA  034504      3285  3333  3386  7940# 8163* 8164* 8165* 8166* 8176* 8208* 8217* 8266* 8271*
                    8310  8318  8342  8377  8381  8508* 8617* 8740  8748  8761  8813* 8820* 8832
                    8889  9005*
DRVTYP  034514      3288  3295  3306  3309  6232  7956# 8209* 8224* 8229* 8234* 8239* 8245* 8345
                    8618*
DRY   = 000200      1719#
DSWR  = 177570      1554# 1945  3206
DTADPB  004164      2767# 3394* 3870* 3873  3876* 3878* 3938* 3939* 3941* 3944* 3945* 4003* 4006
                    4009* 4011* 4012  4017* 4018  4024  4051  4053* 4188  4191  4202  4205  4217
                    4220  4225  4272* 4273* 4279  4287  4290  4297* 4298  4302* 4303  4311  4314
                    4319  4375  4378  4393  4396  4410  4413  4418  4473  4476  4491  4494  4508
                    4511  4516  4568* 4569* 4570* 4571* 4572* 4575* 4577* 4584* 4591* 4592* 4593*
                    4594* 4598* 4599* 4600* 4601* 4639* 4640* 4641* 4642* 4643* 4647* 4649* 4653*
                    4657* 4658* 4660* 4665* 4667* 4670* 4674* 4676* 4677* 4680* 4682  4787* 4788*
                    4803* 4804* 4805* 4810* 4820* 4826* 4836* 4888* 4889* 4890* 4891* 4895* 4900*
                    4901  4903* 4904  4906* 4911  4912  4913* 4915* 4921* 4922* 4927* 4928* 4929*
                    4930* 4935* 4936* 4941* 4942* 4943* 4944* 6398  6400  6406  6407  6408  6409
                    6419  6421  6427* 6430  6432  6436  6437  6438  6439  6447* 6449  6451  6455
                    6457  6462  6463  6464  6465  6473  6643* 6644* 6645* 6648  6650  6654  6655
                    6656  6657  6973  6974  7000  7032  7087  7088  7089  7238* 7239* 7240*
DTE   = 010000      1743#
DTSY  = 000200      1756#
DTUW    034616      4019* 8075# 8159  8395  8451* 8574  8577* 8587  8600  8602* 8611* 8671  8683*
                    8952  8967  8978* 8989  9053
DT00  = 000001      1774#
DT01  = 000002      1775#
DT02  = 000004      1776#
DT03  = 000010      1777#
DT04  = 000020      1778#
DT05  = 000040      1779#
DT06  = 000100      1780#
DT07  = 000200      1781#
DT08  = 000400      1782#
DT1     047374      2860  9771#
DT10    047460      2924  9782#
DT11    047464      2934  9783#
DT12    047470      2947  9784#
DT12A   047506      9787#
DT13    047522      2960  2981  9789#
```

```
DT13A   047540          2968    2989    9792#
DT17    047552          2999    3009    9794#
DT2     047400          2870    2900    9772#
DT21    047572          3022    9797#
DT21A   047606          3030    9799#
DT23    047616          3042    3054    9801#
DT23A   047634          9804#
DT3     047416          2880    9775#
DT4     047426          2890    9777#
DT41    047646          3087    9806#
DT42    047656          3097    9808#
DT43    047674          3109    9811#
DT43A   047712          9814#
DT44    047720          3124    9815#
DT44A   047736          9818#
DT44B   047752          9820#
DT45    047760          3139    3154    9821#
DT45A   047776          9824#
DT45B   050012          9826#
DT5     047440          9779#
DVA   = 004000          1708#
ECM   = 000100          1737#
ECI   = 004000          1851#
EMPTYQ  042742          8571    8616    8704    8827    8896    9018    9259#
EMTVEC= 000030          1643#   3191*   3192*
EM1     044333          2858    9475#
EM10    044561          2922    9501#
EM11    044637          2932    9509#
EM12    044674          2945    9514#
EM13    044721          2958    2979    3020    9518#
EM17    044746          2997    9522#
EM2     044376          2868    9481#
EM20    045000          3007    9527#
EM23    045047          3040    9534#
EM24    045076          3052    9538#
EM3     044434          2878    9486#
EM4     044471          2888    9491#
EM41    045136          3085    3095    3107    3122    3137    9544#
EM46    045162          3152    9548#
EM5     044525          2898    9496#
ERINDX  026364          6287    6318    6361    6410    6440    6466    6517#   6658
ERR   = 040000          1726#
ERRVEC= 000004          1636#   3204    3205*   3216*   5702    5703*   5705*   5708*   6068*   6069*   6074*   6094    9989
                        9990    9994*   9995*   10002*  10003*  10043   10044*  10046*  10052*
ERR.CT  001364          2022#   3430    4563    4634    4763    4883    6475    6955    6986    7101    7160
ES.SAV  043122          8719    8770    8774    9051    9111    9328#
EXIT0   006630          3510    3515#
EXIT1   007006          3546    3551#
EXIT10  012136          3952    4032    4063    4065#
EXIT11  012356          4099    4111#
EXIT12  013114          4164    4169    4232    4236#
EXIT13  013560          4268    4270    4324#
EXIT14  014276          4358    4360    4423#
EXIT15  015014          4456    4458    4521#
EXIT16  015412          4564    4607#
EXIT17  016032          4635    4687#
```

```
EXIT2   007224          3581    3598#
EXIT20  016552          4789    4798    4844#
EXIT21  017330          4885    4948#
EXIT22  017500          4985#
EXIT3   007420          3624    3637#
EXIT4   010006          3665    3715#
EXIT5   010206          3745    3757    3760#
EXIT6   010452          3788    3795    3810#
EXIT7   011032          3839    3883    3885#
EXT1    = 000001        1790#
EXT10   = 000010        1793#
EXT2    = 000002        1791#
EXT20   = 000020        1794#
EXT4    = 000004        1792#
EXT40   = 000040        1795#
FC      001510          2070#   3544    3582    3596    3625    3670    3680    3713    3746    3758    3789    3842    3843
                        3850    3940    3941    3942    4039    4097    4098    4107    4165    4272    4385    4483    4568
                        4640    4792    4888    4980    7215    7233
FEN     = 000200        1816#
FER     = 000020        1735#
FILBUF  030022          4567    4638    6864#
FILRAN  031104          4894    7116#
FMT22   = 010000        1852#   1916#   3397    6606
FS      001524          2076#   3507    3540    3579    3622    3663    3743    3786    4093    4094    4166    4641    4768
                        4782    4787
FT      001516          2073#   3508    3542    3580    3623    3664    3744    3787    3835    3884    3939    4095    4096
                        4167    4569    4802
F1      = 000002        1703#
F2      = 000004        1704#
F3      = 000010        1705#
F4      = 000020        1706#
F5      = 000040        1707#
GETADR  050660          3260    10022#
GETNUM  031634          7259    7281#   10029   10037
GETREG= 000141          1909#
GETREQ  043036          8368    8589    8744    8780    8841    8893    8954    8969    9007    9014    9295#
GETSWR  031542          3265    7251#
GNS     = ****** U      1508    5000    5006    5014    5031    5146    5163    5818    5819    5820    5821    5822    5824
                        5826    5827    5828    5829    5830    7255    7335    7368    7527    7633
GO      = 000001        1702#
GRV     = 000010        1715#
GTSWR   = 104406        3254    5824#
GTTST1  032130          7360    7365#   7414    7427    7438    7441    7467    7474    7483    7489    7498    7515
GTTST2  032214          7382#   7478
GTTST3  032556          7385    7389    7393    7397    7401    7460#
GTTST4  032560          7403    7424    7458    7461#
GTTST5  032670          7462    7484#
GTTST6  032730          7486    7495#
GT.PRM  031772          3327    7331#
GT.PR1  031774          7332#   7340    7341    7342    7343    7344    7350    7351
GT.PR2  032124          7362#   7377    7494    7503    7696
HCE     = 000200        1738#
HCI     = 002000        1850#
HCRC    = 000400        1739#
HT      = 000011        1546#   5238    5279
IAE     = 002000        1741#
```

```
IC      001514          2072#   3587    3595    3629    3690    3712    3754    3755    3808    3943    4030    4035    4037
                        4104    4108    6838
IE    = 000100          1656#
ILF   = 000001          1731#
ILR   = 000002          1732#
INCCYL  027746          4797    6836#
INCSK   007140          3585#   3589
INCTRK  027716          4841    6819#
IOTVEC= 000020          1641#   3189*   3190*
IR    = 000100          1679#
ISR     037544          3986    4227    4321    4420    4518    6640    8168    8668#
IT      001522          2075#   6821
ITEM41  004546          3078#   5130
IXE   = 004000          1820#   1838#
KIPARO= ****** U        9985
LA      037406          8399    8633#
LACNT   034562          8034#   8641*   8642    8659*
LC      001512          2071#   3509    3545    3588    3590    3635    3691    3693    3702    3747    3756    3790    3843
                        3848    4031    4033    4039    4105    4109    4298    4367    4465    4806    4904    4975    6836
                        6839    6841    7228
LDCMD   025104          3395    6259#
LF    = 000012          1547#   1974#   5001    5032    5273    5279    7256    7336    9351    9355    9372    9391    9398
                        9408    9416    9421    9428    9431    9438    9441    9447    9451    9453    9455    9934    9936
                        9943    9952    9962    9965    10061   10063
LKS     001412          2034#   6106    6138*
LKV     001406          2033#   3963    3964    6136*   6137*
LOADRV  050266          3239    9943#
LODFLT  024470          3326    3329    6151#
LODPRM  024726          3501    3535    3573    3616    3657    3737    3780    3830    3930    4087    4153    4257    4347
                        4445    4559    4630    4759    4879    4968    6209#
LOP.CK  026324          6295    6330    6373    6420    6474    6491#
LPB     001422          2038#   5070
LPS     001420          2037#   5069    6070
LPTAVL  001230          1979#   5067    6067*   6071*
LP.AVL  024150          3262    6067#
LS      001526          2077#   3541    4767
LST   = 002000          1722#
LT      001520          2074#   3543    3882    6819    6822    6824
MABOVE  044251          6783    9462#
MBELOW  044222          6767    9458#
MCLK  = 000002          1751#
MCPE  = 020000          1661#
MCPEMX  034630          8093#   9033    9081
MHS   = 001000          1818#   1836#
MINX  = 000004          1752#
MNDLTA  034644          8109#   8657
MOH   = 020000          1784#
MOL   = 010000          1724#
MPE   = 000400          1681#
MRD   = 000020          1754#
MRHVEC  051057          10034   10063#
MRPCS1  051046          10026   10061#
MSE   = 000020          1813#
MSGAVG  044206          6788    9455#
MSGB01  045755          2829    9621#
MSGB02  045705          2828    9614#
```

```
MSGB03  045635          2827    9607#
MSGB04  045615          2826    9604#
MSGB05  045545          2825    9597#
MSGB06  045501          2824    9590#
MSGB08  045437          2823    9584#
MSGB09  045400          2822    9578#
MSGB10  045355          2821    9574#
MSGB11  045327          2820    9570#
MSGB12  045305          2819    9566#
MSGB13  045254          2818    9561#
MSGB14  045212          2817    9555#
MSGFCP  043156          2122    9341#
MSGLCP  043162          2123    9342#
MSGMAX  044177          6772    9453#
MSGMIN  044170          6756    9451#
MSGNON  044315          6809    9469#
MSGNUM  044300          6804    9466#
MSG.CS  043216          7257    9351#
MSG.EQ  043214          7559    7597    7638    9350#
MSG.FC  043150          2120    9339#
MSG.FS  043202          2128    9347#
MSG.FT  043171          2125    9344#
MSG.IC  043166          2124    9343#
MSG.IT  043177          2127    9346#
MSG.LC  043153          2121    9340#
MSG.LS  043205          2129    9348#
MSG.LT  043174          2126    9345#
MSG.PA  043210          7596    9349#
MSG.R   043146          2119    9338#
MSG.SP  044330          3283    3284    3304    3410    5158    5170    5197    5204    6763    6779    6800    7555    7595
                        7600    9471#
MSGOUS  044215          6760    6771    6776    6787    6799    9457#
MSG10A  043742          2385    9421#
MSG10B  044011          2386    9428#
MSG11A  044026          2390    9431#
MSG11B  044074          2391    9438#
MSG12A  044111          2395    9441#
MSG12B  044153          2396    9447#
MSG7    043710          2375    2380    9416#
MSTCK = 000010          1753#
MWR   = 000040          1755#
MXDLTA  034642          8106#   8655
MXF   = 001000          1682#
MXLACT  034640          8103#   8642
MXSTAL  001362          2021#   3687    3709
MXWNDW  034646          8112#   8460
NBA   = 100000          1786#
NC1     001532          2079#   3942*   4030*   4031    4034    4042
NC2     001534          2080#
NED   = 010000          1685#
NEM   = 004000          1684#
NHS   = 002000          1819#   1837#
NOCLOK  043561          3371    9398#
NOLOAD  050464          3246    9965#
NONE    043552          3351    9396#
NOOP  = 000101          1892#
```

```
NOTPRS  043437            3293    9380#
NOTRP   043464            3291    9385#
NOTSAF  043454            3301    9383#
NOUSE   043317            3297    9363#
OCYL  = 100000            1871#   1881#
OFFSET= 000115            1898#
OFREV = 000200            1849#
OF100 = 000004            1845#
OF200 = 000010            1846#
OF25  = 000001            1843#
OF400 = 000020            1847#
OF50  = 000002            1844#
OF800 = 000040            1848#
OPE   = 020000            1879#
OPI   = 020000            1744#
OPNFLG  001240            1982#   7380*   7381*   7470*   7479    7481    7490    7492    7499    7501    7523
OPNPAT  033336            7554    7595#   7604    7613    7616    7620
OPNPRM  033134            7541    7546#   7550    7605    7607    7626
OPNTST  032762            7480    7482    7491    7493    7500    7502    7507#
OPNWDS  033460            7628#   7662
OPN.CT  032766            7509#   7511*   7512    7516    7518    7529
OPN.N1  033672            7591    7622    7685#
OPN.N2  033676            7578    7611    7673    7686#
OPN.X1  033710            7592    7623    7689#
OPN.X2  033714            7583    7615    7677    7690#
OPN.1   032772            7511#   7524    7688
OPN.2   033014            7510    7513    7516#
OPT     035742            8330    8365#   8699    8859    8899
OR    = 000200            1680#
PACK  = 000123            1901#
PAR   = 000010            1734#
PAT     001530            2078#   4800
PAT.PT  003044            2400#   7001    7034    7077
PAT0    003104            2400    2419#   6165    7629    7678
PAT1    003144            2401    2436#
PAT10   003604            2410    2589#
PAT11   003644            2411    2606#
PAT12   003704            2412    2623#
PAT13   003744            2413    2640#
PAT14   004004            2414    2657#
PAT15   004044            2415    2676#
PAT2    003204            2402    2453#
PAT3    003244            2403    2470#
PAT4    003304            2404    2487#
PAT5    003344            2405    2504#
PAT6    003404            2406    2521#
PAT7    003444            2407    2538#
PAT8    003504            2408    2555#   6164
PAT9    003544            2409    2572#   6167
PGE   = 002000            1683#
PGM   = 001000            1721#
PIP   = 020000            1725#
PIRQ  = 177772            1553#
PIRQVE= 000240            1647#
PKB     001402            2031#   4180*   4194*   4278*   4366*   4384*   4464*   4482*   6128*   6590*
PKC     001404            2032#   6706    6708    6709    6713    6715    6716    6720    6725
```

```
PKCS    001400          2030#   4181*   4184*   4196*   4198*   4212*   4281*   4283*   4306*   4369*   4371*   4387*   4389*
                        4405*   4467*   4469*   4485*   4487*   4503*   6100    6129*   6591*
PKV     001374          2029#   3960    3961    4178*   4276*   4364*   4462*   6126*   6127*   6588    6589*   6594*
PLU   = 020000          1822#   1839#
POPQUE  043060          8379    8430    8545    8853    9310#
PRM     001504          2068#   6218    6222*   6235*   6238*   7537    7546    7701
PRMLMT  001606          2105#   3871    4004    4605    4770    4781    6171*   6172*   6175*   6176*   6192    6194    6854
                        6872    6951    7117    7203    7553    7586
PRMMSG  001636          2119#   7556
PRMPT   001536          2083#   6178    6217    7535    7702
PRM0    002330          2083    2188#   6159
PRM1    002344          2084    2196#
PRM10   002544          2091    2274#
PRM11   002564          2092    2284#
PRM12   002606          2093    2295#
PRM13   002622          2094    2303#
PRM14   002636          2095    2311#
PRM15   002652          2096    2319#
PRM16   002666          2097    2327#
PRM17   002700          2098    2334#
PRM2    002372          2085    2209#
PRM20   002712          2099    2341#
PRM21   002744          2100    2356#
PRM22   002760          2101    2364#
PRM3    002414          2086    2220#
PRM4    002436          2087    2231#
PRM5    002460          2088    2242#
PRM6    002502          2089    2253#
PRM7    002524          2090    2264#
PR0   = 000000          1570#
PR1   = 000040          1571#
PR2   = 000100          1572#
PR3   = 000140          1573#
PR4   = 000200          1574#
PR5   = 000240          1575#
PR6   = 000300          1576#
PR7   = 000340          1577#   3273    3956    3978
PS    = 177776          1550#   1551    3263*   3273*   3318*   3956*   3973*   3978*   4025*   4213*   4307*   4406*   4504*
                        6586    6587*   6595*   6639*   8151    8152*   8181*   8189*   8302    8303*   8356*   8366    8415*
                        8648*
PSEL  = 002000          1660#
PSU   = 000001          1865#
PSW   = 177776          1551#
PTRN15  002742          2353#
PWRVEC= 000024          1642#
QCNT    042372          9176#   9235    9259*   9274    9276*   9296    9310*
QDRV0   042464          9187    9198    9207    9219#
QDRV1   042504          9188    9199    9208    9220#
QDRV2   042524          9189    9200    9209    9221#
QDRV3   042544          9190    9201    9210    9222#
QDRV4   042564          9191    9202    9211    9223#
QDRV5   042604          9192    9203    9212    9224#
QDRV6   042624          9193    9204    9213    9225#
QDRV7   042644          9194    9205    9214    9226#
QINPT   042402          9187#   9261    9278*   9279*   9280    9282*
QOUTPT  042422          9198#   9261*   9299    9312    9313*   9314    9316*
```

```
QSTART  042442          9207#   9241    9246    9282    9316
QSTOP   042444          9208#   9280    9314
QTERM = 042664          9215    9227#
RANADR  031360          4910    4920    4934    7197#
RANCK   031126          4926    4940    7131#
RANPAT  031324          4916    7118    7138    7179#
RAW   = 000020          1831#
RDCHR = 104410          5626    5827#
RDLIN = 104411          5828#   7287    7337    7370    7564    7601    7641
RDY   = 000200          1657#
RD.ADR  041654          9035*   9036*   9038#   9775
RD.RP   041630          8220    8255    8261    8493    8517    8531    8635    8649    8691    8714    8782    8792    8807
                        8942    8991    9033#   9088    9102    9108    9142
RD.RP1  041652          9037#   9067
RD.RP2  041776          9033*   9066#
RD.RP3  042002          9044    9058    9068#
RD.RP4  042006          9049    9070#
RD.WRD  041656          9039#   9040    9775    9777
READ  = 000171          1907#   4584    4836    4921    4935
READHD= 000173          1908#   3837    6261    6262    6332    6375
READIN= 000121          1900#
RECAL = 000107          1895#   3403    3506    6620    6645
RELEAS= 000113          1897#
RESREG= 104413          4192    4206    4221    4291    4315    4379    4397    4414    4477    4495    4512    5206    5830#
                        5890    6689    6874    6897    6959    7020    7071    7362    7514    7672    7676    7695    7718
                        8190    8350    8353    8416    8620    8676    8923    9151    9250
RESTAR  006216          3379#   3390    5043
RESVEC= 000010          1637#
RHVEC   001370          2028#
RH.ADR  001366          2027#   9782    10025   10045   10047
RMR   = 000004          1733#
RPADR   034632          3838    3950    6641    8098#   8170    8309    8433    8453    8474    8633    8670    8940    9035
                        9041    9045    9061    9094    9098    9131    9161    10048
RPAS  = 000016          8123#   8260*   8712    8831*   8858*   8876*   8982
RPBA  = 000004          4015*   8118#
RPCA  = 000034          3970*   4165*   4279*   4367*   4385*   4465*   4483*   6668*   8130#   8445    8457    8488
RPCC  = 000036          8131#   8636
RPCS1 = 000000          3972*   4013*   4024*   4368*   4386*   4466*   4484*   8116#   8212*   8218    8249    8371    8387
                        8449    8470    8512    8532    8543    8558    8569    8692    8715    8767    8799    8836    8992
                        9089
RPCS2 = 000010          3861*   3968*   3969*   3984*   3985*   3994*   4060*   4061*   4190*   4191*   4204*   4205*   4219*
                        4220*   4224*   4225*   4289*   4290*   4313*   4314*   4318*   4319*   4377*   4378*   4395*   4396*
                        4412*   4413*   4417*   4418*   4475*   4476*   4493*   4494*   4510*   4511*   4515*   4516*   6642*
                        8120#   8171*   8211*   8213    8322*   8434*   8454*   8475*   8561    8580    8592    8605    8614*
                        8634*   8690*   8781*   8861*   8875*   8941*   8957*   9042    9099    9132*   9138    9162*   9167
RPDA  = 000006          4016*   4168*   6667*   6851    8119#   8441    8466    8481
RPDB  = 000022          8125#   9136
RPDS1 = 000012          3974    3979    4185    4199    4284    4372    4390    4470    4488    8121#   8243    8256    8373
                        8783    8808    8862    8943
RPDT  = 000026          8127#   8221
RPEC1 = 000044          8134#
RPEC2 = 000046          8135#   9146
RPERRS  034464          7918#   8154    8750    8753    8763    8862*   8863*   8864*   8866*   9772    9779
RPER1 = 000014          6324    6326    6367    6369    8122#   8262    8793    8863    9103
RPER2 = 000040          8132#   8864
RPER3 = 000042          8133#   8866
```

```
RPINIT  034650           3267    8150#
RPLA  = 000020           3862    3995    8124#   8650
RPMR  = 000024           8126#
RPOF  = 000032           8129#   8253    8494    8498    8518    8522
RPSN  = 000030           8128#
RPT     001506           2069#   3504    3538    3576    3619    3660    3740    3783    3833    3933    4090    4156    4260
                         4350    4448    4562    4633    4762    4882    4971    7560    7593*   7598    7624*   7685*   7689*
RPTMR   041102           6142    8907#
RPVEC   034634           3966*   3986*   4179*   4227*   4277*   4321*   4365*   4420*   4463*   4518*   6640*   8099#   8167
                         8169    8303
RPWC  = 000002           4014*   8117#
RP.REG  004204           2707    2731    2755    2779    2795#   3413    6324    6326    6367    6369    6423    6851    9794
                         9801    9804    9808    9811    9814    9818    9820    9824    9826
RP04    035450           6276    6307    6350    6397    6429    6448    6454    6647    8302#
RP04B   043502           3305    9388#
RP05    043507           3308    9389#
RP06    043514           3311    9390#
RSTRT1  006162           3368    3370    3372#   3382
RSTRT2  006206           3377#   3383
RTC   = 000117           1899#
SAVCSW  001222           1976#   3321    3325*   7262*
SAVEFG  034572           3268*   8049#   8547    8696    8855
SAVREG= 104412           4187    4201    4216    4286    4310    4374    4392    4409    4472    4490    4507    5115    5829#
                         5864    6680    6864    6883    6907    6998    7030    7331    7507    7627    7700    8150    8305
                         8365    8579    8669    8910    9128    9234
SC      037750           8675    8700    8702    8708    8712#
SCTRWC= 177400           1915#   4592    4642    4906
SC1   = 000100           1796#
SC10  = 001000           1799#
SC11    040602           8752    8789    8846#
SC12    040672           8746    8757    8861#
SC13    040742           8736    8739    8872#
SC2   = 000200           1797#
SC20  = 002000           1800#
SC3     040020           8727#   8731
SC4     040024           8729#   8760    8772    8776    8824    8838    8845    8860    8900
SC5     040036           8728    8734#
SC6     040236           8756    8777#
SC6A    040346           8754    8801#
SC7     040474           8791    8804    8806    8816    8826#
SC8     040552           8768    8784    8794    8800    8809    8837    8839#   8871
SEARCH= 000131           1902#   4182    4195
SEC.DS  001272           1996#   6285*   6316*   6359*   6408*   6438*   6464*   6614*   6656*   7089*   9784    9787    9789
                         9815    9821
SEC.RD  001266           1994#   6613*   9787
SEEK  = 000105           1894#   3836    3937    3972    4280    4368    4386    4466    4484    4976    4981    6264    6265
SEEKFG  034574           3339*   3342*   8057#   8155    8397
SEKCNT  001346           2015#   4027    4029*   4038*
SEKTMR  001344           2014#   3967*   4055*
SELDRV= 000145           1911#
SERIAL  043671           3411    9412#
SETBUF  030476           3947    4815    6998#
SETFOR= 000143           1910#   3401    3403    3506    3836    3937    4976    4981
SETVEC  005434           3267#
SET.IE  042320           8184    8215    8340    8414    8619    8721    9160#
SKI   = 040000           1870#   1880#
```

```
SLASH   043364          7286    7563    9371#
SRCHWT  034546          8003#   8367*   8472*   8850
SRCHOO  027006          4162    4266    4356    4454    6638#
SRTDRV  005706          3272    3319#
SRTINT  005402          3259    3261#
SRVCLK  024456          6126    6136    6141#
STACK = 001100          1541#   3187    3379    3512    3548    3584    3592    3627    3631    3673    3695    3749    3793
                        3841    3857    3860    3955    3993    4100    4175    4271    4361    4459    4566    4574    4579
                        4582    4590    4597    4637    4646    4656    4673    4679    4790    4819    4825    4835    4897
                        4909    4918    4924    4932    4938    4946    4974
STALL   026502          3678    3684    3700    3706    4978    4983    6339    6382    6480    6561#
STALL0  001336          2012#   3671*   3679    3685    3686*   3687    3689*   3701    3707    3708*   3709    3711*
STALL1  001354          2018#   6340    6383
STALL2  001356          2019#   6481
STALL3  001360          2020#   4979    4984
START   004746          3173    3180#
START1  004674          1526    3170#
START2  004730          1528    3177#   3374    5467    5550
START3  004656          1530    3167#
START4  004712          1532    3174#
START5  004642          1534    3164#
START6  004626          1536    3161#
STATBL  004254          2817#   5142
STKLMT= 177774          1552#
STO     041174          8917    8938#
STO1    041224          8947#   8990    8995
STO2    041422          8946    8948    8950    8982#
STO3    041472          8984    8996#
STO5    041516          8944    8963    8993    9002#
STO6    041524          8986    9004#
STO7    041562          8988    9012#
STO8    041612          9011    9018#
STO9    041622          8981    8999    9001    9003    9009    9016    9020#
STRTMR  027172          4177    4275    4363    4461    6680#
STRT1A  004704          3166    3169    3172#
STRT2A  004740          3163    3176    3179#
ST.CLK  024212          3266    4226    4320    4419    4517    6093#
ST.LCL  024422          6108    6133#
ST.PCL  024360          6102    6123#
SVADR   001340          2013#   4911*   4912*   4927    4928    4941    4942
SVRH11  042202          3982    4058    4189    4203    4218    4288    4312    4376    4394    4411    4474    4492    4509
                        8549    8566    8597    8698    8705    8745    8796    8814    8857    8895    8956    9019    9128#
SVSTAT  001260          1990#   5139    6520*
SWR     001140          1945#   3185    3206*   3208    3214*   3252    4811    4816    4822    4830    4832    4838    5065
                        5074    5081    5086    5090    5470    5509    5564*   5697    5715    5722    6425    6491    6623
                        6745    6982    6984    6988    7099    7158    7162    7251
SWREG   000176          1510#   3214    3252    5470    5509    5532
SW0   = 000001          1605#
SW00  = 000001          1595#   1605    4830
SW01  = 000002          1594#   1604    4838    6984    7099    7162
SW02  = 000004          1593#   1603    4832
SW03  = 000010          1592#   1602    4813    4822
SW04  = 000020          1591#   1601    4813    4816    6425
SW05  = 000040          1590#   1600    6124    6134    6988
SW06  = 000100          1589#   1599    4172    4228    6114    6745
SW07  = 000200          1588#   1598    6259    7251
```

```
SW08  = 000400          1587#   1597    3340    5065
SW09  = 001000          1586#   1596    6982    7158
SW1   = 000002          1604#
SW10  = 002000          1585#
SW11  = 004000          1584#
SW12  = 010000          1583#   3667
SW13  = 020000          1582#   6562
SW14  = 040000          1581#   6337    6380    6478
SW15  = 100000          1580#   4892
SW2   = 000004          1603#
SW3   = 000010          1602#
SW4   = 000020          1601#
SW5   = 000040          1600#
SW6   = 000100          1599#
SW7   = 000200          1598#
SW8   = 000400          1597#
SW9   = 001000          1596#   6491    6623
SYSTAT  043370          3276    9372#
TAP   = 040000          1785#
TBITVE= 000014          1638#
TD      037610          8673    8682#
TDF   = 000040          1814#   1832#
TESTNG  043647          3405    9408#
TEST0   006614          3502    3511    3512#
TEST1   006772          3536    3547    3548#
TEST10  011242          3931    3953#   4041    4054
TEST11  012300          4088    4100#
TEST12  012546          4173    4175#
TEST13  013242          4271#
TEST14  013706          4361#
TEST15  014424          4459#
TEST16  015112          4560    4565    4566#
TEST17  015510          4631    4636    4637#
TEST2   007120          3574    3582#
TEST20  016236          4760    4790#
TEST21  017024          4880    4907    4908    4909#
TEST22  017420          4969    4973    4974#
TEST3   007336          3617    3625#
TEST4   007546          3658    3668    3670#
TEST5   010120          3738    3746#
TEST6   010320          3781    3789#
TEST7   010576          3831    3840    3841#
TICKMS  001246          1985#   6116*   6119*   6141
TICKUS  001250          1986#   6117*   6120*
TIMER   034576          4022*   8062#   8275*   8409*   8552*   8598*   8685*   8778*   8822*   8873*   8880*   8913    8915*
                        8973*   9000*   9006*   9012*
TIM.DN  001314          2004#   6688*   6705
TIM.PT  001332          2010#   6682    6723    6725*   6726*
TIM.UP  001276          1998#   6681    6687*   6702    6751
TITLE   050202          3230    9934#
TKVEC = 000060          1645#   5447*   5448*
TPB     001416          2036#   3223    5101
TPS     001414          2035#   3222    5100
TPVEC = 000064          1646#
TRAPVE= 000034          1644#   3193*   3194*
TRCKWC  001352          2017#   4571    4599    4889    6173*   6177*
```

```
TRE    = 040000          1662#
TRK.DS   001274          1997#   6284*   6315*   6358*   6407*   6437*   6463*   6615*   6655*   6974*   7088*   9784    9787
                         9789    9797    9815    9821
TRK.RD   001264          1993#   6612*   9787
TRK1   = 004000          1801#
TRK10  = 040000          1804#
TRK2   = 010000          1802#
TRK20  = 100000          1805#
TRK4   = 020000          1803#
TRNSWT   034544          4018*   7995#   8431*   8576*   8586    8603*   8612*   8687    8688*   8951    8966    8979*
TRTVEC= 000014           1639#
TSTNMS   001234          1981#   3369    3496    3530    3568    3611    3652    3732    3775    3825    3925    4082    4148
                         4252    4342    4440    4554    4625    4754    4874    4963    6156*   6157*   7358*   7359*   7378*
                         7379*   7384*   7388*   7392*   7396*   7400*   7422*   7453*   7463    7465
TSTPGM   035446          3162*   3165*   3168*   3171*   3175*   3178*   3228*   3231    8241    8290#
TST0     006470          3494#
TST1     006632          3498    3528#
TST10    011034          3827    3923#
TST10A   011500          3988    3991#
TST10B   012072          3965    4055#
TST11    012140          392?    4080#
TST12    012360          4084    4146#   4154
TST13    013116          4150    4160    4250#   4258
TST14    013562          4254    4264    4340#   4348
TST15    014300          4344    4354    4438#   4446
TST16    015016          4442    4452    4552#
TST17    015414          4556    4623#
TST2     007010          3532    3566#
TST20    016034          4627    4752#
TST21    016554          4756    4872#
TST22    017332          4876    4961#
TST3     007226          3570    3609#
TST4     007422          3613    3650#
TST5     010010          3654    3730#
TST6     010210          3734    3773#
TST7     010454          3777    3823#
TUF    = 000100          1815#   1833#
TWOMS    026564          4294    4383    4401    4481    4499    6586#
TYPDS  = 104405          5187    5822#
TYPE   = 104401          3230    3233    3239    3246    3261    3275    3276    3283    3284    3291    3293    3297    3299
                         3301    3303    3304    3312    3314    3346    3351    3363    3371    3405    3410    3411    3426
                         3429    4998    5004    5012    5029    5033    5076    5084    5134    5135    5144    5153    5154
                         5158    5161    5167    5168    5177    5190    5194    5197    5199    5201    5204    5243    5340
                         5416    5464    5476    5530    5531    5534    5545    5555    5566    5585    5633    5639    5644
                         5648    5653    5654    5656    5659    5663    5818#   5932    6752    6756    6760    6763    6767
                         6771    6772    6776    6779    6783    6787    6788    6799    6800    6804    6809    7253    7282
                         7286    7333    7366    7525    7534    7555    7557    7559    7563    7595    7596    7597    7600
                         7631    7638
TYPERR   020172          5083    5114#
TYPOC  = 104402          5017    5183    5533    5819#   7285    7599    7640
TYPON  = 104404          5821#
TYPOS  = 104403          3280    3358    3407    5009    5820#   7531
TYPTIM   027370          4230    4234    4322    4421    4519    6744#
T10      003014          2385#   4269    4323
T11      003024          2390#   4359    4422
T12      003034          2395#   4457    4520
```

```
T7A      002774            2375#   4171    4231
T7B      003004            2380#   4174    4235
ULDFLG   034552            8022#   8210*   8312    8327*   8509*   8572*   8604*   8742    8828    8830*   8846    8848*   8959*
                           8972*
UNLOAD=  000103            1893#
UNS    = 040000            1745#
UNTMSG   043410            9375#
UNTOFF   043416            3299    9376#
UNTON    043427            3303    9378#
UPE    = 020000            1686#
USE      043242            3233    9355#
US1    = 000001            1673#
US2    = 000002            1674#
US4    = 000004            1675#
UWR    = 000010            1867#
VERIFY   026644            6334    6377    6604#
VUF    = 000002            1866#
VU30   = 010000            1821#
VV     = 000100            1718#
WAO    = 000002            1876#
WCE    = 040000            1687#
WCEFLG   001334            2011#   4791*   4828    6396*   6460*
WCF    = 000040            1736#
WCU    = 000001            1809#   1827#
WLE    = 004000            1742#
WRCKD  = 000151            1903#   4577    4598    4657    4680    4826    4929    4943    6421    6447
WRCKHD = 000153            1904#
WRITE  = 000161            1905#   3938    4575    4591    4639    4674    4820    4891    4913    6427
WRL    = 004000            1723#
WRTHD  = 000163            1906#
WRT.AD   042100            9084*   9094*   9097#   9777
WRT.RP   042010            8248    8252    8386    8440    8444    8448    8456    8465    8469    8480    8487    8497    8511
                           8521    8542    8557    8568    8766    8798    8835    9081#
WRT.R1   042074            9095#   9115
WRT.R2   042164            9081*   9114#
WRT.R3   042172            9090    9101    9104    9110    9117#
WRT.R4   042176            9106    9119#
WRT.R5   042200            9118    9120#
WRT.WD   042076            9082*   9086    9093*   9096#   9777
WRU    = 000400            1817#   1835#
WSU    = 000004            1811#   1829#
$AUTOB   001134            1942#   3256*   5523    5680
$BDADR   001122            1937#   7081*   7146*   9792
$BDDAT   001126            1939#   6970*   7083*   7148*   9792    9799
$BELL    001210            1968#   3220    5076    5103    5476    5673
$CHARC   020774            5245*   5255*   5262    5271*   5276#
$CKSWR   021750            5509#   5826
$CMTAG   001100            1925#   3182    3183    3191    3197    3198
$CM1   = 000006            1957#   1958#   1959#   1960#   1961#   1962#   1963#
$CM2   = 000014            1957#   1958#   1959#   1960#   1961#   1962#   1963#
$CM3   = 000006            1955#   1957
$CM4   = 000003            1963#   1964#   1965#   1966#
$CNTLC   022662            5464    5545    5673#
$CNTLG   022674            5530    5675#
$CNTLU   022667            5555    5648    5674#
$CRLF    001215            1970#   3261    3275    3314    3429    5084    5103    5134    5153    5167    5177    5194    5201
```

```
                    5244    5279    5566    5653    5673    7534
$DBLK   021442      5382    5416    5424#
$DB2D   023372      5845    5864#
$DECVL  023552      5866    5912#
$DIV    023730      3851    4048    6000#   6792
$DOAGN  017742      5025    5035    5041#
$DTBL   021432      5385    5420#
$ENDAD  017732      1518    5037#
$ENDCT  017664      3347*   3356*   3366    5027#
$ENULL  017746      5033    5044#
$EOP    017502      3353    3396    4965    4997#   6498    10059
$EOPCT  017656      3366*   5024#   5028    10058*
$ERFLG  001103      1928#   5071*   5103    5687    5711    5713    5719*   5740    6475    6955    6986    7101    7160
$ERMAX  001115      1934#   3199*   3430*   3505*   3539*   3577*   3620*   3661*   3741*   3784*   3834*   3934*   4091*
                    4157*   4261*   4351*   4449*   4563*   4634*   4763*   4883*   4972*   5713    5735*   5740
$ERROR  017752      3191    5063#
$ERRPC  001116      1935#   5078*   5079*   5080    5103    9771    9772    9775    9777    9779    9782    9783    9784
                    9789    9794    9797    9801    9806    9808    9811    9815    9821
$ERRTB  004306      2844#   5130    5131
$ERTTL  001112      1932#   5016    5018*   5077*   5103
$ESCAP  001206      1967#   3198*   4169*   4270*   4360*   4458*   5093    5095    5103    5734*   6275*   6282*   6306*
                    6313*   6349*   6356*   6395*   6405*   6428*   6435*   6461*   6494*   6617*   6622*   6646*   6653*
                    6975*   6980*   7091*   7097*   7144*   8719    8770    8774    9051    9111    9320    9329    9330*
                    9332*
$FILLC  001156      1953#   5248    5279
$FILLS  001155      1952#   5279
$GDADR  001120      1936#   7082*   7147*   9792
$GDDAT  001124      1938#   7084*   7149*   9792
$GET42  017722      5034#
$GTSWR  022040      5531#   5824
$HD   = 000000      1459
$HINUM  023724      3195*   3846    4044    4886*   5954    5962    5967*   5972#   7131    7137*   7140*   7184    7200
$ICNT   001104      1929#   3264*   5726*   5727    5729*   5739
$INTAG  001135      1943#   5528*   5547    5567    5680
$ITEMB  001114      1933#   5080*   5103    5123
$LF     001216      1971#   5103    5279    5663    5673
$LONUM  023726      3196*   4887*   5953    5960    5966*   5973#   6565    7132    7136*   7139*   7183    7198    7199
$LPADR  001106      1930#   3200*   3511*   3547*   3578*   3621*   3662*   3742*   3785*   3840*   3935*   4092*   4161*
                    4265*   4355*   4453*   4565*   4636*   4764*   4884*   4907*   4973*   5717*   5732*   5737    5739
$LPERR  001110      1931#   3201*   3502*   3536*   3574*   3583*   3591*   3617*   3626*   3630*   3658*   3672*   3694*
                    3738*   3748*   3781*   3792*   3831*   3856*   3859*   3931*   3954*   3992*   4088*   4154*   4258*
                    4348*   4446*   4560*   4573*   4578*   4581*   4589*   4596*   4631*   4645*   4655*   4672*   4678*
                    4760*   4818*   4824*   4834*   4880*   4896*   4908*   4917*   4923*   4931*   4937*   4945*   4969*
                    5092    5717    5733*   5739    6493    6625
$LSTAD  050656      3244    10004*  10008#
$MAIL = ******  !J  3218    3252    5086    5232    5732
$MNEW   022712      5534    5678#
$MSWR   022701      5531    5676#
$MXCNT  023154      5730    5739#
$NULL   001154      1951#   5250    5279
$NWTST= 000001      3485#   3487    3517#   3519    3553#   3555    3600#   3602    3639#   3641    3717#   3719    3762#
                    3764    3812#   3814    3888#   3890    4067#   4069    4134#   4136    4239#   4241    4327#   4329
                    4425#   4427    4539#   4541    4609#   4611    4690#   4692    4847#   4849    4952#   4954
$OCNT   021222      5312*   5341*   5354#
$OMODE  021224      5307*   5311*   5316    5319*   5330*   5356#
$OVER   023140      5698    5718    5728    5736#
```

```
$PASS   001100          1926#   5021*   5022*   5044    5724    5740
$QUES   001214          1969#   5103    5279    5585    5656    5673
$RAND   023626          3845    4043    5949#   6564    7182    7197
$RDCHR  022312          5598#   5827
$RDDEC= ****** U        5829
$RDLIN  022402          5621#   5828
$RDOCT= ****** U        5829
$RDSZ = 000024          5614#
$REGAD  001160          1955#   3218
$REG0   001162          1957#   5117*   9784    9789    9797    9806    9808    9811    9815    9821
$REG1   001164          1958#   5118*   9772    9779    9799
$REG2   001166          1959#   5119*   9783
$REG3   001170          1960#   5120*   9771    9772
$REG4   001172          1961#   5121*   9792    9799
$REG5   001174          1962#   5122*   9779
$RESRE  023214          5774#   5830
$RTNAD  017744          5043#
$R2A  = ****** U        5831
$SAVRE  023156          5758#   5829
$SB2D   023336          5843#   6758    6765    6769    6774    6781    6785    6797    6802    7561    7636
$SCOPE  022724          3189    5695#
$SETUP= 000147          3161#   3188    3189    3191    3193    3195    3197    3198    3200    3249    5019    5064    5089
                        5097    5468    5473    5474    5504    5680    5696
$SIZE   050562          3243    9987#
$STUP = 177777          3161#
$SUPRS  023566          5924#   6759    6766    6770    6775    6782    6786    6798    6803    7562    7637
$SVLAD  023112          5706    5731#
$SVPC = 000200          1516#   1521
$SWR  = 167000          1449#   1459    1487    1488    1489    1490    1491    1492    1918#   1966    1967    1968    3197
                        3198    3200    3201    3506    3540    3578    3621    3662    3742    3785    3835    3935    4092
                        4158    4262    4352    4450    4564    4635    4764    4884    4973    4994    5020    5036    5042
                        5044    5055    5056    5057    5058    5059    5074    5081    5086    5090    5103    5688    5689
                        5690    5691    5692    5697    5709    5711    5712    5713    5720    5721    5722    5733    5736
                        5739
$SWRMK= 000000          5692
$TIMES  001204          1966#   3197*   3504*   3538*   3576*   3619*   3660*   3740*   3783*   3833*   3933*   4090*   4156*
                        4260*   4350*   4448*   4562*   4633*   4762*   4882*   4971*   5020*   5720*   5727    5730*   5739
$TKB    001146          1948#   5429    5449    5460    5485    5513    5540
$TKCNT  021452          5430#   5444*   5474    5491*   5605    5607*
$TKINT  021462          3248    5444#   5465    5526
$TKQEN= 021462          5434#   5499    5610
$TKQIN  021454          5431#   5445*   5446    5497*   5498*   5499    5501*
$TKQOU  021456          5432#   5446*   5608    5609*   5610    5612*
$TKQSR  021460          5433#   5445    5501    5612
$TKS    001144          1947#   5429    5450*   5481*   5483    5489*   5511    5527*   5537    5549*   5569*
$TKSRV  021532          5447    5460#
$TMP0   001176          1963#   5114*   9775    9777    9779    9784    9789    9794    9797    9801    9806    9808    9811
                        9815    9821
$TMP1   001200          1964#
$TMP2   001202          1965#
$TN   = 000023          1449#   1459    3485    3494    3496    3505    3506#   3517    3528    3530    3539    3540#   3553
                        3566    3568    3577    3578#   3600    3609    3611    3620    3621#   3639    3650    3652    3661
                        3662#   3717    3730    3732    3741    3742#   3762    3773    3775    3784    3785#   3812    3823
                        3825    3834    3835#   3888    3923    3925    3934    3935#   4067    4080    4082    4091    4092#
                        4134    4146    4148    4157    4158#   4239    4250    4252    4261    4262#   4327    4340    4342
                        4351    4352#   4425    4438    4440    4449    4450#   4539    4552    4554    4563    4564#   4609
```

```
                    4623    4625    4634    4635#   4690    4752    4754    4763    4764#   4847    4872    4874    4883
                    4884#   4952    4961    4963    4972    4973#   7413    7437    7512
$TNPWR  023502      5871    5872    5892#
$TPB    001152      1950#   3223*   5070*   5101*   5268*   5279
$TPFLG  001157      1954#   5226    5279
$TPS    001150      1949#   3222*   5069*   5100*   5266    5279
$TRAP   023252      3193    5795#
$TRAP2  023274      5806#   5817
$TRP  = 000014      5810#   5819#   5820#   5821#   5822#   5823#   5824    5825#   5826    5827#   5828#   5829#   5830#
                    5831#
$TRPAD  023306      5800    5817#
$TSTNM  001102      1927#   3499*   3503    3533*   3537    3571*   3575    3614*   3618    3655*   3659    3735*   3739
                    3778*   3782    3828*   3832    3928*   3932    4085*   4089    4151*   4155    4255*   4259    4345*
                    4349    4443*   4447    4557*   4561    4628*   4632    4757*   4761    4877*   4881    4966*   4970
                    5019*   5073    5103    5114    5687    5731*   5736    5740    6215    6417
$TTYIN  022636      5623    5624    5636    5654    5668    5672#   10072
$TYPBN= ****** U    5823
$TYPDS  021226      5370#   5822
$TYPE   020560      5226#   5810    5818
$TYPEC  020730      5247    5254    5261    5266#   5267    5571
$TYPEX  020776      5272    5274    5277#
$TYPOC  021024      5310#   5819
$TYPON  021040      5309    5312#   5821
$TYPOS  021000      5305#   5820
$XTSTR  022736      5700#
$$GET4= 000000      5036#
$OFILL  021223      5306*   5310*   5320    5355#
$40CAT= ****** U    5083    5697
.     = 051120      1504#   1508#   1516    1517#   1519#   1521#   1524#   1924#   1972    3186    3200    3201    3583
                    3591    3626    3630    3672    3694    3748    3792    3856    3859    3954    3992    4573    4578
                    4581    4589    4596    4645    4655    4672    4678    4764    4818    4824    4834    4884    4896
                    4917    4923    4931    4937    4945    5001#   5032#   5044    5045#   5103    5147#   5279    5424#
                    5429    5433#   5434    5435    5672#   5673    5680#   5739    5740    5912#   7528#   9219#   9220#
                    9221#   9222#   9223#   9224#   9225#   9226#   9227    9932
```

```
CKCHR    3157#   7289    7339    7566    7603    7643    7806    7819    7830    7881
CKDIG    3157#   7347    7585
CKNUM    3157#   7297    7618    7651
COMMEN      1#   1648#   3434    3469    4116    4525
COMND    3157#   3403    3506    3836    3936    4976    4981
DO       3157#   3402    3404    3513    3514    3549    3550    3586    3594    3628    3633    3675    3681    3697    3703
         3751    3753    3797    3799    3801    3803    3805    3807    3858    3879    4101    4103    4977    4982    6621
DODTA    3157#   4576    4580    4585    4595    4602    4648    4659    4675    4681    4821    4827    4837    4898    4919
         4925    4933    4939    4947
DRV.IN   3157#   4223    4317    4416    4514
ENDCOM      1#   1648#   3462    3480    4129    4534
ENDPAS   3157#   5029
ERRCAL   7901#   8719    8770    8773    9050    9111
ERREND   3157#   5097
ERROR    1542#   3388    3983    4059    4193    4207    4222    4292    4316    4380    4398    4415    4478    4496    4513
         6289    6290    6291    6292    6293    6320    6321    6322    6323    6328    6363    6364    6365    6366    6371
         6412    6413    6414    6415    6416    6442    6443    6444    6445    6446    6468    6469    6470    6471    6472
         6619    6660    6661    6662    6663    6664    6976    6981    7092    7098    7156    7157    8720    8771    8775
         9052    9112    10054
ERRTYP   3157#   5065
ER.NDX   3157#   6283    6314    6357    6406    6436    6462    6654
ESCAPE      1#   1648#   4169    4270    4360    4458    6282    6313    6356    6404    6435    6461    6617    6653    6975
         6979    7091    7096    7144
GETPRI      1#   1648#   6000    9993
GETSWR      1#   1449#   1648#   3249
LOOP     3157#   3583    3591    3626    3630    3672    3694    3748    3792    3855    3859    3953    3991    4573    4578
         4581    4589    4596    4645    4655    4672    4678    4818    4824    4834    4896    4917    4923    4931    4937
         4945
MORETA   1918#   1973
MORE.S   3157#   3494    3528    3566    3609    3650    3730    3773    3823    3923    4080    4146    4250    4340    4438
         4552    4623    4752    4872    4961
MSG      3485#   3487    3517#   3519    3553#   3555    3600#   3602    3639#   3641    3717#   3719    3762#   3764    3812#
         3814    3887#   3890    4067#   4069    4133#   4136    4239#   4241    4327#   4329    4425#   4427    4539#   4541
         4609#   4611    4690#   4692    4847#   4849    4952#   4954
MULT        1#   1648#
NEWTST      1#   1648#   3485    3517    3553    3600    3639    3717    3762    3812    3888    4067    4134    4239    4327
         4425    4539    4609    4690    4847    4952
POP         1#   1648#   5411    5779    5968    6051    6196    6245
PUSH        1#   1648#   5370    5759    5949    6003    6151    6209
REPORT      1#   1648#   3157#   4230    4233    4322    4421    4519
RP04.D      2#   7901
SAV.RH   3157#   4187    4201    4216    4286    4310    4374    4392    4409    4472    4490    4507
SCOPE    1543#   3515    3551    3598    3637    3715    3760    3810    3885    4065    4111    4236    4324    4423    4521
         4607    4687    4844    4948    4985
SETPRI      1#   1648#   5601
SETTRA   5810#   5819    5820    5821    5822    5824    5826    5827    5828    5829    5830
SETUP       1#   1648#   3181
SET.TN   3157#   3495    3529    3567    3610    3651    3731    3774    3824    3924    4081    4147    4251    4341    4439
         4553    4624    4753    4873    4962
SKIP        1#   1648#
SLASH       1#   1648#
SPACE    1648#
STARS       1#   1514    1648#   1652    1694    1698    1889    1920    1972    3059    3485    3493    3517    3527    3553
         3565    3600    3608    3639    3649    3717    3729    3762    3772    3812    3822    3888    3922    4067    4079
         4134    4145    4239    4249    4327    4339    4425    4437    4539    4551    4609    4622    4690    4751    4847
         4871    4952    4960    4991    5051    5104    5211    5282    5360    5428    5504    5519    5590    5614    5684
```

```
        5743    5789    5834    5853    5916    5940    5977    7900    7910    9981   10010
SWRSU      1#   1648#   3202#
TRMTRP  5810#
TYPBIN     1#   1648#
TYPB2D  3157#   6757    6764    6768    6773    6780    6784    6796    6801    7560    7635
TYPDEC     1#   1648#   5185
TYPEND  3157#   4998
TYPNAM     1#   1648#
TYPNUM     1#   1648#
TYPOCS     1#   1648#   3277    3357    3406    5008    7529
TYPOCT     1#   1648#   5016    5182    5532    7284    7598    7639
TYPTXT     1#   1648#   4998    5004    5012    5029    5144    5161    7253    7333    7365    7525    7630
$$CMRE  1918#   1957    1958    1959    1960    1961    1962
$$CMTM  1918#   1963    1964    1965
$$ESCA     1#   1648#
$$NEWT     1#   1648#   3485    3517    3553    3600    3639    3717    3762    3812    3888    4067    4134    4239    4327
        4425    4539    4609    4690    4847    4952
$$SET   5810#   5819    5820    5821    5822    5824    5826    5827    5828    5829    5830
$$SKIP     1#   1648#
.EQUAT     1#   1449#   1538
.HEADE     1#   1449#
.KT11      1#
.SETUP     1#   1449#   3161
.SWRHI     1#   1449#   1483
.SWRLO  1493#   1494    1495    1496    1497    1498    1499    1500
.$ACT1     1#   1449#   1512
.$APTB     1#
.$APTH     1#
.$APTY     1#
.$ASTA     1#
.$CATC     1#   1449#   1502
.$CMTA     1#   1449#   1918
.$DB2D     1#   1449#   5851
.$DB2O     1#
.$DIV      1#   1449#   5975
.$EOP      1#   1449#   4989
.$ERRO     1#   1449#   5049
.$ERRT     1#
.$MULT     1#
.$POWE     1#
.$RAND     1#   1449#   5938
.$RDDE     1#   1449#
.$RDOC     1#   1449#
.$READ     1#   1449#   5426
.$R2AZ     1#
.$SAVE     1#   1449#   5741
.$SB2D     1#   1449#   5832
.$SB2O     1#
.$SCOP     1#   1449#   5682
.$SIZE     1#   1449#   9979
.$SUPR     1#   1449#   5914
.$TRAP     1#   1449#   5787
.$TYPB     1#
.$TYPD     1#   1449#   5358
.$TYPE     1#   1449#   5209
.$TYPO     1#   1449#   5280
```

```
.$40CA      1#
.1170       1#


. ABS.  051120     000


 ERRORS DETECTED:  0

 DSKZ:CZRJAD.BIN,DSKZ:CZRJAD.LST/CRF/SOL/NL:TOC:MD:MC:CND/LI:ME=CZRJAD.SML,CZRJAD.010,CZRJAD.P11
 RUN-TIME: 77 108 7 SECONDS
 RUN-TIME RATIO: 429/194=2.2
 CORE USED:  52K  (103 PAGES)
```