RH11/RP04

READ/WRITE TEST

MD-11-DZRPK-B

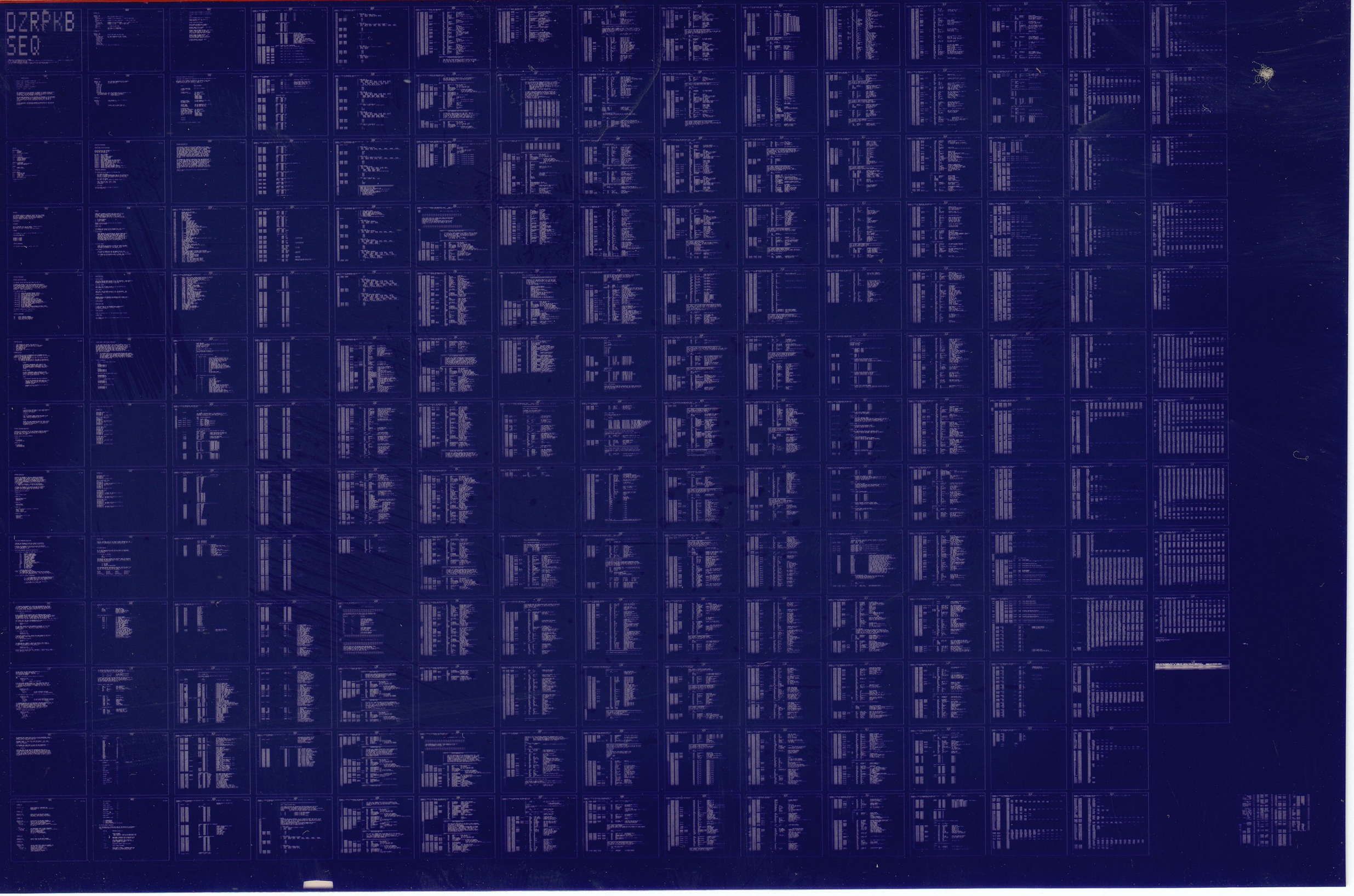
EP-DZRPK-B-DL-A

MAR 1976

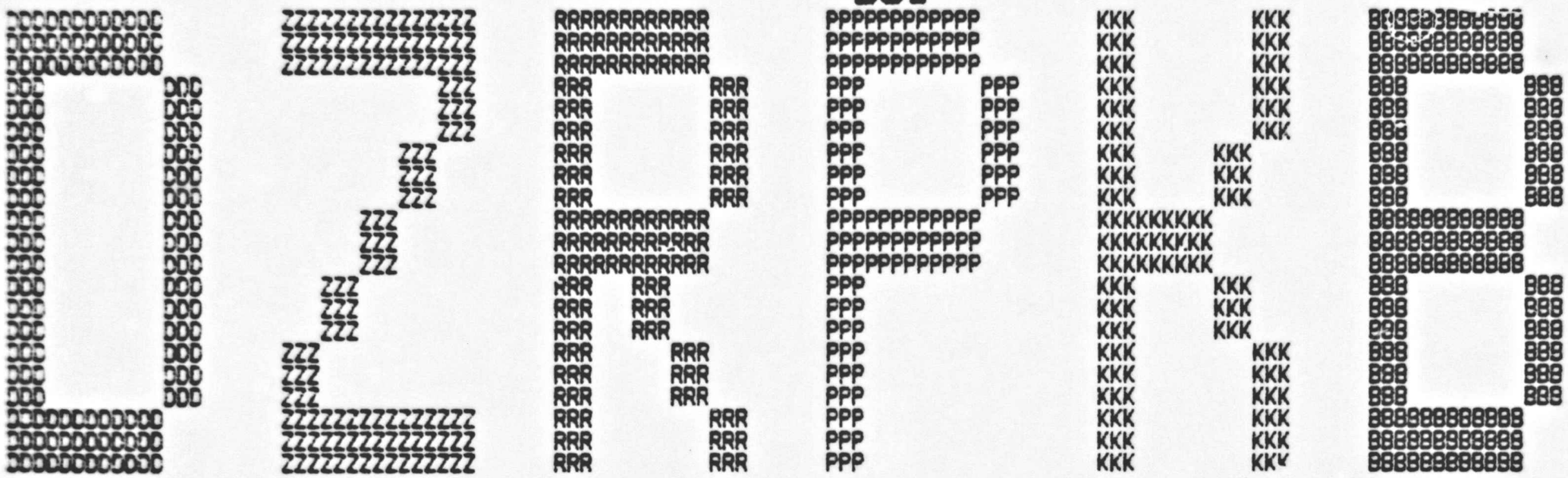
FICHE 1 OF 1

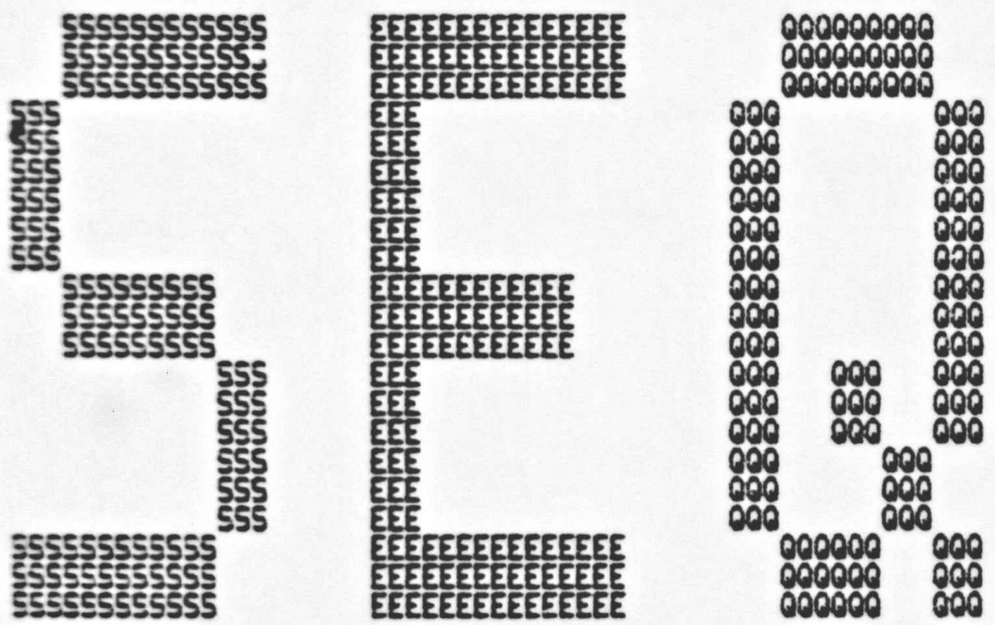
digital

Made In U.S.A.



B01





LPTSPL Version 101(2111) Running on MTA260
 START User WELLINGHAM,A[1,2] Job DZRPKB Seq. 1900 Date 13-Feb-76 18:05:18 Monitor   RZ231D KL10 SYS#1026 *START*
Request created: 13-Feb-76 17:53:30
File: WHTN:DZRPKB.SEQ<055>[404,3722] Created: 13-Feb-76 15:07:30 Printed: 13-Feb-76 18:05:19
QUEUE Switches: /FILE:ASCII /COPIES:1 /SPACING:1 /LIMIT:400 /FORMS:NORMAL

IDENTIFICATION

SEQ 0001

PRODUCT CODE: MAINDEC-11-DZRPK-B-D

PRODUCT NAME: RH11/RP04 MECHANICAL AND READ/WRITE TEST

DATE CREATED: FEBRUARY 21, 1975

MAINTAINER: DIAGNOSTIC ENGINEERING

AUTHOR: J. LACEY/C. HESS

CONTENTS

E01

SEQ 0003

1. ABSTRACT

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

2. REQUIREMENTS

2.1 EQUIPMENT

PDP-11 COMPUTER WITH 16K OF MEMORY, CONSOLE TELETYPE,
RH11/RP04 DISK WITH FORMATTED PACK,
AND A KW11-P CLOCK (IF TIMING TESTS ARE DESIRED).

2.2 STORAGE

THIS PROGRAM WILL LOAD IN 12K BUT REQUIRES. 16K TO RUN

2.3 PRELIMINARY PROGRAMS

MAINDEC-11-DZRPS
MAINDEC-11-DZRPT
MAINDEC-11-DZRPU
MAINDEC-11-DZRPV

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR LOADING .ABS TAPES

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

THE CONTROL SWITCH SETTINGS ARE INPUT VIA THE TTY.

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 A SLASH (/), 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)
```

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

REFER TO 4.4.1 FOR C.SWR SELECTION

4.2 STARTING ADDRESSES

```
200     NORMAL STARTING ADDRESS
204     SELECT OPERATING PARAMETERS
210     SELECT RH11/RP04 ADDRESSES
214     COMBINATION OF 204 AND 210
```

4.3 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 4.1 & 5.1)
6. PRESS START.

4.4 PROGRAM ACTION

IN AN EFFORT TO ALLOW CONVERSATION WITH A PROGRAM FOR THE PURPOSE OF CONTROLLING ITS OPERATION AND PARAMETERS THE FOLLCWING 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

AN INPUT TERMINATOR: WHEN A PERIOD IS TYPED AFTER A "STATEMENT TERMINATOR PERIOD" IT TELLS THE PSI THIS IS THE END OF ALL CHANGES (LEGAL ON ALL LINES) BEGIN EXECUTION.

<,><CR> COMMA

A CONTINUATION INDICATOR AND FIELD SEPARATOR:

1) THIS IS THE CONSTRUCTION USED WHEN NOT USING A PERIOD; THAT IS TO SAY THE COMMA SAYS "TERMINATE THE LINE BUT NOT THE INPUT FOR PARAMETER CHANGES".

2) SEPARATE MULTIPLE ENTRIES ON THE SAME TEXT STRING, I.E., A,B,C,D.

⟨/⟩ SLASH

A MODIFICATION INDICATOR: AT ANY GIVEN PARAMETER STOP IF A SLASH IS TYPED IT SAYS "OPEN THIS PARAMETER FOR CHANGES".

⟨↑U⟩ CONTROL-U

DUMP 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.

4.4.1 CONTROL SWITCHES 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".

4.4.1.1 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..
```

4.4.2 ADDRESS SELECTION

STARTING THE PROGRAM AT 200 WILL RESULT IN AUTOMATIC
SELECT OF THE DEFAULT VALUES OF BUS ADDRESS (RHCS1),
VECTOR ADDRESS, AND PRIORITY LEVEL OF THE RH11/RP04.
IF THE DEFAULT VAULE OF THE BUS ADDRESS DOES NOT RESPOND
(TIMES OUT) TO ADDRESSING IT IS REPORTED AS AN ERROR.
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 CONTROL AND STATUS REGISTER #1 (RHCS1)
ADDRESS, THE VECTOR ADDRESS, AND THE PRIORITY LEVEL.

4.4.2.1 ADDRESS SELECTION EXAMPLES

EXAMPLE #1

```
RHCS1=177200/176300..
```

EXAMPLE #2

```
RHCS1=177200/176300,
RHVEC=254/260,
RHPRIO=5/6..
```

EXAMPLE #3

```
RHCS1=177200,
RHVEC=254/260..
```

EXAMPLE #4

```
RH11/RP04 FAILED TO RESPOND TO ADDRESSING
RHCS1    ERR PC
177200   XXXXXX
RHCS1=177200/176300..
```

EXAMPLE #5

```
RHCS1=177200/1776\67\6300,
RHVEC=254,
RHPRIO=5,
RHCS1=176300..
```

4.4.3 DRIVE AND PARAMETER SELECTION

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

STARTING THE PROGRAM AT 210 OR 214 ALLOWS THE OPERATOR TO SELECT THE DRIVE(S) TO BE TESTED, THE TESTS TO BE EXECUTED, AND THE PARAMETERS TO USE.

4.4.3.1 DRIVE AND PARAMETER SELECTION DESCRIPTION

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

"R"	REPEATS (ITERATIONS)
"FC"	FIRST CYLINDER ADDRESS
"LC"	LAST CYLINDER ADDRESS
"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
*"T"	ALL TIMING TESTS
*"A"	ALL ADDRESS TESTS
*"D"	THE DATA TESTS
*"E"	THE EXERCISER

* USED BY USER AS INPUT.

NOTE1: ALL OTHERS ARE OUTPUT ONLY.

NOTE2: 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(B)=PATTERN 5) 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 OPENED, THEY WILL RECIEVE THE PARAMETERS OF THE TEST THAT IS OPEN WHEN <..> IS TYPED.

<,> IS ILLEGAL AS A "TEST" LINE TERMINATOR UNLESS A TEST IS TO BE OPENED.

ON STARTING THE PROGRAM ALL TESTS WILL RUN AGAINST ALL DRIVES IN A WORST CASE MANNER. BUT IF THE USER DESIRES TO SELECT THE DRIVES TO BE TESTED, THE TESTS TO BE EXECUTED, AND THE PARAMETERS TO BE USED HE ENTERS CONVERSATION MODE BY TYPING CONTROL C (↑C).

THE PROGRAM WILL RESPOND WITH:

```
DRIVE(S)=
```

WE WILL ASSUME FOR OUR EXAMPLE THAT THE OPERATOR WISHES 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‹,›" WHICH SAYS "THIS IS THE END OF ANY CHANGES TO THIS LINE BUT TAKE ME DEEPER FOR MORE CHANGES", AND ‹CR› "TAKE IT."

FOR CLARITY THE LINE WILL BE REPEATED AS IT LOOKS AFTER USER RESPONSE WITH THE LINE BELOW IT ADDED, THE PROGRAMS RESPONSE:

```
DRIVE(S)=3,‹CR›
TEST=
```

NOW THE USER INSERTS THE TEST NUMBERS HE DESIRES. IN THIS CASE HE WANTS TESTS 2 THRU 7 AND TEST 11 SO HE TYPES 2-7‹,› (TO SEPERATE FIELD), 11‹..› (END OF ANY CHANGES START EXECUTION), ‹CR› (TAKE IT).

IT NOW LOOKS LIKE THIS

```
DRIVE(S)=3,‹CR›
TEST=2-7,11..‹CR›
```

IN OUR NEXT EXAMPLE WE WILL ASSUME THE USER WISHES TO TEST DRIVE 4 AND RUN TESTS 1 AND 3 THRU 11 WITH THE PARAMETER FOR TESTS 3 AND 10 MODIFIED.
THIS IS HOW HE WOULD RESPOND:

```
DRIVE(S)=4,‹CR›
TEST=
```

THE USER NOW WILL INSERT A LINE TO GET TESTS 1 AND 3 THRU 11 BUT REMEMBER HE WISHES TO "OPEN" TESTS 3 AND 10 FOR PARAMETER CHANGES. THE ENTIRE ENTRY IS SHOWN BELOW:

```
DRIVE(S)=4,‹CR›
TEST=1,3/4-7,10/11.‹CR›
```

NOTICE THIS SAYS SELECT TEST ONE, CONTINUE‹,›; SELECT TEST 3, OPEN‹/›; SELECT TESTS 4-7, CONTINUE‹,›; SELECT TEST 10, OPEN‹/›; SELECT TEST 11, END OF INPUT‹.›.

THE PSI SCANS THE TEST STRING AND DETERMINES THAT TEST #1'S
PARAMETERS WILL REMAIN THE SAME BUT
TEST 3'S MUST BE CHANGED. THEREFORE, TEST
3 IS OPENED FOR CHANGE.

RESPONDS: ‹FOR CLARITY ENTIRE TRANSACTION REPEATED›

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

AS IN THE NORMAL MANNER TYPING A ‹/› WILL OPEN THIS LINE FOR
MODIFICATION OF THE ITERATION COUNT, ENDING THE LINE WITH A
‹.› (PERIOD) WOULD TERMINATE THE CHANGES FOR THIS TEST, AND AS
IN THE EXAMPLE SHOWN USING A ‹,› (COMMA) WILL GET US 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
```

THE USER DOES NOT WISH TO CHANGE "FC" SO THE FOLLOWING ACTION IS TAKEN:

```
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 USER INTENDED TO MODIFY AND IS WHY HE OPENED
TEST #3. HE WANTS TO CHANGE IT TO CYLINDER 20. THEREFORE
HE TYPES ‹/› (OPEN LINE FOR CHANGE), "20" (THE NEW VALUE),
‹.› (THIS IS THE END OF CHANGES FOR THIS DRIVE), ‹CR›
(TAKE IT).
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
```

M01

SEQ 0011

THE PROGRAM HAS LOADED TEST #3 WITH ITS NEW PARAMETERS, TESTS NUMBER 4 THRU 7 WITH THEIR PREVIOUSLY ASSIGNED PARAMETERS AND IS NOW WAITING FOR CHANGES TO TEST 10.

THE USER TYPES </> (OPEN THIS LINE FOR CHANGE), "10" (MAKE ITERATION COUNT 10), <.> (THIS IS THE END OF CHANGES TO THIS TEST) <CR> (TAKE IT).

THE PROGRAM NOW LOADS TEST 10 WITH THE NEW PARAMETERS, TEST 11 WITH PREVIOUSLY ASSIGNED PARAMETERS AND RESPONDS WITH:

DRIVE(S)=

SINCE THE USER DID NOT END THE CONVERSATION MODE WITH A <..> 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 CHANGES MADE WE MUST NOTIFY THE PROGRAM WE ARE FINISHED BY TYPING <..>.

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 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 THEREFORE BE
                           ;SET TO THE SAME PARAMETERS
                           ;AS TEST 7 -- START EXECUTION
```

EXAMPLE #4

```
DRIVE=0,<CR>               ;SELECT DRIVE #0 AND MAKE CHANGES
TEST=5,E..<CR>             ;RUN ALL SEEK TESTS AND THE EXERCISER
```

EXAMPLE #5

```
DRIVE=1,<CR>
TEST=5/0.<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 1-6 WILL BE ASSIGNED THE
                           ;SAME PARAMETERS AS TEST 0
```

EXAMPLE #6

```
DRIVE=1,<CR>
TEST=5/.<CR>              ;OPEN THE SEEK TESTS (TST 0-6)
  TEST 0
    R=10/100.<CR>         ;CHANGE TO 100 ITERATIONS; <.> MOVE TO 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      ; <,> MOVE TO NEXT LINE
    FC=0/50,<CR>          ;CHANGE FC TO 50; <,> MOVE TO NEXT LINE
    LC=410/51.<CR>        ;CHANGE LC TO 51; <.> MOVE TO NEXT TEST
  TEST 3
    R=1.<CR>              ;<.> MOVE TO NEXT TEST
  TEST 4
    R=1..<CR>             ;USE TESTS 4'S PARAMETERS
                          ;FOR TEST 5 & 6 START EXECUTION
```

EXAMPLE #7

```
DRIVE=1,<CR>
TEST=0/.<CR>              ;SELECT AND OPEN THE DATA TEST
  TEST 15
    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>
    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=0/.<CR>                ;#7, BUT ALSO SPECIFY YOUR OWN DATA
  TEST 15                   ;PATTERN (PAT #0).
    R=1000,<CR>
    FC=10,<CR>
    LC=10,<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 PATTERN #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,                ;TEST DRIVES 0,1, AND 4 IN SEQUENCE
TEST=0-5/                   ;CHANGE TEST 0-5
  TEST 0
    R=10,
    FC=0,
    LC=410/1..              ;CHANGE LAST CYLINDER FROM 410
                            ;TO 1 FOR TEST #0-5; START TESTING
```

5. OPERATING PROCEDURE

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 7-12)
SW<05>=1...REPORT ONE ERROR PER SECTOR (TESTS 13&14)
SW<04>=1...INHIBIT WRITES (TEST 15)
SW<03>=1...INHIBIT WRITE CHECKS (TEST 15)
SW<02>=1...INHIBIT READ AND SOFTWARE COMPARES (TEST 15)
SW<01>=1...INHIBIT SOFTWARE COMPARES (TEST 15)
SW<00>=1...PERFORM READ AFTER WRITE CHECK ERROR (TEST 15)

5.2 SUBROUTINE ABSTRACTS

THE SUBROUTINE CALLS USED BY THIS PROGRAM ARE:

1. IOT (USED FOR SCOPE)
   THIS CALL IS PLACED BETWEEN EACH TEST IN THE INSTRUCTION
   SECTION. IT RECORDS THE STARTING ADDRESS OF EACH TEST IN
   LOCATIONS "$LPADR" AND "$LPERR" AS IT IS BEING ENTERED.

2. EMT (USED FOR ERROR)
   THIS CALL IS USED TO REPORT ALL ERRORS (REFER TO 6).

3. TRAP (USED FOR TYPE, TYPOC, TYPOS,
   TYPON, TYPDS, RDCHR, RDLIN, SAVREG,
   AND RESREG)

4. JSR R,ADR

THE FOLLOWING SECTION CONTAINS AN ABSTRACT
FOR EACH SUBROUTINE

6. ERRORS

THERE ARE A NUMBER 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 TYPE OUT 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 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 ERROR

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 ERROR

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

BEFORE STARTING THE PROGRAM THE OPERATOR MUST INSURE THAT A FORMATTED PACK IS LOADED AND WRITE ENABLED IN THE DRIVE(S) TO BE TESTED.

8. MISCELLANEOUS

8.1 EXECUTION TIME

TO MAKE ONE PASS OF THE PROGRAM TAKES APPROXIMATELY 5 MINUTES. THE RUN TIME INCREASES LINEARALLY FOR EACH DRIVE TESTED.

8.2 STACK POINTER

STACK IS INITIALLY SET TO 1100 AND EXTENDS DOWN IN MEMORY.

8.3 COMMON TAGS

THE COMMON TAGS USED IN THIS PROGRAM START AT 1100 AND EXTEND UP IN MEMORY.

8.4 TIMING TESTS (TST7-TST12) 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 BY THE VENDOR ARE POSITIONER MOVEMENT TIMES ONLY AND ARE NOT A MEASUREMENT OF EFFECTIVE SEEK TIME.

8.4.1 TIMING TOLERANCES

1. TEST 7 -- 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 10 -- ONE CYLINDER SEEK TIMES

   MINIMUM=3000 US
   MAXIMUM=10000 US
   NOMINAL=7000 US

3. TEST 11 -- AVERAGE SEEK TIMES

   MINIMUM=26000 US
   MAXIMUM=30000 US
   NOMINAL=28000 US

4. TEST 12 -- MAXIMUM SEEK TIMES

   MINIMUM=46000 US
   MAXIMUM=52000 US
   NOMINAL=50000 US

8.4.2. 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
** 0, 1, 2,...410 **
MIN=5350 US
MAX=6920 US
AVG=5550 US  409 SEEKS TIMED
**  410, 409, 408,...0 **
MIN=5140 US
MAX=5960 US
AVG=5430 US  410 SEEKS TIMED

AVERAGE SEEK TIMES
** 0 TO 136 **
MIN=27770 US
MAX=28640 US
AVG=28230 US  128 SEEKS TIMED
** 136 TO 0 **
MIN=27990 US
MAX=28550 US
AVG=28220 US  128 SEEKS TIMED

MAXIMUM SEEK TIMES
** 0 TO 410 **
MIN=49990 US
MAX=51980 US
AVG=51010 US  128 SEEKS TIMED
** 410 TO 0 **
MIN=48120 US
MAX=50650 US
AVG=49340 US  128 SEEKS TIMED
```

EXAMPLE #2 SEQ 0020

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

ONE CYLINDER SEEK TIMES
** 0, 1, 2,...410 **
MIN=5470 US
MAX=10940 US  3 ABOVE THE MAXIMUM OF 10000 US
AVG=5830 US  409 SEEKS TIMED
** 410, 409, 408,...0 **
MIN=5040 US
MAX=5970 US
AVG=5330 US 410 SEEKS TIMED

AVERAGE SEEK TIMES
** 0 TO 136 **
MIN=29730 US
MAX=31620 US  73 ABOVE THE MAXIMUM OF 30000 US
AVG=30320 US  128 SEEKS TIMED
** 136 TO 0 **
MIN=28620
MAX=31230 US  128 ABOVE THE MAXIMUM OF 30000 US
AVG=30800 US  128 SEEKS TIMED

MAXIMUM SEEK TIMES
** 0 TO 410 **
MIN=53510 US
MAX=54240 US  128 ABOVE THE MAXIMUM OF 52000 US
AVG=54020 US  128 SEEKS TIMED
** 410 TO 0 **
MIN=52050 US
MAX=54550 US  128 ABOVE THE MAXIMUM OF 52000 US
AVG=52210 US  128 SEEKS TIMED
```

8.5 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.

8.6 RH11/RP04 DRIVER

ALL OF THE COMMUNICATION WITH THE RH11/RP04 IS PERFORMED VIA A COMMON DRIVER. THIS SECTION CONTAINS THE USERS GUIDE FOR THE DRIVER.

8.6.1. TO INITIALIZE THE DRIVER:

```
        JSR     PC,RPINIT
        RETURN
```

UPON RETURN YOU MUST EXAMINE THE "DRVSTA" TABLE TO DETERMINE THE DRIVES THAT ARE ONLINE FOR TESTING. THE DRVSTA TABLE IS EIGHT BYTES; ONE BYTE PER DRIVE. THE STATE OF EACH DRIVE WILL BE INDICATED AS FOLLOWS:

```
        >0  ON LINE
        =0  OFFLINE
        <0  UNSAFE OR NONEXISTENT
```

WITH NON-RP04 DRIVES INDICATED AS OFFLINE. AND THE DRIVE TYPE WILL BE INDICATED IN THE "DRVTYP" TABLE. USING "DRVSTA" AND "DRVTYP" THE FOLLOWING CAN BE DETERMINED:

ONLINE	OFFLINE	UNSAFE	NONEXISTENT
DRVSTA>0	DRVSTA=0	DRVSTA<0	DRVSTA<0
DRVTYP=(RHDT)	DRVTYP=(RHDT)	DRVTYP=(RHDT)	DRVTYP=0

THE RPINIT ROUTINE WILL DO A READIN PRESET AND WILL SET FMT22.

8.6.2. AFTER THE DRIVER HAS BEEN INITIALIZED YOU CAN USE IT TO PERFORM FOR YOU.


```
CALL:
        JSR     R0,RP04         ;MAKE THE CALL
        PNTDPB                  ;ADDRESS OF DPB*
        RETURN1                 ;RETURN IF QUEUE IS FULL
        RETURN2                 ;RETURN IF REQUEST IS IN
                                ;QUEUE OR THERE IS AN
                                ;ERROR CONDITION

*DPB (DATA PARAMETER BLOCK)

PNTDPB: .BYTE   0               ;(0) DRIVE NUMBER
        .BYTE   0               ;(1) OFFSET VALUE OR FMT22, ECT, AND HCI
        .BYTE   0               ;(2) COMMAND
        .BYTE   0               ;(3) PSEL AND A17 AND A16
        .WORD   0               ;(4) WORD COUNT (MUST BE NEG.)
        .WORD   0               ;(6) BUFFER ADDRESS OR
                                ;REGISTER TABLE POINTER
        .BYTE   0               ;(10) SECTOR ADDRESS OR
                                ;FIRST REG. INDEX
        .BYTE   0               ;(11) TRACK ADDRESS OR
                                ;LAST REG. INDEX
        .WORD   0               ;(12) CYLINDER ADDRESS
        .WORD   0               ;(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.
        .WORD   0               ;(16) STATUS/ERROR INDICATOR
                                ;BIT15=1=>ERROR OCCURRED
                                ;BIT07=1=>DONE
                                ;BIT14-BIT09 AND BIT06-BIT03
                                ;INDICATE TYPE OF ERROR
```

8.6.3. THE DRIVER PROVIDES A SOFTWARE TIMEOUT CAPABILITY.
TO UTILIZE THIS CAPABILITY YOU MUST SUPPLY THE "RP TIMER" ROUTINE
WITH THE ELAPSED TIME IN THE FOLLOWING MANNER:

```
        MOV     #16.,-(SP)      ;16 MILLISECONDS BETWEEN
                                ;CLOCK TICKS
        JSR     R0,RPTMR        ;CALL THE TIMER ROUTINE
```

IT SHOULD BE NOTED THAT YOU MUST PROVIDE THE CODE TO DRIVE THE
CLOCK. AND THE ELAPSED TIME MUST BE IN MILLISECONDS.
THE DRIVER WILL SET THE TIMEOUT TO 1 SECOND FOR ALL POSITIONING
AND DATA TRANSFER OPERATIONS AND WILL SET THE TIMOUT TO 30
SECONDS FOR ERROR RECOVERY OPERATIONS.

8.6.4. EXAMPLE - WRITE 1000. WORDS

```
1$:     JSR     R0,RP03         ;CALL THE DRIVER
        WRTDPB                  ;DPB ADDRESS
        BR      1$              ;WAIT FOR QUEUE IF FULL
2$:     TST     WRTDPB+16       ;WAIT FOR COMMAND TO COMPLETE
        BEQ     2$
        BMI     ERROR1          ;ERROR OCCURRED
        .
        .
        .

WRTDPB: .BYTE   5               ;DRIVE #5
        .BYTE   0               ;
        .BYTE   161             ;WRITE COMMAND
        .BYTE   0               ;
        .WORD   -1000.          ;WORD COUNT
        .WORD   WRTBUF          ;BUFFER ADDRESS
        .BYTE   3               ;SECTOR
        .BYTE   5               ;TRACK
        .WORD   400             ;CYLINDER
        .WORD   ERRTB5          ;ERROR TABLE
        .WORD   0               ;STATUS/ERROR INDICATOR

        ATERNATE DPB SETUP

WRTDPB: .WORD   5               ;THIS SETUP ACHIEVED
        .WORD   WRITE           ;EVERYTHING THE
        .WORD   -1000.          ;ABOVE TABLE DID, BUT
        .WORD   WRTBUF          ;IN A CLEANER FORMAT
        .BYTE   3,5
        .WORD   400,ERRTB5,0
```

8.6.5. RH11/RP04 REGISTERS

MNEMONIC	INDEX
RHCS1	0
RHWC	2
RHBA	4
RHDA	6
RHCS2	10
RHDS1	12
RHER1	14
RHAS	16
RHLA	20
RHDB	22
RHMR	24
RHDT	26
RHSN	30
RHOF	32
RHCA	34
RHCC	36
RHER2	40
RHER3	42
RHEC1	44
RHEC2	46

8.6.6. COMMANDS PERFORMED BY THE DRIVER

COMMAND	CODE	COMMAND TYPE
NO OPERATION	101	N
UNLOAD	103	N
SEEK	105	P
RECALIRATE	107	P
DRIVE CLEAR	111	N
RELEASE	113	N
OFFSET	115	P
RETURN TO CENTER	117	P
READIN PRESET	121	N
PACK ACKNOWLEDGE	123	N
SEARCH	131	P

GET REGISTER(S)	141	S
SET FORMAT	143	S
SELECT DRIVE	145	S
WRITE CHECK DATA	151	D
WRITE CHECK HEADER AND DATA	153	D
WRITE DATA	161	D
WRITE HEADER & DATA	163	D
READ DATA	171	D
READ HEADER & DATA	173	D

N = HOUSEKEEPING
P = POSITIONING
D = DATA TRANSFER
S = SPECIAL PROVIDED BY THE DRIVER

8.6.7. DPB STATUS/ERROR INDICATOR

THIS INDICATOR WILL INFORM THE USER OF THE RESULTS OF THE REQUEST. THIS IS ACCOMPLISHED BY SETTING VARIES BITS OF THE INDICATOR TO A ONE.

BIT NO.	MEANING IF ON A "1"
15	ERROR OCCURRED DONE (BIT07=0); BITS 14-10 SPECIFIES TYPE DONE (BIT07=1); BITS 06-03 SPECIFIES TYPE
14(1)	USER MADE A REQUEST FOR A FUNCTION TO BE PERFORMED ON AN OFFLINE OR UNSAFE DRIVE
13(1)	USER MADE A REQUEST FOR A FUNCTION TO BE PERFORMED ON A DRIVE THAT HAS AN UNLOAD REQUEST IN QUEUE.
12(2)	PERSISTENT UNSAFE CONDITION EXIST.
11(2)	PARITY ERROR OCCURRED
10(2)(4)	FATAL PARITY ERROR. A MASSBUS CLEAR WAS PERFORMED, ALL QUEUES WERE EMPTIED, AND ALL DRVACT'S SET TO THE IDLE STATE

9(3)(4)	SOFTWARE TIMEOUT OCCURRED ON THIS DRIVE
8(4)	SOFTWARE TIMEOUT OCCURRED ON ANOTHER DRIVE
7	DONE
6(2)	ERROR OCCURRED DURING AN I/O OPERATION
5(2)	ERROR OCCURRED DURING AN OPERATION OTHER THAN I/O.
4(2)	UNSAFE CONDITION OCCURRED
3(2)	DRIVE ERROR OCCURRED THAT CAUSED AN AUTOMATIC "RECALIBRATE" SEQUENCE
(1) =>	REQUEST WASN'T PUT IN QUEUE. (RH11/RP04 REGISTERS WERE NOT SAVED)
(2) =>	REQUEST QUEUE HAS BEEN EMPTIED. THE DRIVER ISSUED A "DRIVE CLEAR" TO THE DRIVE. NOTE: ALL RH11/RP04 REGISTERS ARE SAVED AS PER DPB+14 BEFORE THE "DRIVE CLEAR".
(3) =>	REQUEST QUEUE HAS BEEN EMPTIED. THE DRIVER ISSUED A MASSBUS INIT. ALL RH11/RP04 REGISTERS FOR THE DRIVE WERE SAVED AS PER DPB+14 BEFORE THE INIT.
(4) =>	A "RECALIBRATE" SHOULD BE ISSUED BEFORE ANY OTHER COMMAND.

8.6.8. ERROR CALLS MADE BY THE DRIVER. SEQ 0027

THERE ARE A FEW ERRORS THAT CAN OCCUR THAT CAN NOT BE INDICATED IN A DPB.

WHEN THIS TYPE OF ERROR IS DETECTED BY THE DRIVER IT WILL MAKE AN ERROR CALL OF THE FORM "ERROR N", WHERE "N" IS THE ERROR NUMBER AND THE ERROR WILL BE AN EMT INSTRUCTION.

N	TYPE	DATA AVAILABLE
1	ILLEGAL SUBSYSTEM INTERRUPT OCCURRED (SC=0)	*R4= RHCS1'S ADDRESS
2	ILLEGAL DRIVE INTERRUPT OCCURRED	R1= DRIVE NUMBER R3= ATA BIT *R4= RHCS1'S ADDRESS R5= (RHAS) RPERRS =RHDS1 RPERRS+2=RHER1 RPERRS+4=RHER2 RPERRS+6=RHER3
3	MASSBUS CONTROL BUS PARITY ERROR OCCURRED(MCPE=1)	RD.ADR= ADDRESS OF REG. READ RD.WRD= WORD READ
4	CONTROL BUS PARITY ERROR OCCURRED(PAR=1)	WRT.AD= ADDRESS OF REG. WRITTEN WRT.WD= WORD WRITTEN RD.WRD= WORD READ BACK
5	ATTENTION FROM AN OFFLINE, NONEXISTENT OR PREVIOUSLY DETECT UNSAFE DRIVE OCCURRED. NOTE: COULD HAVE BEEN CAUSED BY A BIT PICK OR DROP WHEN READING RHAS.	R1= DRIVE NUMBER R3= ATA BIT *R4= RHCS1'S ADDRESS R5= (RHAS) RPERRS =RHDS1 RPERRS+2=RHER1 RPERRS+4=RHER2 RPERRS+6=RHER3
6	RESERVED	
7.	RESERVED	

* THIS IS THE ACTUAL UNIBUS ADDRESS (177200)

9. PROGRAM DESCRIPTION

THIS PROGRAM CONTAINS FIFTEEN TESTS NUMBERED 0-16 IN OCTAL. TESTS 0-6 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. TEST 7-12 WILL MEASURE THE ROTATIONAL SPEED, THE ONE CYLINDER SEEK, THE AVERAGE SEEK, AND THE MAXIMUM SEEK TIMES TO INSURE THEY ARE ALL WITHIN THE TOLERANCES ALLOWED. TEST 13 AND 14 INSURES THE SECTOR AND TRACK ADDRESSING CIRCUITRY WORKS PROPERLY. TEST 15 VERIFIES THE DATA STORAGE AND RETRIEVAL CAPABILITIES ARE FUNCTIONAL. AND TEST 16 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 SEQUENCIAL ORDER ALL OF THE DRIVES WILL BE TESTED. ONE PASS THROUGH THE TEST SEQUENCE (TEST 0-15) WILL BE PERFORMED ON EACH DRIVE BEFORE MOVING TO THE NEXT DRIVE IN SEQUENCE. THE DRIVE TO BE TESTED WILL BE TYPED AT THE BEGINNING OF EACH PASS, 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 SECTION FOR A DETAILED DESCRIPTION OF EACH TEST.

```
                        .ENABLE ABS,AMA
                        .TITLE  MAINDEC-11-DZRPK-B "MECHANICAL AND READ/WRITE TEST"
                        ;*COPYRIGHT (C) 1974,1975
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*PROGRAM BY J. LACEY/C. HESS
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-A2).
                        ;*

                        .SBTTL  CONTROL SWITCH SETTINGS
                        ;*
                        ;*    SWITCH  STATE           USE
                        ;*    ------  -----   ----------------------------------
                        ;*      15      0     WRITE PACK BEFORE TESTING (TEST 16)
                        ;*              1     INHIBIT WRITING PACK BEFORE TESTING (TEST 16)
                        ;*      14      0     NO STALL BETWEEN DRIVE FUNCTIONS
                        ;*              1     STALL AFTER EVERY DRIVE FUNCTION
                        ;*      13      0     USE SPECIFIC STALL TIME
                        ;*              1     USE RANDOM STALL
                        ;*      12      0     NO INCREMENTING STALL IN TEST 4
                        ;*              1     DO INCREMENTING STALL IN TEST 4
                        ;*       8      0     DO IMPLIED SEEKS WITH DATA TRANSFERS
                        ;*              1     DO EXPLICIT SEEKS BEFORE DATA TRANSFERS
                        ;*       7      0     DO "READ HEADER AND DATA" IN TESTS 0-6
                        ;*              1     DO EXPLICIT SEEKS IN TESTS 0-6
                        ;*       6      0     60 HZ
                        ;*              1     50 HZ
                        ;*       5      0     RUN WATCHDOG TIMER
                        ;*              1     INHIBIT WATCHDOG TIMER


                        .SBTTL  OPERATIONAL SWITCH SETTINGS
                        ;*
                        ;*      SWITCH              USE
                        ;*      ------      --------------------
                        ;*        15        HALT ON ERROR
                        ;*        14        LOOP ON TEST
                        ;*        13        INHIBIT ERROR TYPEOUTS
                        ;*        11        INHIBIT ITERATIONS
                        ;*        10        BELL ON ERROR
                        ;*         9        LOOP ON ERROR
                        ;*         8        PRINT ERROR MESSAGE ON LINE PRINTER
                        ;*         7        READ CONTROL SWITCH SETTINGS FROM TTY
                        ;*         6        INHIBIT TIME REPORTS (TESTS 7-12)
                        ;*         5        REPORT ONE ERROR PER SECTOR (TESTS 13 & 14)
                        ;*         4        INHIBIT WRITES (TEST 15)
                        ;*         3        INHIBIT WRITE CHECKS (TEST 15)
                        ;*         2        INHIBIT READ AND SOFTWARE COMPARES (TEST 15)
                        ;*         1        INHIBIT SOFTWARE COMPARES (TEST 15)
                        ;*         0        PERFORM READ AFTER WRITE CHECK ERROR (TEST 15)
```

```
4954
4955
 (1)                                    .SBTTL  TRAP CATCHER
 (1)
 (1)                                            .=0
 (1)            000000                  ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
 (1)                                    ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
 (1)                                    ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
4956
4957                                    .SBTTL  STARTING ADDRESSES
4958            000200                  .=200
4959                                    ;*200 = NORMAL START
4960  000200   000137  004120                    JMP     @#START1
4961                                    ;*204 = SELECT OPERATING PARAMETERS
4962  000204   000137  004210                    JMP     @#START2
4963                                    ;*210 = SELECT RH11/RP04 ADDRESSES
4964  000210   000137  004110                    JMP     @#START3
4965                                    ;*214 = COMBINATION OF 204 AND 210
4966  000214   000137  004200                    JMP     @#START4
4967
4968
 (1)                                    .SBTTL  BASIC DEFINITIONS
 (1)
 (1)                                    ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
 (1)            001100                  STACK=  1100
 (1)                                    .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
 (1)                                    .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL
 (1)            177776                  PS=     177776          ;;PROCESSOR STATUS WORD
 (1)                                    .EQUIV  PS,PSW
 (1)            177774                  STKLMT= 177774          ;;STACK LIMIT REGISTER
 (1)            177772                  PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
 (1)            177570                  SWR=    177570          ;;SWITCH REGISTER
 (1)            177570                  DISPLAY=SWR
 (1)
 (1)                                    ;*GENERAL PURPOSE REGISTER DEFINITIONS
 (1)            000000                  R0=     %0              ;;GENERAL REGISTER
 (1)            000001                  R1=     %1              ;;GENERAL REGISTER
 (1)            000002                  R2=     %2              ;;GENERAL REGISTER
 (1)            000003                  R3=     %3              ;;GENERAL REGISTER
 (1)            000004                  R4=     %4              ;;GENERAL REGISTER
 (1)            000005                  R5=     %5              ;;GENERAL REGISTER
 (1)            000006                  R6=     %6              ;;GENERAL REGISTER
 (1)            000007                  R7=     %7              ;;GENERAL REGISTER
 (1)                                    .EQUIV  R6,SP           ;;STACK POINTER
 (1)                                    .EQUIV  R7,PC           ;;PROGRAM COUNTER
 (1)
 (1)                                    ;*PRIORITY LEVEL DEFINITIONS
 (1)            000000                  PR0=    0               ;;PRIORITY LEVEL 0
 (1)            000040                  PR1=    40              ;;PRIORITY LEVEL 1
 (1)            000100                  PR2=    100             ;;PRIORITY LEVEL 2
 (1)            000140                  PR3=    140             ;;PRIORITY LEVEL 3
 (1)            000200                  PR4=    200             ;;PRIORITY LEVEL 4
 (1)            000240                  PR5=    240             ;;PRIORITY LEVEL 5
 (1)            000300                  RP6=    300             ;;PRIORITY LEVEL 6
```

```
(1)  000340                  PR7=    340            ;;PRIORITY LEVEL 7
(1)
(1)                          ;*"SWITCH REGISTER" SWITCH DEFINITIONS
(1)  100000                  SW15=   100000
(1)  040000                  SW14=   40000
(1)  020000                  SW13=   20000
(1)  010000                  SW12=   10000
(1)  004000                  SW11=   4000
(1)  002000                  SW10=   2000
(1)  001000                  SW09=   1000
(1)  000400                  SW08=   400
(1)  000200                  SW07=   200
(1)  000100                  SW06=   100
(1)  000040                  SW05=   40
(1)  000020                  SW04=   20
(1)  000010                  SW03=   10
(1)  000004                  SW02=   4
(1)  000002                  SW01=   2
(1)  000001                  SW00=   1
(1)                          .EQUIV  SW09,SW9
(1)                          .EQUIV  SW08,SW8
(1)                          .EQUIV  SW07,SW7
(1)                          .EQUIV  SW06,SW6
(1)                          .EQUIV  SW05,SW5
(1)                          .EQUIV  SW04,SW4
(1)                          .EQUIV  SW03,SW3
(1)                          .EQUIV  SW02,SW2
(1)                          .EQUIV  SW01,SW1
(1)                          .EQUIV  SW00,SW0
(1)
(1)                          ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
(1)  100000                  BIT15=  100000
(1)  040000                  BIT14=  40000
(1)  020000                  BIT13=  20000
(1)  010000                  BIT12=  10000
(1)  004000                  BIT11=  4000
(1)  002000                  BIT10=  2000
(1)  001000                  BIT09=  1000
(1)  000400                  BIT08=  400
(1)  000200                  BIT07=  200
(1)  000100                  BIT06=  100
(1)  000040                  BIT05=  40
(1)  000020                  BIT04=  20
(1)  000010                  BIT03=  10
(1)  000004                  BIT02=  4
(1)  000002                  BIT01=  2
(1)  000001                  BIT00=  1
(1)                          .EQUIV  BIT09,BIT9
(1)                          .EQUIV  BIT08,BIT8
(1)                          .EQUIV  BIT07,BIT7
(1)                          .EQUIV  BIT06,BIT6
(1)                          .EQUIV  BIT05,BIT5
(1)                          .EQUIV  BIT04,BIT4
(1)                          .EQUIV  BIT03,BIT3
```

```
(1)                                 .EQUIV  BIT02,BIT2
(1)                                 .EQUIV  BIT01,BIT1
(1)                                 .EQUIV  BIT00,BIT0
(1)
(1)                                 ;*BASIC "CPU" TRAP VECTOR ADDRESSES
(1)          000004                 ERRVEC= 4                   ;;TIME OUT AND OTHER ERRORS
(1)          000010                 RESVEC= 10                  ;;RESERVED AND ILLEGAL INSTRUCTIONS
(1)          000014                 TBITVEC=14                  ;;"T" BIT
(1)          000014                 TRTVEC= 14                  ;;TRACE TRAP
(1)          000014                 BPTVEC= 14                  ;;BREAKPOINT TRAP (BPT)
(1)          000020                 IOTVEC= 20                  ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
(1)          000024                 PWRVEC= 24                  ;;POWER FAIL
(1)          000030                 EMTVEC= 30                  ;;EMULATOR TRAP (EMT) **ERROR**
(1)          000034                 TRAPVEC=34                  ;;"TRAP" TRAP
(1)          000060                 TKVEC=  60                  ;;TTY KEYBOARD VECTOR
(1)          000064                 TPVEC=  64                  ;;TTY PRINTER VECTOR
(1)          000240                 PIRQVEC=240                 ;;PROGRAM INTERRUPT REQUEST VECTOR
```

```
                                ;OP CODE DEFINITIONS
000101                          NOOP=101
000103                          UNLOAD=103
000105                          SEEK=105
000107                          RECAL=107
000111                          DRVCLR=111
000113                          RELEASE=113
000115                          OFFSET=115
000117                          RTC=117
000121                          READIN=121
000123                          PACK=123
000131                          SEARCH=131
000151                          WRCKD=151
000153                          WRCKHD=153
000161                          WRITE=161
000163                          WRTHD=163
000171                          READ=171
000173                          READHD=173
000141                          GETREG=141
000143                          SETFORM=143
000145                          SELDRV=145

                                ;OTHER EQUATES

165000                          TRCKWC = -<256.*22.>            ;WORD COUNT FOR TRACK
177400                          SCTRWC = -256.                  ;WORD COUNT FOR SECTOR
010000                          FMT22=10000                     ;FORMAT 22 BIT
```

```
5715
 (1)
 (1)                            ;*****************************************************************
 (1)
 (1)                            .SBTTL  COMMON TAGS
 (1)
 (1)                            ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
 (1)                            ;*USED IN THE PROGRAM.
 (1)
 (1)      000046                        .=46
 (1)  000046  013742                    $ENDAD                  ;;LOGICAL END OF PROGRAM
 (1)
 (1)      001100                        .=1100
 (1)
 (1)  001100                    $CMTAG:                         ;;START OF COMMON TAGS
 (1)  001100  000000            $PASS:  .WORD   0               ;;CONTAINS PASS COUNT
 (1)  001102     000            $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
 (1)  001103     000            $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
 (1)  001104  000000            $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
 (1)  001106  000000            $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP
 (1)  001110  000000            $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
 (1)  001112  000000            $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
 (1)  001114     000            $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
 (1)  001115     001            $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
 (1)  001116  000000            $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
 (1)  001120  000000            $GDADR: .WORD   0               ;;CONTAINS  OF 'GOOD' DATA
 (1)  001122  000000            $BDADR: .WORD   0               ;;CONTAINS  OF 'BAD' DATA
 (1)  001124  000000            $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
 (1)  001126  000000            $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
 (1)  001130  000000  000000  000000    .WORD   0,0,0           ;;RESERVED--NOT TO BE USED
 (1)  001136  177560            $TKS:   177560                  ;;TTY KBD STATUS
 (1)  001140  177562            $TKB:   177562                  ;;TTY KBD BUFFER
 (1)  001142  177564            $TPS:   177564                  ;;TTY PRINTER STATUS REG.
 (1)  001144  177566            $TPB:   177566                  ;;TTY PRINTER BUFFER REG.
 (1)  001146     000            $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
 (1)  001147     002            $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
 (1)  001150     012            $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
 (1)  001151     000            $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
 (1)  001152  000000            $REGAD: .WORD   0               ;;CONTAINS THE  FROM
 (1)                                                            ;;WHICH  ($REG0) WAS OBTAINED
 (3)  001154  000000            $REG0:  .WORD   0               ;;CONTAINS (($REGAD)+0)
 (3)  001156  000000            $REG1:  .WORD   0               ;;CONTAINS (($REGAD)+2)
 (3)  001160  000000            $REG2:  .WORD   0               ;;CONTAINS (($REGAD)+4)
 (3)  001162  000000            $REG3:  .WORD   0               ;;CONTAINS (($REGAD)+6)
 (3)  001164  000000            $REG4:  .WORD   0               ;;CONTAINS (($REGAD)+10)
 (3)  001166  000000            $REG5:  .WORD   0               ;;CONTAINS (($REGAD)+12)
 (3)  001170  000000            $TMP0:  .WORD   0               ;;USER DEFINED
 (3)  001172  000000            $TMP1:  .WORD   0               ;;USER DEFINED
 (3)  001174  000000            $TMP2:  .WORD   0               ;;USER DEFINED
 (1)  001176  000000            $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
 (1)  001200  000000            $ESCAPE:0                       ;;ESCAPE ON ERROR
 (1)  001202  177607  000377    $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
 (1)  001206     077            $QUES:  .ASCII  /?/             ;;QUESTION MARK
 (1)  001207     015            $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
 (1)  001210  000012            $LF:    .ASCIZ  <12>            ;;LINE FEED
 (3)  001212  000000            C.SWR:  .WORD   0               ;CONTROL SWITCHES
```

```
(3)  001214  000000                CNTRLC: .WORD   0           ;CONTROL "C" FLAG
(3)  001216  000000                BUSADR: .WORD   0           ;GET ADDRESSES FROM THE TTY FLAG (0=NO, -1=YES)
(3)  001220  000000                LPTAVL: .WORD   0           ;LPT AVAILABLE STATUS (0=NO,1=YES)
(3)  001222  000000                DRVSEL: .WORD   0           ;DRIVES SELECTED FOR TESTING
(3)  001224  037777                TSTNMS: .WORD   37777       ;RUN TESTS 0-15
(3)  001226  000000                CLKSTA: .WORD   0           ;CLOCK STATUS (0=NO CLOCK,+1=KW11-P,
(3)                                                            ;AND -1=KW11-L)
(3)  001230  000020                TICKMS: .WORD   ^D16        ;16 MILLISECONDS PER CLOCK TICK
(3)  001232  040432                TICKUS: .WORD   ^D16666     ;16666 MIRCOSECONDS PER CLOCK TICK
(3)  001234  000000                BYPASS: .WORD   0
(3)  001236  000000                CHKDRV: .WORD   0           ;DRIVE UNDER TEST
(3)  001240  000000                DRVMSK: .WORD   0           ;DRIVE MASK BIT
(3)  001242  000000                SVSTAT: .WORD   0           ;STATUS/ERROR INDICATOR IS
(3)                                                            ;SAVED HERE ON AN ERROR
(3)  001244  000000                CYL.RD: .WORD   0           ;CYLINDER READ
(3)  001246  000000                TRK.RD: .WORD   0           ;TRACK READ
(3)  001250  000000                SEC.RD: .WORD   0           ;SECTOR READ
(3)  001252  000000                CYL.DS: .WORD   0           ;CYLINDER DESIRED
(3)  001254  000000                SEC.DS: .WORD   0           ;SECTOR DESIRED
(3)  001256  000000                TRK.DS: .WORD   0           ;TRACK DESIRED
(3)  001260  000000                TIM.UP: .WORD   0           ;MINIMUM TIME
(3)  001262  000000                        .WORD   0           ;NUMBER OF COUNTS BELOW MIN. LIMIT
(3)  001264  000000                        .WORD   0           ;MAXIMUM TIME
(3)  001266  000000                        .WORD   0           ;NUMBER OF COUNTS ABOVE MAX. LIMIT
(3)  001270  000000  000000               .WORD   0,0         ;TOTAL TIME OF ALL SEEKS
(3)  001274  000000                        .WORD   0           ;NUMBER OF SEEKS PERFORMED
(3)  001276  000000                TIM.DN: .WORD   0           ;MINIMUM TIME
(3)  001300  000000                        .WORD   0           ;NUMBER OF COUNTS BELOW MIN. LIMIT
(3)  001302  000000                        .WORD   0           ;MAXIMUM TIME
(3)  001304  000000                        .WORD   0           ;NUMBER OF COUNTS ABOVE MAX. LIMIT
(3)  001306  000000  000000               .WORD   0,0         ;TOTAL TIME OF ALL SEEKS
(3)  001312  000000                        .WORD   0           ;NUMBER OF SEEKS PERFORMED
(3)  001314  000000                TIM.PT: .WORD   0           ;POINTS TO TABLE OF TIMES
(3)  001316  000000                WCEFLG: .WORD   0           ;FATAL WRITE CHECK ERROR FLAG (TST 15)
(3)  001320  000000                STALLO: .WORD   0           ;VARIABLE STALL (TEST 4)
(3)  001322  000000  000000        SVADR:  .WORD   0,0         ;SAVE DISK ADDRESS (TST16)
(3)
(3)                                ;CONSTANTS STORAGE
(3)  001326  176700                RH.ADR: .WORD   176700      ;RH11/RP04 CONTROL AND STATUS REG. #1
(3)  001330  000254  000240        RHVEC:  .WORD   254,5*32.   ;RH11/RP04 VECTOR AND PRIORITY
(3)  001334  000104  000106        PKV:    .WORD   104,106     ;KW11-P VECTOR ADDRESS
(3)  001340  172540                PKCS:   .WORD   172540      ;KW11-P CONTROL AND STATUS REG.
(3)  001342  172542                PKB:    .WORD   172542      ;KW11-P COUNT SET BUFFER
(3)  001344  172544                PKC:    .WORD   172544      ;KW11-P COUNTER
(3)  001346  000100  000102        LKV:    .WORD   100,102     ;KW11-L VECTOR ADDRESS
(3)  001352  177546                LKS:    .WORD   177546      ;KW11-L STATUS REGISTER
(3)  001354  177564                TPS:    .WORD   177564      ;TTY PRINTER STATUS
(3)  001356  177566                TPB:    .WORD   177566      ;TTY PRINTER BUFFER
(3)  001360  177514                LPS:    .WORD   177514      ;LINE PRINTER STATUS
(3)  001362  177516                LPB:    .WORD   177516      ;LINE PRINTER BUFFER
(3)  001364  000012                STALL1: .WORD   ^D10        ;10 MILLISECONDS STALL (TEST 0-6)
(3)  001366  000012                STALL2: .WORD   ^D10        ;10 MILLISECONDS STALL (TEST 13-16)
(3)  001370  000031                MXSTAL: .WORD   ^D25        ;MAX. INCREMENTING STALL ALLOWED IN TEST 4
(3)  001372     006                ERR.CT: .BYTE   6           ;NUMBER OF ERRORS ALLOWED IN TESTS 13-16
```

```
(3)
(3)  001373  000                        .BYTE   0               ;BEFORE GOING TO THE NEXT TEST
(3)                                                             ;RESERVED
(3)
(3)                                 ;BIT TABLE
(3)  001374  000001                 BITS:   .WORD   BIT00
(3)  001376  000002                         .WORD   BIT01
(3)  001400  000004                         .WORD   BIT02
(3)  001402  000010                         .WORD   BIT03
(3)  001404  000020                         .WORD   BIT04
(3)  001406  000040                         .WORD   BIT05
(3)  001410  000100                         .WORD   BIT06
(3)  001412  000200                         .WORD   BIT07
(3)  001414  000400                         .WORD   BIT08
(3)  001416  001000                         .WORD   BIT09
(3)  001420  002000                         .WORD   BIT10
(3)  001422  004000                         .WORD   BIT11
(3)  001424  010000                         .WORD   BIT12
(3)  001426  020000                         .WORD   BIT13
(3)  001430  040000                         .WORD   BIT14
(3)  001432  100000                         .WORD   BIT15
(3)
(3)
(3)                                 ;COMMON STORAGE FOR TEST PARAMETER
(3)  001434  000000                 PRM:    .WORD   0
(3)  001436  000000                 RPT:    .WORD   0               ;REPEAT COUNTS FOR ALL TESTS
(3)  001440  000000                 FC:     .WORD   0               ;FIRST CYLINDER
(3)  001442  000000                 LC:     .WORD   0               ;LAST CYLINDER
(3)  001444  000000                 IC:     .WORD   0               ;INCREMENT CYLINDER
(3)  001446  000000                 FT:     .WORD   0               ;FIRST TRACK
(3)  001450  000000                 LT:     .WORD   0               ;LAST TRACK
(3)  001452  000000                 IT:     .WORD   0               ;INCREMENT TRACK
(3)  001454  000000                 FS:     .WORD   0               ;FIRST SECTOR
(3)  001456  000000                 LS:     .WORD   0               ;LAST SECTOR
(3)  001460  000000                 PAT:    .WORD   0               ;PATTERN CODE
(3)
(3)
(3)                                 ;TABLE OF PARAMETER POINTERS
(3)  001462  002062                 PRMPT:  .WORD   PRM0
(3)  001464  002074                         .WORD   PRM1
(3)  001466  002116                         .WORD   PRM2
(3)  001470  002134                         .WORD   PRM3
(3)  001472  002152                         .WORD   PRM4
(3)  001474  002170                         .WORD   PRM5
(3)  001476  002206                         .WORD   PRM6
(3)  001500  002224                         .WORD   PRM7
(3)  001502  002236                         .WORD   PRM10
(3)  001504  002242                         .WORD   PRM11
(3)  001506  002246                         .WORD   PRM12
(3)  001510  002252                         .WORD   PRM13
(3)  001512  002262                         .WORD   PRM14
(3)  001514  002272                         .WORD   PRM15
(3)  001516  002320                         .WORD   PRM16
(3)
(3)
(3)                                 ;PARAMETER UPPER LIMIT
(3)  001520  077777                 PRMLMT: .WORD   ^D32767         ;"R"
```

```
(3)  001522  000632                        .WORD   ↑D410           ;"FC"
(3)  001524  000632                        .WORD   ↑D410           ;"LC"
(3)  001526  000632                        .WORD   ↑D410           ;"IC"
(3)  001530  000022                        .WORD   ↑D18            ;"FT"
(3)  001532  000022                        .WORD   ↑D18            ;"LT"
(3)  001534  000022                        .WORD   ↑D18            ;"IT"
(3)  001536  000025                        .WORD   ↑D21            ;"FS"
(3)  001540  000025                        .WORD   ↑D21            ;"LS"
(3)  001542  177777                        .WORD   177777          ;"PAT"
(3)
(3)                                 ;TABLE OF MESSAGE POINTERS
(3)  001544  034376                 PRMMSG: .WORD   MSG.R
(3)  001546  034400                         .WORD   MSG.FC
(3)  001550  034403                         .WORD   MSG.LC
(3)  001552  034406                         .WORD   MSG.IC
(3)  001554  034411                         .WORD   MSG.FT
(3)  001556  034414                         .WORD   MSG.LT
(3)  001560  034417                         .WORD   MSG.IT
(3)  001562  034422                         .WORD   MSG.FS
(3)  001564  034425                         .WORD   MSG.LS
(3)
(3)                                 ;STATUS/ERROR INDICATOR MESSAGES POINTER TABLE
(3)  001566  036021                 STATBL: .WORD   MSGB14          ;OFFLINE OR UNSAFE DRIVE REQUESTED
(3)  001570  036063                         .WORD   MSGB13          ;UNLOAD DRIVE REQUESTED
(3)  001572  036114                         .WORD   MSGB12          ;PERSISTENT UNSAFE
(3)  001574  036136                         .WORD   MSGB11          ;PARITY ERROR OCCURRED
(3)  001576  036164                         .WORD   MSGB10          ;FATAL PARITY ERROR
(3)  001600  036207                         .WORD   MSGB09          ;SOFTWARE TIMEOUT ON THIS DRIVE
(3)  001602  036246                         .WORD   MSGB08          ;SOFTWARE TIMEOUT ON ANOTHER DRIVE
(3)  001604  036310                         .WORD   MSGB06          ;ERROR OCCURRED DURING I/O OPERATION
(3)  001606  036354                         .WORD   MSGB05          ;ERROR OCCURRED DURING NON-I/O OPERATION
(3)  001610  036424                         .WORD   MSGB04          ;UNSAFE OCCURRED
(3)  001612  036444                         .WORD   MSGB03          ;AUTOMATIC RECALIBRATE SEQUENCE OCCURRED
(3)
(3)                                 ;DEFAULT VALUES OF TESTS PARAMETERS
(3)  001614  000225  000012  000632 DFLT:   .WORD   225,10.,410.,0,0 ;RECAL/SEEK (T0)
(3)  001622  000000  000000
(3)  001626  000677  000144  000000         .WORD   677,100.,0,128.,0,0,0,0,0 ;SEEK/SEEK (T1)
(3)  001634  000200  000000  000000
(3)  001642  000000  000000  000000
(3)  001650  000237  000001  000000         .WORD   237,1,0,410.,1,0,0 ;INCREMENT SEEK (T2)
(3)  001656  000632  000001  000000
(3)  001664  000000
(3)  001666  000237  000001  000000         .WORD   237,1,0,256.,1,0,0 ;STEPPING SEEK (T3)
(3)  001674  000400  000001  000000
(3)  001702  000000
(3)  001704  000237  000001  000000         .WORD   237,1,0,410.,1,0,0 ;OSCILLATING SEEK (T4)
(3)  001712  000632  000001  000000
(3)  001720  000000
(3)  001722  000237  000001  000000         .WORD   237,1,0,410.,1,0,0 ;CONVERGING/DIVERGING SEEK (T5)
(3)  001730  000632  000001  000000
(3)  001736  000000
(3)  001740  000237  000001  000000         .WORD   237,1,0,410.,1,0,0 ;SERVO ADDRESSING LOGIC NOISE (T6)
(3)  001746  000632  000001  000000
```

```
(3)  001754  000000
(3)  001756  000223  000001  000000          .WORD   223,1,0,0,0     ;ROTATIONAL SPEED TIMING TEST (T7)
(3)  001764  000000  000000
(3)  001770  000001  000001                  .WORD   1,1             ;ONE CYLINDER SEEK TIMING TEST (T10)
(3)  001774  000001  000001                  .WORD   1,1             ;AVERAGE SEEK TIMING TEST (T11)
(3)  002000  000001  000001                  .WORD   1,1             ;MAXIMUM SEEK TIMEING TEST (T12)
(3)  002004  000023  000001  000000          .WORD   23,1,0,0        ;SECTOR ADDRESSING TEST (T13)
(3)  002012  000000
(3)  002014  000203  000001  000000          .WORD   203,1,0,0       ;TRACK ADDRESSING TEST (T14)
(3)  002022  000000
(3)  002024  001777  000001  000000          .WORD   1777,1,0,410.,64.,0,18.,1,1,0,177777 ;DATA TEST (T15)
(3)  002032  000632  000100  000000
(3)  002040  000022  000001  000001
(3)  002046  000000  177777
(3)  002052  000007  047040  000000          .WORD   7,20000.,0,410. ;EXERCISER (T16)
(3)  002060  000632
(3)
(3)                                  ;PARAMETER TABLES
(3)
(3)                                  ;RECAL/SEEK (T0)
(3)  002062  000225                  PRM0:   .WORD   225
(3)  002064  000012                  RPT0:   .WORD   10.
(3)  002066  000632                  LC0:    .WORD   410.
(3)  002070  000000                  FT0:    .WORD   0
(3)  002072  000000                  FS0:    .WORD   0
(3)
(3)                                  ;SEEK/SEEK (T1)
(3)  002074  000677                  PRM1:   .WORD   677
(3)  002076  000144                  RPT1:   .WORD   100.
(3)  002100  000000                  FC1:    .WORD   0
(3)  002102  000200                  LC1:    .WORD   128.
(3)  002104  000000                  IC1:    .WORD   0
(3)  002106  000000                  FT1:    .WORD   0
(3)  002110  000000                  LT1:    .WORD   0
(3)  002112  000000                  FS1:    .WORD   0
(3)  002114  000000                  LS1:    .WORD   0
(3)
(3)                                  ;INCREMENT SEEK (T2)
(3)  002116  000237                  PRM2:   .WORD   237
(3)  002120  000001                  RPT2:   .WORD   1
(3)  002122  000000                  FC2:    .WORD   0
(3)  002124  000632                  LC2:    .WORD   410.
(3)  002126  000001                  IC2:    .WORD   1
(3)  002130  000000                  FT2:    .WORD   0
(3)  002132  000000                  FS2:    .WORD   0
(3)
(3)                                  ;STEPPING SEEK (T3)
(3)  002134  000237                  PRM3:   .WORD   237
(3)  002136  000001                  RPT3:   .WORD   1
(3)  002140  000000                  FC3:    .WORD   0
(3)  002142  000400                  LC3:    .WORD   256.
(3)  002144  000001                  IC3:    .WORD   1
(3)  002146  000000                  FT3:    .WORD   0
(3)  002150  000000                  FS3:    .WORD   0
```

```
(3)
(3)                                     ;OSCILLATING SEEK (T4)
(3)  002152  000237                     PRM4:   .WORD   237
(3)  002154  000001                     RPT4:   .WORD   1
(3)  002156  000000                     FC4:    .WORD   0
(3)  002160  000632                     LC4:    .WORD   410.
(3)  002162  000001                     IC4:    .WORD   1
(3)  002164  000000                     FT4:    .WORD   0
(3)  002166  000000                     FS4:    .WORD   0
(3)
(3)                                     ;CONVERGING/DIVERGING SEEK (T5)
(3)  002170  000237                     PRM5:   .WORD   237
(3)  002172  000001                     RPT5:   .WORD   1
(3)  002174  000000                     FC5:    .WORD   0
(3)  002176  000632                     LC5:    .WORD   410.
(3)  002200  000001                     IC5:    .WORD   1
(3)  002202  000000                     FT5:    .WORD   0
(3)  002204  000000                     FS5:    .WORD   0
(3)
(3)                                     ;SERVO ADDRESSING LOGIC NOISE GENERATOR (T6)
(3)  002206  000237                     PRM6:   .WORD   237
(3)  002210  000001                     RPT6:   .WORD   1
(3)  002212  000000                     FC6:    .WORD   0
(3)  002214  000632                     LC6:    .WORD   410.
(3)  002216  000001                     IC6:    .WORD   1
(3)  002220  000000                     FT6:    .WORD   0
(3)  002222  000000                     FS6:    .WORD   0
(3)
(3)                                     ;ROTATIONAL SPEED TIMING TEST (T7)
(3)  002224  000223                     PRM7:   .WORD   223
(3)  002226  000001                     RPT7:   .WORD   1
(3)  002230  000000                     FC7:    .WORD   0
(3)  002232  000000                     FT7:    .WORD   0
(3)  002234  000000                     FS7:    .WORD   0
(3)
(3)                                     ;ONE CYLINDER SEEK TIMING TEST (T10)
(3)  002236  000001                     PRM10:  .WORD   1
(3)  002240  000001                     RPT10:  .WORD   1
(3)
(3)                                     ;AVERAGE SEEK TIMING TEST (T11)
(3)  002242  000001                     PRM11:  .WORD   1
(3)  002244  000001                     RPT11:  .WORD   1
(3)
(3)                                     ;MAXIMUM SEEK TIMING TEST (T12)
(3)  002246  000001                     PRM12:  .WORD   1
(3)  002250  000001                     RPT12:  .WORD   1
(3)
(3)                                     ;SECTOR ADDRESSING TEST (T13)
(3)  002252  000023                     PRM13:  .WORD   23
(3)  002254  000001                     RPT13:  .WORD   1
(3)  002256  000000                     FC13:   .WORD   0
(3)  002260  000000                     FT13:   .WORD   0
(3)
(3)                                     ;TRACK ADDRESSING TEST (T14)
```

```
(3)  002262  000203                PRM14:  .WORD   203
(3)  002264  000001                RPT14:  .WORD   1
(3)  002266  000000                FC14:   .WORD   0
(3)  002270  000000                FS14:   .WORD   0
(3)
(3)                                ;DATA TEST (T15)
(3)  002272  001777                PRM15:  .WORD   1777
(3)  002274  000001                RPT15:  .WORD   1
(3)  002276  000000                FC15:   .WORD   0
(3)  002300  000632                LC15:   .WORD   410.
(3)  002302  000100                IC15:   .WORD   64.
(3)  002304  000000                FT15:   .WORD   0
(3)  002306  000022                LT15:   .WORD   18.
(3)  002310  000001                IT15:   .WORD   1
(3)  002312  000001                FS15:   .WORD   1
(3)  002314  000000                LS15:   .WORD   0
(3)  002316  177777                PTRN15: .WORD   177777
(3)
(3)                                ;EXERCISER (T16)
(3)  002320  000007                PRM16:  .WORD   7
(3)  002322  047040                RPT16:  .WORD   20000.
(3)  002324  000000                FC16:   .WORD   0
(3)  002326  000632                LC16:   .WORD   410.
(3)
(3)                                ;SEEK TIMING LIMITS
(3)
(3)  002330  034516                T7A:    .WORD   MSG7
(3)  002332  000000                        .WORD   0
(3)  002334  003142                        .WORD   1634.           ;(16.67-2%)**2
(3)  002336  003244                        .WORD   1700.           ;(16.67+2%)**2
(3)
(3)  002340  034516                T7B:    .WORD   MSG7
(3)  002342  000000                        .WORD   0
(3)  002344  003131                        .WORD   1625.           ;(16.67-2.5%)**2
(3)  002346  003255                        .WORD   1709.           ;(16.67+2.5%)**2
(3)
(3)  002350  034550                T10:    .WORD   MSG10A
(3)  002352  034631                        .WORD   MSG10B
(3)  002354  000454                        .WORD   300.            ;(7-4)**2
(3)  002356  001750                        .WORD   1000.           ;(7+3)**2
(3)
(3)  002360  034664                T11:    .WORD   MSG11A
(3)  002362  034732                        .WORD   MSG11B
(3)  002364  005050                        .WORD   2600.           ;(28-2)**2
(3)  002366  005670                        .WORD   3000.           ;(28+2)**2
(3)
(3)  002370  034753                T12:    .WORD   MSG12A
(3)  002372  035021                        .WORD   MSG12B
(3)  002374  010770                        .WORD   4600.           ;(50-4)**2
(3)  002376  012120                        .WORD   5200.           ;(50+2)**2
(3)
(3)  002400  002440                PAT.PT: .WORD   PAT0            ;TABLE OF POINTERS WHICH POINT TO THE
(3)  002402  002500                        .WORD   PAT1            ;PATTERNS USED BY THE DATA TEST
(3)  002404  002540                        .WORD   PAT2
```

```
(3)  002406  002600                        .WORD   PAT3
(3)  002410  002640                        .WORD   PAT4
(3)  002412  002700                        .WORD   PAT5
(3)  002414  002740                        .WORD   PAT6
(3)  002416  003000                        .WORD   PAT7
(3)  002420  003040                        .WORD   PAT8
(3)  002422  003100                        .WORD   PAT9
(3)  002424  003140                        .WORD   PAT10
(3)  002426  003200                        .WORD   PAT11
(3)  002430  003240                        .WORD   PAT12
(3)  002432  003300                        .WORD   PAT13
(3)  002434  003340                        .WORD   PAT14
(3)  002436  003400                        .WORD   PAT15
(3)
(3)                                 ;PATTERNS 0 THRU 15
(3)
(3)  002440  165555                 PAT0:  .WORD   165555          ;PATTERN 0
(3)  002442  133333                        .WORD   133333
(3)  002444  165555                        .WORD   165555
(3)  002446  133333                        .WORD   133333
(3)  002450  165555                        .WORD   165555
(3)  002452  133333                        .WORD   133333
(3)  002454  165555                        .WORD   165555
(3)  002456  133333                        .WORD   133333
(3)  002460  165555                        .WORD   165555
(3)  002462  133333                        .WORD   133333
(3)  002464  165555                        .WORD   165555
(3)  002466  133333                        .WORD   133333
(3)  002470  165555                        .WORD   165555
(3)  002472  133333                        .WORD   133333
(3)  002474  165555                        .WORD   165555
(3)  002476  133333                        .WORD   133333
(3)
(3)  002500  000001                 PAT1:  .WORD   000001          ;PATTERN 1
(3)  002502  000003                        .WORD   000003
(3)  002504  000007                        .WORD   000007
(3)  002506  000017                        .WORD   000017
(3)  002510  000037                        .WORD   000037
(3)  002512  000077                        .WORD   000077
(3)  002514  000177                        .WORD   000177
(3)  002516  000377                        .WORD   000377
(3)  002520  000777                        .WORD   000777
(3)  002522  001777                        .WORD   001777
(3)  002524  003777                        .WORD   003777
(3)  002526  007777                        .WORD   007777
(3)  002530  017777                        .WORD   017777
(3)  002532  037777                        .WORD   037777
(3)  002534  077777                        .WORD   077777
(3)  002536  177777                        .WORD   177777
(3)
(3)  002540  177776                 PAT2:  .WORD   177776          ;PATTERN 2
(3)  002542  177774                        .WORD   177774
(3)  002544  177770                        .WORD   177770
(3)  002546  177760                        .WORD   177760
```

```
(3)  002550  177740                        .WORD   177740
(3)  002552  177700                        .WORD   177700
(3)  002554  177600                        .WORD   177600
(3)  002556  177400                        .WORD   177400
(3)  002560  177000                        .WORD   177000
(3)  002562  176000                        .WORD   176000
(3)  002564  174000                        .WORD   174000
(3)  002566  170000                        .WORD   170000
(3)  002570  160000                        .WORD   160000
(3)  002572  140000                        .WORD   140000
(3)  002574  100000                        .WORD   100000
(3)  002576  000000                        .WORD   000000
(3)
(3)  002600  000000                PAT3:   .WORD   000000          ;PATTERN 3
(3)  002602  000000                        .WORD   000000
(3)  002604  000000                        .WORD   000000
(3)  002606  177777                        .WORD   177777
(3)  002610  177777                        .WORD   177777
(3)  002612  177777                        .WORD   177777
(3)  002614  000000                        .WORD   000000
(3)  002616  000000                        .WORD   000000
(3)  002620  177777                        .WORD   177777
(3)  002622  177777                        .WORD   177777
(3)  002624  000000                        .WORD   000000
(3)  002626  177777                        .WORD   177777
(3)  002630  000000                        .WORD   000000
(3)  002632  177777                        .WORD   177777
(3)  002634  000000                        .WORD   000000
(3)  002636  177777                        .WORD   177777
(3)
(3)  002640  000000                PAT4:   .WORD   000000          ;PATTERN 4
(3)  002642  010421                        .WORD   010421
(3)  002644  021042                        .WORD   021042
(3)  002646  031463                        .WORD   031463
(3)  002650  042104                        .WORD   042104
(3)  002652  052525                        .WORD   052525
(3)  002654  063146                        .WORD   063146
(3)  002656  073567                        .WORD   073567
(3)  002660  104210                        .WORD   104210
(3)  002662  114631                        .WORD   114631
(3)  002664  125252                        .WORD   125252
(3)  002666  135673                        .WORD   135673
(3)  002670  146314                        .WORD   146314
(3)  002672  156735                        .WORD   156735
(3)  002674  167356                        .WORD   167356
(3)  002676  177777                        .WORD   177777
(3)
(3)  002700  052525                PAT5:   .WORD   052525          ;PATTERN 5
(3)  002702  052525                        .WORD   052525
(3)  002704  052525                        .WORD   052525
(3)  002706  125252                        .WORD   125252
(3)  002710  125252                        .WORD   125252
(3)  002712  125252                        .WORD   125252
(3)  002714  052525                        .WORD   052525
```

```
(3)  002716  052525                    .WORD   052525
(3)  002720  125252                    .WORD   125252
(3)  002722  125252                    .WORD   125252
(3)  002724  052525                    .WORD   052525
(3)  002726  125252                    .WORD   125252
(3)  002730  052525                    .WORD   052525
(3)  002732  125252                    .WORD   125252
(3)  002734  052525                    .WORD   052525
(3)  002736  125252                    .WORD   125252
(3)
(3)  002740  007417            PAT6:   .WORD   007417          ;PATTERN 6
(3)  002742  007417                    .WORD   007417
(3)  002744  007417                    .WORD   007417
(3)  002746  170360                    .WORD   170360
(3)  002750  170360                    .WORD   170360
(3)  002752  170360                    .WORD   170360
(3)  002754  007417                    .WORD   007417
(3)  002756  007417                    .WORD   007417
(3)  002760  170360                    .WORD   170360
(3)  002762  170360                    .WORD   170360
(3)  002764  007417                    .WORD   007417
(3)  002766  170360                    .WORD   170360
(3)  002770  007417                    .WORD   007417
(3)  002772  170360                    .WORD   170360
(3)  002774  007417                    .WORD   007417
(3)  002776  170360                    .WORD   170360
(3)
(3)  003000  026455            PAT7:   .WORD   026455          ;PATTERN 7
(3)  003002  026455                    .WORD   026455
(3)  003004  026455                    .WORD   026455
(3)  003006  151322                    .WORD   151322
(3)  003010  151322                    .WORD   151322
(3)  003012  151322                    .WORD   151322
(3)  003014  026455                    .WORD   026455
(3)  003016  026455                    .WORD   026455
(3)  003020  151322                    .WORD   151322
(3)  003022  151322                    .WORD   151322
(3)  003024  026455                    .WORD   026455
(3)  003026  151322                    .WORD   151322
(3)  003030  026455                    .WORD   026455
(3)  003032  151322                    .WORD   151322
(3)  003034  026455                    .WORD   026455
(3)  003036  151322                    .WORD   151322
(3)
(3)  003040  165555            PAT8:   .WORD   165555          ;PATTERN 8
(3)  003042  133333                    .WORD   133333
(3)  003044  165555                    .WORD   165555
(3)  003046  133333                    .WORD   133333
(3)  003050  165555                    .WORD   165555
(3)  003052  133333                    .WORD   133333
(3)  003054  165555                    .WORD   165555
(3)  003056  133333                    .WORD   133333
(3)  003060  165555                    .WORD   165555
(3)  003062  133333                    .WORD   133333
```

```
(3)  003064  165555                    .WORD   165555
(3)  003066  133333                    .WORD   133333
(3)  003070  165555                    .WORD   165555
(3)  003072  133333                    .WORD   133333
(3)  003074  165555                    .WORD   165555
(3)  003076  133333                    .WORD   133333
(3)
(3)  003100  000001          PAT9:     .WORD   000001          ;PATTERN 9
(3)  003102  000002                    .WORD   000002
(3)  003104  000004                    .WORD   000004
(3)  003106  000010                    .WORD   000010
(3)  003110  000020                    .WORD   000020
(3)  003112  000040                    .WORD   000040
(3)  003114  000100                    .WORD   000100
(3)  003116  000200                    .WORD   000200
(3)  003120  000400                    .WORD   000400
(3)  003122  001000                    .WORD   001000
(3)  003124  002000                    .WORD   002000
(3)  003126  004000                    .WORD   004000
(3)  003130  010000                    .WORD   010000
(3)  003132  020000                    .WORD   020000
(3)  003134  040000                    .WORD   040000
(3)  003136  100000                    .WORD   100000
(3)
(3)  003140  177776          PAT10:    .WORD   177776          ;PATTERN 10
(3)  003142  177775                    .WORD   177775
(3)  003144  177773                    .WORD   177773
(3)  003146  177767                    .WORD   177767
(3)  003150  177757                    .WORD   177757
(3)  003152  177737                    .WORD   177737
(3)  003154  177677                    .WORD   177677
(3)  003156  177577                    .WORD   177577
(3)  003160  177377                    .WORD   177377
(3)  003162  176777                    .WORD   176777
(3)  003164  175777                    .WORD   175777
(3)  003166  173777                    .WORD   173777
(3)  003170  167777                    .WORD   167777
(3)  003172  157777                    .WORD   157777
(3)  003174  137777                    .WORD   137777
(3)  003176  077777                    .WORD   077777
(3)
(3)  003200  172666          PAT11:    .WORD   172666          ;PATTERN 11
(3)  003202  155555                    .WORD   155555
(3)  003204  172666                    .WORD   172666
(3)  003206  155555                    .WORD   155555
(3)  003210  172666                    .WORD   172666
(3)  003212  155555                    .WORD   155555
(3)  003214  172666                    .WORD   172666
(3)  003216  155555                    .WORD   155555
(3)  003220  172666                    .WORD   172666
(3)  003222  155555                    .WORD   155555
(3)  003224  172666                    .WORD   172666
(3)  003226  155555                    .WORD   155555
(3)  003230  172666                    .WORD   172666
```

```
(3)  003232  155555                        .WORD   155555
(3)  003234  172666                        .WORD   172666
(3)  003236  155555                        .WORD   155555
(3)
(3)  003240  077777              PAT12:    .WORD   077777          ;PATTERN 12
(3)  003242  137777                        .WORD   137777
(3)  003244  157777                        .WORD   157777
(3)  003246  167777                        .WORD   167777
(3)  003250  173777                        .WORD   173777
(3)  003252  175777                        .WORD   175777
(3)  003254  176777                        .WORD   176777
(3)  003256  177377                        .WORD   177377
(3)  003260  177577                        .WORD   177577
(3)  003262  177677                        .WORD   177677
(3)  003264  177737                        .WORD   177737
(3)  003266  177757                        .WORD   177757
(3)  003270  177767                        .WORD   177767
(3)  003272  177773                        .WORD   177773
(3)  003274  177775                        .WORD   177775
(3)  003276  177776                        .WORD   177776
(3)
(3)  003300  153333              PAT13:    .WORD   153333          ;PATTERN 13
(3)  003302  066667                        .WORD   066667
(3)  003304  153333                        .WORD   153333
(3)  003306  066667                        .WORD   066667
(3)  003310  153333                        .WORD   153333
(3)  003312  066667                        .WORD   066667
(3)  003314  153333                        .WORD   153333
(3)  003316  066667                        .WORD   066667
(3)  003320  153333                        .WORD   153333
(3)  003322  066667                        .WORD   066667
(3)  003324  153333                        .WORD   153333
(3)  003326  066667                        .WORD   066667
(3)  003330  153333                        .WORD   153333
(3)  003332  066667                        .WORD   066667
(3)  003334  153333                        .WORD   153333
(3)  003336  066667                        .WORD   066667
(3)
(3)  003340  000000              PAT14:    .WORD   000000          ;PATTERN 14
(3)  003342  177777                        .WORD   177777
(3)  003344  177777                        .WORD   177777
(3)  003346  177777                        .WORD   177777
(3)  003350  177777                        .WORD   177777
(3)  003352  177777                        .WORD   177777
(3)  003354  177777                        .WORD   177777
(3)  003356  177777                        .WORD   177777
(3)  003360  177777                        .WORD   177777
(3)  003362  177777                        .WORD   177777
(3)  003364  177777                        .WORD   177777
(3)  003366  177777                        .WORD   177777
(3)  003370  177777                        .WORD   177777
(3)  003372  177777                        .WORD   177777
(3)  003374  177777                        .WORD   177777
(3)  003376  177777                        .WORD   177777
```

```
(3)
(3)  003400  177777          PAT15:  .WORD   177777          ;PATTERN 15
(3)  003402  000000                  .WORD   000000
(3)  003404  000000                  .WORD   000000
(3)  003406  000000                  .WORD   000000
(3)  003410  000000                  .WORD   000000
(3)  003412  000000                  .WORD   000000
(3)  003414  000000                  .WORD   000000
(3)  003416  000000                  .WORD   000000
(3)  003420  000000                  .WORD   000000
(3)  003422  000000                  .WORD   000000
(3)  003424  000000                  .WORD   000000
(3)  003426  000000                  .WORD   000000
(3)  003430  000000                  .WORD   000000
(3)  003432  000000                  .WORD   000000
(3)  003434  000000                  .WORD   000000
(3)  003436  000000                  .WORD   000000
(3)
(3)                           ;DPB (DATA PARAMETER BLOCK)
(3)
(3)  003440      000          DPB.A:  .BYTE   0               ;(0) DRIVE NUMBER
(3)  003441      000                  .BYTE   0               ;(1) OFFSET VALUE OR FMT22, ECT, AND HCI
(3)  003442      000                  .BYTE   3               ;(2) COMMAND
(3)  003443      000                  .BYTE   0               ;(3) PSEL AND A17 AND A16
(3)  003444  000000                   .WORD   0               ;(4) WORD COUNT (MUST BE NEG.)
(3)  003446  040524                   .WORD   BUFFER          ;(6) BUFFER ADDRESS OR
(3)                                                           ;REGISTER TABLE POINTER
(3)  003450      000                  .BYTE   0               ;(10) SECTOR ADDRESS OR
(3)                                                           ;FIRST REG. INDEX
(3)  003451      000                  .BYTE   0               ;(11) TRACK ADDRESS OR
(3)                                                           ;LAST REG. INDEX
(3)  003452  000000                   .WORD   0               ;(12) CYLINDER ADDRESS
(3)  003454  003540                   .WORD   RP.REG          ;(14) ERROR TABLE POINTER
(3)                                                           ;POINTS TO THE FIRST OF TWENTY
(3)                                                           ;LOCATIONS OF WHERE THE DRIVER
(3)                                                           ;IS TO STORE THE RH11/RP04
(3)                                                           ;REGISTERS ON AN ERROR. IF LEFT
(3)                                                           ;ZERO REGISTERS ARE NOT SAVED.
(3)  003456  000000                   .WORD   0               ;(16) STATUS/ERROR INDICATOR
(3)                                                           ;BIT15=1=>ERROR OCCURRED
(3)                                                           ;BIT07=1=>DONE
(3)                                                           ;BIT14-BIT09 AND BIT06-BIT03
(3)                                                           ;INDICATE TYPE OF ERROR
(3)
(3)  003460      000          DPB.B:  .BYTE   0               ;(0) DRIVE NUMBER
(3)  003461      000                  .BYTE   0               ;(1) OFFSET VALUE OR FMT22, ECT, AND HCI
(3)  003462      000                  .BYTE   0               ;(2) COMMAND
(3)  003463      000                  .BYTE   0               ;(3) PSEL AND A17 AND A16
(3)  003464  177776                   .WORD   -2              ;(4) WORD COUNT (MUST BE NEG.)
(3)  003466  040524                   .WORD   BUFFER          ;(6) BUFFER ADDRESS OR
(3)                                                           ;REGISTER TABLE POINTER
(3)  003470      000                  .BYTE   0               ;(10) SECTOR ADDRESS OR
(3)                                                           ;FIRST REG. INDEX
(3)  003471      000                  .BYTE   0               ;(11) TRACK ADDRESS OR
```

```
(3)
(3)  003472  000000                        .WORD   0           ;LAST REG. INDEX
(3)  003474  003540                        .WORD   RP.REG      ;(12) CYLINDER ADDRESS
(3)                                                            ;(14) ERROR TABLE POINTER
(3)                                                            ;POINTS TO THE FIRST OF TWENTY
(3)                                                            ;LOCATIONS OF WHERE THE DRIVER
(3)                                                            ;IS TO STORE THE RH11/RP04
(3)                                                            ;REGISTERS ON AN ERROR. IF LEFT
(3)                                                            ;ZERO REGISTERS ARE NOT SAVED.
(3)  003476  000000                        .WORD   0           ;(16) STATUS/ERROR INDICATOR
(3)                                                            ;BIT15=1=>ERROR OCCURRED
(3)                                                            ;BIT07=1=>DONE
(3)                                                            ;BIT14-BIT09 AND BIT06-BIT03
(3)                                                            ;INDICATE TYPE OF ERROR
(3)
(3)  003500     000        DPB.C:  .BYTE   0           ;(0) DRIVE NUMBER
(3)  003501     000                .BYTE   0           ;(1) OFFSET VALUE OR FMT22, ECT, AND HCI
(3)  003502     000                .BYTE   0           ;(2) COMMAND
(3)  003503     000                .BYTE   0           ;(3) PSEL AND A17 AND A16
(3)  003504  177776                .WORD   -2          ;(4) WORD COUNT (MUST BE NEG.)
(3)  003506  040524                .WORD   BUFFER      ;(6) BUFFER ADDRESS OR
(3)                                                    ;REGISTER TABLE POINTER
(3)  003510     000                .BYTE   0           ;(10) SECTOR ADDRESS OR
(3)                                                    ;FIRST REG. INDEX
(3)  003511     000                .BYTE   0           ;(11) TRACK ADDRESS OR
(3)                                                    ;LAST REG. INDEX
(3)  003512  000000                .WORD   0           ;(12) CYLINDER ADDRESS
(3)  003514  003540                .WORD   RP.REG      ;(14) ERROR TABLE POINTER
(3)                                                    ;POINTS TO THE FIRST OF TWENTY
(3)                                                    ;LOCATIONS OF WHERE THE DRIVER
(3)                                                    ;IS TO STORE THE RH11/RP04
(3)                                                    ;REGISTERS ON AN ERROR. IF LEFT
(3)                                                    ;ZERO REGISTERS ARE NOT SAVED.
(3)  003516  000000                        .WORD   0           ;(16) STATUS/ERROR INDICATOR
(3)                                                            ;BIT15=1=>ERROR OCCURRED
(3)                                                            ;BIT07=1=>DONE
(3)                                                            ;BIT14-BIT09 AND BIT06-BIT03
(3)                                                            ;INDICATE TYPE OF ERROR
(3)
(3)  003520     000        DTADPB: .BYTE   0           ;(0) DRIVE NUMBER
(3)  003521     000                .BYTE   0           ;(1) OFFSET VALUE OR FMT22, ECT, AND HCI
(3)  003522     000                .BYTE   0           ;(2) COMMAND
(3)  003523     000                .BYTE   0           ;(3) PSEL AND A17 AND A16
(3)  003524  000000                .WORD   0           ;(4) WORD COUNT (MUST BE NEG.)
(3)  003526  040524                .WORD   BUFFER      ;(6) BUFFER ADDRESS OR
(3)                                                    ;REGISTER TABLE POINTER
(3)  003530     000                .BYTE   0           ;(10) SECTOR ADDRESS OR
(3)                                                    ;FIRST REG. INDEX
(3)  003531     000                .BYTE   0           ;(11) TRACK ADDRESS OR
(3)                                                    ;LAST REG. INDEX
(3)  003532  000000                .WORD   0           ;(12) CYLINDER ADDRESS
(3)  003534  003540                .WORD   RP.REG      ;(14) ERROR TABLE POINTER
(3)                                                    ;POINTS TO THE FIRST OF TWENTY
(3)                                                    ;LOCATIONS OF WHERE THE DRIVER
(3)                                                    ;IS TO STORE THE RH11/RP04
```

```
(3)                                                     ;REGISTERS ON AN ERROR. IF LEFT
(3)                                                     ;ZERO REGISTERS ARE NOT SAVED.
(3)  003536  000000                     .WORD   0       ;(16) STATUS/ERROR INDICATOR
(3)                                                     ;BIT15=1=>ERROR OCCURRED
(3)                                                     ;BIT07=1=>DONE
(3)                                                     ;BIT14-BIT09 AND BIT06-BIT03
(3)                                                     ;INDICATE TYPE OF ERROR
(3)
(3)
(3)
(3)
(3)                             ;SAVE RH11/RP04 REGISTERS HERE ON ERROR
(3)
(3)  003540  000000             RP.REG: .WORD   0       ;RHCS1 (776700) CONTROL & STATUS #1
(3)  003542  000000                     .WORD   0       ;RHWC (776702) WORD COUNT
(3)  003544  000000                     .WORD   0       ;RHBA (776704) BUS ADDRESS
(3)  003546  000000                     .WORD   0       ;RHDA (776706) DESIRED SECTOR/TRACK
(3)  003550  000000                     .WORD   0       ;RHCS2 (776710) CONTROL & STATUS #2
(3)  003552  000000                     .WORD   0       ;RHDS1 (776712) DISK STATUS
(3)  003554  000000                     .WORD   0       ;RHER1 (776714) ERROR REG. #1
(3)  003556  000000                     .WORD   0       ;RHAS (776716) ATTENTION SUMMARY
(3)  003560  000000                     .WORD   0       ;RHLA (776720) LOOK AHEAD
(3)  003562  000000                     .WORD   0       ;RHDB (776722) DATA BUFFER
(3)  003564  000000                     .WORD   0       ;RHMR (776724) MAINTAINABILITY
(3)  003566  000000                     .WORD   0       ;RHDT (776726) DRIVE TYPE
(3)  003570  000000                     .WORD   0       ;RHSN (776730) SERIAL NUMBER
(3)  003572  000000                     .WORD   0       ;RHOF (776732) OFFSET
(3)  003574  000000                     .WORD   0       ;RHCA (776734) DESIRED CYLINDER
(3)  003576  000000                     .WORD   0       ;RHCC (776736) CURRENT CYLINDER
(3)  003600  000000                     .WORD   0       ;RHER2 (776740) ERROR REG #2
(3)  003602  000000                     .WORD   0       ;RHER3 (776742) ERROR REG #3
(3)  003604  000000                     .WORD   0       ;RHEC1 (776744) ECC POSITION
(3)  003606  000000                     .WORD   0       ;RHEC2 (776746) ECC PATTERN
```

```
(2)                             ;*****************************************************************
(1)
(1)                             .SBTTL  ERROR POINTER TABLE
(1)
(1)                             ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
(1)                             ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
(1)                             ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
(1)                             ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
(1)                             ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:
(1)
(1)                             ;*      EM              ;;POINTS TO THE ERROR MESSAGE
(1)                             ;*      DH              ;;POINTS TO THE DATA HEADER
(1)                             ;*      DT              ;;POINTS TO THE DATA
(1)                             ;*      DF              ;;POINTS TO THE DATA FORMAT
(1)
(1)
(1)  003610                     $ERRTB:
5716                            ;*EM AND DH ARE ASCIZ MESSAGES, DT IS A STRING OF WORDS THAT POINT TO THE
5717                            ;*DATA TO BE TYPED AND DF IS A STRING OF DATA THAT TELL HOW THE DT WORDS
5718                            ;*ARE TO BE TYPED.  IF ANY OF THE POINTERS ARE NOT NEEDED FOR A PARTICULAR
5719                            ;*ERROR IT IS REPLACED WITH A ZERO.
5720                            ;*EACH OF THE ITEMS BELOW REFER TO THE ERROR NUMBER AND INDICATE
5721                            ;*THE INFORMATION THAT WILL BE TYPED WHEN THE ERROR OCCURS.
5722                            ;*UNLESS STATED OTHER ALL NUMBERS ARE OCTAL
5723
5724                            ;*  ERROR ITEM 1
5725                            ;*    ILLEGAL RH11(SC=0)INTERRUPT OCCURRED
5726                            ;*    TEST    ERR PC
5727                            ;*    $TMP0   $ERRPC
5728
5729  003610  035205                    EM1
5730  003612  036514                    DH1
5731  003614  037754                    DT1
5732  003616  040360                    DF1
5733
5734                            ;*  ERROR ITEM 2
5735                            ;*    ILLEGAL RP04 INTERRUPT OCCURRED
5736                            ;*    TEST    ERR PC  DRIVE   RHAS    RHCS    RHER1     RHER2     RHER3
5737                            ;*    $TMP0   $ERRPC  $REG1   $REG5   RPERR$  RPERR$+2  RPERR$+4  RPERR$+6
5738
5739  003620  035253                    EM2
5740  003622  036533                    DH2
5741  003624  037760                    DT2
5742  003626  040364                    DF2
5743
5744                            ;*  ERROR ITEM 3
5745                            ;*    MASSBUS PARITY ERROR (MCPE=1)
5746                            ;*    TEST    ERR PC  ADDRESS DATA
5747                            ;*    $TMP0   $ERRPC  RD.ADR  RD.WRD
5748
5749  003630  035313                    EM3
5750  003632  036631                    DH3
5751  003634  040000                    DT3
5752  003636  040370                    DF3
```

```
                                        ;*  ERROR ITEM 4
                                        ;*    MASSBUS PARITY ERROR (PAR=1)
                                        ;*    TEST    ERR PC  ADDRESS GODATA  BDDATA
                                        ;*    $TMPO   $ERRPC  WRT.ADR WRT.WD  RD.WRD

003640  035350                              EM4
003642  036666                              DH4
003644  040010                              DT4
003646  040374                              DF4

                                        ;*  ERROR ITEM 5
                                        ;*    OFFLINE, NONEXISTENT, OR UNSAFE DRIVE INTERRUPTED
                                        ;*    TEST    ERR PC  DRIVE   RHAS    RHDS    RHER1     RHER2     RHER3
                                        ;*    $TMPO   $ERRPC  $REG1   $REG5   RPERRS  RPERRS+2  RPERRS+4 RPERRS+6

003650  035404                              EM5
003652  036533                              DH2
003654  037760                              DT2
003656  040364                              DF2

                                        ;*  ERROR ITEM 6 -- NOT USED

003660  000000                              0
003662  000000                              0
003664  000000                              0
003666  000000                              0

                                        ;*  ERROR ITEM 7 -- NOT USED

003670  000000                              0
003672  000000                              0
003674  000000                              0
003676  000000                              0

                                        ;*  ERROR ITEM 10
                                        ;*    RH11/RP04 FAILED TO RESPOND TO ADDRESSING
                                        ;*    RHCS1   ERR PC
                                        ;*    RH.ADR  $ERRPC

003700  035467                              EM10
003702  036735                              DH10
003704  040042                              DT10
003706  040400                              DF10

                                        ;*  ERROR ITEM 11
                                        ;*    DRIVE SELECTED IS NOT ONLINE
                                        ;*    DRIVE   ERR PC
                                        ;*    $REG2   $ERRPC

003710  035541                              EM11
003712  036754                              DH11
003714  040046                              DT11
003716  040404                              DF11
```

```
5807
5808                                    ;*  ERROR ITEM 12
5809                                    ;*    IMPROPER HEADER DATA
5810                                    ;*    TEST    ERR PC  TST PC  DRIVE   CYLNDR  TRACK   SECTOR
5811                                    ;*    $TMPO   $ERRPC  $REGD   CHKDRV  CYL.DS  TRK.DS  SEC.DS
5812                                    ;*        GDCYL   GDTRK   GDSCTR  BDCYL   BDTRK   BDSCTR
5813                                    ;*        CYL.DS  TRK.DS  SEC.DS  CYL.RD  TRK.RD  SEC.RD
5814                                    ;*      CYLNDR, TRACK, AND SECTOR ARE DECIMAL
5815
5816  003720  035576                        EM12
5817  003722  036773                        DH12
5818  003724  040052                        DT12
5819  003726  040410                        DF12
5820
5821                                    ;*  ERROR ITEM 13
5822                                    ;*    DATA COMPARE FAILURE
5823                                    ;*    TEST    ERR PC  TST PC  DRIVE   CYLNDR  TRACK   SECTOR
5824                                    ;*    $TMPO   $ERRPC  $REGD   CHKDRV  CYL.DS  TRK.DS  SEC.DS
5825                                    ;*        GDDAT   BDDAT   WRDCNT  GDADR   BDADR
5826                                    ;*        $GDDAT  $BDDAT  $REG4   $GDADR  $BDADR
5827                                    ;*      CYLNDR, TRACK, SECTOR, AND WRDCNT  ARE DECIMAL
5828
5829  003730  035623                        EM13
5830  003732  036773                        DH12
5831  003734  040104                        DT13
5832  003736  040420                        DF13
5833
5834                                    ;*  ERROR ITEM 14 -- FOLLOWS #13
5835                                    ;*        $GDDAT  $BDDAT  $REG4   $GDADR  $BDADR
5836
5837  003740  000000                        0
5838  003742  000000                        0
5839  003744  040122                        DT13A
5840  003746  040430                        DF14
5841
5842                                    ;*  ERROR ITEM 15
5843                                    ;*    DATA COMPARE FAILURE
5844                                    ;*    TEST    ERR PC  TST PC  DRIVE   CYLNDR  TRACK   SECTOR
5845                                    ;*    $TMPO   $ERRPC  $REGD   CHKDRV  CYL.DS  TRK.DS  SEC.DS
5846                                    ;*        GDDAT   BDDAT   WRDCNT  GDADR   BDADR
5847                                    ;*        $GDDAT  $BDDAT  $REG4   $GDADR  $BDADR
5848                                    ;*        CYLNDR, TRACK, SECTOR, AND WRDCNT ARE DECIMAL
5849
5850  003750  035623                        EM13
5851  003752  036773                        DH12
5852  003754  040104                        DT13
5853  003756  040420                        DF13
5854
5855                                    ;*  ERROR ITEM 16 -- FOLLOWS #15
5856                                    ;*        $GDDAT  $BDDAT  $REG4   $GDADR  $BDADR
5857
5858  003760  000000                        0
5859  003762  000000                        0
5860  003764  040122                        DT13A
```

```
 5861  003766  040430                        DF14
 5862
 5863                                ;*  ERROR ITEM 17
 5864                                ;*    DISK ERROR IN TIMING TEST
 5865                                ;*    TEST    ERR PC  DRIVE   RHCS1   RHDS      RHER1     RHER2     RHER3
 5866                                ;*    $TMPO   $ERRPC  CHKDRV  RP.REG  RP.REG+12 RP.REG+14 RP.REG+40 RP.REG+42
 5867
 5868  003770  035650                        EM17
 5869  003772  037207                        DH17
 5870  003774  040134                        DT17
 5871  003776  040434                        DF17
 5872
 5873                                ;*  ERROR ITEM 20
 5874                                ;*    CLOCK (KW11-P) OVERFLOW IN TIMING TEST
 5875                                ;*    TEST    ERR PC  DRIVE   RHCS1   RHDS      RHER1     RHER2     RHER3
 5876                                ;*    $TMPO   $ERRPC  CHKDRV  RP.REG  RP.REG+12 RP.REG+14 RP.REG+40 RP.REG+42
 5877
 5878  004000  035702                        EM20
 5879  004002  037207                        DH17
 5880  004004  040134                        DT17
 5881  004006  040434                        DF17
 5882
 5883                                ;*  ERROR ITEM 21
 5884                                ;*    DATA COMPARE FAILURE
 5885                                ;*    TEST      ERR PC  TST PC  DRIVE   CYLNDR  TRACK
 5886                                ;*    $TMPO     $ERRPC  $REG0   CHKDRV  CYL.DS  TRK.DS
 5887                                ;*        GDDAT   BDDAT   WRDCNT  SECTOR
 5888                                ;*        $REG1   $BDDAT  $REG4   $REG1
 5889                                ;*        CYLINDR, TRACK, WRDCNT, AND SECTOR ARE DECIMAL
 5890
 5891  004010  035623                        EM13
 5892  004012  037305                        DH21
 5893  004014  040154                        DT21
 5894  004016  040440                        DF21
 5895
 5896                                ;*  ERROR ITEM 22--FOLLOWS #21
 5897                                ;*        $REG1   $BDDAT   $REG4   $REG1
 5898
 5899  004020  000000                        0
 5900  004022  000000                        0
 5901  004024  040170                        DT21A
 5902  004026  040450                        DF22
 5903
 5904                                ;*****************************************************************
  (2)                                ;*****************************************************************
 5905                                ;*  ERROR ITEMS 23-40 NOT USED
 5906                                ;*  ERROR ITEMS 41-46 WILL HAVE AN EM THAT
 5907                                ;*  VARIES DEPENDING ON THE ERROR, IT WILL BE IN THE FORM:
 5908                                ;*  RH11/RP04 ERROR (MESSAGE)
 5909                                ;*  WHERE MESSAGE WILL BE ONE OR MORE OF THE FOLLOWING:
 5910                                ;*  1)  OFFLINE OR UNSAFE DRIVE REQUESTED
 5911                                ;*  2)  UNLOADED DRIVE REQUESTED
 5912                                ;*  3)  PERSISTENT UNSAFE
 5913                                ;*  4)  PARITY ERROR OCCURRED
```

```
                                        ;*  5)  FATAL PARITY ERROR
                                        ;*  6)  SOFTWARE TIMEOUT ON THIS DRIVE
                                        ;*  7)  SOFTWARE TIMEOUT ON ANOTHER DRIVE
                                        ;*  8)  ERROR OCCURRED DURING I/O OPERATION
                                        ;*  9)  ERROR OCCURRED DURING NON-I/O OPERATION
                                        ;*  10) UNSAFE OCCURRED
                                        ;*  11) AUTOMATIC RECALIBRATE SEQUENCE OCCURRED

004030                                  ITEM41:

                                        ;*  ERROR ITEM 41
                                        ;*    RH11/RP04 ERROR (MESSAGE)
                                        ;*    TEST    ERR PC  TST PC  DRIVE
                                        ;*    $TMPO   $ERRPC  $REGO   CHKDRV

004030  035751                                  EM41
004032  037422                                  DH41
004034  040200                                  DT41
004036  040454                                  DF41

                                        ;*  ERROR ITEM 42
                                        ;*    RH11/RP04 ERROR (MESSAGE)
                                        ;*    TEST    ERR PC  TST PC  DRIVE   RHCS1   RHCS2     RHDS
                                        ;*    $TMPO   $ERRPC  $REGO   CHKDRV  RP.REG  RP.REG+10 RP.REG+12

004040  035751                                  EM41
004042  037460                                  DH42
004044  040210                                  DT42
004046  040460                                  DF42

                                        ;*  ERROR ITEM 43
                                        ;*    RH11/RP04 ERROR (MESSAGE)
                                        ;*    TEST    ERR PC  TST PC  DRIVE   RHCS1   RHCS2     RHDS
                                        ;*    $TMPO   $ERRPC  $REGO   CHKDRV  RP.REG  RP.REG+10 RP.REG+12
                                        ;*       RHER1     RHER2     RHER3
                                        ;*       RP.REG+14 RP.REG+40 RP.REG+42

004050  035751                                  EM41
004052  037460                                  DH42
004054  040226                                  DT43
004056  040464                                  DF43

                                        ;*  ERROR ITEM 44
                                        ;*    RH11/RP04 ERROR (MESSAGE)
                                        ;*    TEST    ERR PC  TST PC  DRIVE   CYLNDR  TRACK     SECTOR
                                        ;*    $TMPO   $ERRPC  $REGO   CHKDRV  CYL.DS  TRK.DS    SEC.DS
                                        ;*       RHCS1     RHCS2     RHDS      RHCC      RHCA      RHDA
                                        ;*       RP.REG    RP.REG+10 RP.REG+12 RP.REG+36 RP.REG+34 RP.REG+06
                                        ;*     RHER1       RHER2       RHER3
                                        ;*     RF.REG+14   RP.REG+40   RP.REG+42
                                        ;*     CYLNDR,TRACK, AND SECTOR ARE DECIMAL

004060  035751                                  EM41
004062  036773                                  DH12
```

```
5968  004064  040252                        DT44
5969  004066  040474                        DF44
5970
5971                                ;*  ERROR ITEM 45
5972                                ;*    RH11/RP04 ERROR (MESSAGE)
5973                                ;*    TEST    ERR PC  TST PC  DRIVE   CYLNDR  TRACK     SECTOR
5974                                ;*    $TMPO   $ERRPC  $REGO   CHKDRV  CYL.DS  TRK.DS    SEC.DS
5975                                ;*       RHCS1      RHCS2      RHDS       RHCC       RHCA       RHDA
5976                                ;*       RP.REG     RP.REG+10  RP.REG+12  RP.REG+36  RP.REG+34  RP.REG+06
5977                                ;*     RHER1      RHER2      RHER3      RHWC       RHBA       RHDB
5978                                ;*     RP.REG+14  RP.REG+40  RP.REG+42  RP.REG+2   RP.REG+4   RP.REG+22
5979                                ;*       CYLNDR,TRACK, AND SECTOR ARE DECIMAL
5980
5981  004070  035751                        EM41
5982  004072  036773                        DH12
5983  004074  040312                        DT45
5984  004076  040510                        DF45
5985
5986                                ;*  ERROR ITEM 46
5987                                ;*    FATAL WRITE CHECK ERROR (MESSAGE)
5988                                ;*    TEST    ERR PC  TST PC  DRIVE   CYLNDR  TRACK     SECTOR
5989                                ;*    $TMPO   $ERRPC  $REGO   CHKDRV  CYL.DS  TRK.DS    SEC.DS
5990                                ;*       RHCS1      RHCS2      RHDS       RHCC       RHCA       RHDA
5991                                ;*       RP.REG     RP.REG+10  RP.REG+12  RP.REG+36  RP.REG+34  RP.REG+06
5992                                ;*     RHER1      RHER2      RHER3      RHWC       RHBA       RHDB
5993                                ;*     RP.REG+14  RP.REG+40  RP.REG+42  RP.REG+2   RP.REG+4   RP.REG+22
5994                                ;*       CYLNDR,TRACK, AND SECTOR ARE DECIMAL
5995
5996  004100  035771                        EM46
5997  004102  036773                        DH12
5998  004104  040312                        DT45
5999  004106  040510                        DF45
6000
```

```
                                        .SBTTL  START OF PROGRAM

004110  012737  177777  001216  START3: MOV     #-1,@#BUSADR    ;GET BUSADR FLAG
004116  000402                  START1A BR      START1A
004120  005037  001216          START1: CLR     @#BUSADR        ;CLR BUSADR FLAG
004124  012737  037777  001224  START1A: MOV     #37777,TSTNMS   ;SELECT TESTS 0-15
004132  005037  001214                  CLR     @#CNTRLC        ;NO CONTROL "C"
004136  012700  001614                  MOV     #DFLT,R0        ;DEFAULT PARAMETERS POINTER
004142  012701  002062                  MOV     #PRM0,R1        ;TABLE POINTER
004146  010102                          MOV     R1,R2           ;STOP ADDRESS
004150  012021                  1$:     MOV     (R0)+,(R1)+     ;MOVE DEFAULT PARAMETERS INTO
004152  020002                          CMP     R0,R2           ;RUN TIME TABLES ** DONE?
004154  103775                          BLO     1$              ;NO--BRANCH
004156  012700  003040                  MOV     #PAT8,R0        ;PAT0 DEFAULTS TO PATTERN 8
004162  012701  002440                  MOV     #PAT0,R1
004166  012021                  2$:     MOV     (R0)+,(R1)+
004170  020027  003100                  CMP     R0,#PAT9
004174  103774                          BLO     2$
004176  000411                          BR      START
004200  012737  177777  001216  START4: MOV     #-1,@#BUSADR    ;SET BUSADR FLAG
004206  000402                  START2A BR      START2A
004210  005037  001216          START2: CLR     @#BUSADR        ;CLR BUSADR FLAG
004214  012737  177777  001214  START2A: MOV     #-1,@#CNTRLC    ;SET CONTROL "C" FLAG
004222  000005          START:  RESET
004224  012737  000340  177776          MOV     #340,@#PS       ;;LOCK OUT ALL INTERRUPTS
(1) 004232  012706  001100              MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
(1) 004236  005026                      CLR     (R6)+           ;;CLEAR MEMORY LOCATION
(1) 004240  022706  001136              CMP     #$TKS,R6        ;;DONE?
(1) 004244  001374                      BNE     .-6             ;;LOOP BACK IF NO
(1) 004246  012706  001100              MOV     #$STACK,SP      ;;SETUP THE STACK POINTER
(1) 004252  012737  016102  000020      MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
(1) 004260  012737  000340  000022      MOV     #340,@#IOTVEC+2 ;;LEVEL 7
(1) 004266  012737  013762  000030      MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
(1) 004274  012737  000340  000032      MOV     #340,@#EMTVEC+2 ;;LEVEL 7
(1) 004302  012737  016416  000034      MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
(1) 004310  012737  000340  000036      MOV     #340,@#TRAPVEC+2;LEVEL 7
(1) 004316  005037  001176              CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
(1) 004322  005037  001200              CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
(1) 004326  112737  000001  001115      MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
(1) 004334  012737  004334  001106      MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
(1) 004342  012737  004342  001110      MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
6178 004350  012700  001152              MOV     #$REGAD,R0      ;FIRST ADDRESS
6179 004354  005020                  1$: CLR     (R0)+           ;CLEAR VARIABLE STORAGE
6180 004356  022700  001202              CMP     #$BELL,R0       ;DONE?
6181 004362  001374                      BNE     1$              ;NO--BRANCH
6182 004364  013737  001354  001144      MOV     @#TPS,@#$TPB    ;SETUP THE STATUS AND BUFFER REG'S
6183 004372  013737  001356  001144      MOV     @#TPB,@#$TPB    ;FOR THE TYPE ROUTINE
6184 004400  063727  001214  000000      ADD     @#CNTRLC,#0     ;TYPE "ID" IF START AT 200 OR
6185 004406  103433                      BCS     SRTINT          ;THE 1ST START AT 204
6186 004410  104400  004416              TYPE    .+4             ;;TYPE ASCIZ STRING
(1) 004414  000430                      BR      64$             ;;GET OVER THE ASCIZ
(1)                                     ;;.ASCIZ        <15><12>/MD-11-DZRPK-B "MECHANICAL & READ-WRITE" TEST/
```

```
  (1)  004476                                    64$:
6187   004476  004737  017316                   SRTINT: JSR     PC,@#LP.AVL     ;CHECK FOR A LINE PRINTER
6188   004502  004737  015346                           JSR     PC,@#$TKINT     ;TURN ON THE TTY KEYBOARD INTERRUPT
6189   004506  005037  177776                           CLR     @#PS            ;INSURE THE PRIORITY = 0
6190   004512  004737  024200                           JSR     PC,@#GETADR     ;CHECK RH11/RP04 (RHCS1) ADDRESS
6191   004516  004737  024112                           JSR     PC,@#GETSWR     ;CO CHECK FOR CONTROL SWITCHES
6192   004522  004737  027204                           JSR     PC,@#RPINIT     ;SETUP DRIVE STATUS
6193   004526  004737  017360                           JSR     PC,@#ST.CLK     ;INITIALIZE THE CLOCK
6194   004532  005737  001214                           TST     @#CNTRLC        ;CONTROL "C" START/RESTART?
6195   004536  001403                                   BEQ     1$              ;NO--BRANCH
6196   004540  004737  024550                           JSR     PC,@#GT.PRM     ;YES--GET PARAMETERS
6197   004544  000416                                   BR      4$
6198   004546  005037  001222                   1$:     CLR     DRVSEL          ;NO DRIVES SELECTED
6199   004552  005000                                   CLR     R0              ;DETERMINE THE DRIVES THAT
6200   004554  012701  000001                           MOV     #1,R1           ;ARE AVAILABLE FOR TESTING
6201   004560  105760  027050                   2$:     TSTB    DRVSTA(R0)
6202   004564  003403                                   BLE     3$
6203   004566  156037  027154  001222                   BISB    ATABIT(R0),@#DRVSEL
6204   004574  005200                           3$:     INC     R0
6205   004576  106301                                   ASLB    R1
6206   004600  001367                                   BNE     2$
6207   004602  005037  027130                   4$:     CLR     @#SEEKFG        ;CLEAR SEEK FLAG
6208   004606  032737  000400  001212                   BIT     #SW08,@#C.SWR   ;DO SEEK BEFORE DATA TRANSFER?
6209   004614  001002                                   BNE     5$              ;YES--BRANCH
6210   004616  005137  027130                           COM     @#SEEKFG        ;NO
6211   004622                                   5$:
  (1)  004622  104400  004630                           TYPE    ,.+4            ;;TYPE ASCIZ STRING
  (1)  004626  000415                                   BR      64$             ;;GET OVER THE ASCIZ
  (1)                                                   ;;.ASCIZ        <15><12>/DRIVE(S) TO BE TESTED /
  (1)  004662                                    64$:
6212   004662  005037  013660                           CLR     @#$ENDCT        ;DETERMINE PASSES TO MAKE AND
6213   004666  005000                                   CLR     R0              ;THE DRIVES TO BE TESTED
6214   004670  013701  001222                           MOV     @#DRVSEL,R1     ;ANY DRIVES SELECTED?
6215   004674  001010                                   BNE     6$              ;YES--BRANCH
6216   004676  104400  004704                           TYPE    ,.+4            ;;TYPE ASCIZ STRING
  (1)  004702  000403                                   BR      65$             ;;GET OVER THE ASCIZ
  (1)                                                   ;;.ASCIZ        /NONE/
  (1)  004712                                    65$:
6217   004712  000137  013476                           JMP     @#$EOP          ;GO TO END OF PROGRAM
6218   004716  006201                           6$:     ASR     R1              ;REPORT THE DRIVES TO BE TESTED
6219   004720  103013                                   BCC     7$
6220   004722  005237  013660                           INC     @#$ENDCT        ;GIVE THIS DRIVE A PASS
6221   004726  010046                                   MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
  (1)  004730  104404                                   TYPOS                   ;;GO TYPE--OCTAL ASCII
  (1)  004732     001                                   .BYTE   1               ;;TYPE 1 DIGIT(S)
  (1)  004733     000                                   .BYTE   0               ;;SUPPRESS LEADING ZEROS
6222   004734  005701                                   TST     R1              ;MORE DRIVES?
6223   004736  001406                                   BEQ     8$              ;NO--BRANCH
6224   004740  104400  004746                           TYPE    ,.+4            ;;TYPE ASCIZ STRING
  (1)  004744  000401                                   BR      66$             ;;GET OVER THE ASCIZ
  (1)                                                   ;;.ASCIZ        "."
  (1)  004750                                    66$:
6225   004750  005200                           7$:     INC     R0              ;FORM DRIVE NUMBER
6226   004752  000761                                   BR      6$
```

```
 6227  004754  013737  013660  013652  8$:     MOV     @#$ENDCT,@#$EOPCT
 6228  004762  005737  001226                  TST     @#CLKSTA        ;KW11-P AVAILABLE
 6229  004766  003041                          BGT     RSTRT1          ;YES--BRANCH
 6230  004770  032737  003600  001224          BIT     #3600,@#TSTNMS  ;NO--ANY TIMING TESTS TO BE PERFORMED?
 6231  004776  001435                          BEQ     RSTRT1          ;NO--BRANCH
 6232  005000  104400  005006                  TYPE    .+4             ;;TYPE ASCIZ STRING
  (1)  005004  000432                          BR      67$             ;;GET OVER THE ASCIZ
  (1)                                          ;;.ASCIZ      <15><12>/NO KW11-P CLOCK--TEST 7-12 WILL NOT BE PERFORMED/
  (1)  005072                          67$:
 6233  005072  005037  001236          RSTRT1: CLR     CHKDRV          ;INIT. THE CHECK DRIVE KEY
 6234  005076  012737  000001  001240          MOV     #1,DRVMSK       ;START TO CHECK DESIRED DRIVES
 6235  005104  033737  001240  001222  RSTRT2: BIT     DRVMSK,DRVSEL   ;IS THIS DRIVE SELECTED?
 6236  005112  001006                  RSTRT3: BNE     DRVOK           ;YES--GO CHECK IF DRIVE IS READY FOR TESTING
 6237  005114  005237  001236          RESTART:INC     CHKDRV          ;MOVE  TO NEXT DRIVE NUMBER
 6238  005120  106337  001240                  ASLB    DRVMSK          ;POSITION THE MASK
 6239  005124  103762                          BCS     RSTRT1          ;BRANCH IF THE DRIVE NUMBER NEEDS INITIALIZED
 6240  005126  000766                          BR      RSTRT2
 6241
 6242  005130  013702  001236          DRVOK:  MOV     @#CHKDRV,R2     ;PICKUP THE DRIVE NUMBER
 6243  005134  105762  027050                  TSTB    DRVSTA(R2)      ;IS DESIRED DRIVE ON-LINE?
 6244  005140  003002                          BGT     1$              ;YES, BRANCH
 6245  005142  104011                          ERROR   11              ;DRIVE SELECTED IS NOT ONLINE
 6246  005144  000763                          BR      RESTART         ;NO-- RETURN
 6247  005146  010237  003440          1$:     MOV     R2,@#DPB.A      ;SET THE DRIVE NUMBER INTO THE DPB'S
 6248  005152  010237  003460                  MOV     R2,@#DPB.B
 6249  005156  010237  003500                  MOV     R2,@#DPB.C
 6250  005162  010237  003520                  MOV     R2,@#DTADPB
 6251  005166  004737  017636                  JSR     PC,@#LDCMD      ;LOAD COMMAND INTO DPB.B AND DPB.C
 6252  005172  012737  013476  001234          MOV     #$EOP,@#BYPASS  ;IF ERROR GO TO END OF PROGRAM
 6253  005200  112737  000020  003441          MOVB    #FMT22/^D256,@#DPB.A+1
  (1)  005206  012737  000143  003442          MOV     #SETFORM,@#DPB.A+2      ;SETFORM=COMMAND
 6254  005214  004037  017702                  JSR     R0,@#CALL.A             ;GO EXECUTE THE COMMAND
 6255  005220  012737  000107  003442          MOV     #RECAL,@#DPB.A+2        ;RECAL=COMMAND
 6256  005226  004037  017702                  JSR     R0,@#CALL.A             ;GO EXECUTE THE COMMAND
 6257  005232  012737  000030  003450          MOV     #RHSN,@#DPB.A+10 ;FIRST REG. INDEX
 6258  005240  112737  000030  003451          MOVB    #RHSN,@#DPB.A+11 ;LAST REG. INDEX
 6259  005246  012737  000141  003442          MOV     #GETREG,@#DPB.A+2       ;GETREG=COMMAND
 6260  005254  004037  017702                  JSR     R0,@#CALL.A             ;GO EXECUTE THE COMMAND
 6261  005260  104400  005266                  TYPE    .+4             ;;TYPE ASCIZ STRING
  (1)  005264  000411                          BR      64$             ;;GET OVER THE ASCIZ
  (1)                                          ;;.ASCIZ      <15><12><12>/TESTING DRIVE /
  (1)  005310                          64$:
 6262  005310  010246                          MOV     R2,-(SP)        ;;SAVE R2 FOR TYPEOUT
  (1)  005312  104404                          TYPOS                   ;;GO TYPE--OCTAL ASCII
  (1)  005314     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
  (1)  005315     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
 6263  005316  104400  035202                  TYPE    ,MSG.SP         ;TYPE SPACES
 6264  005322  104400  005330                  TYPE    .+4             ;;TYPE ASCIZ STRING
  (1)  005326  000410                          BR      65$             ;;GET OVER THE ASCIZ
  (1)                                          ;;.ASCIZ      /SERIAL NUMBER /
  (1)  005350                          65$:
 6265  005350  012700  000004                  MOV     #4,R0           ;FOUR DIGITS TO TYPE
 6266  005354  013701  040524                  MOV     @#BUFFER,R1     ;SERIAL NUMBER
 6267  005360  005002                  2$:     CLR     R2              ;ZERO
```

```
 6268  005362  006101                        ROL     R1              ;PUT THE NEXT DIGIT
 6269  005364  C06102                        ROL     R2              ;INTO R2
 6270  005366  006101                        ROL     R1
 6271  005370  006102                        ROL     R2
 6272  005372  006101                        ROL     R1
 6273  005374  006102                        ROL     R2
 6274  005376  006101                        ROL     R1
 6275  005400  006102                        ROL     R2
 6276  005402  062702  000060                ADD     #'0,R2          ;MAKE IT ASCII
 6277  005406  010227                        MOV     R2,(PC)+        ;SAVE IT
 6278  005410  000000              3$:       .WORD   0
 6279  005412  104400  005410                TYPE    ,3$             ;TYPE
 6280  005416  005300                        DEC     R0              ;ALL DIGITS TYPED?
 6281  005420  003357                        BGT     2$              ;NO -- BRANCH
 6282  005422  104400  001207                TYPE    ,$CRLF
```

```
6284
6285                                    .SBTTL  #### TESTS ####
6286

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

        ;*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             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

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

6313
6314                                    .SBTTL  *** SEEK TESTS (0-6) ***
6315

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

        ;*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.

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

6325
```

```
6335                                    ;*********************************************************************
 (3)                                    ;*TEST 0        RECAL/SEEK TEST
 (4)
 (4)                                    ;*      THIS TEST WILL CAUSE THE DRIVE TO EXECUTE A RECALIBRATE
 (4)                                    ;*      COMMAND CYCLE AND THEN SEEK FORWARD TO CYLINDER "LC" AT
 (4)                                    ;*      THE COMPLETION OF BOTH COMMANDS STATUS INDICATIONS ARE
 (4)                                    ;*      CHECKED TO INSURE NO ERRORS OCCURRED.
 (4)                                    ;*      THIS TEST WILL BE REPEATED 10 TIMES.
 (4)
 (3)                                    ;*********************************************************************
 (3)  005426                            TSTO:
 (4)  005426  033737  001374  001224            BIT     @#BITS+<0*2>,TSTNMS ;DO THIS TEST?
 (4)  005434  001002                            BNE     64$             ;YES--BRANCH
 (4)  005436  000137  005554                    JMP     TST1            ;NO--GO TO THE NEXT TEST
 (4)  005442  013737  002064  001176  64$:     MOV     @#RPTO,$TIMES   ;GET THE ITERATION COUNT
 (3)  005450  012737  005536  001106            MOV     #TESTO,@#$LPADR
 (3)  005456  012737  005426  001110            MOV     #TSTO,@#$LPERR  ;SETUP THE ERROR LOOP ADDRESS
 (3)  005464  012737  000000  001102            MOV     #0,@#$TSTNM             ;SET UP TEST NUMBER AND
 (3)                                                                    ;CLEAR THE ERROR FLAG ($ERFLG)
 (3)  005472  013737  001102  177570            MOV     $TSTNM,@#DISPLAY        ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
6336  005500  012737  000107  003442            MOV     #RECAL,@#DPB.A+2        ;RECAL=COMMAND
6337  005506  113737  002072  003470            MOVB    @#FSO,@#DPB.B+10 ;FS
6338  005514  113737  002070  003471            MOVB    @#FTO,@#DPB.B+11 ;FT
6339  005522  013737  002066  003472            MOV     @#LCO,@#DPB.B+12 ;LC
6340  005530  012737  005552  001234            MOV     #EXITO,@#BYPASS ;GO TO EXITO ON ERROR
6341  005536  012706  001100            TESTO:  MOV     #STACK,SP       ;SET UP STACK POINTER
6342  005542  004037  017702                    JSR     R0,@#CALL.A             ;GO EXECUTE THE COMMAND
6343  005546  004037  020010                    JSR     R0,@#CALL.B             ;GO EXECUTE THE COMMAND
6344  005552  000004                    EXITO:  SCOPE                   ;LOOP
6345
6357                                    ;*********************************************************************
 (3)                                    ;*TEST 1        SEEK/SEEK TEST
 (4)
 (4)                                    ;*      THIS TEST WILL CAUSE THE DRIVE TO EXECUTE A FORWARD SEEK
 (4)                                    ;*      CYCLE TO "LC", "LT", "LS" FOLLOWED BY A REVERSE SEEK CYCLE TO
 (4)                                    ;*      "FC", "FT", "FS". AT THE COMPLETION OF EACH SEEK, THE PROPER
 (4)                                    ;*      INDICATORS ARE EXAMINED TO INSURE PROPER OPERATION.
 (4)                                    ;*      "LC" WILL DEFAULT TO 128 AND "FC", "FT", "LT", "FS", AND "LS"
 (4)                                    ;*      WILL DEFAULT TO 0
 (4)                                    ;*      THIS TEST WILL BE REPEATED 100 TIMES
 (4)
 (3)                                    ;*********************************************************************
 (3)  005554                            TST1:
 (4)  005554  033737  001376  001224            BIT     @#BITS+<1*2>,TSTNMS ;DO THIS TEST?
 (4)  005562  001002                            BNE     64$             ;YES--BRANCH
 (4)  005564  000137  005716                    JMP     TST2            ;NO--GO TO THE NEXT TEST
 (4)  005570  013737  002076  001176  64$:     MOV     @#RPT1,$TIMES   ;GET THE ITERATION COUNT
 (3)  005576  012737  005700  001106            MOV     #TEST1,@#$LPADR
 (3)  005604  012737  005554  001110            MOV     #TST1,@#$LPERR  ;SETUP THE ERROR LOOP ADDRESS
 (3)  005612  012737  000001  001102            MOV     #1,@#$TSTNM             ;SET UP TEST NUMBER AND
 (3)                                                                    ;CLEAR THE ERROR FLAG ($ERFLG)
 (3)  005620  013737  001102  177570            MOV     $TSTNM,@#DISPLAY        ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
6358  005626  113737  002112  003470            MOVB    @#FS1,@#DPB.B+10 ;FS
6359  005634  113737  002114  003510            MOVB    @#LS1,@#DPB.C+10 ;LS
```

```
 6360  005642  113737  002106  003471          MOVB    @#FT1,@#DPB.B+11 ;FT
 6361  005650  113737  002110  003511          MOVB    @#LT1,@#DPB.C+11 ;LT
 6362  005656  013737  002100  003472          MOV     @#FC1,@#DPB.B+12 ;FC
 6363  005664  013737  002102  003512          MOV     @#LC1,@#DPB.C+12 ;LC
 6364  005672  012737  005714  001234          MOV     #EXIT1,@#BYPASS ;GO TO EXIT1 ON ERROR
 6365  005700  012706  001100          TEST1:  MOV     #STACK,SP       ;SET THE STACK POINTER
 6366  005704  004037  020154                  JSR     R0,@#CALL.C             ;GO EXECUTE THE COMMAND
 6367  005710  004037  020010                  JSR     R0,@#CALL.B             ;GO EXECUTE THE COMMAND
 6368  005714  000004                  EXIT1:  SCOPE                   ;LOOP
 6369
 6382                                  ;*****************************************************************
  (3)                                  ;*TEST 2        INCREMENT/SEEK TEST
  (4)
  (4)                                  ;*
  (4)                                  ;*      THIS TEST WILL COMMAND FORWARD SEEK CYCLES TO ADVANCE THE
  (4)                                  ;*      CYLINDER ADDRESS FROM "FC" TO "LC" BY THE INCREMENT "IC".
  (4)                                  ;*      WHEN THE RESULTANT CYLINDER ADDRESS (NC) EXCEEDS
  (4)                                  ;*      "LC" REVERSE SEEK CYCLES ARE INITIATED; STARTING
  (4)                                  ;*      AT THE LAST LEGAL "NC" AND DECREMENTING BY "IC"
  (4)                                  ;*      UNTIL "NC" IS LESS THAN "FC". AT THE COMPLETION OF EACH
  (4)                                  ;*      SEEK COMMAND THE PROPER INDICATORS ARE EXAMINED TO
  (4)                                  ;*      INSURE PROPER OPERATION.
  (4)
  (3)                                  ;*****************************************************************
  (3)  005716                          TST2:
  (4)  005716  033737  001400  001224          BIT     @#BITS+<2*2>,TSTNMS ;DO THIS TEST?
  (4)  005724  001002                          BNE     64$             ;YES--BRANCH
  (4)  005726  000137  006100                  JMP     TST3            ;NO--GO TO THE NEXT TEST
  (4)  005732  013737  002120  001176  64$:    MOV     @#RPT2,$TIMES   ;GET THE ITERATION COUNT
  (3)  005740  012737  006012  001106          MOV     #TEST2,@#$LPADR
  (3)  005746  012737  005716  001110          MOV     #TST2,@#$LPERR  ;SETUP THE ERROR LOOP ADDRESS
  (3)  005754  012737  000002  001102          MOV     #2,@#$TSTNM             ;SET UP TEST NUMBER AND
  (3)                                                                  ;CLEAR THE ERROR FLAG ($ERFLG)
  (3)  005762  013737  001102  177570          MOV     $TSTNM,@#DISPLAY        ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
 6383  005770  113737  002132  003470          MOVB    @#FS2,@#DPB.B+10 ;FS
 6384  005776  113737  002130  003471          MOVB    @#FT2,@#DPB.B+11 ;FT
 6385  006004  012737  006076  001234          MOV     #EXIT2,@#BYPASS ;GO TO EXIT2 ON ERROR
 6386  006012  012706  001100          TEST2:  MOV     #STACK,SP       ;SET UP THE STACK POINTER
 6387  006016  013737  002122  003472          MOV     @#FC2,@#DPB.B+12 ;FC
 6388  006024                          INCSK:
 6389  006024  004037  020010                  JSR     R0,@#CALL.B             ;GO EXECUTE THE COMMAND
 6390  006030  063737  002126  003472          ADD     @#IC2,@#DPB.B+12 ;MOVE TO NEXT CYLINDER
 6391  006036  023737  002124  003472          CMP     @#LC2,@#DPB.B+12 ;OUT OF CYLINDERS?
 6392  006044  002367                          BGE     INCSK           ;NO--BRANCH
 6393  006046  013737  002124  003472          MOV     @#LC2,@#DPB.B+12
 6394  006054                          DECSK:
 6395  006054  004037  020010                  JSR     R0,@#CALL.B             ;GO EXECUTE THE COMMAND
 6396  006060  163737  002126  003472          SUB     @#IC2,@#DPB.B+12
 6397  006066  023737  002122  003472          CMP     @#FC2,@#DPB.B+12
 6398  006074  003767                          BLE     DECSK
 6399  006076  000004                  EXIT2:  SCOPE                   ;LOOP
 6400
 6409                                  ;*****************************************************************
  (3)                                  ;*TEST 3        STEPPING SEEK TEST
  (4)
```

```
(4)                                     ;*      THIS TEST WILL COMMAND SEEK CYCLES TO CYLINDER 0,1,2,4,
(4)                                     ;*      8,16,32,64,128, AND 256.  AT THE COMPLETION OF EACH SEEK
(4)                                     ;*      COMMAND THE PROPER INDICATORS ARE EXPLAINED TO INSURE PROPER
(4)                                     ;*      OPERATION.
(4)
(3)                                     ;*****************************************************************
(3)  006100                             TST3:
(4)  006100  033737  001402  001224             BIT     @#BITS+<3*2>,TSTNMS ;DO THIS TEST?
(4)  006106  001002                             BNE     64$             ;YES--BRANCH
(4)  006110  000137  006240                     JMP     TST4            ;NO--GO TO THE NEXT TEST
(4)  006114  013737  002136  001176  64$:      MOV     @#RPT3,$TIMES   ;GET THE ITERATION COUNT
(3)  006122  012737  006174  001106             MOV     #TEST3,@#$LPADR
(3)  006130  012737  006100  001110             MOV     #TST3,@#$LPERR  ;SETUP THE ERROR LOOP ADDRESS
(3)  006136  012737  000003  001102             MOV     #3,@#$TSTNM             ;SET UP TEST NUMBER AND
(3)                                                                     ;CLEAR THE ERROR FLAG ($ERFLG)
(3)  006144  013737  001102  177570             MOV     $TSTNM,@#DISPLAY        ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
6410 006152  113737  002150  003470             MOVB    @#FS3,@#DPB.B+10 ;FS
6411 006160  113737  002146  003471             MOVB    @#FT3,@#DPB.B+11 ;FT
6412 006166  012737  006236  001234             MOV     #EXIT3,@#BYPASS ;GO TO BYPASS ON ERROR
6413 006174  012706  001100             TEST3:  MOV     #STACK,SP       ;SET UP THE STACK
6414 006200  013737  002140  003472             MOV     @#FC3,@#DPB.B+12 ;FC
6415 006206  004037  020010                     JSR     R0,@#CALL.B             ;GO EXECUTE THE COMMAND
6416 006212  013701  002144                     MOV     IC3,R1          ;CYLINDER 1
6417 006216  010137  003472             1$:     MOV     R1,@#DPB.B+12   ;DESIRED CYLINDER
6418 006222  004037  020010                     JSR     R0,@#CALL.B             ;GO EXECUTE THE COMMAND
6419 006226  006301                             ASL     R1              ;MOVE TO NEXT CYLINDER
6420 006230  020137  002142                     CMP     R1,@#LC3        ;DONE?
6421 006234  003770                             BLE     1$              ;NO--LOOP
6422 006236  000004                     EXIT3:  SCOPE
6423
6433                                    ;*****************************************************************
(3)                                     ;*TEST 4        OSCILLATING SEEK TEST
(4)
(4)                                     ;*      THIS TEST WILL COMMAND SEEK CYCLES FROM "FC" TO "NC" AND BACK
(4)                                     ;*      TO "FC".  "NC" STARTS AT "FC" AND INCREMENTS BY "IC" UP TO CYLINDER
(4)                                     ;*      "LC", THEN IS DECREMENTED BY "IC" BACK TO CYLINDER "FC".  AT THE
(4)                                     ;*      COMPLETION OF EVERY SEEK COMMAND THE PROPER INDICATORS ARE
(4)                                     ;*      EXAMINED TO INSURE PROPER OPERATION.
(4)
(3)                                     ;*****************************************************************
(3)  006240                             TST4:
(4)  006240  033737  001404  001224             BIT     @#BITS+<4*2>,TSTNMS ;DO THIS TEST?
(4)  006246  001002                             BNE     64$             ;YES--BRANCH
(4)  006250  000137  006572                     JMP     TST5            ;NO--GO TO THE NEXT TEST
(4)  006254  013737  002154  001176  64$:      MOV     @#RPT4,$TIMES   ;GET THE ITERATION COUNT
(3)  006262  012737  006350  001106             MOV     #TEST4,@#$LPADR
(3)  006270  012737  006240  001110             MOV     #TST4,@#$LPERR  ;SETUP THE ERROR LOOP ADDRESS
(3)  006276  012737  000004  001102             MOV     #4,@#$TSTNM             ;SET UP TEST NUMBER AND
(3)                                                                     ;CLEAR THE ERROR FLAG ($ERFLG)
(3)  006304  013737  001102  177570             MOV     $TSTNM,@#DISPLAY        ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
6434 006312  113737  002166  003470             MOVB    @#FS4,@#DPB.B+10 ;FS
6435 006320  113737  002164  003471             MOVB    @#FT4,@#DPB.B+11 ;FT
6436 006326  012737  006570  001234             MOV     #EXIT4,@#BYPASS ;GO TO EXIT4 ON ERROR
6437 006334  005002                             CLR     R2              ;CLEAR STALL SWITCH (NO STALL)
```

```
 6438  006336  032737  010000  001212          BIT     #SW12,@#C.SWR   ;STALL REQUIRED?
 6439  006344  001401                          BEQ     TEST4           ;NO--BRANCH
 6440  006346  005102                          COM     R2              ;YES--SET SWITCH
 6441  006350  012706  001100          TEST4:  MOV     #STACK,SP       ;SET UP THE STACK POINTER
 6442  006354  013701  002156                  MOV     @#FC4,R1        ;SET NC TO FC
 6443  006360  005037  001320                  CLR     @#STALLO        ;START AT ZERO IF STALLS REQUIRED
 6444  006364  010137  003472          1$:     MOV     R1,@#DPB.B+12   ;NC
 6445  006370  004037  020010                  JSR     R0,@#CALL.B             ;GO EXECUTE THE COMMAND
 6446  006374  005702                          TST     R2              ;STALL?
 6447  006376  001403                          BEQ     2$              ;NO--BRANCH
 6448  006400  004037  021104                  JSR     R0,@#STALL      ;YES-GO TO STALL ROUTINE
 6449  006404  001320                          .WORD   STALLO          ;TIME POINTER
 6450  006406  013737  002156  003472  2$:     MOV     FC4,@#DPB.B+12  ;FC
 6451  006414  004037  020010                  JSR     R0,@#CALL.B             ;GO EXECUTE THE COMMAND
 6452  006420  005702                          TST     R2              ;STALL?
 6453  006422  001413                          BEQ     3$              ;NO--BRANCH
 6454  006424  004037  021104                  JSR     R0,@#STALL      ;YES--GO TO STALL ROUTINE
 6455  006430  001320                          .WORD   STALLO          ;TIME POINTER
 6456  006432  005237  001320                  INC     @#STALLO        ;UPDATE THE TIME
 6457  006436  023737  001370  001320          CMP     @#MXSTAL,@#STALLO ;TIME TO BIG?
 6458  006444  003347                          BGT     1$              ;NO--BRANCH
 6459  006446  005037  001320                  CLR     @#STALLO        ;YES--START OVER AT ZERO
 6460  006452  063701  002162          3$:     ADD     @#IC4,R1        ;MOVE TO NEXT CYLINDER
 6461  006456  020137  002160                  CMP     R1,@#LC4        ;LAST CYLINDER COMPLETED?
 6462  006462  003740                          BLE     1$              ;NO--BRANCH
 6463  006464  013701  002160                  MOV     @#LC4,R1        ;SET NC TO LC
 6464  006470  010137  003472          4$:     MOV     R1,@#DPB.B+12   ;NC
 6465  006474  004037  020010                  JSR     R0,@#CALL.B             ;GO EXECUTE THE COMMAND
 6466  006500  005702                          TST     R2              ;STALL?
 6467  006502  001403                          BEQ     5$              ;NO--BRANCH
 6468  006504  004037  021104                  JSR     R0,@#STALL      ;YES--GO TO STALL ROUTINE
 6469  006510  001320                          .WORD   STALLO          ;TIME POINTER
 6470  006512  013737  002160  003472  5$:     MOV     @#LC4,@#DPB.B+12 ;LC
 6471  006520  004037  020010                  JSR     R0,@#CALL.B             ;GO EXECUTE THE COMMAND
 6472  006524  005702                          TST     R2              ;STALL?
 6473  006526  001413                          BEQ     6$              ;NO--BRANCH
 6474  006530  004037  021104                  JSR     R0,@#STALL      ;YES--GO TO STALL ROUTINE
 6475  006534  001320                          .WORD   STALLO          ;TIME POINTER
 6476  006536  005237  001320                  INC     @#STALLO        ;UPDATE STALL TIME
 6477  006542  023737  001370  001320          CMP     @#MXSTAL,@#STALLO ;TIME TOO BIG?
 6478  006550  003347                          BGT     4$              ;NO--BRANCH
 6479  006552  005037  001320                  CLR     @#STALLO        ;YES--SET STALL TIME BACK TO ZERO
 6480  006556  163701  002162          6$:     SUB     @#IC4,R1        ;NEXT CYLINDER
 6481  006562  020137  002156                  CMP     R1,@#FC4        ;DONE?
 6482  006566  002373                          BGE     6$              ;NO--BRANCH
 6483  006570  000004                  EXIT4:  SCOPE                   ;LOOP
 6484
 6497                                  ;*****************************************************************
   (3)                                 ;*TEST 5        CONVERING/DIVERGING SEEK TEST
   (4)
   (4)                                 ;*      THIS TEST WILL CAUSE THE DRIVE TO EXECUTE FORWARD AND REVERSE
   (4)                                 ;*      SEEKS FROM "NC1" AND "NC2" RESPECTIVELY, "NC1" WILL BE INCREMENTED
   (4)                                 ;*      BY "IC" AND "NC2" WILL BE DECREMENTED BY "IC" UNTIL "NC1" IS
   (4)                                 ;*      GREATER THAN THE INITIAL VALUE OF "NC2" AND "NC2" IS
```

```
(4)                                     ;*      LESS THAN THE INITIAL VALUE OF "NC1".  AT THE COMPLETION OF
(4)                                     ;*      EACH SEEK COMMAND THE PROPER INDICATORS ARE EXAMINED TO
(4)                                     ;*      INSURE PROPER OPERATION.  "NC1" AND "NC2" DEFAULT TO
(4)                                     ;*      "FC" AND "LC" RESPECTIVELY.
(4)
(3)                                     ;*****************************************************************
(3)  006572                             TST5:
(4)  006572  033737  001406  001224             BIT     @#BITS+<5*2>,TSTNMS ;DO THIS TEST?
(4)  006600  001002                             BNE     64$             ;YES--BRANCH
(4)  006602  000137  006750                     JMP     TST6            ;NO--GO TO THE NEXT TEST
(4)  006606  013737  002172  001176  64$:      MOV     @#RPT5,$TIMES   ;GET THE ITERATION COUNT
(3)  006614  012737  006666  001106             MOV     #TEST5,@#$LPADR
(3)  006622  012737  006572  001110             MOV     #TST5,@#$LPERR  ;SETUP THE ERROR LOOP ADDRESS
(3)  006630  012737  000005  001102             MOV     #5,@#$TSTNM             ;SET UP TEST NUMBER AND
(3)                                                                     ;CLEAR THE ERROR FLAG ($ERFLG)
(3)  006636  013737  001102  177570             MOV     $TSTNM,@#DISPLAY        ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
6498 006644  113737  002204  003470             MOVB    @#FS5,@#DPB.B+10 ;FS
6499 006652  113737  002202  003471             MOVB    @#FT5,@#DPB.B+11 ;FT
6500 006660  012737  006746  001234             MOV     #EXIT5,@#BYPASS ;GO TO EXIT5 ON ERROR
6501 006666  012706  001100         TEST5:      MOV     #STACK,SP
6502 006672  013701  002174                     MOV     @#FC5,R1        ;START NC1 AT FC
6503 006676  013702  002176                     MOV     @#LC5,R2        ;START NC2 AT LC
6504 006702  010137  003472         1$:         MOV     R1,@#DPB.B+12   ;NC1
6505 006706  004037  020010                     JSR     R0,@#CALL.B             ;GO EXECUTE THE COMMAND
6506 006712  010237  003472                     MOV     R2,@#DPB.B+12   ;NC2
6507 006716  004037  020010                     JSR     R0,@#CALL.B             ;GO EXECUTE THE COMMAND
6508 006722  063701  002200                     ADD     @#IC5,R1        ;NEXT NC1
6509 006726  163702  002200                     SUB     @#IC5,R2        ;NEXT NC2
6510 006732  020137  002176                     CMP     R1,@#LC5        ;DONE?
6511 006736  003003                             BGT     EXIT5           ;YES--BRANCH
6512 006740  020237  002174                     CMP     R2,@#FC5        ;?
6513 006744  002356                             BGE     1$              ;NO--BRANCH
6514 006746  000004         EXIT5:      SCOPE                           ;LOOP
6515
6526                                    ;*****************************************************************
(3)                                     ;*TEST 6        SERVO ADDRESSING LOGIC NOISE GENERATOR
(4)
(4)                                     ;*      IN THIS TEST A SEEK IS DONE TO CYL "NC" THEN A SEEK TO
(4)                                     ;*      NC+4 THEN NC+1 THEN NC+3 THEN NC+2 THEN NC+5.  NOW "NC" IS UPDATED
(4)                                     ;*      BY "IC" AND THE ABOVE SEQUENCE IS REPEATED UNITL "LC" IS
(4)                                     ;*      EXCEEDED BY ANY OF THE ABOVE VALUES.  THE INITIAL VALUE OF "NC"
(4)                                     ;*      IS "FC".  AT THE COMPLETION OF EACH SEEK COMMAND THE
(4)                                     ;*      PROPER INDICATORS ARE EXAMINED TO INSURE PROPER OPERATION.
(4)
(3)                                     ;*****************************************************************
(3)  006750                             TST6:
(4)  006750  033737  001410  001224             BIT     @#BITS+<6*2>,TSTNMS ;DO THIS TEST?
(4)  006756  001002                             BNE     64$             ;YES--BRANCH
(4)  006760  000137  007172                     JMP     TST7            ;NO--GO TO THE NEXT TEST
(4)  006764  013737  002210  001176  64$:      MOV     @#RPT6,$TIMES   ;GET THE ITERATION COUNT
(3)  006772  012737  007044  001106             MOV     #TEST6,@#$LPADR
(3)  007000  012737  006750  001110             MOV     #TST6,@#$LPERR  ;SETUP THE ERROR LOOP ADDRESS
(3)  007006  012737  000006  001102             MOV     #6,@#$TSTNM             ;SET UP TEST NUMBER AND
(3)                                                                     ;CLEAR THE ERROR FLAG ($ERFLG)
```

```
  (3)  007014  013737  001102  177570          MOV     $TSTNM,@#DISPLAY        ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
 6527  007022  113737  002222  003470          MOVB    @#FS6,@#DPB.B+10 ;FS
 6528  007030  113737  002220  003471          MOVB    @#FT6,@#DPB.B+11 ;FT
 6529  007036  012737  007170  001234          MOV     #EXIT6,@#BYPASS ;GO TO EXIT6 ON ERROR
 6530  007044  012706  001100          TEST6:  MOV     #STACK,SP       ;SETUP STACK
 6531  007050  013701  002212                  MOV     @#FC6,R1        ;PICKUP "FC"
 6532  007054  013702  002214                  MOV     @#LC6,R2                ;FORM LAST CYLINDER THAT
 6533  007060  162702  000005                  SUB     #5,R2           ;IS AVAILABLE FOR TESTING
 6534  007064  020102                  1$:     CMP     R1,R2           ;LAST CYLINDER
 6535  007066  003040                          BGT     EXIT6           ;YES--BRANCH
 6536  007070  010137  003472                  MOV     R1,@#DPB.B+12   ;NC
 6537  007074  004037  020010                  JSR     R0,@#CALL.B             ;GO EXECUTE THE COMMAND
 6538  007100  062737  000004  003472          ADD     #4,@#DPB.B+12   ;NC+4
 6539  007106  004037  020010                  JSR     R0,@#CALL.B             ;GO EXECUTE THE COMMAND
 6540  007112  162737  000003  003472          SUB     #3,@#DPB.B+12   ;NC+1
 6541  007120  004037  020010                  JSR     R0,@#CALL.B             ;GO EXECUTE THE COMMAND
 6542  007124  062737  000002  003472          ADD     #2,@#DPB.B+12   ;NC+3
 6543  007132  004037  020010                  JSR     R0,@#CALL.B             ;GO EXECUTE THE COMMAND
 6544  007136  162737  000001  003472          SUB     #1,@#DPB.B+12   ;NC+2
 6545  007144  004037  020010                  JSR     R0,@#CALL.B             ;GO EXECUTE THE COMMAND
 6546  007150  062737  000003  003472          ADD     #3,@#DPB.B+12   ;NC+5
 6547  007156  004037  020010                  JSR     R0,@#CALL.B             ;GO EXECUTE THE COMMAND
 6548  007162  063701  002216                  ADD     @#IC6,R1
 6549  007166  000736                          BR      1$
 6550  007170  000004          EXIT6:  SCOPE                   ;LOOP
```

```
                                                  .SBTTL  *** TIMING TESTS (7-12) ***

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

        ;*THE TIMING TESTS WILL INSURE 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.

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

                                          ;*************************************************************
 (3)                                      ;*TEST 7          ROTATIONAL SPEED TIMING TEST
 (4)
 (4)                                      ;*        THIS TEST WILL START A SEARCH TO CYLINDER 0, TRACK 0, SECTOR
 (4)                                      ;*        0.  AS SOON AS THE INTERRUPT OCCURS, THE GO BIT IS SET AGAIN
 (4)                                      ;*        AND THE OPERATION IS TIMED.  THIS PROCEDURE IS REPEATED 10
 (4)                                      ;*        TIMES THEN THE AVERAGE TIME IS CALCULATED AND CHECKED TO
 (4)                                      ;*        INSURE IT IS WITHIN TOLERANCE:
 (4)                                      ;*                16.67 MS/REV + OR - 2% IF 60HZ
 (4)                                      ;*                16.67 MS/REV + OR - 2.5% IF 50HZ.
 (4)
 (3)                                      ;*************************************************************
 (3)   007172                             TST7:
 (4)   007172  033737  001412  001224             BIT     @#BITS+<7*2>,TSTNMS ;DO THIS TEST?
 (4)   007200  001002                             BNE     64$                 ;YES--BRANCH
 (4)   007202  000137  007712                     JMP     TST10               ;NO--GO TO THE NEXT TEST
 (4)   007206  013737  002226  001176     64$:    MOV     @#RPT7,$TIMES       ;GET THE ITERATION COUNT
 (3)   007214  012737  007172  001106             MOV     #TST7,@#$LPADR      ;SETUP THE SCOPE LOOP ADDRESS
 (3)   007222  012737  007172  001110             MOV     #TST7,@#$LPERR      ;SETUP THE ERROR LOOP ADDRESS
 (3)   007230  012737  000007  001102             MOV     #7,@#$TSTNM                 ;SET UP TEST NUMBER AND
 (3)                                                                                  ;CLEAR THE ERROR FLAG ($ERFLG)
 (3)   007236  013737  001102  177570             MOV     $TSTNM,@#DISPLAY            ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
       007244  005737  001226                     TST     @#CLKSTA            ;KW11-P CLOCK?
       007250  003002                             BGT     1$                  ;YES--START TEST
       007252  000137  007712                     JMP     TST10               ;NO--GO TO NEXT TEST
       007256  004037  021356             1$:     JSR     R0,@#SRCH00         ;DO A MASSBUS INIT & RECAL
       007262  000401                             BR      2$                  ;RETURN HERE IF NO ERROR
       007264  000563                             BR      EXIT7               ;RETURN HERE IF ERROR
       007266  013764  002230  000034     2$:     MOV     @#FC7,RHCA(R4)      ;FC
       007274  013746  002234                     MOV     @#FS7,-(SP)         ;FS
       007300  113766  002232  000001             MOVB    @#FT7,1(SP)         ;FT
       007306  012664  000006                     MOV     (SP)+,RHDA(R4)      ;LOAD FT/FS
       007312  012737  007634  001200             MOV     #EXIT7,$ESCAPE      ;;ESCAPE TO EXIT7 ON ERROR
```

```
 6590  007320  005005                          CLR     R5              ;COUNT UP
 6591  007322  012703  002330                  MOV     #T7A,R3         ;60HZ PARAMETERS
 6592  007326  032737  000100  001212          BIT     #SW06,@#C.SWR   ;60 HZ?
 6593  007334  001402                          BEQ     TEST7           ;YES--BRANCH
 6594  007336  012703  002340                  MOV     #T7B,R3         ;NO--50 HZ PARAMETERS
 6595  007342  012706  001100          TEST7:  MOV     #STACK,SP       ;SETUP STACK
 6596  007346  012701  000012                  MOV     #^D10,R1        ;TIME 10 SEARCHES
 6597  007352  004737  021542                  JSR     PC,@#STRTMR     ;INITIALIZE THE TIMERS
 6598  007356  012777  007564  171750          MOV     #7$,@PKV        ;SETUP VECTOR IN CASE OF OVERFLOW
 6599  007364  012777  021540  017576          MOV     #DORTI,@RPVEC   ;SETUP RP04 VECTOR
 6600  007372  005077  171744          1$:     CLR     @PKB            ;START COUNTING AT ZERO
 6601  007376  012777  000131  171734          MOV     #131,@PKCS      ;INT.EN., COUNT UP AT 100KHZ
 6602  007404  012714  000131                  MOV     #SEARCH,(R4)    ;START A SEARCH
 6603  007410  000001                          WAIT                    ;WAIT ON INTERRUPT
 6604  007412  042777  000101  171720          BIC     #101,@PKCS      ;STOP THE CLOCK
 6605  007420  032764  040000  000012          BIT     #BIT14,RHDS1(R4) ;ERROR?
 6606  007426  001415                          BEQ     2$              ;NO--BRANCH
 6607  007430  104416                          SAVREG                  ;SAVE R0-R5
   (1) 007432  012702  003520                  MOV     #DTADPB,R2      ;DPB POINTER
   (1) 007436  004737  033434                  JSR     PC,@#SVRH11     ;SAVE ALL THE RH11/RP04 REGISTERS
   (1) 007442  012764  000040  000010          MOV     #BIT05,RHCS2(R4) ;MASSBUS CLEAR
   (1) 007450  013764  003520  000010          MOV     @#DTADPB,RHCS2(R4) ;SELECT DRIVE
   (1) 007456  104420                          RESREG                  ;RESTORE R0-R5
 6608  007460  104017                          ERROR   17
 6609  007462  005077  171654          2$:     CLR     @PKB            ;START THE COUNT AT ZERO
 6610  007466  012714  000131                  MOV     #SEARCH,(R4)    ;START A SEARCH
 6611  007472  012777  000131  171640          MOV     #131,@PKCS      ;START THE CLOCK
 6612  007500  000001                          WAIT                    ;WAIT ON INTERRUPT
 6613  007502  042777  000101  171630          BIC     #101,@PKCS      ;STOP THE CLOCK
 6614  007510  032764  040000  000012          BIT     #BIT14,RHDS1(R4) ;IS "ERR=1"?
 6615  007516  001415                          BEQ     3$              ;NO--BRANCH
 6616  007520  104416                          SAVREG                  ;SAVE R0-R5
   (1) 007522  012702  003520                  MOV     #DTADPB,R2      ;DPB POINTER
   (1) 007526  004737  033434                  JSR     PC,@#SVRH11     ;SAVE ALL THE RH11/RP04 REGISTERS
   (1) 007532  012764  000040  000010          MOV     #BIT05,RHCS2(R4) ;MASSBUS CLEAR
   (1) 007540  013764  003520  000010          MOV     @#DTADPB,RHCS2(R4) ;SELECT DRIVE
   (1) 007546  104420                          RESREG                  ;RESTORE R0-R5
 6617  007550  104017                          ERROR   17              ;DISK ERROR OCCURRED
 6618  007552  004737  021606          3$:     JSR     PC,@#COUNT      ;UPDATE THE COUNT
 6619  007556  005301                          DEC     R1              ;DONE?
 6620  007560  003304                          BGT     1$              ;NO--BRANCH
 6621  007562  000424                          BR      EXIT7           ;YES--GO TO THE EXIT
 6622  007564  042777  000101  171546  7$:     BIC     #101,@PKCS      ;STOP THE CLOCK
 6623  007572  005037  177776                  CLR     @#PS            ;DROP THE PRIORITY
 6624  007576  012600                          MOV     (SP)+,R0        ;PC OF WAIT+2
 6625  007600  005726                          TST     (SP)+           ;POP THE PS FROM THE STACK
 6626  007602  104416                          SAVREG                  ;SAVE R0-R5
   (1) 007604  012702  003520                  MOV     #DTADPB,R2      ;DPB POINTER
   (1) 007610  004737  033434                  JSR     PC,@#SVRH11     ;SAVE ALL THE RH11/RP04 REGISTERS
   (1) 007614  012764  000040  000010          MOV     #BIT05,RHCS2(R4) ;MASSBUS CLEAR
   (1) 007622  013764  003520  000010          MOV     @#DTADPB,RHCS2(R4) ;SELECT DRIVE
   (1) 007630  104420                          RESREG                  ;RESTORE R0-R5
 6627  007632  104020                          ERROR   20              ;CLOCK OVERFLOWED
 6628  007634  000004                  EXIT7:  SCOPE
```

```
 6629  007636  012764  000040  000010          MOV     #BIT05,RHCS2(R4) ;MASSBUS INIT.
  (1)  007644  013764  003520  000010          MOV     @#DTADPB,RHCS2(R4) ;SELECT DRIVE
 6630  007652  004737  017360                  JSR     PC,@#ST.CLK     ;INITIALIZE THE CLOCK
 6631  007656  012777  031356  017304          MOV     #ISR,@RPVEC     ;RESTORE RH11/RP04 INT. VECTOR
 6632  007664  032737  000100  001212          BIT     #SW06,@#C.SWR   ;60 HZ?
 6633  007672  001004                          BNE     1$              ;NO -- BRANCH
 6634  007674  004037  021740                  JSR     R0,@#TYPTIM     ;GO TYPE THE TIMES
  (1)  007700  002330                          T7A                     ;POINTER
 6635  007702  000403                          BR      TST10           ;GO TO NEXT TEST
 6636  007704                          1$:
  (1)  007704  004037  021740                  JSR     R0,@#TYPTIM     ;GO TYPE THE TIMES
  (1)  007710  002340                          T7B                     ;POINTER
 6637
 6638
 6651                                  ;*****************************************************************
  (3)                                  ;*TEST 10       ONE CYLINDER SEEK TIMING TEST
  (4)
  (4)                                  ;*      THIS TEST WILL COMMAND FORWARD SEEK CYCLES TO ADVANCE THE
  (4)                                  ;*      CYLINDER BY ONE UNTIL THE INCREMENT IS GREATER THAN THE
  (4)                                  ;*      NUMBER OF CYLINDERS TO CYLINDER 410, THEN REVERSE SEEK
  (4)                                  ;*      CYCLES TO RETRACT THE CYLINDER BY ONE UNTIL THE INCREMENT
  (4)                                  ;*      IS GREATER THAN THE NUMBER OF CYLINDERS TO CYLINDER 0.  THE
  (4)                                  ;*      TIME TO PERFORM EACH SEEK IS CHECKED TO INSURE IT DOES NOT
  (4)                                  ;*      EXCEED THE MAXIMUM TIME PERMITTED FOR A ONE CYLINDER SEEK.
  (4)                                  ;*      THE TIME MUST BE LESS THAN 10MS BUT GREATER THAN 3MS.
  (4)
  (3)                                  ;*****************************************************************
  (3)  007712                          TST10:
  (4)  007712  033737  001414  001224          BIT     @#BITS+<10*2>,TSTNMS ;DO THIS TEST?
  (4)  007720  001002                          BNE     64$             ;YES--BRANCH
  (4)  007722  000137  010334                  JMP     TST11           ;NO--GO TO THE NEXT TEST
  (4)  007726  013737  002240  001176  64$:     MOV     @#RPT10,$TIMES  ;GET THE ITERATION COUNT
  (3)  007734  012737  007712  001106          MOV     #TST10,@#$LPADR ;SETUP THE SCOPE LOOP ADDRESS
  (3)  007742  012737  007712  001110          MOV     #TST10,@#$LPERR ;SETUP THE ERROR LOOP ADDRESS
  (3)  007750  012737  000010  001102          MOV     #10,@#$TSTNM    ;SET UP TEST NUMBER AND
  (3)                                                                  ;CLEAR THE ERROR FLAG ($ERFLG)
  (3)  007756  013737  001102  177570          MOV     $TSTNM,@#DISPLAY ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
 6652  007764  005737  001226                  TST     @#CLKSTA        ;KW11-P CLOCK?
 6653  007770  003002                          BGT     1$              ;YES--START TEST
 6654  007772  000137  010334                  JMP     TST11           ;NO--GO TO NEXT TEST
 6655  007776  004037  021356          1$:     JSR     R0,@#SRCH00     ;DO A MASSBUS INIT. AND RECAL
 6656  010002  000401                          BR      2$              ;NO ERROR RETURN
 6657  010004  000534                          BR      EXIT10          ;ERROR RETURN--SCOPE LOOP CALL
 6658  010006  012703  002350          2$:     MOV     #T10,R3         ;PARAMETER POINTER
 6659  010012  012737  010276  001200          MOV     #EXIT10,$ESCAPE ;;ESCAPE TO EXIT10 ON ERROR
 6660  010020  012706  001100          TEST10: MOV     #STACK,SP       ;SETUP STACK
 6661  010024  012737  000001  003532          MOV     #1,@#DTADPB+12  ;START WITH CYLINDER #1
 6662  010032  005005                          CLR     R5              ;SET THE UP/DOWN SWITCH TO UP
 6663  010034  004737  021542                  JSR     PC,@#STRTMR     ;INITIALIZE THE TIMERS
 6664  010040  012777  010226  171266          MOV     #7$,@PKV        ;SETUP INCASE OF OVERFLOW
 6665  010046  012777  021540  017114          MOV     #DORTI,@RPVEC   ;SET RP04 VECTOR
 6666  010054  005077  171262          1$:     CLR     @PKB            ;START THE COUNTER AT ZERO
 6667  010060  013764  003532  000034          MOV     @#DTADPB+12,RHCA(R4) ;LOAD DESIRED CYLINDER
 6668  010066  012714  000105                  MOV     #SEEK,(R4)      ;START A SEEK
```

```
 6669  010072  012777  000131  171240          MOV     #131,@PKCS      ;START THE CLOCK
 6670  010100  000001                          WAIT                    ;WAIT ON INTERRUPT
 6671  010102  042777  000101  171230          BIC     #101,@PKCS      ;STOP THE CLOCK
 6672  010110  032764  040000  000012          BIT     #BIT14,RHDS1(R4) ;ANY DISK ERRORS?
 6673  010116  001415                          BEQ     2$              ;NO--BRANCH
 6674  010120  104416                          SAVREG                  ;SAVE R0-R5
   (1) 010122  012702  003520                  MOV     #DTADPB,R2      ;DPB POINTER
   (1) 010126  004737  033434                  JSR     PC,@#SVRH11     ;SAVE ALL THE RH11/RP04 REGISTERS
   (1) 010132  012764  000040  000010          MOV     #BIT05,RHCS2(R4) ;MASSBUS CLEAR
   (1) 010140  013764  003520  000010          MOV     @#DTADPB,RHCS2(R4) ;SELECT DRIVE
   (1) 010146  104420                          RESREG                  ;RESTORE R0-R5
 6675  010150  104017                          ERROR   17              ;REPORT THE ERROR
 6676  010152  004737  021606          2$:     JSR     PC,@#COUNT      ;COUNT THIS SEEKS TIME
 6677  010156  004737  021166                  JSR     PC,@#TWOMS      ;STALL FOR 2 MILLISECONDS
 6678  010162  005705                          TST     R5              ;UP OR DOWN?
 6679  010164  001011                          BNE     4$              ;DOWN--BRANCH
 6680  010166  005237  003532          3$:     INC     @#DTADPB+12     ;MOVE TO NEXT CYLINDER
 6681  010172  023727  003532  000632          CMP     @#DTADPB+12,#410 ;OUT OF CYLINDERS?
 6682  010200  002725                          BLT     1$              ;NO--GO DO THE NEXT SEEK
 6683  010202  012705  177777                  MOV     #-1,R5          ;SET UP/DOWN SWITCH TO DOWN
 6684  010206  000722                          BR      1$              ;GO DO THE NEXT SEEK
 6685  010210  005337  003532          4$:     DEC     @#DTADPB+12     ;MOVE TO NEXT CYLINDER
 6686  010214  023727  003532  000000          CMP     @#DTADPB+12,#0  ;OUT OF CYLINDERS?
 6687  010222  003314                          BGT     1$              ;NO--GO DO THE NEXT SEEK
 6688  010224  000424                          BR      EXIT10          ;GO TO THE EXIT
 6689  010226  042777  000101  171104  7$:     BIC     #101,@PKCS      ;STOP THE CLOCK
 6690  010234  005037  177776                  CLR     @#PS            ;DROP THE PRIORITY
 6691  010240  012600                          MOV     (SP)+,R0        ;PC OF WAIT+2
 6692  010242  005726                          TST     (SP)+           ;POP THE PS FROM THE STACK
 6693  010244  104416                          SAVREG                  ;SAVE R0-R5
   (1) 010246  012702  003520                  MOV     #DTADPB,R2      ;DPB POINTER
   (1) 010252  004737  033434                  JSR     PC,@#SVRH11     ;SAVE ALL THE RH11/RP04 REGISTERS
   (1) 010256  012764  000040  000010          MOV     #BIT05,RHCS2(R4) ;MASSBUS CLEAR
   (1) 010264  013764  003520  000010          MOV     @#DTADPB,RHCS2(R4) ;SELECT DRIVE
   (1) 010272  104420                          RESREG                  ;RESTORE R0-R5
 6694  010274  104020                          ERROR   20              ;REPORT CLOCK OVERFLOW
 6695  010276  000004                  EXIT10: SCOPE
 6696  010300  012764  000040  000010          MOV     #BIT05,RHCS2(R4) ;MASSBUS INIT.
   (1) 010306  013764  003520  000010          MOV     @#DTADPB,RHCS2(R4) ;SELECT DRIVE
 6697  010314  004737  017360                  JSR     PC,@#ST.CLK     ;INITIALIZE THE CLOCK
 6698  010320  012777  031356  016642          MOV     #ISR,@RPVEC     ;RESTORE RH11/RP04 INT. VECTOR
 6699  010326  004037  021740                  JSR     R0,@#TYPTIM     ;GO TYPE THE TIMES
   (1) 010332  002350                          T10                     ;POINTER
 6700
 6701
 6712                          ;*****************************************************************
  (3)                          ;*TEST 11       AVERAGE SEEK TIMING TEST
  (4)
  (4)                          ;*      THIS TEST WILL COMMAND A FORWARD SEEK FROM CYLINDER 0 TO
  (4)                          ;*      CYLINDER 136, THEN A REVERSE SEEK FROM CYLINDER 136 TO
  (4)                          ;*      CYLINDER 0.  BOTH SEEKS ARE TIMED AND CHECKED TO INSURE THEY
  (4)                          ;*      ARE WITHIN THE TOLERANCE ALLOWED FOR THE AVERAGE SEEK TIME.
  (4)                          ;*      THIS SEQUENCE IS REPEATED 128 TIMES (FOR A TOTAL
  (4)                          ;*      OF 256 SEEKS).  THE AVERAGE SEEK TIME IS 28 MS + OR - 2MS.
```

```
(4)
(3)                                     ;*****************************************************************
(3)  010334                             TST11:
(4)  010334  033737  001416  001224              BIT     @#BITS+(11*2),TSTNMS ;DO THIS TEST?
(4)  010342  001002                              BNE     64$             ;YES--BRANCH
(4)  010344  000137  011034                      JMP     TST12           ;NO--GO TO THE NEXT TEST
(4)  010350  013737  002244  001176  64$:       MOV     @#RPT11,$TIMES  ;GET THE ITERATION COUNT
(3)  010356  012737  010334  001106              MOV     #TST11,@#$LPADR ;SETUP THE SCOPE LOOP ADDRESS
(3)  010364  012737  010334  001110              MOV     #TST11,@#$LPERR ;SETUP THE ERROR LOOP ADDRESS
(3)  010372  012737  000011  001102              MOV     #11,@#$TSTNM            ;SET UP TEST NUMBER AND
(3)                                                                             ;CLEAR THE ERROR FLAG ($ERFLG)
(3)  010400  013737  001102  177570              MOV     $TSTNM,@#DISPLAY        ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
6713 010406  005737  001226                      TST     @#CLKSTA        ;KW11-P CLOCK?
6714 010412  003002                              BGT     1$              ;YES--START TEST
6715 010414  000137  011034                      JMP     TST12           ;NO--GO TO NEXT TEST
6716 010420  004037  021356              1$:     JSR     R0,@#SRCH00     ;DO A MASSBUS INIT & RECAL
6717 010424  000401                              BR      2$              ;RETURN HERE IF NO ERROR
6718 010426  000563                              BR      EXIT11          ;RETURN HERE ON ERROR
6719 010430  012703  002360              2$:     MOV     #T11,R3         ;PARAMETER POINTER
6720 010434  012737  010776  001200              MOV     #EXIT11,$ESCAPE ;;ESCAPE TO EXIT11 ON ERROR
6721 010442  012706  001100              TEST11: MOV     #STACK,SP       ;SETUP STACK
6722 010446  012701  000200                      MOV     #^D128,R1       ;REPEAT "D-136-0" 128 TIMES
6723 010452  004737  021542                      JSR     PC,@#STRTMR     ;INIT. THE COUNTERS
6724 010456  012777  010726  170650              MOV     #7$,@PKV        ;SET UP VECTOR IN CASE OF OVERFLOW
6725 010464  012777  021540  016476              MOV     #DORTI,@RPVEC   ;SETUP RP04 VECTOR
6726 010472  005077  170644              1$:     CLR     @PKB            ;START COUNT AT ZERO
6727 010476  012764  000210  000034              MOV     #^D136,RHCA(R4) ;CYLINDER=136
6728 010504  012764  000105  000000              MOV     #SEEK,RHCS1(R4) ;START A SEEK
6729 010512  012777  000131  170620              MOV     #131,@PKCS      ;START THE CLOCK
6730 010520  000001                              WAIT                    ;WAIT ON INTERRUPT
6731 010522  042777  000101  170610              BIC     #101,@PKCS      ;STOP CLOCK
6732 010530  032764  040000  000012              BIT     #BIT14,RHDS1(R4) ;ERR=1?
6733 010536  001415                              BEQ     2$              ;NO--BRANCH
6734 010540  104416                              SAVREG                  ;SAVE R0-R5
 (1) 010542  012702  003520                      MOV     #DTADPB,R2      ;DPB POINTER
 (1) 010546  004737  033434                      JSR     PC,@#SVRH11     ;SAVE ALL THE RH11/RP04 REGISTERS
 (1) 010552  012764  000040  000010              MOV     #BIT05,RHCS2(R4) ;MASSBUS CLEAR
 (1) 010560  013764  003520  000010              MOV     @#DTADPB,RHCS2(R4) ;SELECT DRIVE
 (1) 010566  104420                              RESREG                  ;RESTORE R0-R5
6735 010570  104017                              ERROR   17
6736 010572  005005                      2$:     CLR     R5              ;SET UP/DOWN SWITCH TO UP
6737 010574  004737  021606                      JSR     PC,@#COUNT      ;UPDATE THE COUNT
6738 010600  004737  021166                      JSR     PC,@#TWOMS      ;STALL FOR 2 MILLISECONDS
6739 010604  005077  170532                      CLR     @PKB            ;START THE COUNT AT ZERO
6740 010610  012764  000000  000034              MOV     #0,RHCA(R4)     ;CYLINDER=0
6741 010616  012764  000105  000000              MOV     #SEEK,RHCS1(R4) ;START A SEEK
6742 010624  012777  000131  170506              MOV     #131,@PKCS      ;START THE CLOCK
6743 010632  000001                              WAIT                    ;WAIT ON INTERRUPT
6744 010634  042777  000101  170476              BIC     #101,@PKCS      ;STOP THE CLOCK
6745 010642  032764  040000  000012              BIT     #BIT14,RHDS1(R4) ;ERR=1?
6746 010650  001415                              BEQ     3$              ;NO--BRANCH
6747 010652  104416                              SAVREG                  ;SAVE R0-R5
 (1) 010654  012702  003520                      MOV     #DTADPB,R2      ;DPB POINTER
 (1) 010660  004737  033434                      JSR     PC,@#SVRH11     ;SAVE ALL THE RH11/RP04 REGISTERS
```

```
  (1)  010664  012764  000040  000010          MOV     #BIT05,RHCS2(R4) ;MASSBUS CLEAR
  (1)  010672  013764  003520  000010          MOV     @#DTADPB,RHCS2(R4) ;SELECT DRIVE
  (1)  010700  104420                          RESREG                   ;RESTORE R0-R5
6748   010702  104017                          ERROR   17
6749   010704  012705  177777          3$:     MOV     #-1,R5           ;SET UP/DOWN SWITCH TO DOWN
6750   010710  004737  021606                  JSR     PC,@#COUNT       ;UPDATE THE COUNT
6751   010714  004737  021166                  JSR     PC,@#TWOMS       ;STALL FOR 2 MILLISECONDS
6752   010720  005301                          DEC     R1               ;DONE?
6753   010722  003263                          BGT     1$               ;NO--BRANCH
6754   010724  000424                          BR      EXIT11           ;YES--EXIT
6755   010726  042777  000101  170404  7$:     BIC     #101,@PKCS       ;STOP THE CLOCK
6756   010734  005037  177776                  CLR     @#PS             ;DROP THE PRIORITY
6757   010740  012600                          MOV     (SP)+,R0         ;PC OF WAIT+2
6758   010742  005726                          TST     (SP)+            ;POP THE PS FROM THE STACK
6759   010744  104416                          SAVREG                   ;SAVE R0-R5
  (1)  010746  012702  003520                  MOV     #DTADPB,R2       ;DPB POINTER
  (1)  010752  004737  033434                  JSR     PC,@#SVRH11      ;SAVE ALL THE RH11/RP04 REGISTERS
  (1)  010756  012764  000040  000010          MOV     #BIT05,RHCS2(R4) ;MASSBUS CLEAR
  (1)  010764  013764  003520  000010          MOV     @#DTADPB,RHCS2(R4) ;SELECT DRIVE
  (1)  010772  104420                          RESREG                   ;RESTORE R0-R5
6760   010774  104020                          ERROR   20               ;CLOCK OVERFLOWED
6761   010776  000004                  EXIT11: SCOPE
6762   011000  012764  000040  000010          MOV     #BIT05,RHCS2(R4) ;MASSBUS INIT.
  (1)  011006  013764  003520  000010          MOV     @#DTADPB,RHCS2(R4) ;SELECT DRIVE
6763   011014  004737  017360                  JSR     PC,@#ST.CLK      ;INITIALIZE THE CLOCK
6764   011020  012777  031356  016142          MOV     #ISR,@RPVEC      ;RESTORE RH11/RP04 INT. VECTOR
6765   011026  004037  021740                  JSR     R0,@#TYPTIM      ;GO TYPE THE TIMES
  (1)  011032  002360                          T11                      ;POINTER
6766
6777                                   ;*****************************************************************
  (3)                                  ;*TEST 12       MAXIMUM SEEK TIMING TEST
  (4)
  (4)                                  ;*      THIS TEST WILL COMMAND A FORWARD SEEK FROM CYLINDER 0 TO
  (4)                                  ;*      CYLINDER 410, THEN A REVERSE SEEK FROM CYLINDER 410 TO
  (4)                                  ;*      CYLINDER 0.  BOTH SEEKS ARE TIMED AND CHECKED TO INSURE
  (4)                                  ;*      THEY ARE WITHIN THE TOLERANCE ALLOWED FOR THE MAXIMUM SEEK
  (4)                                  ;*      TIME.  THIS SEQUENCE IS REPEATED 128 TIMES (FOR
  (4)                                  ;*      A TOTAL OF 256 SEEKS).  THE MAXIMUM SEEK TIME IS 50MS + 4MS OR - 2MS.
  (4)
  (3)                                  ;*****************************************************************
  (3)  011034                          TST12:
  (4)  011034  033737  001420  001224          BIT     @#BITS+<12*2>,TSTNMS ;DO THIS TEST?
  (4)  011042  001002                          BNE     64$              ;YES--BRANCH
  (4)  011044  000137  011534                  JMP     TST13            ;NO--GO TO THE NEXT TEST
  (4)  011050  013737  002250  001176  64$:     MOV     @#RPT12,$TIMES   ;GET THE ITERATION COUNT
  (3)  011056  012737  011034  001106          MOV     #TST12,@#$LPADR  ;SETUP THE SCOPE LOOP ADDRESS
  (3)  011064  012737  011034  001110          MOV     #TST12,@#$LPERR  ;SETUP THE ERROR LOOP ADDRESS
  (3)  011072  012737  000012  001102          MOV     #12,@#$TSTNM             ;SET UP TEST NUMBER AND
  (3)                                                                   ;CLEAR THE ERROR FLAG ($ERFLG)
  (3)  011100  013737  001102  177570          MOV     $TSTNM,@#DISPLAY         ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
6778   011106  005737  001226                  TST     @#CLKSTA         ;KW11-P CLOCK
6779   011112  003002                          BGT     1$               ;YES--START TEST
6780   011114  000137  011534                  JMP     TST13            ;NO--GO TO NEXT TEST
6781   011120  004037  021356          1$:     JSR     R0,@#SRCH00      ;DO A MASSBUS INIT & RECAL
```

```
 6782  011124  000401                          BR      2$              ;RETURN HERE IF NO ERROR
 6783  011126  000563                          BR      EXIT12          ;RETURN HERE ON ERROR
 6784  011130  012703  002370          2$:     MOV     #T12,R3         ;PARAMETER POINTER
 6785  011134  012737  011476  001200        MOV     #EXIT12,$ESCAPE ;;ESCAPE TO EXIT12 ON ERROR
 6786  011142  012706  001100          TEST12: MOV     #STACK,SP       ;SETUP STACK
 6787  011146  012701  000200                  MOV     #^D128,R1       ;REPEAT "0-410-0" 128 TIMES
 6788  011152  004737  021542                  JSR     PC,@#STRTMR     ;INIT. THE TIMERS
 6789  011156  012777  011426  170150          MOV     #7$,@PKV        ;SETUP VECTOR IN CASE OF OVERFLOW
 6790  011164  012777  021540  015776          MOV     #DORTI,@RPVEC   ;SETUP RP04 VECTOR
 6791  011172  005077  170144          1$:     CLR     @PKB            ;START COUNTING FROM ZERO
 6792  011176  012764  000632  000034          MOV     #410.,RHCA(R4)  ;CYLINDER=410
 6793  011204  012764  000105  000000          MOV     #SEEK,RHCS1(R4) ;START A SEEK
 6794  011212  012777  000131  170120          MOV     #131,@PKCS      ;START THE CLOCK
 6795  011220  000001                          WAIT                    ;WAIT ON INTERRUPT
 6796  011222  042777  000101  170110          BIC     #101,@PKCS      ;STOP THE CLOCK
 6797  011230  032764  040000  000012          BIT     #BIT14,RHDS1(R4) ;ERR=1?
 6798  011236  001415                          BEQ     2$              ;NO--BRANCH
 6799  011240  104416                          SAVREG                  ;SAVE R0-R5
   (1) 011242  012702  003520                  MOV     #DTADPB,R2      ;DPB POINTER
   (1) 011246  004737  033434                  JSR     PC,@#SVRH11     ;SAVE ALL THE RH11/RP04 REGISTERS
   (1) 011252  012764  000040  000010          MOV     #BIT05,RHCS2(R4) ;MASSBUS CLEAR
   (1) 011260  013764  003520  000010          MOV     @#DTADPB,RHCS2(R4) ;SELECT DRIVE
   (1) 011266  104420                          RESREG                  ;RESTORE R0-R5
 6800  011270  104017                          ERROR   17
 6801  011272  005005                  2$:     CLR     R5              ;SET THE UP/DOWN SWITCH TO UP
 6802  011274  004737  021606                  JSR     PC,@#COUNT      ;UP THE COUNT
 6803  011300  004737  021166                  JSR     PC,@#TWOMS      ;STALL FOR 2 MILLISECONDS
 6804  011304  005077  170032                  CLR     @PKB            ;START COUNT AT ZERO
 6805  011310  012764  000000  000034          MOV     #0,RHCA(R4)     ;CYLINDER=0
 6806  011316  012764  000105  000000          MOV     #SEEK,RHCS1(R4) ;START A SEEK
 6807  011324  012777  000131  170006          MOV     #131,@PKCS      ;START THE CLOCK
 6808  011332  000001                          WAIT                    ;WAIT ON INTERRUPT
 6809  011334  042777  000101  167776          BIC     #101,@PKCS      ;STOP THE CLOCK
 6810  011342  032764  040000  000012          BIT     #BIT14,RHDS1(R4) ;"ERR"=1?
 6811  011350  001415                          BEQ     3$              ;NO--BRANCH
 6812  011352  104416                          SAVREG                  ;SAVE R0-R5
   (1) 011354  012702  003520                  MOV     #DTADPB,R2      ;DPB POINTER
   (1) 011360  004737  033434                  JSR     PC,@#SVRH11     ;SAVE ALL THE RH11/RP04 REGISTERS
   (1) 011364  012764  000040  000010          MOV     #BIT05,RHCS2(R4) ;MASSBUS CLEAR
   (1) 011372  013764  003520  000010          MOV     @#DTADPB,RHCS2(R4) ;SELECT DRIVE
   (1) 011400  104420                          RESREG                  ;RESTORE R0-R5
 6813  011402  104017                          ERROR   17              ;REPORT THE ERROR
 6814  011404  012705  177777          3$:     MOV     #-1,R5          ;SET THE UP/DOWN SWITCH TO DOWN
 6815  011410  004737  021606                  JSR     PC,@#COUNT      ;UPDATE THE COUNT
 6816  011414  004737  021166                  JSR     PC,@#TWOMS      ;STALL FOR 2 MILLISECONDS
 6817  011420  005301                          DEC     R1              ;DONE?
 6818  011422  003263                          BGT     1$              ;NO--BRANCH
 6819  011424  000424                          BR      EXIT12          ;YES--EXIT
 6820  011426  042777  000101  167704  7$:     BIC     #101,@PKCS      ;STOP THE CLOCK
 6821  011434  005037  177776                  CLR     @#PS            ;DROP THE PRIORITY
 6822  011440  012600                          MOV     (SP)+,R0        ;PC OF WAIT+2
 6823  011442  005726                          TST     (SP)+           ;POP THE PS FROM THE STACK
 6824  011444  104416                          SAVREG                  ;SAVE R0-R5
   (1) 011446  012702  003520                  MOV     #DTADPB,R2      ;DPB POINTER
```

```
  (1)  011452  004737  033434                    JSR     PC,@#SVRH11      ;SAVE ALL THE RH11/RP04 REGISTERS
  (1)  011456  012764  000040  000010            MOV     #BIT05,RHCS2(R4) ;MASSBUS CLEAR
  (1)  011464  013764  003520  000010            MOV     @#DTADPB,RHCS2(R4) ;SELECT DRIVE
  (1)  011472  104420                            RESREG                   ;RESTORE R0-R5
6825   011474  104020                            ERROR   20               ;CLOCK OVERFLOWED
6826   011476  000004                    EXIT12: SCOPE
6827   011500  012764  000040  000010            MOV     #BIT05,RHCS2(R4) ;MASSBUS INIT.
  (1)  011506  013764  003520  000010            MOV     @#DTADPB,RHCS2(R4) ;SELECT DRIVE
6828   011514  004737  017360                    JSR     PC,@#ST.CLK      ;INITIALIZE THE CLOCK
6829   011520  012777  031356  015442            MOV     #ISR,@RPVEC      ;RESTORE RH11/RP04 INT. VECTOR
6830   011526  004037  021740                    JSR     R0,@#TYPTIM      ;GO TYPE THE TIMES
  (1)  011532  002370                            T12                      ;POINTER
```

```
6832
6833                                    .SBTTL  *** ADDRESSING TESTS (13-14) ***
6834

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

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

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

6842
6855                                    ;*****************************************************************
 (3)                                    ;*TEST 13       SECTOR ADDRESSING TEST
 (4)
 (4)                                    ;*      THIS TEST WRITES DATA INTO ALL SECTORS OF TRACK "FT".  THE
 (4)                                    ;*      DATA WILL BE 256 WORDS OF THE SECTOR ADDRESS OF THE SECTOR
 (4)                                    ;*      BEING WRITTEN.  A WRITE CHECK IS PERFORMED, THE BUFFER IS
 (4)                                    ;*      CLEARED (TO 177400) AND THE DATA IS READ AND COMPARED.  THEN SECTOR 0
 (4)                                    ;*      IS REWRITTEN AND SECTORS 0 - 21 ARE WRITE CHECKED.  THEN
 (4)                                    ;*      SECTOR 1 IS REWRITTEN AND SECTORS 0 - 21 ARE WRITE CHECKED.
 (4)                                    ;*      THIS REWRITE AND WRITE CHECK PROCEDURE IS CONTINUED UP THROUGH
 (4)                                    ;*      REWRITE SECTOR 21 AND WRITE CHECK SECTORS 0-21.
 (4)
 (3)                                    ;*****************************************************************
 (3)                                    ;
 (3)  011534                            TST13:
 (4)  011534  033737  001422  001224            BIT     @#BITS+<13*2>,TSTNMS ;DO THIS TEST?
 (4)  011542  001002                            BNE     64$             ;YES--BRANCH
 (4)  011544  000137  012034                    JMP     TST14           ;NO--GO TO THE NEXT TEST
 (4)  011550  013737  002254  001176  64$:     MOV     @#RPT13,$TIMES  ;GET THE ITERATION COUNT
 (3)  011556  012737  011614  001106          MOV     #TEST13,@#$LPADR
 (3)  011564  012737  011534  001110          MOV     #TST13,@#$LPERR ;SETUP THE ERROR LOOP ADDRESS
 (3)  011572  012737  000013  001102          MOV     #13,@#$TSTNM            ;SET UP TEST NUMBER AND
 (3)                                                                    ;CLEAR THE ERROR FLAG ($ERFLG)
 (3)  011600  013737  001102  177570          MOV     $TSTNM,@#DISPLAY        ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
6856  011606  012737  012032  001234          MOV     #EXIT13,@#BYPASS
6857  011614  012706  001100          TEST13: MOV     #STACK,SP       ;SET THE STACK POINTER
6858  011620  004737  022346                  JSR     PC,@#FILBUF     ;FILL THE BUFFER WITH SECTOR ADDRESSES
6859  011624  012737  000161  003522          MOV     #WRITE,@#DTADPB+2 ;COMMAND=WRITE DATA
6860  011632  013737  002256  003532          MOV     @#FC13,@#DTADPB+12 ;CYLINDER
6861  011640  113737  002260  003531          MOVB    @#FT13,@#DTADPB+11 ;TRACK
6862  011646  105037  003530                  CLRB    @#DTADPB+10     ;SECTOR
6863  011652  012737  165000  003524          MOV     #TRCKWC,@#DTADPB+4 ;WORD COUNT
6864  011660  012737  040524  003526          MOV     #BUFFER,@#DTADPB+6 ;BUFFER ADDRESS
6865  011666  004037  020320                  JSR     R0,@#DRVCAL     ;START A DATA TRANSFER
6866  011672  012737  000151  003522          MOV     #WRCKD,@#DTADPB+2 ;COMMAND=WRITE CHECK DATA
6867  011700  004037  020320                  JSR     R0,@#DRVCAL     ;START A DATA TRANSFER
6868  011704  004037  022410                  JSR     R0,@#CLRBUF     ;CLEAR BUFFER
6869  011710  012737  000171  003522          MOV     #READ,@#DTADPB+2 ;COMMAND = READ
```

```
 6870  011716  004037  020320                        JSR     R0,@#DRVCAL     ;START A DATA TRANSFER
 6871  011722  004037  022462                        JSR     R0,@#CKSCTR     ;CHECK THE SECTOR DATA READ
 6872  011726  012700  040524                        MOV     #BUFFER,R0      ;BUFFER ADDRESS
 6873  011732  005001                                CLR     R1              ;FIRST SECTOR
 6874  011734  012737  000161  003522  1$:           MOV     #WRITE,@#DTADPB+2 ;COMMAND=WRITE DATA
 6875  011742  012737  177400  003524                MOV     #SCTRWC,@#DTADPB+4 ;WORD COUNT
 6876  011750  010037  003526                        MOV     R0,@#DTADPB+6   ;BUFFER ADDRESS
 6877  011754  110137  003530                        MOVB    R1,@#DTADPB+10  ;SECTOR
 6878  011760  004037  020320                        JSR     R0,@#DRVCAL     ;START A DATA TRANSFER
 6879  011764  012737  000151  003522                MOV     #WRCKD,@#DTADPB+2 ;COMMAND=WRITE CHECK DATA
 6880  011772  012737  165000  003524                MOV     #TRCKWC,@#DTADPB+4 ;WORD COUNT
 6881  012000  012737  040524  003526                MOV     #BUFFER,@#DTADPB+6 ;BUFFER ADDRESS
 6882  012006  105037  003530                        CLRB    @#DTADPB+10     ;SECTOR
 6883  012012  004037  020320                        JSR     R0,@#DRVCAL     ;START A DATA TRANSFER
 6884  012016  062700  001000                        ADD     #^D256*2,R0     ;MOVE TO NEXT SECTOR
 6885  012022  005201                                INC     R1
 6886  012024  022701  000026                        CMP     #^D22,R1        ;DONE?
 6887  012030  003341                                BGT     1$              ;NO--BRANCH
 6888  012032  000004                        EXIT13: SCOPE                   ;LOOP
 6889
 6903                                   ;*****************************************************************
   (3)                                  ;*TEST 14       TRACK ADDRESSING TEST
   (4)
   (4)                                  ;*      THIS TEST WILL WRITE DATA IN THE FORM OF TRACK ADDRESSES
   (4)                                  ;*      IN CYLINDER "FC" SECTOR "FS" OF EVERY TRACK WITH EACH TRACK
   (4)                                  ;*      GETTING ITS OWN TRACK ADDRESS.
   (4)                                  ;*      A WRITE CHECK IS THEN PERFORMED ON EACH TRACK TO INSURE
   (4)                                  ;*      THE DATA IS VALID.  THEN TRACK 0 IS REWRITTEN AND TRACK 1
   (4)                                  ;*      THROUGH TRACK 18 IS WRITE CHECKED.  THEN TRACK 1 IS
   (4)                                  ;*      REWRITTEN AND TRACK 2 THROUGH TRACK 18 IS WRITE CHECKED.
   (4)                                  ;*      THIS PROCEDURE IS CONTINUED UP THROUGH REWRITING TRACK 17
   (4)                                  ;*      AND WRITE CHECKING TRACK 18.
   (4)
   (3)                                  ;*****************************************************************
   (3)  012034                          TST14:
   (4)  012034  033737  001424  001224          BIT     @#BITS+<14*2>,TSTNMS ;DO THIS TEST?
   (4)  012042  001002                          BNE     64$             ;YES--BRANCH
   (4)  012044  000137  012370                  JMP     TST15           ;NO--GO TO THE NEXT TEST
   (4)  012050  013737  002264  001176  64$:     MOV     @#RPT14,$TIMES  ;GET THE ITERATION COUNT
   (3)  012056  012737  012114  001106          MOV     #TEST14,@#$LPADR
   (3)  012064  012737  012034  001110          MOV     #TST14,@#$LPERR ;SETUP THE ERROR LOOP ADDRESS
   (3)  012072  012737  000014  001102          MOV     #14,@#$TSTNM            ;SET UP TEST NUMBER AND
   (3)                                                                  ;CLEAR THE ERROR FLAG ($ERFLG)
   (3)  012100  013737  001102  177570          MOV     $TSTNM,@#DISPLAY        ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
 6904  012106  012737  012366  001234          MOV     #EXIT14,@#BYPASS
 6905  012114  012706  001100                  TEST14: MOV     #STACK,SP       ;SET THE STACK POINTER
 6906  012120  004737  022346                  JSR     PC,@#FILBUF     ;FILL THE BUFFER WITH TRACK ADDRESS
 6907  012124  012737  000161  003522          MOV     #WRITE,@#DTADPB+2 ;COMMAND=WRITE DATA
 6908  012132  013737  002266  003532          MOV     @#FC14,@#DTADPB+12 ;CYLINDER
 6909  012140  113737  002270  003530          MOVB    @#FS14,@#DTADPB+10 ;SECTOR
 6910  012146  012737  177400  003524          MOV     #SCTRWC,@#DTADPB+4 ;WORD COUNT
 6911  012154  012737  040524  003526          MOV     #BUFFER,@#DTADPB+6 ;BUFFER ADDRESS
 6912  012162  005000                          CLR     R0              ;TRACK=0
 6913  012164  110037  003531          1$:     MOVB    R0,@#DTADPB+11  ;TRACK ADDRESS
```

```
6914  012170  004037  020320                    JSR     R0,@#DRVCAL     ;START A DATA TRANSFER
6915  012174  062737  001000  003526            ADD     #^D256*2,@#DTADPB+6 ;UPDATE BUFFER ADDRESS
6916  012202  005200                            INC     R0              ;UPDATE TRACK NUMBER
6917  012204  022700  000023                    CMP     #^D19,R0        ;OUT OF TRACKS?
6918  012210  003365                            BGT     1$              ;NO--BRANCH
6919  012212  012737  040524  003526            MOV     #BUFFER,@#DTADPB+6 ;BUFFER ADDRESS
6920  012220  005000                            CLR     R0
6921  012222  012737  000151  003522            MOV     #WRCKD,@#DTADPB+2 ;COMMAND=WRITE CHECK
6922  012230  110037  003531          2$:       MOVB    R0,@#DTADPB+11  ;TRACK ADDRESS
6923  012234  004037  020320                    JSR     R0,@#DRVCAL     ;START A DATA TRANSFER
6924  012240  062737  001000  003526            ADD     #^D256*2,@#DTADPB+6 ;UPDATE BUFFER ADDRESS
6925  012246  005200                            INC     R0              ;UPDATE TRACK NUMBER
6926  012250  022700  000023                    CMP     #^D19,R0        ;OUT OF TRACKS?
6927  012254  003365                            BGT     2$              ;NO--BRANCH
6928  012256  005000                            CLR     R0              ;FIRST TRACK ADDRESS
6929  012260  110037  003531          3$:       MOVB    R0,@#DTADPB+11  ;TRACK
6930  012264  010001                            MOV     R0,R1           ;FORM BUFFER ADDRESS
6931  012266  012737  040524  003526            MOV     #BUFFER,@#DTADPB+6 ;BUFFER ADDRESS
6932  012274  005301                    4$:     DEC     R1
6933  012276  002404                            BLT     5$
6934  012300  062737  001000  003526            ADD     #^D256*2,@#DTADPB+6
6935  012306  000772                            BR      4$
6936  012310  012737  000161  003522  5$:       MOV     #WRITE,@#DTADPB+2 ;COMMAND=WRITE DATA
6937  012316  004037  020320                    JSR     R0,@#DRVCAL     ;START A DATA TRANSFER
6938  012322  062737  001000  003526  6$:       ADD     #^D256*2,@#DTADPB+6 ;UPDATE BUFFER ADDRESS
6939  012330  105237  003531                    INCB    @#DTADPB+11     ;MOVE TO NEXT TRACK
6940  012334  012737  000151  003522            MOV     #WRCKD,@#DTADPB+2 ;COMMAND=WRITE CHECK DATA
6941  012342  004037  020320                    JSR     R0,@#DRVCAL     ;START A DATA TRANSFER
6942  012346  122737  000022  003531            CMPB    #^D18,@#DTADPB+11 ;OUT OF TRACKS?
6943  012354  003362                            BGT     6$              ;NO--BRANCH
6944  012356  005200                            INC     R0              ;NEXT TRACK TO WRITE
6945  012360  022700  000022                    CMP     #^D19,R0        ;OUT OF TRACKS?
6946  012364  003335                            BGT     3$              ;NO--BRANCH
6947  012366  000004                    EXIT14: SCOPE
```

```
                        .SBTTL  *** DATA TEST (15) ***

                        ;*********************************************************************
                        ;*TEST 15        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
```

```
(4)                                     ;*      133333  001000  176777  155555  177677  066667  177777  000000
(4)                                     ;*      165555  002000  175777  172666  177737  153333  177777  000000
(4)                                     ;*      133333  004000  173777  155555  177757  066667  177777  000000
(4)                                     ;*      165555  010000  167777  172666  177767  153333  177777  000000
(4)                                     ;*      133333  020000  157777  155555  177773  066667  177777  000000
(4)                                     ;*      165555  040000  137777  172666  177775  153333  177777  000000
(4)                                     ;*      133333  100000  077777  155555  177776  066667  177777  000000
(4)                                     ;*
(4)
(3)                                     ;*****************************************************************
(3)  012370                             TST15:
(4)  012370  033737  001426  001224             BIT     @#BITS+<15*2>,TSTNMS ;DO THIS TEST?
(4)  012376  001002                             BNE     64$             ;YES--BRANCH
(4)  012400  000137  013030                     JMP     TST16           ;NO--GO TO THE NEXT TEST
(4)  012404  013737  002274  001176  64$:      MOV     @#RPT15,$TIMES  ;GET THE ITERATION COUNT
(3)  012412  012737  012554  001106             MOV     #TEST15,@#$LPADR
(3)  012420  012737  012370  001110             MOV     #TST15,@#$LPERR ;SETUP THE ERROR LOOP ADDRESS
(3)  012426  012737  000015  001102             MOV     #15,@#$TSTNM            ;SET UP TEST NUMBER AND
(3)                                                                             ;CLEAR THE ERROR FLAG (SERFLG)
(3)  012434  013737  001102  177570             MOV     $TSTNM,@#DISPLAY        ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
7013  012442  005000                             CLR     R0              ;CLEAR SWITCH
7014  012444  005004                             CLR     R4              ;FORM WORD COUNT IN R4
7015  012446  013701  002314                     MOV     @#LS15,R1
7016  012452  163701  002312                     SUB     @#FS15,R1
7017  012456  002003                             BGE     1$              ;BRANCH IF FS < OR = LS
7018  012460  062701  000026                     ADD     #^D22,R1        ;MAKE DIFFERENCE POSITIVE
7019  012464  005100                             COM     R0              ;SET SWITCH
7020  012466  062704  000400             1$:     ADD     #^D256,R4
7021  012472  005301                             DEC     R1
7022  012474  002374                             BGE     1$
7023  012476  005404                             NEG     R4
7024  012500  010405                             MOV     R4,R5           ;COPY NORMAL WORD COUNT INTO SMALL WC
7025  012502  005700                             TST     R0              ;SWITCH SET?
7026  012504  001412                             BEQ     3$              ;NO--BRANCH
7027  012506  005005                             CLR     R5              ;FORM WORD COUNT FOR LS < FS
7028  012510  012701  000025                     MOV     #^D21,R1
7029  012514  163701  002312                     SUB     @#FS15,R1
7030  012520  062705  000400             2$:     ADD     #^D256,R5
7031  012524  005301                             DEC     R1
7032  012526  002374                             BGE     2$
7033  012530  005405                             NEG     R5
7034  012532  113737  002312  003530  3$:      MOVB    @#FS15,@#DTADPB+10 ;SECTOR
7035  012540  012737  040524  003526             MOV     #BUFFER,@#DTADPB+6 ;DATA BUFFER
7036  012546  012737  013026  001234             MOV     #EXIT15,@#BYPASS
7037  012554  005037  001316             TEST15: CLR     @#WCEFLG        ;CLEAR THE WRITE CHECK ERROR FLAG
7038  012560  013701  002276                     MOV     @#FC15,R1       ;PICKUP FIRST CYLINDER
7039  012564  000407                             BR      2$
7040  012566  005720                     1$:     TST     (R0)+           ;MOVE TO NEXT DATA PATTERN
7041  012570  022700  000040                     CMP     #^D16*2,R0      ;OUT OF PATTERNS?
7042  012574  003004                             BGT     3$              ;NO--BRANCH
7043  012576  004037  022316                     JSR     R0,@#INCCYL     ;MOVE TO NEXT CYLINDER
7044  012602  000511                             BR      EXIT15          ;OUT OF CYLINDERS
7045  012604  005000                     2$:     CLR     R0              ;START WITH PATTERN 0
7046  012606  036037  001374  002316  3$:      BIT     BITS(R0),@#PTRN15 ;THIS PATTERN SELECTED?
```

```
 7047  012614  001764                          BEQ     1$              ;NO--BRANCH
 7048  012616  013702  002304                  MOV     @#FT15,R2       ;FIRST TRACK
 7049  012622  010137  003532                  MOV     R1,@#DTADPB+12  ;CYLINDER
 7050  012626  110237  003531          4$:     MOVB    R2,@#DTADPB+11  ;TRACK
 7051  012632  010437  003524                  MOV     R4,@#DTADPB+4   ;WORD COUNT
 7052  012636  022701  000632                  CMP     #410.,R1        ;LAST DISK CYLINDER?
 7053  012642  003005                          BGT     5$              ;NO--BRANCH
 7054  012644  022702  000022                  CMP     #18.,R2         ;LAST DISK TRACK?
 7055  012650  003002                          BGT     5$              ;NO--BRANCH
 7056  012652  010537  003524                  MOV     R5,@#DTADPB+4   ;SHORT WORD COUNT
 7057  012656  013703  177570          5$:     MOV     @#SWR,R3        ;INHIBIT WRITE AND
 7058  012662  005103                          COM     R3              ;WRITE CHECK?
 7059  012664  032703  000030                  BIT     #SW04!SW03,R3
 7060  012670  001424                          BEQ     7$              ;YES--BRANCH
 7061  012672  004737  023036                  JSR     PC,@#SETBUF     ;MOVE DATA PATTERN INTO THE BUFFER
 7062  012676  032737  000020  177570          BIT     #SW04,@#SWR     ;INHIBIT WRITE?
 7063  012704  001005                          BNE     6$              ;YES--BRANCH
 7064  012706  012737  000161  003522          MOV     #WRITE,@#DTADPB+2 ;COMMAND=WRITE DATA
 7065  012714  004037  020320                  JSR     R0,@#DRVCAL     ;START A DATA TRANSFER
 7066  012720  032737  000010  177570  6$:     BIT     #SW03,@#SWR     ;INHIBIT WRITE CHECK?
 7067  012726  001005                          BNE     7$              ;YES--BRANCH
 7068  012730  012737  000151  003522          MOV     #WRCKD,@#DTADPB+2 ;COMMAND=WRITE CHECK DATA
 7069  012736  004037  020320                  JSR     R0,@#DRVCAL     ;START A DATA TRANSFER
 7070  012742  005737  001316          7$:     TST     @#WCEFLG        ;WRITE CHECK ERROR FLAG SET?
 7071  012746  001404                          BEQ     8$              ;NO--BRANCH
 7072  012750  032737  000001  177570          BIT     #SW00,@#SWR     ;PERFORM READ AFTER FATAL "WCE"?
 7073  012756  001417                          BEQ     9$              ;NO--BRANCH
 7074  012760  032737  000004  177570  8$:     BIT     #SW02,@#SWR     ;INHIBIT READ DATA AND SOFTWARE COMPARE?
 7075  012766  001013                          BNE     9$              ;YES--BRANCH
 7076  012770  012737  000171  003522          MOV     #READ,@#DTADPB+2 ;COMMAND=READ
 7077  012776  004037  020320                  JSR     R0,@#DRVCAL     ;START A DATA TRANSFER
 7078  013002  032737  000002  177570          BIT     #SW01,@#SWR     ;COMPARE THE DATA?
 7079  013010  001002                          BNE     9$              ;NO--BRANCH
 7080  013012  004737  023132                  JSR     PC,@#DATCMP     ;YES--DO IT
 7081  013016  004037  022266          9$:     JSR     R0,@#INCTRK     ;MOVE TO NEXT TRACK
 7082  013022  000661                          BR      1$              ;OUT OF TRACKS GO TO NEXT PATTERN
 7083  013024  000700                          BR      4$              ;LOOP
 7084  013026  000004                  EXIT15: SCOPE                   ;SCOPE LOOP
```

```
                                .SBTTL  *** EXERCISE TEST (16) ***

                                ;*********************************************************************
                                ;*TEST 16       RANDOM ADDRESS AND RANDOM PATTERN TEST (TST16)

                                ;*      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

                                ;*********************************************************************
013030                          TST16:
013030  033737  001430  001224          BIT     @#BITS+<16*2>,TSTNMS ;DO THIS TEST?
013036  001002                        BNE     64$             ;YES--BRANCH
013040  000137  013476              JMP     $EOP            ;NO--GO TO THE END OF THE PROGRAM
013044  013737  002322  001176  64$:   MOV     @#RPT16,$TIMES  ;GET THE ITERATION COUNT
013052  012737  013030  001106          MOV     #TST16,@#$LPADR ;SETUP THE SCOPE LOOP ADDRESS
013060  012737  013030  001110          MOV     #TST16,@#$LPERR ;SETUP THE ERROR LOOP ADDRESS
013066  012737  000016  001102          MOV     #16,@#$TSTNM            ;SET UP TEST NUMBER AND
                                                                        ;CLEAR THE ERROR FLAG (SERFLG)
013074  013737  001102  177570          MOV     $TSTNM,@#DISPLAY        ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
013102  012737  013474  001234          MOV     #EXIT16,@#BYPASS
013110  012737  176543  017312          MOV     #176543,@#$HINUM ;PRIME THE RANDOM NUMBER GENERATOR
013116  012737  123456  017314          MOV     #123456,@#$LONUM
013124  013737  002324  003532          MOV     @#FC16,@#DTADPB+12 ;CYLINDER
013132  012737  165000  003524          MOV     #TRCKWC,@#DTADPB+4 ;WORD COUNT
013140  012737  040524  003526          MOV     #BUFFER,@#DTADPB+6 ;BUFFER ADDRESS
013146  012737  000161  003522          MOV     #WRITE,@#DTADPB+2 ;COMMMAND
013154  032737  100000  001212          BIT     #SW15,@#C.SWR   ;WRITE THE DISK PACK BEFORE TESTING?
013162  001022                        BNE     3$              ;NO--BRANCH
013164  004037  023454                JSR     R0,@#FILRAN     ;FILL DATA BUFFER WITH RANDOM DATA
013170  005037  003530          1$:   CLR     @#DTADPB+10     ;SECTOR AND TRACK
013174                          2$:
013174  004037  020320                JSR     R0,@#DRVCAL     ;START A DATA TRANSFER
013200  105237  003531                INCB    @#DTADPB+11     ;NEXT TRACK
013204  122737  000023  003531        CMPB    #^D19,@#DTADPB+11 ;TIME FOR NEXT CYLINDER
013212  003370                        BGT     2$              ;NO--BRANCH
013214  005237  003532                INC     @#DTADPB+12
```

```
7129  013220  023737  002326  003532          CMP     @#LC16,@#DTADPB+12 ;OUT OF CYLINDERS?
7130  013226  002360                          BGE     I$                 ;NO--BRANCH
7131  013230  012737  177400  003524  3$:     MOV     #SCTRWC,@#DTADPB+4 ;WORD COUNT
7132  013236  012737  013252  001106          MOV     #TEST16,@#$LPADR
7133  013244  012737  013252  001110          MOV     #TEST16,@#$LPERR
7134  013252  012706  001100          TEST16: MOV     #STACK,SP          ;SET STACK POINTER
7135  013256  004037  023730                  JSR     R0,@#RANADR        ;GENERATE A RANDOM ADDRESS
7136  013262  013737  003530  001322          MOV     @#DTADPB+10,@#SVADR ;SAVE THE TRACK/SECTOR
7137  013270  013737  003532  001324          MOV     @#DTADPB+12,@#SVADR+2 ;SAVE THE CYLINDER
7138  013276  012737  000161  003522          MOV     #WRITE,@#DTADPB+2  ;COMMAND=WRITE DATA
7139  013304  012701  040524                  MOV     #BUFFER,R1         ;BUFFER ADDRESS
7140  013310  010137  003526                  MOV     R1,@#DTADPB+6
7141  013314  004037  023674                  JSR     R0,@#RANPAT        ;GENERATE RANDOM PATTERN
7142  013320  004037  020320                  JSR     R0,@#DRVCAL        ;START A DATA TRANSFER
7143  013324  004037  023730                  JSR     R0,@#RANADR
7144  013330  012737  000171  003522          MOV     #READ,@#DTADPB+2   ;COMMAND=READ DATA
7145  013336  012737  041524  003526          MOV     #BUFFER+512.,@#DTADPB+6 ;BUFFER ADDRESS
7146  013344  004037  020320                  JSR     R0,@#DRVCAL        ;START A DATA TRANSFER
7147  013350  004037  023476                  JSR     R0,@#RANCK         ;CHECK THE DATA
7148  013354  013737  001322  003530          MOV     @#SVADR,@#DTADPB+10 ;GET ADDRESS OF WHERE THE LAST
7149  013362  013737  001324  003532          MOV     @#SVADR+2,@#DTADPB+12 ;WRITE WAS PERFORMED
7150  013370  012737  000151  003522          MOV     #WRCKD,@#DTADPB+2  ;COMMAND=WRITE CHECK DATA
7151  013376  012737  040524  003526          MOV     #BUFFER,@#DTADPB+6 ;DATA BUFFER ADDRESS
7152  013404  004037  020320                  JSR     R0,@#DRVCAL        ;START A DATA TRANSFER
7153  013410  004037  023730                  JSR     R0,@#RANADR        ;GENERATE A RANDOM ADDRESS
7154  013414  012737  000171  003522          MOV     #READ,@#DTADPB+2   ;COMMAND=READ
7155  013422  012737  041524  003526          MOV     #BUFFER+512.,@#DTADPB+6 ;DATA BUFFER ADDRESS
7156  013430  004037  020320                  JSR     R0,@#DRVCAL        ;START A DATA TRANSFER
7157  013434  004037  023476                  JSR     R0,@#RANCK         ;CHECK THE DATA
7158  013440  013737  001322  003530          MOV     @#SVADR,@#DTADPB+10 ;GET DISK ADDRESS OF THE
7159  013446  013737  001324  003532          MOV     @#SVADR+2,@#DTADPB+12 ;LAST WRITE
7160  013454  012737  000151  003522          MOV     #WRCKD,@#DTADPB+2  ;COMMAND=WRITE CHECK DATA
7161  013462  012737  040524  003526          MOV     #BUFFER,@#DTADPB+6 ;DATA BUFFER ADDRESS
7162  013470  004037  020320                  JSR     R0,@#DRVCAL        ;START A DATA TRANSFER
7163  013474  000004                  EXIT16: SCOPE                      ;LOOP
```

```
7165                                    ;*****************************************************************
 (1)
 (1)                                    .SBTTL  END OF PASS ROUTINE
 (1)
 (1)                                    ;*INCREMENT THE PASS NUMBER ($PASS)
 (1)                                    ;*INDICATE END-OF-PROGRAM AFTER 8. PASSES THRU THE PROGRAM
 (1)                                    ;*IF THERES A MONITOR GO TO IT
 (1)                                    ;*IF THERE ISN'T JUMP TO RESTART
 (1)
 (1)  013476                            $EOP:
 (4)  013476  104400  013504                    TYPE    .+4             ;;TYPE ASCIZ STRING
 (4)  013502  000410                            BR      64$             ;;GET OVER THE ASCIZ
 (4)                                            ;;.ASCIZ        <15><12><12>/END OF PASS/
 (4)  013524                            64$:
 (3)  013524  005737  001222                    TST     @#DRVSEL        ;ANY DRIVES SELECTED?
 (3)  013530  001434                            BEQ     1$              ;NO--BRANCH
 (4)  013532  104400  013540                    TYPE    .+4             ;;TYPE ASCIZ STRING
 (4)  013536  000405                            BR      65$             ;;GET OVER THE ASCIZ
 (4)                                            ;;.ASCIZ        / ON DRIVE/
 (4)  013552                            65$:
 (4)  013552  013746  001236                    MOV     @#CHKDRV,-(SP)  ;;SAVE @#CHKDRV FOR TYPEOUT
 (4)  013556  104404                            TYPOS                   ;;GO TYPE--OCTAL ASCII
 (4)  013560     002                            .BYTE   2               ;;TYPE 2 DIGIT(S)
 (4)  013561     000                            .BYTE   0               ;;SUPPRESS LEADING ZEROS
 (4)  013562  104400  013570                    TYPE    .+4             ;;TYPE ASCIZ STRING
 (4)  013566  000412                            BR      66$             ;;GET OVER THE ASCIZ
 (4)                                            ;;.ASCIZ        /   ERRORS DETECTED=/
 (4)  013614                            66$:
 (4)  013614  013746  001112                    MOV     @#SERTTL,-(SP)  ;;SAVE @#SERTTL FOR TYPEOUT
 (4)  013620  104402                            TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 (3)  013622  005037  001112            1$:     CLR     @#SERTTL        ;ZERO ERROR TOTAL
 (1)  013626  005037  001102                    CLR     $TSTNM          ;;ZERO THE TEST NUMBER
 (1)  013632  005037  001176                    CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
 (1)  013636  005237  001100                    INC     $PASS           ;;INCREMENT THE PASS NUMBER
 (1)  013642  042737  100000  001100            BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
 (1)  013650  005327                            DEC     (PC)+           ;;LOOP?
 (1)  013652  000010                    $EOPCT: .WORD   8.
 (1)  013654  003036                            BGT     $DOAGN          ;;YES
 (1)  013656  012737                            MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
 (1)  013660  000010                    $ENDCT: .WORD   8.
 (1)  013662  013652                            $EOPCT
 (3)  013664  104400  013672                    TYPE    .+4             ;;TYPE ASCIZ STRING
 (3)  013670  000407                            BR      64$             ;;GET OVER THE ASCIZ
 (3)                                            ;;.ASCIZ        <15><12>/END OF TEST/
 (3)  013710                            64$:
 (2)  013710  104400  013756                    TYPE    ,$ENULL         ;TYPE NULL CHARACTER
 (1)  013714  013700  000042            $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
 (1)  013720  001414                            BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
 (1)  013722  022700  013742                    CMP     #$ENDAD,R0      ;;IS MONITOR ACT11?
 (1)  013726  001005                            BNE     $ENDAD          ;;NO--BRANCH
 (1)  013730  022760  177777  000002            CMP     #-1,2(R0)       ;;YES--IS THIS THE LAST PASS?
 (1)  013736  001005                            BNE     $DOAGN          ;;NO--MAKE ANOTHER PASS
 (1)  013740  000005                            RESET                   ;;CLEAR THE WORLD
 (1)  013742  004710                    $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
```

```
(1)  013744  000240                          NOP                        ;;SAVE ROOM
(1)  013746  000240                          NOP                        ;;FOR
(1)  013750  000240                          NOP                        ;;ACT11
(1)  013752  000137  005114          $DOAGN: JMP        @#RESTART                ;;RETURN
(1)  013756     377     377     000  $ENULL: .BYTE      -1,-1,0          ;;NULL CHARACTER STRING
(1)          013762                       .EVEN
```

```
                                        .SBTTL  *** 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

013762                                  $ERROR:
013762  032737  000400  177570          BIT     #SW08,@#SWR     ;SEND ERROR MESSAGE TO TTY?
013770  001411                          BEQ     7$              ;YES--BRANCH
013772  005737  001220                  TST     @#LPTAVL        ;IS THERE A LINE PRINTER AVAILABLE?
013776  001406                          BEQ     7$              ;NO--BRANCH
014000  013737  001360  001142          MOV     @#LPS,@#$TPS    ;YES--SETUP STATUS
014006  013737  001362  001144          MOV     @#LPB,@#$TPB    ;AND BUFFER REG.'S FOR LINE PRINTER
014014  105237  001103          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
014020  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
014022  013737  001102  177570          MOV     $TSTNM,@#DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
014030  032737  002000  177570          BIT     #BIT10,@#SWR    ;;BELL ON ERROR?
014036  001402                          BEQ     1$              ;;NO - SKIP
014040  104400  001202                  TYPE    ,$BELL          ;;RING BELL
014044  005237  001112          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
014050  011637  001116                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
014054  162737  000002  001116          SUB     #2,$ERRPC
014062  117737  165030  001114          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
014070  032737  020000  177570          BIT     #BIT13,@#SWR    ;;SKIP TYPEOUT IF SET
014076  001004                          BNE     2$              ;;SKIP TYPEOUTS
014100  004737  014164                  JSR     PC,@#TYPERR     ;;GO TO USER ERROR ROUTINE
014104  104400  001207                  TYPE    ,$CRLF
014110  005737  177570          2$:     TST     @#SWR           ;;HALT ON ERROR
014114  100001                          BPL     3$              ;;SKIP IF CONTINUE
014116  000000                          HALT                    ;;HALT ON ERROR!
014120  032737  001000  177570  3$:     BIT     #BIT09,@#SWR    ;;LOOP ON ERROR SWITCH SET?
014126  001402                          BEQ     4$              ;;BR IF NO
014130  013716  001110                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
014134  005737  001200          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
014140  001402                          BEQ     5$              ;;BR IF NONE
014142  013716  001200                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
014146                          5$:
014146  013737  001354  001142          MOV     @#TPS,@#$TPS    ;SET STATUS AND BUFFER REG.'S
014154  013737  001356  001144          MOV     @#TPB,@#$TPB    ;FOR TTY
014162  000002                          RTI                     ;RETURN FROM ERROR CALL

                                        ;*****************************************************************
                                        .SBTTL  TYPERR - TYPE ERROR ROUTINE
```

```
                                        ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE
                                        ;*WHICH ERROR IS TO BE REPORTED, IT THEN OBTAINS, FROM THE "ERROR
                                        ;*TABLE" ($ERRTB), AND REPORTS THE APPROPIATE INFORMATION
                                        ;*CONCERNING THE ERROR.
                                        ;*CALL
                                        ;*      JSR     PC,@#TYPERR
                                        ;*      RETURN

014164  113737  001102  001170  TYPERR: MOVB    @#$TSTNM,@#$TMPO ;SAVE THE TEST NUMBER
014172  004037  034334                  JSR     R0,@#SAVR05     ;SAVE R0-R5
014176  162700  000004                  SUB     #4,R0           ;FORM TEST PC
014202  010037  001154                  MOV     R0,@#$REG0      ;COPY R0-R5 IN $REG0-$REG5
014206  010137  001156                  MOV     R1,@#$REG1
014212  010237  001160                  MOV     R2,@#$REG2
014216  010337  001162                  MOV     R3,@#$REG3
014222  010437  001164                  MOV     R4,@#$REG4
014226  010537  001166                  MOV     R5,@#$REG5
014232  113700  001114                  MOVB    @#$ITEMB,R0     ;PICKUP ERROR ITEM NUMBER
014236  010001                          MOV     R0,R1           ;AND COPY IT INTO R1
014240  005300                          DEC     R0              ;FORM INDEX FOR ERROR TABLE
014242  106300                          ASLB    R0
014244  106300                          ASLB    R0
014246  106300                          ASLB    R0
014250  103002                          BCC     1$              ;IS ERROR > 37?
014252  062700  000220                  ADD     #ITEM41-$ERRTB,R0 ;YES--FORM OFFSET
014256  062700  003610          1$:     ADD     #$ERRTB,R0      ;FORM ADDRESS
014262  012037  014276                  MOV     (R0)+,2$        ;GET ERROR MESSAGE (EM) POINTER
014266  001447                          BEQ     7$              ;BRANCH IF THERE ISN'T ONE
014270  104400  001207                  TYPE    ,$CRLF          ;"CARRIAGE RETURN - LINE FEED
014274  104400                          TYPE
014276  000000          2$:     .WORD   0                       ;"EM" POINTER GOES HERE
014300  162701  000041                  SUB     #41,R1          ;SPECIAL ERROR ITEM NUMBER?
014304  100440                          BMI     7$              ;NO--BRANCH
014306  013701  001242                  MOV     @#SVSTAT,R1     ;GET STATUS/ERROR INDICATOR
014312  106301                          ASLB    R1              ;STRIP "DONE" BIT (BIT07)
014314  006301                          ASL     R1              ;STRIP "ERROR" BIT (BIT15)
014316  012702  001566                  MOV     #STATBL,R2      ;1ST ADDRESS ON STATUS MESSAGE POINTERS
014322  005003                          CLR     R3              ;CARRIAGE RETURN-LINE FEED SWITCH
014324  104400  014332                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
014330  000402                          BR      64$             ;;GET OVER THE ASCIZ
                                        ;;.ASCIZ      / (/
014336                          64$:
014336  012237  014360          3$:     MOV     (R2)+,5$        ;MESSAGE POINTER
014342  006301                          ASL     R1              ;TYPE THIS MESSAGE?
014344  103013                          BCC     6$              ;NO--BRANCH
014346  005103                          COM     R3              ;YES--TYPE A "CR" & "LF"?
014350  001002                          BNE     4$              ;NO--BRANCH
014352  104400  001207                  TYPE    ,$CRLF          ;YES
014356  104400                  4$:     TYPE
014360  000000          5$:     .WORD   0                       ;MESSAGE POINTER GOES HERE
014362  005701                          TST     R1              ;MORE TO TYPE?
014364  001403                          BEQ     6$              ;NO--BRANCH
014366  104400  035202                  TYPE    ,MSG.SP         ;YES--SPACES
014372  000761                          BR      3$              ;LOOP
```

```
 7224  014374  001360                6$:   BNE     3$              ;BRANCH IF NOT FINISHED
 7225  014376  104400  014404              TYPE    .+4             ;;TYPE ASCIZ STRING
  (1)  014402  000401                      BR      65$             ;;GET OVER THE ASCIZ
  (1)                                      ;;.ASCIZ      /)/
  (1)  014406                       65$:
 7226  014406  012037  014422        7$:   MOV     (R0)+,8$        ;PICK UP DATA HEADER (DH) POINTER
 7227  014412  001404                      BEQ     9$              ;BRANCH IF NONE
 7228  014414  104400  001207              TYPE    ,$CRLF          ;CARRIAGE RETURN-LINE FEED
 7229  014420  104400                      TYPE
 7230  014422  000000                8$:   .WORD   0               ;"DH" POINTER GOES HERE
 7231  014424  012001                9$:   MOV     (R0)+,R1        ;PICKUP DATA TABLE (DT) POINTER
 7232  014426  001450                      BEQ     20$             ;BRANCH IF NONE
 7233  014430  005005                      CLR     R5              ;SET INDENT SWITCH
 7234  014432  012000                      MOV     (R0)+,R0        ;DATA FORMAT (DF) POINTER
 7235  014434  012002                      MOV     (R0)+,R2        ;NUMBER OF DH'S TO TYPE
 7236  014436  001441                      BEQ     17$             ;BRANCH IF DH NUMBER IS 0
 7237  014440  005105                      COM     R5              ;NO INDENT
 7238  014442  104400  001207              TYPE    ,$CRLF          ;CARRIAGE RETURN-LINE FEED
 7239  014446  112003                10$:  MOVB    (R0)+,R3        ;NUMBER OF DATA WORDS TO TYPE
 7240  014450  112004                      MOVB    (R0)+,R4        ;AND HOW TO TYPE THEM
 7241  014452  006004                11$:  ROR     R4              ;OCTAL OR DECIMAL?
 7242  014454  103403                      BCS     12$             ;DECIMAL--BRANCH
 7243  014456  013146                      MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
  (1)  014460  104402                      TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 7244  014462  000402                      BR      13$
 7245  014464                       12$:
  (1)  014464  013146                      MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
  (1)  014466  104410                      TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
 7246  014470  005303                13$:  DEC     R3              ;MORE NUMBERS TO TYPE?
 7247  014472  001403                      BEQ     14$             ;NO--BRANCH
 7248  014474  104400  035202              TYPE    ,MSG.SP         ;YES--TYPE SEPERATORS
 7249  014500  000764                      BR      11$             ;LOOP
 7250  014502  005302                14$:  DEC     R2              ;MORE DH'S?
 7251  014504  003421                      BLE     20$             ;NO--BRANCH
 7252  014506  104400  001207              TYPE    ,$CRLF          ;YES--START A NEW LINE
 7253  014512  005105                      COM     R5              ;INDENT?
 7254  014514  001002                      BNE     15$             ;NO--BRANCH
 7255  014516  104400  035202              TYPE    ,MSG.SP         ;YES--TYPE SPACES
 7256  014522  012037  014530        15$:  MOV     (R0)+,16$       ;GET NEXT DH
 7257  014526  104400                      TYPE                    ;AND TYPE IT
 7258  014530  000000                16$:  .WORD   0               ;DH POINTER GOES HERE
 7259  014532  104400  001207              TYPE    ,$CRLF          ;CARRIAGE RETURN-LINE FEED
 7260  014536  005705                      TST     R5              ;INDENT?
 7261  014540  001342                      BNE     10$             ;NO--BRANCH
 7262  014542  104400  035202        17$:  TYPE    ,MSG.SP         ;YES--TYPE SPACES
 7263  014546  000737                      BR      10$             ;LOOP
 7264  014550  004037  034360        20$:  JSR     R0,@#GETR05     ;RESTORE R0-R5
 7265  014554  000207                      RTS     PC              ;RETURN
 7266
 7267
 7268
 7269                                ;******************************************************************
  (1)
  (1)                                .SBTTL  TYPE ROUTINE
```

```
(1)
(1)                                     ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
(1)                                     ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
(1)                                     ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
(1)                                     ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
(1)                                     ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
(1)                                     ;*
(1)                                     ;*CALL:
(1)                                     ;*1) USING A TRAP INSTRUCTION
(1)                                     ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
(1)                                     ;*OR
(1)                                     ;*      TYPE
(1)                                     ;*      MESADR
(1)                                     ;*
(1)                                     ;*2) USING A JSR INSTRUCTION
(1)                                     ;*      MOV     PS,-(SP)        ;;PUSH PROCESSOR STATUS WORD ON THE STACK
(1)                                     ;*      JSR     PC,$TYPE        ;;CALL TYPE ROUTINE
(1)                                     ;*      MESADDR                 ;;FIRST ADRESS OF MESSAGE
(1)
(1)  014556  105737  001151             $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
(1)  014562  100002                             BPL     1$              ;;BR IF YES
(1)  014564  000000                             HALT                    ;;HALT HERE IF NO TERMINAL
(1)  014566  000407                             BR      3$              ;;LEAVE
(1)  014570  010046                     1$:     MOV     R0,-(SP)        ;;SAVE R0
(1)  014572  017600  000002                     MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
(1)  014576  112046                     2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
(1)  014600  001005                             BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
(1)  014602  005726                             TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
(1)  014604  012600                             MOV     (SP)+,R0        ;;RESTORE R0
(1)  014606  062716  000002             3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
(1)  014612  000002                             RTI                     ;;RETURN
(1)  014614  004737  014646             4$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
(1)  014620  123726  001150             5$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
(1)  014624  001364                             BNE     2$              ;;IF NO GO GET NEXT CHAR.
(1)  014626  013746  001146                     MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
(1)                                                                     ;;AND THE NULL CHAR.
(1)  014632  105366  000001             6$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
(1)  014636  002770                             BLT     5$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
(1)  014640  004737  014646                     JSR     PC,$TYPEC       ;;GO TYPE A NULL
(1)  014644  000772                             BR      6$              ;;LOOP
(1)  014646  105777  164270             $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
(1)  014652  100375                             BPL     $TYPEC
(1)  014654  116677  000002  164262             MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
(1)  014662  000207                             RTS     PC
7270                                    ;******************************************************************
(1)
(1)                                     .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
(1)
(1)                                     ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
(1)                                     ;*OCTAL (ASCII) NUMBER AND TYPE IT.
(1)                                     ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
(1)                                     ;*CALL:
(1)                                     ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
(1)                                     ;*      TYPOS                   ;;CALL FOR TYPEOUT
```

```
(1)                                     ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
(1)                                     ;*      .BYTE   M               ;;M=1 OR 0
(1)                                     ;*                              ;;1=TYPE LEADING ZEROS
(1)                                     ;*                              ;;0=SUPPRESS LEADING ZEROS
(1)                                     ;*
(1)                                     ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
(1)                                     ;*$TYPOS OR $TYPOC
(1)                                     ;*CALL:
(1)                                     ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
(1)                                     ;*      TYPON                   ;;CALL FOR TYPEOUT
(1)                                     ;*
(1)                                     ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
(1)                                     ;*CALL:
(1)                                     ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
(1)                                     ;*      TYPOC                   ;;CALL FOR TYPEOUT
(1)
(1)  014664  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
(1)  014670  116637  000001  015107          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
(1)  014676  112637  015111                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
(1)  014702  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
(1)  014706  000406                          BR      $TYPON
(1)  014710  112737  000001  015107  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
(1)  014716  112737  000006  015111          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
(1)  014724  112737  000005  015106  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
(1)  014732  010346                          MOV     R3,-(SP)        ;;SAVE R3
(1)  014734  010446                          MOV     R4,-(SP)        ;;SAVE R4
(1)  014736  010546                          MOV     R5,-(SP)        ;;SAVE R5
(1)  014740  113704  015111                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
(1)  014744  005404                          NEG     R4
(1)  014746  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
(1)  014752  110437  015110                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
(1)  014756  113704  015107                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
(1)  014762  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
(1)  014766  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
(1)  014770  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
(1)  014772  000404                          BR      3$              ;;GO DO MSB
(1)  014774  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
(1)  014776  006105                          ROL     R5
(1)  015000  006105                          ROL     R5
(1)  015002  010503                          MOV     R5,R3
(1)  015004  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
(1)  015006  105337  015110                  DECB    $OMODE          ;;TYPE THIS DIGIT?
(1)  015012  100016                          BPL     7$              ;;BR IF NO
(1)  015014  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
(1)  015020  001002                          BNE     4$              ;;TEST FOR 0
(1)  015022  005704                          TST     R4              ;;SUPPRESS THIS 0?
(1)  015024  001403                          BEQ     5$              ;;BR IF YES
(1)  015026  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
(1)  015030  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
(1)  015034  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
(1)  015040  110337  015104                  MOVB    R3,8$           ;;SAVE FOR TYPING
(1)  015044  104400  015104                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
(1)  015050  105337  015106          7$:     DECB    $OCNT           ;;COUNT BY 1
(1)  015054  003347                          BGT     2$              ;;BR IF MORE TO DO
```

```
(1)  015056  002402                        BLT     6$              ;;BR IF DONE
(1)  015060  005204                        INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
(1)  015062  000744                        BR      2$              ;;GO DO THE LAST DIGIT
(1)  015064  012605                6$:     MOV     (SP)+,R5        ;;RESTORE R5
(1)  015066  012604                        MOV     (SP)+,R4        ;;RESTORE R4
(1)  015070  012603                        MOV     (SP)+,R3        ;;RESTORE R3
(1)  015072  016666  000002  000004        MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
(1)  015100  012616                        MOV     (SP)+,(SP)
(1)  015102  000002                        RTI                     ;;RETURN
(1)  015104     000                8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
(1)  015105     000                        .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
(1)  015106     000                $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
(1)  015107     000                $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
(1)  015110  000000                $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
7271                                ;*****************************************************************
(1)                                 .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
(1)
(1)                                 ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
(1)                                 ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
(1)                                 ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
(1)                                 ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
(1)                                 ;*REPLACED WITH SPACES.
(1)                                 ;*CALL:
(1)                                 ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
(1)                                 ;*      TYPDS                   ;;GO TO THE ROUTINE
(1)
(2)  015112                         $TYPDS:
(3)  015112  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
(3)  015114  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
(3)  015116  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
(3)  015120  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
(3)  015122  010546                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
(1)  015124  012746  020200                MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
(1)  015130  016605  000020                MOV     20(SP),R5       ;;GET THE INPUT NUMBER
(1)  015134  100004                        BPL     1$              ;;BR IF INPUT IS POS.
(1)  015136  005405                        NEG     R5              ;;MAKE THE BINARY NUMBER POS.
(1)  015140  112766  000055  000001        MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
(1)  015146  005000                1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
(1)  015150  012703  015326                MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
(1)  015154  112723  000040                MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
(1)  015160  005002                2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
(1)  015162  016001  015316                MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
(1)  015166  160105                3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
(1)  015170  002402                        BLT     4$              ;;BR IF DONE
(1)  015172  005202                        INC     R2              ;;INCREASE THE BCD DIGIT BY 1
(1)  015174  000774                        BR      3$
(1)  015176  060105                4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
(1)  015200  005702                        TST     R2              ;;CHECK IF BCD DIGIT=0
(1)  015202  001002                        BNE     5$              ;;FALL THROUGH IF 0
(1)  015204  105716                        TSTB    (SP)            ;;STILL DOING LEADING 0'S?
(1)  015206  100407                        BMI     7$              ;;BR IF YES
(1)  015210  106316                5$:     ASLB    (SP)            ;;MSD?
(1)  015212  103003                        BCC     6$              ;;BR IF NO
```

```
(1)  015214  116663  000001  177777          MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
(1)  015222  052702  000060          6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
(1)  015226  052702  000040          7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
(1)  015232  110223                          MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
(1)  015234  005720                          TST     (R0)+           ;;JUST INCREMENTING
(1)  015236  020027  000010                  CMP     R0,#10          ;;CHECK THE TABLE INDEX
(1)  015242  002746                          BLT     2$              ;;GO DO THE NEXT DIGIT
(1)  015244  003002                          BGT     8$              ;;GO TO EXIT
(1)  015246  010502                          MOV     R5,R2           ;;GET THE LSD
(1)  015250  000764                          BR      6$              ;;GO CHANGE TO ASCII
(1)  015252  105726                  8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
(1)  015254  100003                          BPL     9$              ;;BR IF NO
(1)  015256  116663  177777  177776          MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
(1)  015264  105013                  9$:     CLRB    (R3)            ;;SET THE TERMINATOR
(3)  015266  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
(3)  015270  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
(3)  015272  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
(3)  015274  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
(3)  015276  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
(1)  015300  104400  015326                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
(1)  015304  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
(1)  015312  012616                          MOV     (SP)+,(SP)
(1)  015314  000002                          RTI                     ;;RETURN TO USER
(1)  015316  023420                  $DTBL:  10000.
(1)  015320  001750                          1000.
(1)  015322  000144                          100.
(1)  015324  000012                          10.
(1)  015326  000004                  $DBLK:  .BLKW   4
7272                                 ;******************************************************************
(1)                                  .SBTTL  TTY INPUT ROUTINE
(1)
(1)  015336  000000                  $TKCNT: .WORD   0               ;;NUMBER OF ITEMS IN QUEUE
(1)  015340  000000                  $TKQIN: .WORD   0               ;;INPUT POINTER
(1)  015342  000000                  $TKQOUT: .WORD  0               ;;OUTPUT POINTER
(1)  015344  000002                  $TKQSRT: .BLKB  2               ;;TTY KEYBOARD QUEUE
(1)          015346                  $TKQEND=.
(1)
(1)                                  ;*TK INITIALIZE ROUTINE
(1)                                  ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
(1)                                  ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
(1)                                  ;
(1)                                  ;*CALL
(1)                                  ;*      JSR     PC,$TKINT
(1)                                  ;*      RETURN
(1)                                  ;
(1)  015346  005037  015336  $TKINT: CLR     $TKCNT          ;;CLEAR COUNT OF ITEMS IN QUEUE
(1)  015352  012737  015344  015340          MOV     #$TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
(1)  015360  013737  015340  015342          MOV     $TKQIN,$TKQOUT  ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
(1)  015366  012737  015416  000060          MOV     #$TKSRV,@#TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
(1)  015374  012737  000200  000062          MOV     #200,@#TKVEC+2  ;;"BR" LEVEL 4
(1)  015402  005777  163532                  TST     @$TKB           ;;CLEAR DONE FLAG
(1)  015406  012777  000100  163522          MOV     #BIT06,@$TKS    ;;ENABLE INTERRUPT
(1)  015414  000207                          RTS     PC              ;;RETURN TO CALLER
```

```
(1)
(1)                                     ;*TK SERVICE ROUTINE
(1)                                     ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
(1)                                     ;
(1)  015416  117746  163516             $TKSRV: MOVB    @$TKB,-(SP)     ;;PICKUP THE CHARACTER
(1)  015422  042716  177600                     BIC     #^C177,(SP)     ;;STRIP THE JUNK
(1)  015426  021627  000003                     CMP     (SP),#3         ;;IS IT A CONTROL C?
(1)  015432  001006                             BNE     1$              ;;BRANCH IF NO
(1)  015434  104400  016075                     TYPE    ,$CNTLC         ;;TYPE A CONTROL-C (^C)
(1)  015440  004737  015346                     JSR     PC,$TKINT       ;;INIT THE KEYBOARD
(1)  015444  000137  004210                     JMP     START2          ;;CONTROL C RESTART
(1)  015450  022737  000002  015336     1$:    CMP     #2,$TKCNT       ;;IS THE QUEUE FULL?
(1)  015456  001004                             BNE     2$              ;;BRANCH IF NO
(1)  015460  104400  001202                     TYPE    ,$BELL          ;;RING THE TTY BELL
(1)  015464  005726                             TST     (SP)+           ;;CLEAN CHARACTER OFF OF STACK
(1)  015466  000002                             RTI                     ;;RETURN
(1)  015470  005237  015336             2$:    INC     $TKCNT          ;;COUNT THIS CHARACTER
(1)  015474  112677  177640                     MOVB    (SP)+,@$TKQIN   ;;AND PUT IT IN QUEUE
(1)  015500  005237  015340                     INC     $TKQIN          ;;UPDATE THE POINTER
(1)  015504  023727  015340  015346             CMP     $TKQIN,#$TKQEND ;;GO OFF THE END?
(1)  015512  001003                             BNE     3$              ;;BRANCH IF NO
(1)  015514  012737  015344  015340             MOV     #$TKQSRT,$TKQIN ;;RESET THE POINTER
(1)  015522  000002                     3$:    RTI                     ;;RETURN
(2)                                     ;*****************************************************************
(1)                                     ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
(1)                                     ;*CALL:
(1)                                     ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
(1)                                     ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
(1)                                     ;
(1)
(1)  015524  011646                     $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC AND
(1)  015526  016666  000004  000002             MOV     4(SP),2(SP)     ;;THE PS
(1)  015534  005066  000004                     CLR     4(SP)           ;;GET READY FOR A CHARACTER
(1)  015540  005037  177776                     CLR     @#PS            ;;ALLOW INTERRUPTS
(1)  015544  005737  015336             1$:    TST     $TKCNT          ;;WAIT ON A CHARACTER
(1)  015550  001775                             BEQ     1$
(1)  015552  005337  015336                     DEC     $TKCNT          ;;DECREMENT THE COUNTER
(1)  015556  117766  177560  000004             MOVB    @$TKQOUT,4(SP)  ;;GET ONE CHARACTER
(1)  015564  005237  015342                     INC     $TKQOUT         ;;UPDATE THE POINTER
(1)  015570  023727  015342  015346             CMP     $TKQOUT,#$TKQEND ;;DID IT GO OFF OF THE END?
(1)  015576  001003                             BNE     2$              ;;BRANCH IF NO
(1)  015600  012737  015344  015342             MOV     #$TKQSRT,$TKQOUT ;;RESET THE POINTER
(1)  015606  000002                     2$:    RTI                     ;;RETURN
(2)                                     ;*****************************************************************
(1)                                     ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
(1)                                     ;*CALL:
(1)                                     ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
(1)                                     ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
(1)                                     ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
(1)
(1)  015610  010346                     $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
(1)  015612  005046                             CLR     -(SP)           ;;CLEAR THE RUBOUT KEY
(1)  015614  012703  016051             1$:    MOV     #$TTYIN,R3      ;;GET ADDRESS
(1)  015620  022703  016075             2$:    CMP     #$TTYIN+20.,R3  ;;BUFFER FULL?
```

```
(1)  015624  101456                          BLOS    4$              ;;BR IF YES
(1)  015626  104412                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
(1)  015630  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
(1)  015632  122713  000177                  CMPB    #177,(R3)       ;;IS IT A RUBOUT
(1)  015636  001022                          BNE     5$              ;;BR IF NO
(1)  015640  005716                          TST     (SP)            ;;IS THIS THE FIRST RUBOUT?
(1)  015642  001007                          BNE     6$              ;;BR IF NO
(1)  015644  112737  000134  016042          MOVB    #'\,9$          ;;TYPE A BACK SLASH
(1)  015652  104400  016042                  TYPE    ,9$
(1)  015656  012716  177777                  MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
(1)  015662  005303                  6$:     DEC     R3              ;;BACKUP BY ONE
(1)  015664  020327  016051                  CMP     R3,#$TTYIN      ;;STACK EMPTY?
(1)  015670  103434                          BLO     4$              ;;BR IF YES
(1)  015672  111337  016042                  MOVB    (R3),9$         ;;SETUP TO TYPEOUT THE DELETED CHAR.
(1)  015676  104400  016042                  TYPE    ,9$             ;;GO TYPE
(1)  015702  000746                          BR      2$              ;;GO READ ANOTHER CHAR.
(1)  015704  005716                  5$:     TST     (SP)            ;;RUBOUT KEY SET?
(1)  015706  001406                          BEQ     7$              ;;BR IF NO
(1)  015710  112737  000134  016042          MOVB    #'\,9$          ;;TYPE A BACK SLASH
(1)  015716  104400  016042                  TYPE    ,9$
(1)  015722  005016                          CLR     (SP)            ;;CLEAR THE RUBOUT KEY
(1)  015724  122713  000025          7$:     CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
(1)  015730  001003                          BNE     8$              ;;BR IF NO
(1)  015732  104400  016044                  TYPE    ,$CNTLU         ;;TYPE A CONTROL "U"
(1)  015736  000726                          BR      1$              ;;GO START OVER
(1)  015740  122713  000012          8$:     CMPB    #12,(R3)        ;;IS CHARACTER A "LF"?
(1)  015744  001011                          BNE     3$              ;;BRANCH IF NO
(1)  015746  105013                          CLRB    (R3)            ;;CLEAR THE CHARACTER
(1)  015750  104400  001207                  TYPE    ,$CRLF          ;;TYPE A "CR" & "LF"
(1)  015754  104400  016051                  TYPE    ,$TTYIN         ;;TYPE THE INPUT STRING
(1)  015760  000717                          BR      2$              ;;GO PICKUP ANOTHER CHACTER
(1)  015762  104400  001206          4$:     TYPE    ,$QUES          ;;TYPE A '?'
(1)  015766  000712                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
(1)  015770  111337  016042          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
(1)  015774  104400  016042                  TYPE    ,9$
(1)  016000  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
(1)  016004  001305                          BNE     2$              ;;LOOP IF NOT RETURN
(1)  016006  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
(1)  016012  104400  001210                  TYPE    ,$LF            ;;TYPE A LINE FEED
(1)  016016  005726                          TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
(1)  016020  012603                          MOV     (SP)+,R3        ;;RESTORE R3
(1)  016022  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
(1)  016024  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
(1)  016032  012766  016051  000004          MOV     #$TTYIN,4(SP)
(1)  016040  000002                          RTI                     ;;RETURN
(1)  016042     000                  9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
(1)  016043     000                          .BYTE   0               ;;TERMINATOR
(1)  016044  052536  005015     000  $CNTLU: .ASCIZ   /↑U/<15><12>   ;;CONTROL "U"
(1)  016051  000024                  $TTYIN: .BLKB    20.            ;;RESERVE 20. BYTES FOR TTY INPUT
(1)  016075     136  006503  000012  $CNTLC: .ASCIZ   /↑C/<15><12>   ;;CONTROL "C"
7273                                 ;******************************************************************
 (1)
 (1)                                 .SBTTL  SCOPE HANDLER ROUTINE
 (1)
```

```
(1)                                     ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
(1)                                     ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
(1)                                     ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
(1)                                     ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
(1)                                     ;*SW14=1        LOOP ON TEST
(1)                                     ;*SW11=1        INHIBIT ITERATIONS
(1)                                     ;*SW09=1        LOOP ON ERROR
(1)                                     ;*CALL
(1)                                     ;*      SCOPE           ;;SCOPE=IOT
(1)
(1)  016102                             $SCOPE:
(1)  016102  006137  177570                     ROL     @#SWR           ;;LOOP ON PRESENT TEST?
(1)  016106  100476                             BMI     $OVER           ;;YES IF SW14=1
(1)                                     ;#####START OF CODE FOR THE XOR TESTER#####
(1)  016110  000416                     $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
(1)                                                                     ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
(1)  016112  013746  000004                     MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
(1)  016116  012737  016136  000004             MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
(1)  016124  005737  177060                     TST     @#177060        ;;TIME OUT ON XOR?
(1)  016130  012637  000004                     MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
(1)  016134  000450                             BR      $SVLAD          ;;GO TO THE NEXT TEST
(1)  016136  022626                     5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
(1)  016140  012637  000004                     MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
(1)  016144  000413                             BR      7$              ;;LOOP ON THE PRESENT TEST
(1)  016146                             6$:;#####END OF CODE FOR THE XOR TESTER#####
(1)  016146  105737  001103             2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
(1)  016152  001421                             BEQ     3$              ;;BR IF NO
(1)  016154  123737  001115  001103             CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
(1)  016162  101015                             BHI     3$              ;;BR IF NO
(1)  016164  032737  001000  177570             BIT     #BIT09,@#SWR    ;;LOOP ON ERROR?
(1)  016172  001404                             BEQ     4$              ;;BR IF NO
(1)  016174  013737  001110  001106     7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
(1)  016202  000440                             BR      $OVER
(1)  016204  105037  001103             4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
(1)  016210  005037  001176                     CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
(1)  016214  000412                             BR      1$              ;;ESCAPE TO THE NEXT TEST
(1)  016216  032737  004000  177570     3$:     BIT     #BIT11,@#SWR    ;;INHIBIT ITERATIONS?
(1)  016224  001006                             BNE     1$              ;;BR IF YES
(1)  016226  005237  001104                     INC     $ICNT           ;;INCREMENT ITERATION COUNT
(1)  016232  023737  001176  001104             CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
(1)  016240  002021                             BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
(1)  016242  012737  000001  001104     1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
(1)  016250  013737  016320  001176             MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
(1)  016256  105237  001102             $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
(1)  016262  011637  001106                     MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
(1)  016266  011637  001110                     MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
(1)  016272  005037  001200                     CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
(1)  016276  112737  000001  001115             MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
(1)  016304  013737  001102  177570     $OVER:  MOV     $TSTNM,@#DISPLAY;;DISPLAY TEST NUMBER
(1)  016312  013716  001106                     MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
(1)  016316  000002                             RTI                     ;;FIXES PS
(1)  016320  000001                     $MXCNT: 1                       ;;MAX. NUMBER OF ITERATIONS
7274                                    ;******************************************************************
 (1)
```

```
(1)                                     .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES
(1)
(1)                                     ;*SAVE R0-R5
(1)                                     ;*CALL:
(1)                                     ;*      SAVREG
(1)                                     ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
(1)                                     ;*
(1)                                     ;*TOP---(+16)
(1)                                     ;* +2---(+18)
(1)                                     ;* +4---R5
(1)                                     ;* +6---R4
(1)                                     ;* +8---R3
(1)                                     ;*+10---R2
(1)                                     ;*+12---R1
(1)                                     ;*+14---R0
(1)
(1)  016322                             $SAVREG:
(3)  016322  010046                             MOV     R0,-(SP)        ;;PUSH R0 ON STACK
(3)  016324  010146                             MOV     R1,-(SP)        ;;PUSH R1 ON STACK
(3)  016326  010246                             MOV     R2,-(SP)        ;;PUSH R2 ON STACK
(3)  016330  010346                             MOV     R3,-(SP)        ;;PUSH R3 ON STACK
(3)  016332  010446                             MOV     R4,-(SP)        ;;PUSH R4 ON STACK
(3)  016334  010546                             MOV     R5,-(SP)        ;;PUSH R5 ON STACK
(1)  016336  016646  000022                     MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
(1)  016342  016646  000022                     MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
(1)  016346  016646  000022                     MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
(1)  016352  016646  000022                     MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
(1)  016356  000002                             RTI
(1)
(1)                                     ;*RESTORE R0-R5
(1)                                     ;*CALL:
(1)                                     ;*      RESREG
(1)  016360                             $RESREG:
(1)  016360  012666  000022                     MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
(1)  016364  012666  000022                     MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
(1)  016370  012666  000022                     MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
(1)  016374  012666  000022                     MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
(3)  016400  012605                             MOV     (SP)+,R5        ;;POP STACK INTO R5
(3)  016402  012604                             MOV     (SP)+,R4        ;;POP STACK INTO R4
(3)  016404  012603                             MOV     (SP)+,R3        ;;POP STACK INTO R3
(3)  016406  012602                             MOV     (SP)+,R2        ;;POP STACK INTO R2
(3)  016410  012601                             MOV     (SP)+,R1        ;;POP STACK INTO R1
(3)  016412  012600                             MOV     (SP)+,R0        ;;POP STACK INTO R0
(1)  016414  000002                             RTI
7275                                    ;******************************************************************
(1)
(1)                                     .SBTTL  TRAP DECODER
(1)
(1)                                     ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
(1)                                     ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
(1)                                     ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
(1)                                     ;*GO TO THAT ROUTINE.
(1)
(1)  016416  010046                     $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
```

```
(1)  016420  016600  000002                MOV     2(SP),R0        ;;GET TRAP ADDRESS
(1)  016424  005740                        TST     -(R0)           ;;BACKUP BY 2
(1)  016426  111000                        MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
(1)  016430  016000  016436                MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
(1)  016434  000200                        RTS     R0              ;;GO TO ROUTINE
(1)
(3)
(3)                                .SBTTL  TRAP TABLE
(3)
(3)                                ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
(3)                                ;*BY THE "TRAP" INSTRUCTION.
(3)
(3)                                ;       ROUTINE
(3)                                ;       -------
(3)  016436                        $TRPAD:
(3)  016436  014556                        $TYPE   ;;CALL=TYPE     TRAP+0(104400)  TTY TYPEOUT ROUTINE
(3)  016440  014710                        $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
(3)  016442  014664                        $TYPOS  ;;CALL=TYPOS    TRAP+4(104404)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
(3)  016444  014724                        $TYPON  ;;CALL=TYPON    TRAP+6(104406)  TYPE OCTAL NUMBER (AS PER LAST CALL)
(3)  016446  015112                        $TYPDS  ;;CALL=TYPDS    TRAP+10(104410) TYPE DECIMAL NUMBER (WITH SIGN)
(3)  016450  015524                        $RDCHR  ;;CALL=RDCHR    TRAP+12(104412) TTY TYPEIN CHARACTER ROUTINE
(3)  016452  015610                        $RDLIN  ;;CALL=RDLIN    TRAP+14(104414) TTY TYPEIN STRING ROUTINE
(3)  016454  016322                        $SAVREG ;;CALL=SAVREG   TRAP+16(104416) SAVE R0-R5 ROUTINE
(3)  016456  016360                        $RESREG ;;CALL=RESREG   TRAP+20(104420) RESTORE R0-R5 ROUTINE
7276                               ;*****************************************************************
(1)
(1)                                .SBTTL  SINGLE LENGTH BINARY TO DECIMAL ASCIZ ROUTINE
(1)
(1)                                ;*THIS ROUTINE WILL CONVERT A 16-BIT UNSIGNED BINARY NUMBER TO AN
(1)                                ;*UNSIGNED DECIMAL ASCIZ NUMBER.
(1)                                ;*CALL
(1)                                ;*      MOV     NUMBER,-(SP)    ;;PUT BINARY NUMBER ON THE STACK
(1)                                ;*      JSR     PC,@#$SB2D      ;;CALL
(1)                                ;*      RETURN                  ;;ADDRESS OF THE 1ST ASCIZ CHAR.IS ON THE STACK
(1)
(1)
(1)  016460  016637  000002  016504 $SB2D:  MOV     2(SP),1$        ;;SAVE BINARY NUMBER
(1)  016466  012746  016504                MOV     #1$,-(SP)       ;;SET POINTER
(1)  016472  004737  016510                JSR     PC,@#$DB2D      ;;CALL DOUBLE LENGTH CONVERT
(1)  016476  012666  000002                MOV     (SP)+,2(SP)     ;;PICKUP POINTER
(1)  016502  000207                        RTS     PC              ;;RETURN
(1)  016504  000000  000000          1$:   .WORD   0,0
7277                               ;*****************************************************************
(1)
(1)                                .SBTTL  DOUBLE LENGTH BINARY TO DECIMAL ASCII CONVERT ROUTINE
(1)
(1)                                ;*THIS ROUTINE WILL CONVERT A 32-BIT BINARY NUMBER TO AN UNSIGNED
(1)                                ;*DECIMAL (ASCII) NUMBER. THE SIGN OF THE BINARY NUMBER MUST BE
(1)                                ;*POSITIVE.
(1)                                ;*CALL
(1)                                ;*      MOV     #PNTR,-(SP)     ;;POINTER TO LOW WORD OF BINARY NUMBER
(1)                                ;*      JSR     PC,@#$DB2D
(1)                                ;*      RETURN                  ;;THE FIRST ADDRESS OF ASCIZ
(1)                                                                ;;IS ON THE STACK
```

```
(1)
(1)
(1)  016510  104416                    $DB2D:  SAVREG                  ;;SAVE REGISTERS
(1)  016512  016602  000002                    MOV     2(SP),R2        ;;PICKUP THE DATA POINTER
(1)  016516  012700  016670                    MOV     #$DECVL,R0      ;;GET ADDRESS OF "$DECVL" STRING
(1)  016522  010066  000002                    MOV     R0,2(SP)        ;;PUT ADDRESS OF ASCIZ STRING ON STACK
(1)  016526  012201                            MOV     (R2)+,R1        ;;PICKUP THE BINARY NUMBER
(1)  016530  012202                            MOV     (R2)+,R2
(1)  016532  012737  000012  016606            MOV     #10.,4$         ;;SET UP TO DO 10 CONVERSIONS
(1)  016540  012704  016620                    MOV     #$TNPWR,R4      ;;ADDRESS OF TEN POWER
(1)  016544  012705  016622                    MOV     #$TNPWR+2,R5
(1)  016550  005003                    1$:     CLR     R3              ;;CLEAR PARTIAL
(1)  016552  161401                    2$:     SUB     (R4),R1         ;;SUBTRACT TEN POWER
(1)  016554  005602                            SBC     R2
(1)  016556  161502                            SUB     (R5),R2
(1)  016560  002402                            BLT     3$              ;;BR IF TEN POWER TO LARGE
(1)  016562  005203                            INC     R3              ;;ADD 1 TO PARTIAL
(1)  016564  000772                            BR      2$              ;;LOOP
(1)  016566  062401                    3$:     ADD     (R4)+,R1        ;;RESTORE SUBTRACTED VALUE
(1)  016570  005502                            ADC     R2
(1)  016572  062402                            ADD     (R4)+,R2
(1)  016574  022525                            CMP     (R5)+,(R5)+     ;;MOVE TO NEXT TEN POWER
(1)  016576  052703  000060                    BIS     #'0,R3          ;;CHANGE PARTIAL TO ASCII
(1)  016602  110320                            MOVB    R3,(R0)+        ;;SAVE IT
(1)  016604  005327                            DEC     (PC)+           ;;DONE?
(1)  016606  000000                    4$:     .WORD   0
(1)  016610  001357                            BNE     1$              ;;BR IF NO
(1)  016612  105020                            CLRB    (R0)+           ;;TERMINATOR
(1)  016614  104420                            RESREG                  ;;RESTORE REGISTERS
(1)  016616  000207                            RTS     PC              ;;RETURN
(1)  016620  145000                    $TNPWR: 145000                  ;;1.0E09
(1)  016622  035632                            35632
(1)  016624  160400                            160400                  ;;1.0E08
(1)  016626  002765                            2765
(1)  016630  113200                            113200                  ;;1.0E07
(1)  016632  000230                            230
(1)  016634  041100                            041100                  ;;1.0E06
(1)  016636  000017                            17
(1)  016640  103240                            103240                  ;;1.0E05
(1)  016642  000001                            1
(1)  016644  023420                            23420                   ;;1.0E04
(1)  016646  000000                            0
(1)  016650  001750                            1750                    ;;1.0E03
(1)  016652  000000                            0
(1)  016654  000144                            144                     ;;1.0E02
(1)  016656  000000                            0
(1)  016660  000012                            12                      ;;1.0E01
(1)  016662  000000                            0
(1)  016664  000001                            1                       ;;1.0E00
(1)  016666  000000                            0
(1)  016670  000014                    $DECVL: .BLKB   12.             ;;RESERVE STORAGE FOR ASCIZ STRING
7278                                   ;******************************************************************
 (1)
 (1)                                   .SBTTL  TYPE NUMERICAL ASCIZ STRING SUPPRESS LEADING ZEROS
```

```
(1)
(1)                                     ;*THIS ROUTINE IS USED TO TYPE AN ASCIZ NUMBER SUPPRESSING THE
(1)                                     ;*LEADING NUMBERS.
(1)                                     ;*CALL
(1)                                     ;*      MOV     #NUMADR,-(SP)   ;;FIRST ADDRESS OF ASCIZ STRING
(1)                                     ;*      JSR     PC,@#$SUPRS
(1)
(1)
(1)  016704  010046                     $SUPRS: MOV     R0,-(SP)        ;;SAVE R0
(1)  016706  016600  000004                     MOV     4(SP),R0        ;;PICKUP THE POINTER
(1)  016712  105710             1$:     TSTB    (R0)            ;;TERMINATEOR?
(1)  016714  001403                     BEQ     2$              ;;BR IF YES
(1)  016716  122720  000060             CMPB    #'0,(R0)+       ;;IS THIS AN ASCII "0" ?
(1)  016722  001773                     BEQ     1$              ;;BR IF YES
(1)  016724  005300             2$:     DEC     R0              ;;BACKUP BY "1"
(1)  016726  010037  016734             MOV     R0,3$           ;;SAVE FOR TYPING
(1)  016732  104400                     TYPE                    ;;GO TYPE
(1)  016734  000000             3$:     .WORD   0               ;;ASCIZ POINTER GOES HERE
(1)  016736  012600                     MOV     (SP)+,R0        ;;RESTORE R0
(1)  016740  012616                     MOV     (SP)+,(SP)      ;;RESTORE THE STACK
(1)  016742  000207                     RTS     PC              ;;RETURN
7279                                    ;******************************************************************
(1)
(1)                                     .SBTTL  INTEGER DIVIDE ROUTINE
(1)
(1)                                     ;*THIS ROUTINE WILL DIVIDE A 32-BIT TWO'S COMPLEMENT INTEGER
(1)                                     ;*DIVIDEND BY A 16-BIT TWO'S COMPLEMENT INTEGER DIVISOR GIVING
(1)                                     ;*A 16-BIT TWO'S COMPLEMENT INTEGER QUOTIENT AND A 16-BIT REMAINDER.
(1)                                     ;*DIVISION WILL BE PERFORMED SO THAT THE REMAINDER IS OF THE
(1)                                     ;*SAVE SIGN AS THE DIVIDEND.
(1)                                     ;*CALL:
(1)                                     ;*      MOV     LOW DIVIDEND,-(SP) ;;THE HIGH DIVIDEND MUST BE < 1/2
(1)                                     ;*      MOV     HIGH DIVIDEND,-(SP);    AS LARGE AS THE DIVISOR
(1)                                     ;*      MOV     DIVISOR,-(SP)
(1)                                     ;*      JSR     PC,$DIV
(1)                                     ;*      RETURN                  ;;QUOTIENT & REMAINDER ARE ON THE STACK
(1)                                     ;*      "V"=0   IMPLIES NO ERROR
(1)                                     ;*      "V"=1   IMPLIES ERROR OCCURRED
(1)                                     ;*              "C"=0   DIVIDE OVERFLOW OCCURRED
(1)                                     ;*              "C"=1   ATTEMPTED TO DIVIDE BY ZERO
(1)                                     ;*
(1)                                     ;*
(1)                                     ;*      STACK   NO ERROR        OVERFLOW        DIVIDE BY ZERO
(1)                                     ;*      -----   --------        --------        -------------
(1)                                     ;*      TOP     REMAINDER       ALL ZEROS       ALL ONES
(1)                                     ;*      +2      QUOTIENT        ALL ZEROS       ALL ONES
(1)
(1)  016744  013746  177776             $DIV:   MOV     @#PS,-(SP)      ;;SAVE THE PROCESSOR STATUS
(1)  016750  042716  000017                     BIC     #17,(SP)        ;;STRIP AWAY CONDITION CODES
(3)  016754  010046                     MOV     R0,-(SP)        ;;PUSH R0 ON STACK
(3)  016756  010146                     MOV     R1,-(SP)        ;;PUSH R1 ON STACK
(3)  016760  010246                     MOV     R2,-(SP)        ;;PUSH R2 ON STACK
(3)  016762  010346                     MOV     R3,-(SP)        ;;PUSH R3 ON STACK
(1)  016764  005046                     CLR     -(SP)           ;;SAVE A PLACE FOR SIGNS
```

```
(1)  016766  012746  000021                    MOV     #17.,-(SP)      ;;SETUP THE ITERATION COUNTER
(1)  016772  016601  000024                    MOV     24(SP),R1       ;;PICKUP THE DIVIDEND
(1)  016776  016600  000022                    MOV     22(SP),R0
(1)  017002  100005                            BPL     1$              ;;CHECK THE SIGN
(1)  017004  105366  000003                    DECB    3(SP)           ;;KEEP TRACK OF THE SIGN
(1)  017010  005400                            NEG     R0              ;;AND NEGATE THE ORIGINAL
(1)  017012  005401                            NEG     R1              ;;NUMBER
(1)  017014  005600                            SBC     R0
(1)  017016  016602  000020            1$:     MOV     20(SP),R2       ;;PICKUP THE DIVISOR
(1)  017022  002407                            BLT     2$              ;;CHECK THE SIGN
(1)  017024  003011                            BGT     3$              ;;DIVISOR OF 0 IS A NO-NO
(1)  017026  052766  000003  000014            BIS     #3,14(SP)       ;;SET "V" & "C"
(1)  017034  012700  177777                    MOV     #-1,R0          ;;SET REMAINDER TO ALL ONES
(1)  017040  000424                            BR      7$              ;;EXIT
(1)  017042  005266  000002            2$:     INC     2(SP)           ;;KEEP TRACK OF DIVISORS SIGN
(1)  017046  000401                            BR      4$
(1)  017050  005402                    3$:     NEG     R2              ;;NEGATE THE ORIGINAL NUMBER
(1)  017052  000241                    4$:     CLC                     ;;CLEAR "C"
(1)  017054  000405                            BR      6$              ;;START FORMING QUOTIENT
(1)  017056  006100                    5$:     ROL     R0              ;;POSITION MSB'S
(1)  017060  010003                            MOV     R0,R3           ;;COPY
(1)  017062  060203                            ADD     R2,R3           ;;COMPARE DIVIDEND & DIVISOR
(1)  017064  103001                            BCC     6$              ;;BR IF DIVIDEND > DIVISOR
(1)  017066  010300                            MOV     R3,R0           ;;REMAINDER AFTER THIS LOOP
(1)  017070  006101                    6$:     ROL     R1              ;;QUOTIENT BIT ENTERS HERE
(1)  017072  005316                            DEC     (SP)            ;;DONE?
(1)  017074  001370                            BNE     5$              ;;BR IF NO
(1)  017076  005701                            TST     R1              ;;OVERFLOW?
(1)  017100  100005                            BPL     8$              ;;BR IF NO
(1)  017102  052766  000002  000014            BIS     #2,14(SP)       ;;SET "V" IN RETURN STATUS WORD
(1)  017110  005000                            CLR     R0              ;;SET REMAINDER TO ALL ZEROS
(1)  017112  010001                    7$:     MOV     R0,R1           ;;COPY REMAINDER INTO QUOTIENT
(1)  017114  005726                    8$:     TST     (SP)+           ;;CLEAR COUNTER FROM STACK
(1)  017116  005716                            TST     (SP)            ;;REMAINDER SIGN CORRECTION NEEDED?
(1)  017120  002004                            BGE     9$              ;;BR IF NO
(1)  017122  005400                            NEG     R0              ;;NEGATE REMAINDER
(1)  017124  105066  000001                    CLRB    1(SP)           ;;CLEAR SIGN
(1)  017130  005316                            DEC     (SP)            ;;BUT DON'T FORGET QUOTIENT
(1)  017132  005726                    9$:     TST     (SP)+           ;;QUOTIENT SIGN CORRECTION NEEDED?
(1)  017134  001401                            BEQ     10$             ;;BR IF NO
(1)  017136  005401                            NEG     R1              ;;NEGATE QUOTIENT
(1)  017140  010166  000020            10$:    MOV     R1,20(SP)       ;;RETURN QUOTIENT AND
(1)  017144  010066  000016                    MOV     R0,16(SP)       ;;REMAINDER TO USER
(3)  017150  012603                            MOV     (SP)+,R3        ;;POP STACK INTO R3
(3)  017152  012602                            MOV     (SP)+,R2        ;;POP STACK INTO R2
(3)  017154  012601                            MOV     (SP)+,R1        ;;POP STACK INTO R1
(3)  017156  012600                            MOV     (SP)+,R0        ;;POP STACK INTO R0
(1)  017160  012666  000002                    MOV     (SP)+,2(SP)     ;;SETUP TO RETURN CONDITION CODES
(1)  017164  000002                            RTI                     ;;RETURN
7280                            ;*****************************************************************
(1)
(1)                             .SBTTL  RANDOM NUMBER GENERATOR ROUTINE
(1)
(1)                             ;*THIS ROUTINE IS A DOUBLE PRECISION PSEUDO RANDOM NUMBER GENERATOR
```

```
(1)                                     ;*WITH A RANGE OF 0 TO 2(+33)-1.
(1)                                     ;*CALL:
(1)                                     ;*      JSR     PC,$RAND        ;;CALL THE ROUTINE
(1)                                     ;*      RETURN                  ;;RETURN HERE THE RANDOM
(1)                                     ;*                              ;;NUMBER WILL BE IN
(1)                                     ;*                              ;;$HINUM,$LONUM
(1)
(2)  017166                             $RAND:
(3)  017166  010046                             MOV     R0,-(SP)        ;;PUSH R0 ON STACK
(3)  017170  010146                             MOV     R1,-(SP)        ;;PUSH R1 ON STACK
(3)  017172  010246                             MOV     R2,-(SP)        ;;PUSH R2 ON STACK
(3)  017174  010346                             MOV     R3,-(SP)        ;;PUSH R3 ON STACK
(1)  017176  013700  017314                     MOV     $LONUM,R0       ;;SET R0 WITH LOW
(1)  017202  013701  017312                     MOV     $HINUM,R1       ;;SET R1 WITH HIGH
(1)  017206  012703  177771                     MOV     #-7,R3          ;;SET SHIFT COUNT
(1)  017212  005002                             CLR     R2              ;;ZERO R2
(1)  017214  006300                     1$:     ASL     R0              ;;SHIFT R0 LEFT AND
(1)  017216  006101                             ROL     R1              ;;ROTATE CARRY INTO R1 AND
(1)  017220  006102                             ROL     R2              ;;ROTATE CARRY INTO R2
(1)  017222  005203                             INC     R3              ;;CHECK FOR DONE
(1)  017224  001373                             BNE     1$              ;;CONTINUE SHIFT LOOP
(1)  017226  063700  017314                     ADD     $LONUM,R0       ;;ADD NUMBER TO MAKE X 129
(1)  017232  005501                             ADC     R1              ;;PROPOGATE CARRY
(1)  017234  063701  017312                     ADD     $HINUM,R1       ;;ADD NUMBER TO MAKE X 129
(1)  017240  005502                             ADC     R2              ;;PROPOGATE CARRY
(1)  017242  062700  001057                     ADD     #1057,R0        ;;ADD LOW CONSTANT
(1)  017246  005501                             ADC     R1              ;;PROPOGATE CARRY
(1)  017250  005502                             ADC     R2              ;;PROPOGATE CARRY
(1)  017252  062701  047401                     ADD     #47401,R1       ;;ADD HIGH CONSTANT
(1)  017256  005502                             ADC     R2              ;;PROPOGATE CARRY
(1)  017260  062702  000006                     ADD     #6,R2           ;;ADD HIGHEST CONSTART
(1)  017264  060200                             ADD     R2,R0           ;;REPRIME R0 WITH HIGHEST DIGIT
(1)  017266  005501                             ADC     R1              ;;PROPOGATE CARRY
(1)  017270  010037  017314                     MOV     R0,$LONUM       ;;SAVE R0
(1)  017274  010137  017312                     MOV     R1,$HINUM       ;;SAVE R1
(3)  017300  012603                             MOV     (SP)+,R3        ;;POP STACK INTO R3
(3)  017302  012602                             MOV     (SP)+,R2        ;;POP STACK INTO R2
(3)  017304  012601                             MOV     (SP)+,R1        ;;POP STACK INTO R1
(3)  017306  012600                             MOV     (SP)+,R0        ;;POP STACK INTO R0
(1)  017310  000207                             RTS     PC              ;;RETURN
(1)  017312  176543                     $HINUM: .WORD   176543
(1)  017314  123456                     $LONUM: .WORD   123456
7281
7282                                    ;******************************************************************
7283                                    .SBTTL  LP.AVL - LINE PRINTER AVAILABLE
7284                                    ;*THIS ROUTINE WILL CHECK IF THERE IS A LINE PRINTER AVAILABLE AND SET
7285                                    ;*"LPTAVL" TO THE PROPER STATE.
7286                                    ;* LPTAVL = 0 IF NO LINE PRINTER AVAILABLE
7287                                    ;* LPTAVL = 1 IF LINE PRINTER IS AVAILABLE
7288                                    ;*CALL
7289                                    ;*      JSR     PC,@#LP.AVL
7290                                    ;*      RETURN
7291
7292  017316  005037  001220            LP.AVL: CLR     @#LPTAVL        ;START WITH NO PRINTER AVAIABLE
```

```
7293  017322  012737  [illegible]346  000004          MOV     #1$,@#ERRVEC    ;SETUP THE TIMEOUT VECTOR
7294  017330  005037  [illegible]006                  CLR     @#ERRVEC+2
7295  017334  005777  162020                          TST     @LPS            ;IS THERE A LINE PRINTER?
7296  017340  005237  001220                          INC     @#LPTAVL        ;YES--SET AVAILABLE SWITCH
7297  017344  000401                                  BR      2$
7298  017346  022626                          1$:     CMP     (SP)+,(SP)+     ;NO--POP STACK
7299  017350  012737  000006  000004        2$:     MOV     #ERRVEC+2,@#ERRVEC ;RESTORE TIMEOUT VECTOR
7300  017356  000207                                  RTS     PC              ;RETURN
7301
7302                                  ;*****************************************************************
7303                                  .SBTTL  ST.CLK - CLOCK STARTUP ROUTINE
7304                                  ;*THIS ROUTINE WILL DETERMINE IF THERE IS A CLOCK ON THE SYSTEM
7305                                  ;*AND IF THERE IS IT WILL SETUP THE VECTOR AND START THE CLOCK
7306                                  ;*"CLKSTA" WILL INDICATE THE CLOCK TYPE
7307                                  ;* 0= NO CLOCK
7308                                  ;*+1= KW11-P
7309                                  ;*-1= KW11-L
7310                                  ;*THIS ROUTINE WILL ALSO SETUP "TICKMS" (TIME
7311                                  ;*PER CLOCK TICK IN MILLISECONDS) AND "TICKUS"
7312                                  ;*(TIME PER CLOCK TICK IN MICROSECONDS) AS
7313                                  ;*PER SW00.
7314                                  ;*SW00=0 -- 60HZ
7315                                  ;*SW00=1 -- 50HZ
7316                                  ;*CALL
7317                                  ;*      JSR     PC,@#ST.CLK
7318                                  ;*      RETURN
7319
7320  017360  010146                  ST.CLK: MOV     R1,-(SP)        ;SAVE R1
7321  017362  012701  000006                  MOV     #ERRVEC+2,R1    ;SAVE AND SETUP TIMEOUT VECTOR
7322  017366  011146                          MOV     (R1),-(SP)
7323  017370  005011                          CLR     (R1)            ;LEVEL 0
7324  017372  014146                          MOV     -(R1),-(SP)
7325  017374  012711  017424                  MOV     #1$,(R1)        ;GO TO 1$ ON TIMEOUT
7326  017400  005037  001226                  CLR     CLKSTA          ;SET CLOCK STATUS TO NO CLOCK
7327  017404  005777  161730                  TST     @PKCS           ;IS THERE A KW11-P?
7328  017410  012737  000001  001226          MOV     #1,CLKSTA       ;YES--SET STATUS TO KW11-P
7329  017416  004737  017526                  JSR     PC,ST.PCLK      ;START THE KW11-P
7330  017422  000414                          BR      3$              ;GO TO EXIT
7331  017424  022626                  1$:     CMP     (SP)+,(SP)+     ;CLEAN UP THE STACK
7332  017426  012711  017452                  MOV     #2$,(R1)        ;IF TIMEOUT GO TO 2$
7333  017432  005777  161714                  TST     @LKS            ;IS THERE A KW11-L?
7334  017436  012737  177777  001226          MOV     #-1,CLKSTA      ;YES-- SET STATUS TO KW11-L
7335  017444  004737  017570                  JSR     PC,ST.LCLK      ;START THE KW11-L
7336  017450  000401                          BR      3$              ;EXIT
7337  017452  022626                  2$:     CMP     (SP)+,(SP)+     ;CLEAN UP THE STACK
7338  017454  012621                  3$:     MOV     (SP)+,(R1)+     ;RESTORE THE TIMEOUT VECTOR
7339  017456  012621                          MOV     (SP)+,(R1)+
7340  017460  012601                          MOV     (SP)+,R1        ;RESTORE R1
7341  017462  032737  000100  001212          BIT     #SW06,@#C.SWR   ;50HZ OR 60HZ?
7342  017470  001407                          BEQ     4$              ;BRANCH IF 60
7343  017472  012737  000020  001230          MOV     #20,@#TICKMS    ;SETUP TIME PER
7344  017500  012737  047040  001232          MOV     #20000.,@#TICKUS ;TICK FOR 50HZ
7345  017506  000406                          BR      5$
7346  017510  012737  000016  001230  4$:     MOV     #16,@#TICKMS    ;SETUP TIME PER
```

```
7347  017516  012737  040432  001232          MOV     #16666.,@#TICKUS ;TICK FOR 60HZ
7348  017524  000207                  5$:     RTS     PC               ;RETURN
7349
7350  017526                          ST.PCLK:
7351  017526  032737  000040  001212          BIT     #SW05,@#C.SWR    ;ALLOW SOFTWARE TIMEOUTS?
7352  017534  001014                          BNE     1$               ;NO--BRANCH
7353  017536  012777  017624  161570          MOV     #SRVCLK,@PKV     ;SETUP THE KW11-P VECTOR
7354  017544  012777  000300  161564          MOV     #300,@PKV+2
7355  017552  012777  000001  161562          MOV     #1,@PKB          ;COUNT ONE TICK
7356  017560  012777  000115  161552          MOV     #115,@PKCS       ;"INT.EN." ,COUNT DOWN", "MODE 1 (REPEAT)",
7357                                                                   ;"LINE FREQ", AND "RUN"
7358  017566  000207                  1$:     RTS     PC               ;RETURN
7359
7360  017570                          ST.LCLK:
7361  017570  032737  000040  001212          BIT     #SW05,@#C.SWR    ;ALLOW SOFTWARE TIMEOUTS?
7362  017576  001011                          BNE     1$               ;NO--BRANCH
7363  017600  012777  017624  161540          MOV     #SRVCLK,@LKV     ;SETUP THE KW11-L VECTOR
7364  017606  012777  000300  161534          MOV     #300,@LKV+2
7365  017614  012777  000100  161530          MOV     #100,@LKS        ;START THE KW11-L
7366  017622  000207                  1$:     RTS     PC               ;RETURN
7367
7368  017624  013746  001230          SRVCLK: MOV     @#TICKMS,-(SP)   ;TIME PER TICK IN MILLISECONDS
7369  017630  004037  032534                  JSR     R0,@#RPTMR       ;COUNT THE ELASPED TIME
7370  017634  000002                          RTI                      ;RETURN AFTER INTERRUPT
7371
7372                                  ;*****************************************************************
7373                                  .SBTTL  LDCMD - LOAD COMMAND
7374                                  ;*THIS ROUTINE LOADS A READ HEADER AND DATA COMMAND OR A SEEK COMMAND
7375                                  ;*INTO DPB.B+2 AND DPB.C+2, DEPENDING ON THE STATE OF "CONTROL SWITCH"
7376                                  ;*BIT07.
7377                                  ;*CALL
7378                                  ;*      JSR     PC,@#LDCMD
7379                                  ;*      RETURN
7380
7381  017636  032737  000200  001212  LDCMD:  BIT     #SW07,@#C.SWR    ;DO EXPLICIT SEEKS?
7382  017644  001007                          BNE     1$               ;YES--BRANCH
7383  017646  012737  000173  003462          MOV     #READHD,@#DPB.B+2 ;NO--SET UP FOR READ HEADER AND
7384  017654  012737  000173  003502          MOV     #READHD,@#DPB.C+2 ;DATA COMMAND
7385  017662  000406                          BR      2$
7386  017664  012737  000105  003462  1$:     MOV     #SEEK,@#DPB.B+2  ;SETUP FOR SEEK COMMAND
7387  017672  012737  000105  003502          MOV     #SEEK,@#DPB.C+2
7388  017700  000207                  2$:     RTS     PC
7389
7390                                  ;*****************************************************************
7391                                  .SBTTL  CALL.A - CALL RP04 DRIVER USING "DPB.A"
7392                                  ;*THIS ROUTINE WILL CALL THE RP04 DRIVER AND THEN WAIT ON THE FUNCTION
7393                                  ;*TO COMPLETE. IF AN ERROR OCCURS IT IS REPORTED.
7394                                  ;*CALL
7395                                  ;*      FILL "DPB" WITH COMMAND INFORMATION
7396                                  ;*      JSR     R0,@#CALL.A
7397                                  ;*      RETURN
7398
7399  017702  005037  001200          CALL.A: CLR     @#$ESCAPE        ;NO ESCAPE ADDRESS
7400  017706  004037  027574                  JSR     R0,@#RP04        ;CALL RP04 DRIVER
```

```
7401  017712  003440                          DPB.A
7402  017714  000772                          BR      CALL.A
7403  017716  005737  003456          1$:     TST     @#DPB.A+16      ;DONE?
7404  017722  001775                          BEQ     1$              ;NO--LOOP
7405  017724  100030                          BPL     3$              ;BRANCH IF NO ERROR
7406  017726  012737  020002  001200          MOV     #2$,$ESCAPE     ;;ESCAPE TO 2$ ON ERROR
7407  017734  013737  003452  001252          MOV     @#DPB.A+12,@#CYL.DS ;CYLINDER
 (1)  017742  113737  003451  001256          MOVB    @#DPB.A+11,@#TRK.DS ;TRACK
 (1)  017750  113737  003450  001254          MOVB    @#DPB.A+10,@#SEC.DS ;SECTOR
 (1)  017756  012746  003456                  MOV     #DPB.A+16,-(SP) ;STATUS/ERROR INDICATOR ADDRESS
 (1)  017762  004737  020766                  JSR     PC,@#ERINDX     ;FORM DISPATCH INDEX
 (1)  017766  062607                          ADD     (SP)+,PC        ;REPORT PROPER ERROR
 (1)  017770  104041                          ERROR   41              ;
 (1)  017772  104042                          ERROR   42              ;PARITY ERROR
 (1)  017774  104043                          ERROR   43              ;UNSAFE ERROR
 (1)  017776  104044                          ERROR   44              ;NON-I/O ERROR
 (1)  020000  104045                          ERROR   45              ;I/O ERROR
7408  020002  013700  001234          2$:     MOV     @#BYPASS,R0     ;TAKE ERROR EXIT
7409  020006  000200                  3$:     RTS     R0              ;RETURN
7410
7411                                  ;*****************************************************************
7412                                  .SBTTL  CALL.B - CALL RP04 DRIVER USING "DPB.B"
7413                                  ;*THIS ROUTINE IS THE SAME AS "CALL.A" EXCEPT FOR THE DPB USED AND IF
7414                                  ;*THE COMMAND IS A READ HEADER AND DATA THE HEADER (CYLINDER, TRACK,
7415                                  ;*AND SECTOR) READ IS CHECKED FOR VALIDITY.
7416                                  ;*CALL
7417                                  ;*      FILL DPB
7418                                  ;*      JSR     R0,@#CALL.B
7419                                  ;*      RETURN
7420
7421  020010  005037  001200          CALL.B: CLR     @#$ESCAPE       ;NO ESCAPE ADDRESS
7422  020014  004037  027574                  JSR     R0,@#RP04       ;CALL RP04 DRIVER
7423  020020  003460                          DPB.B
7424  020022  000772                          BR      CALL.B
7425  020024  005737  003476          1$:     TST     DPB.B+16        ;DONE?
7426  020030  001775                          BEQ     1$              ;NO--BRANCH
7427  020032  100031                          BPL     3$              ;BRANCH IF NO ERROR
7428  020034  012737  020110  001200          MOV     #2$,$ESCAPE     ;;ESCAPE TO 2$ ON ERROR
7429  020042  013737  003472  001252          MOV     @#DPB.B+12,@#CYL.DS ;CYLINDER
 (1)  020050  113737  003471  001256          MOVB    @#DPB.B+11,@#TRK.DS ;TRACK
 (1)  020056  113737  003470  001254          MOVB    @#DPB.B+10,@#SEC.DS ;SECTOR
 (1)  020064  012746  003476                  MOV     #DPB.B+16,-(SP) ;STATUS/ERROR INDICATOR ADDRESS
 (1)  020070  004737  020766                  JSR     PC,@#ERINDX     ;FORM DISPATCH INDEX
 (1)  020074  062607                          ADD     (SP)+,PC        ;REPORT PROPER ERROR
 (1)  020076  104041                          ERROR   41              ;
 (1)  020100  104042                          ERROR   42              ;PARITY ERROR
 (1)  020102  104043                          ERROR   43              ;UNSAFE ERROR
 (1)  020104  104044                          ERROR   44              ;NON-I/O ERROR
 (1)  020106  104045                          ERROR   45              ;I/O ERROR
7430  020110  013700  001234          2$:     MOV     @#BYPASS,R0     ;TAKE ERROR EXIT
7431  020114  000407                          BR      4$
7432  020116  123727  003462  000173  3$:     CMPB    @#DPB.B+2,#READHD ;DOING IMPLIED SEEKS?
7433  020124  001003                          BNE     4$              ;NO--BRANCH
7434  020126  004037  021246                  JSR     R0,@#VERIFY     ;YES--GO CHECK THE DATA
```

```
7435  020132  003470                        DPB.B+10
7436  020134  032737  040000  001212  4$:   BIT     #SW14,@#C.SWR   ;STALL?
7437  020142  001403                        BEQ     5$              ;NO--BRANCH
7438  020144  004037  021104                JSR     R0,@#STALL      ;YES--CALL STALL ROUTINE
7439  020150  001364                        .WORD   STALL1          ;STALL TIME POINTER
7440  020152  000200                5$:     RTS     R0              ;RETURN
7441
7442                                ;*****************************************************************
7443                                .SBTTL  CALL.C - CALL RP04 DRIVER USING "DBP.C"
7444                                ;*THIS ROUTINE IS THE SAME AS "CALL.B" EXCEPT FOR THE DPB USED.
7445                                ;*CALL
7446                                ;*      FILL DPB
7447                                ;*      JSR     R0,@#CALL.C
7448                                ;*      RETURN
7449
7450  020154  005037  001200        CALL.C: CLR     @#$ESCAPE       ;NO ESCAPE ADDRESS
7451  020160  004037  027574                JSR     R0,@#RP04       ;CALL RP04 DRIVER
7452  020164  003500                        DPB.C
7453  020166  000772                        BR      CALL.C
7454  020170  005737  003516        1$:     TST     @#DPB.C+16      ;DONE?
7455  020174  001775                        BEQ     1$              ;NO--LOOP
7456  020176  100031                        BPL     3$              ;YES--BRANCH IF NO ERROR
7457  020200  012737  020254  001200        MOV     #2$,$ESCAPE     ;;ESCAPE TO 2$ ON ERROR
7458  020206  013737  003512  001252        MOV     @#DPB.C+12,@#CYL.DS ;CYLINDER
 (1)  020214  113737  003511  001256        MOVB    @#DPB.C+11,@#TRK.DS ;TRACK
 (1)  020222  113737  003510  001254        MOVB    @#DPB.C+10,@#SEC.DS ;SECTOR
 (1)  020230  012746  003516                MOV     #DPB.C+16,-(SP) ;STATUS/ERROR INDICATOR ADDRESS
 (1)  020234  004737  020766                JSR     PC,@#ERINDX     ;FORM DISPATCH INDEX
 (1)  020240  062607                        ADD     (SP)+,PC        ;REPORT PROPER ERROR
 (1)  020242  104041                        ERROR   41              ;
 (1)  020244  104042                        ERROR   42              ;PARITY ERROR
 (1)  020246  104043                        ERROR   43              ;UNSAFE ERROR
 (1)  020250  104044                        ERROR   44              ;NON-I/O ERROR
 (1)  020252  104045                        ERROR   45              ;I/O ERROR
7459  020254  013700  001234        2$:     MOV     @#BYPASS,R0     ;TAKE ERROR EXIT
7460  020260  000400                        BR      3$
7461  020262  123727  003502  000173  3$:   CMPB    @#DPB.C+2,#READHD ;DOING IMPLIED SEEK?
7462  020270  001003                        BNE     4$              ;NO--EXIT
7463  020272  004037  021246                JSR     R0,@#VERIFY     ;YES--CHECK THE DATA
7464  020276  003510                        DPB.C+10
7465  020300  032737  040000  001212  4$:   BIT     #SW14,@#C.SWR   ;STALL?
7466  020306  001403                        BEQ     5$              ;NO--BRANCH
7467  020310  004037  021104                JSR     R0,@#STALL      ;YES--CALL STALL ROUTINE
7468  020314  001364                        .WORD   STALL1          ;STALL TIME POINTER
7469  020316  000200                5$:     RTS     R0
7470
7471
7472                                ;*****************************************************************
7473                                .SBTTL  DRVCAL - DRIVER (RP04) CALL USING "DTADPB"
7474                                ;*THIS ROUTINE IS THE SAME AS "CALL.A" EXCEPT FOR THE DPB USED AND
7475                                ;*ON AN ERROR LOCATION "ERR.CT" IS EXAMINED. IF ERR.CT IS EQUAL TO
7476                                ;*$ERFLG EXIT IS TO THE NEXT TEST.
7477                                ;*CALL
7478                                ;*      FILL DPB
```

```
7479                                    ;*      JSR     R0,@#DRVCAL
7480                                    ;*      RETURN
7481
7482  020320  005037  001200            DRVCAL: CLR     @#$ESCAPE       ;NO ESCAPE ADDRESS
7483  020324  005037  001316                    CLR     @#WCEFLG        ;CLEAR WRITE CHECK ERROR FLAG
7484  020330  004037  027574                    JSR     R0,@#RP04       ;CALL RP04 DRIVER
7485  020334  003520                            DTADPB
7486  020336  000770                            BR      DRVCAL
7487  020340  005737  003536            1$:     TST     @#DTADPB+16     ;DONE
7488  020344  001775                            BEQ     1$              ;NO--LOOP
7489  020346  100177                            BPL     10$             ;YES--BRANCH IF NO ERRORS
7490  020350  012737  020424  001200            MOV     #2$,$ESCAPE     ;;ESCAPE TO 2$ ON ERROR
7491  020356  013737  003532  001252            MOV     @#DTADPB+12,@#CYL.DS ;CYLINDER
  (1) 020364  113737  003531  001256            MOVB    @#DTADPB+11,@#TRK.DS ;TRACK
  (1) 020372  113737  003530  001254            MOVB    @#DTADPB+10,@#SEC.DS ;SECTOR
  (1) 020400  012746  003536                    MOV     #DTADPB+16,-(SP) ;STATUS/ERROR INDICATOR ADDRESS
  (1) 020404  004737  020766                    JSR     PC,@#ERINDX     ;FORM DISPATCH INDEX
  (1) 020410  062607                            ADD     (SP)+,PC        ;REPORT PROPER ERROR
  (1) 020412  104041                            ERROR   41              ;
  (1) 020414  104042                            ERROR   42              ;PARITY ERROR
  (1) 020416  104043                            ERROR   43              ;UNSAFE ERROR
  (1) 020420  104044                            ERROR   44              ;NON-I/O ERROR
  (1) 020422  104045                            ERROR   45              ;I/O ERROR
7492  020424  122737  000015  001102  2$:     CMPB    #15,@#$TSTNM    ;TEST 15?
7493  020432  001133                            BNE     8$              ;NO--BRANCH
7494  020434  122737  000151  003522            CMPB    #WRCKD,@#DTADPB+2 ;DOING A WRITE CHECK?
7495  020442  001127                            BNE     8$              ;NO--BRANCH
7496  020444  032737  040000  003550            BIT     #BIT14,@#RP.REG+10 ;IS "WCE"=1?
7497  020452  001523                            BEQ     8$              ;NO--BRANCH
7498  020454  032737  000020  177570            BIT     #SW04,@#SWR     ;INHIBIT WRITES?
7499  020462  001117                            BNE     8$              ;YES--BRANCH
7500  020464  112737  000161  003522            MOVB    #WRITE,@#DTADPB+2 ;SETUP FOR A WRITE
7501  020472  005037  001200                    CLR     @#$ESCAPE       ;NO ESCAPE ADDRESS
7502  020476  004037  027574                    JSR     R0,@#RP04       ;DO THE WRITE
7503  020502  003520                            DTADPB
7504  020504  000240                            NOP
7505  020506  005737  003536            3$:     TST     @#DTADPB+16     ;DONE?
7506  020512  001775                            BEQ     3$              ;NO--LOOP
7507  020514  100026                            BPL     4$              ;YES--BRANCH IF NO ERROR
7508  020516  012737  020722  001200            MOV     #8$,$ESCAPE     ;;ESCAPE TO 8$ ON ERROR
7509  020524  013737  003532  001252            MOV     @#DTADPB+12,@#CYL.DS ;CYLINDER
  (1) 020532  113737  003531  001256            MOVB    @#DTADPB+11,@#TRK.DS ;TRACK
  (1) 020540  113737  003530  001254            MOVB    @#DTADPB+10,@#SEC.DS ;SECTOR
  (1) 020546  012746  003536                    MOV     #DTADPB+16,-(SP) ;STATUS/ERROR INDICATOR ADDRESS
  (1) 020552  004737  020766                    JSR     PC,@#ERINDX     ;FORM DISPATCH INDEX
  (1) 020556  062607                            ADD     (SP)+,PC        ;REPORT PROPER ERROR
  (1) 020560  104041                            ERROR   41              ;
  (1) 020562  104042                            ERROR   42              ;PARITY ERROR
  (1) 020564  104043                            ERROR   43              ;UNSAFE ERROR
  (1) 020566  104044                            ERROR   44              ;NON-I/O ERROR
  (1) 020570  104045                            ERROR   45              ;I/O ERROR
7510  020572  112737  000151  003522  4$:     MOVB    #WRCKD,@#DTADPB+2 ;COMMAND=WRITE CHECK DATA
7511  020600  004037  027574                    JSR     R0,@#RP04       ;DO THE WRITE CHECK
7512  020604  003520                            DTADPB
```

```
7513  020606  000240                        NOP
7514  020610  005737  003536          5$:   TST     @#DTADPB+16       ;DONE?
7515  020614  001775                        BEQ     5$                ;NO--LOOP
7516  020616  100410                        BMI     7$                ;YES--BRANCH IF ERROR
7517  020620  004037  027574                JSR     R0,@#RP04         ;DO A 2ND WRITE CHECK
7518  020624  003520                        DTADPB
7519  020626  000240                        NOP
7520  020630  005737  003536          6$:   TST     @#DTADPB+16       ;DONE?
7521  020634  001775                        BEQ     6$                ;NO--LOOP
7522  020636  100031                        BPL     8$                ;YES--BRANCH IF NO ERROR
7523  020640  012737  000001  001316  7$:   MOV     #1,@#WCEFLG       ;SET THE WRITE CHECK ERROR FLAG
7524  020646  012737  020722  001200        MOV     #8$,$ESCAPE       ;;ESCAPE TO 8$ ON ERROR
7525  020654  013737  003532  001252        MOV     @#DTADPB+12,@#CYL.DS ;CYLINDER
  (1) 020662  113737  003531  001256        MOVB    @#DTADPB+11,@#TRK.DS ;TRACK
  (1) 020670  113737  003530  001254        MOVB    @#DTADPB+10,@#SEC.DS ;SECTOR
  (1) 020676  012746  003536                MOV     #DTADPB+16,-(SP) ;STATUS/ERROR INDICATOR ADDRESS
  (1) 020702  004737  020766                JSR     PC,@#ERINDX       ;FORM DISPATCH INDEX
  (1) 020706  062607                        ADD     (SP)+,PC          ;REPORT PROPER ERROR
  (1) 020710  104041                        ERROR   41                ;
  (1) 020712  104042                        ERROR   42                ;PARITY ERROR
  (1) 020714  104043                        ERROR   43                ;UNSAFE ERROR
  (1) 020716  104044                        ERROR   44                ;NON-I/O ERROR
  (1) 020720  104046                        ERROR   46                ;FATAL WRITE CHECK
7526  020722  032737  001000  177570  8$:   BIT     #SW09,@#SWR       ;LOOP ON ERROR?
7527  020730  001004                        BNE     9$                ;YES--BRANCH
7528  020732  123737  001372  001103        CMPB    @#ERR.CT,@#SERFLG ;GO TO NEXT TEST?
7529  020740  101002                        BHI     10$               ;NO--BRANCH
7530  020742  013700  001234          9$:   MOV     @#BYPASS,R0       ;YES--GET EXIT ADDRESS
7531  020746  032737  040000  001212  10$:   BIT     #SW14,@#C.SWR     ;STALL?
7532  020754  001403                        BEQ     11$               ;NO--BRANCH
7533  020756  004037  021104                JSR     R0,@#STALL        ;YES--CALL STALL ROUTINE
7534  020762  001366                        .WORD   STALL2            ;STALL TIME POINTER
7535  020764  000200                  11$:  RTS     R0
7536
7537                                  ;*****************************************************************
7538                                  .SBTTL  ERINDX - FORM ERROR INDEX
7539                                  ;*THIS ROUTINE FORMS AN INDEX THAT WILL BE USED TO DISPATCH
7540                                  ;*TO THE PROPER ERROR CALL.  THE INDEX IS FORMED BY EXAMINING
7541                                  ;*THE STATUS/ERROR INDICATOR OF THE APPLICABLE DPB.
7542                                  ;*INDEX          STATUS/ERROR
7543                                  ;*-----          ------------
7544                                  ;*     0         BIT14!BIT13!BIT08
7545                                  ;*     2         BIT11!BIT10
7546                                  ;*     4         BIT12!BIT04
7547                                  ;*     6         BIT05!BIT03!<BIT09 & COMMAND=NON-I/O>
7548                                  ;*    10         BIT06!<BIT09 & COMMAND=I/O>
7549                                  ;*CALL
7550                                  ;*      JSR     #DPB+16,-(SP)   ;ADDRESS OF STATUS/ERROR INDICATOR
7551                                  ;*      JSR     PC,@#ERINDX     ;FORM INDEX
7552                                  ;*      RETURN                  ;INDEX IS ON THE STACK
7553
7554  020766  010046                  ERINDX: MOV     R0,-(SP)          ;SAVE R0
7555  020770  010146                        MOV     R1,-(SP)          ;SAVE R1
7556  020772  016600  000006                MOV     6(SP),R0          ;GET STATUS/ERROR INDICATOR POINTER
```

```
7557  020776  011037  001242                  MOV     (R0),@#SVSTAT   ;SAVE THE STATUS/ERROR INDICATOR
7558  021002  005001                          CLR     R1              ;START INDEX AT ZERO
7559  021004  032710                          BIT     (PC)+,(R0)      ;FORM INDEX OF 0?
7560  021006  060400                          .WORD   BIT14!BIT13!BIT08
7561  021010  001027                          BNE     5$              ;YES--BRANCH
7562  021012  032710                          BIT     (PC)+,(R0)      ;FORM PARITY ERROR INDEX (2)?
7563  021014  006000                          .WORD   BIT11!BIT10
7564  021016  001023                          BNE     4$              ;YES--BRANCH
7565  021020  032710                          BIT     (PC)+,(R0)      ;FORM UNSAFE INDEX (4)?
7566  021022  010020                          .WORD   BIT12!BIT04
7567  021024  001017                          BNE     3$              ;YES--BRANCH
7568  021026  032710                          BIT     (PC)+,(R0)      ;FORM NON-I/O ERROR INDEX (6)?
7569  021030  000050                          .WORD   BIT05!BIT03
7570  021032  001013                          BNE     2$              ;YES--BRANCH
7571  021034  032710                          BIT     (PC)+,(R0)      ;FORM I/O ERROR INDEX (10)?
7572  021036  000100                          .WORD   BIT06
7573  021040  001007                          BNE     1$              ;YES--BRANCH
7574  021042  032710                          BIT     (PC)+,(R0)      ;SOFTWARE TIMEOUT?
7575  021044  001000                          .WORD   BIT09
7576  021046  001410                          BEQ     5$              ;NO--FORM INDEX OF 0
7577  021050  122760  000150  177762          CMPB    #150,-16(R0)    ;YES--I/O?
7578  021056  003001                          BGT     2$              ;NO--BRANCH
7579  021060  005201                  1$:     INC     R1              ;INDEX=10---ERROR=45 OR 46
7580  021062  005201                  2$:     INC     R1              ;INDEX=6---ERROR=44
7581  021064  005201                  3$:     INC     R1              ;INDEX=4---ERROR=43
7582  021066  005201                  4$:     INC     R1              ;INDEX=2---ERROR=42
7583  021070  006301                  5$:     ASL     R1              ;INDEX=0---ERROR=41
7584  021072  010166  000006                  MOV     R1,6(SP)        ;RETURN INDEX TO USER
7585  021076  012601                          MOV     (SP)+,R1        ;RESTORE R1
7586  021100  012600                          MOV     (SP)+,R0        ;RESTORE R0
7587  021102  000207                          RTS     PC              ;RETURN FROM CALL
7588
7589                          ;*****************************************************************
7590                          .SBTTL  STALL - USED TO STALL AFTER A DRIVE FUNCTION IS COMPLETE
7591                          ;*THIS ROUTINE WILL PROVIDE A STALL IN MILLISECONDS FOR A SPECIFIC
7592                          ;*AMOUNT OF TIME IF BIT13 OF C.SWR = 0 OR A RANDOM AMOUNT OF TIME
7593                          ;*IF BIT 13 OF C.SWR = 1.
7594                          ;*STALL1 CONTAINS SPECIFIED TIME FOR TESTS 0-6 AND STALL2
7595                          ;*CONTAINS THE TIME FOR TESTS 13-16.
7596                          ;*CALL
7597                          ;*      JSR     R0,@#STALL
7598                          ;*      TIME POINTER            ;WHERE TO FIND THE STALL TIME
7599
7600  021104  013046          STALL:  MOV     @(R0)+,-(SP)    ;PICKUP STALL TIME
7601  021106  032737  020000  001212          BIT     #SW13,@#C.SWR   ;USE A RANDOM TIME?
7602  021114  001406                          BEQ     1$              ;NO--BRANCH
7603  021116  004737  017166                  JSR     PC,@#$RAND      ;YES--FORM RANDOM NUMBER
7604  021122  013716  017314                  MOV     @#$LONUM,(SP)   ;AND USE IT FOR THE STALL TIME
7605  021126  042716  177700                  BIC     #^C77,(SP)      ;BUT NEVER > 64 MILLISECONDS
7606  021132  005046                  1$:     CLR     -(SP)           ;CLEAR TEMP. LOCATION
7607  021134  162766  000001  000002  2$:     SUB     #1,2(SP)        ;MORE STALL REQUIRED?
7608  021142  103407                          BLO     4$              ;NO--BRANCH
7609  021144  012716  000144                  MOV     #^D100,(SP)     ;STALL FOR ABOUT 1 MILLISECOND
7610  021150  005700                  3$:     TST     R0              ;NOP TO KILL TIME
```

```
7611  021152  005366  000000                DEC     0(SP)           ;COUNT
7612  021156  001374                      BNE     3$              ;LOOP IF MORE COUNTS NEEDED
7613  021160  000765                      BR      2$
7614  021162  022626              4$:     CMP     (SP)+,(SP)+     ;CLEAN OFF THE STACK
7615  021164  000200                      RTS     R0              ;EXIT
7616
7617
7618                                  ;*******************************************************************
7619                                  .SBTTL  TWOMS - STALL FOR 2 MS BETWEEN SEEKS IN TESTS 10- - 12
7620                                  ;*ROUTINE TO PROVIDE A 2 MS STALL AFTER A SEEK OPERATION IN THE SEEK TIMING
7621                                  ;*TESTS (TESTS 10, 11, & 12).  THIS STALL IS SPECIFIED BY THE VENDOR (ISS)
7622                                  ;*OF THE DISK DRIVE.  THIS STALL TIME IS NOT INCLUDED IN THE CALCULATED
7623                                  ;*SEEK TIMES.
7624                                  ;*CALL
7625                                  ;*      JSR     PC,@#TWOMS
7626                                  ;*      RETURN
7627
7628  021166  013746  177776          TWOMS:  MOV     @#PS,-(SP)      ;SAVE THE PRESENT PROCESSOR STATUS
7629  021172  012737  000240  177776          MOV     #(5*32.),@#PS   ;SET THE PROCESSOR PRIORITY TO 5
7630  021200  017746  160130                 MOV     @PKV,-(SP)      ;SAVE THE OLD CLOCK VECTOR ADDRESS
7631  021204  012777  021230  160122         MOV     #1$,@PKV        ;SETUP NEW VECTOR ADDRESS
7632  021212  012777  000310  160122         MOV     #200.,@PKB      ;LOAD THE CLOCK BUFFER
7633  021220  012777  000101  160112         MOV     #101,@PKCS      ;START THE CLOCK
7634  021226  000001                         WAIT                    ;WAIT FOR 2 MS
7635  021230  062706  000004          1$:    ADD     #4,SP           ;INCREMENT STACK FOR RETURN
7636  021234  012677  160074                 MOV     (SP)+,@PKV      ;RESTORE OLD CLOCK VECTOR
7637  021240  012637  177776                 MOV     (SP)+,@#PS      ;RESTORE THE OLD PROCESSOR STATUS
7638  021244  000207                         RTS     PC              ;RETURN
7639
7640                                  ;*******************************************************************
7641                                  .SBTTL  VERIFY - VALIDATE HEADER ON IMPLIED SEEKS
7642                                  ;*ROUTINE TO SOFTWARE COMPARE HEADER ON IMPLIED SEEKS
7643                                  ;*CALL
7644                                  ;*      JSR     R0,@#VERIFY
7645                                  ;*      ADR POINTER             ;ADDRESS OF DPB+10 (SECTOR NUMBER)
7646                                  ;*      RETURN
7647
7648  021246  010146                  VERIFY: MOV     R1,-(SP)        ;SAVE R1
7649  021250  012001                         MOV     (R0)+,R1        ;GET ADDRESS OF DPB+10
7650  021252  042737  010000  040524         BIC     #FMT22,@#BUFFER ;STRIP FORMAT BIT FROM CYLINDER NUMBER
7651  021260  023761  040524  000002         CMP     @#BUFFER,2(R1)  ;CYLINDER NUMBER OK?
7652  021266  001003                         BNE     1$              ;NO--BRANCH
7653  021270  023711  040526                 CMP     @#BUFFER+2,(R1) ;YES--HOW ABOUT TRACK/SECTOR?
7654  021274  001426                         BEQ     3$              ;BRANCH IF GOOD
7655  021276  013737  040524  001244  1$:    MOV     @#BUFFER,@#CYL.RD ;SAVE THE EXPECTED AND THE
7656  021304  113737  040527  001246         MOVB    @#BUFFER+3,@#TRK.RD ;RECIEVED CYLINDER, TRACK,
7657  021312  113737  040526  001250         MOVB    @#BUFFER+2,@#SEC.RD ;AND SECTOR
7658  021320  112137  001254                 MOVB    (R1)+,@#SEC.DS
7659  021324  112137  001256                 MOVB    (R1)+,@#TRK.DS
7660  021330  011137  001252                 MOV     (R1),@#CYL.DS
7661  021334  012737  021346  001200         MOV     #2$,$ESCAPE     ;;ESCAPE TO 2$ ON ERROR
7662  021342  005740                         TST     -(R0)           ;MAKE IT TEST PC+4
7663  021344  104012                         ERROR   12              ;REPORT THE ERROR
7664  021346  013700  001234          2$:    MOV     @#BYPASS,R0     ;TAKE ERROR EXIT
```

```
 7665  021352  012601                    3$:     MOV     (SP)+,R1        ;RESTORE R1
 7666  021354  000200                            RTS     R0              ;EXIT
 7667
 7668                                    ;*******************************************************************
 7669                                    .SBTTL  SRCH00 - INITIALIZE THE DRIVE FOR THE TIMING TESTS
 7670                                    ;*THIS ROUTINE WILL PERFORM A "MASSBUS" INIT. FOLLOWED BY
 7671                                    ;*A "RECALIBRATE" ON THE DRIVE UNDER TEST.
 7672                                    ;*NOTE: THIS ROUTINE DESTROYS R1 AND R4
 7673                                    ;*CALL
 7674                                    ;*      JSR     R0,SRCH00       ;DO A MASSBUS INIT. AND RECAL
 7675                                    ;*      RETURN1                 ;RETURN HERE IF NO ERROR
 7676                                    ;*      RETURN2                 ;RETURN HERE ON ERROR
 7677
 7678  021356  005001                    SRCH00: CLR     R1              ;INCASE OF ERROR (TYPTIM)
 7679  021360  005037  177776                    CLR     @#PS
 7680  021364  012777  031356  005576            MOV     #ISR,@RPVEC     ;SETUP INTERRUPT VECTOR
 7681  021372  013704  027166                    MOV     @#RPADR,R4      ;PICKUP ADDRESS OF RHCS1
 7682  021376  012764  000040  000010            MOV     #BIT05,RHCS2(R4) ;MASSBUS INIT.
 7683  021404  005037  003530                    CLR     @#DTADPB+10     ;TRACK=0; SECTOR=0
 7684  021410  005037  003532                    CLR     @#DTADPB+12     ;CYLINDER =0
 7685  021414  012737  000107  003522            MOV     #RECAL,@#DTADPB+2 ;COMMAND = RECALIBRATE
 7686  021422  005037  001200                    CLR     @#$ESCAPE       ;NO ESCAPE ADDRESS
 7687  021426  004037  027574                    JSR     R0,@#RP04       ;CALL THE DRIVER
 7688  021432  003520                            DTADPB                  ;DPB POINTER
 7689  021434  000440                            BR      4$              ;QUEUE IS FULL
 7690  021436  005737  003536            1$:     TST     DTADPB+16       ;WAIT ON DONE
 7691  021442  001775                            BEQ     1$
 7692  021444  100030                            BPL     3$              ;TAKE NORMAL EXIT IF NO ERROR
 7693  021446  012737  021522  001200            MOV     #2$,$ESCAPE     ;;ESCAPE TO 2$ ON ERROR
 7694  021454  013737  003532  001252            MOV     @#DTADPB+12,@#CYL.DS ;CYLINDER
   (1) 021462  113737  003531  001256            MOVB    @#DTADPB+11,@#TRK.DS ;TRACK
   (1) 021470  113737  003530  001254            MOVB    @#DTADPB+10,@#SEC.DS ;SECTOR
   (1) 021476  012746  003536                    MOV     #DTADPB+16,-(SP) ;STATUS/ERROR INDICATOR ADDRESS
   (1) 021502  004737  020766                    JSR     PC,@#ERINDX     ;FORM DISPATCH INDEX
   (1) 021506  062607                            ADD     (SP)+,PC        ;REPORT PROPER ERROR
   (1) 021510  104041                            ERROR   41              ;
   (1) 021512  104042                            ERROR   42              ;PARITY ERROR
   (1) 021514  104043                            ERROR   43              ;UNSAFE ERROR
   (1) 021516  104044                            ERROR   44              ;NON-I/O ERROR
   (1) 021520  104045                            ERROR   45              ;I/O ERROR
 7695  021522  005720                    2$:     TST     (R0)+           ;ADJUST FOR ERROR EXIT
 7696  021524  000404                            BR      4$              ;GO TO THE EXIT
 7697  021526  005064  000006            3$:     CLR     RHDA(R4)        ;TRACK AND SECTOR = 0
 7698  021532  005064  000034                    CLR     RHCA(R4)        ;CYLINDER = 0
 7699  021536  000200                    4$:     RTS     R0              ;RETURN
 7700
 7701                                    ;*******************************************************************
 7702                                    .SBTTL  DORTI - RETURN FROM INTERRUPT
 7703                                    ;*THIS IS AN RTI WHICH IS USED BY THE TIMING TESTS
 7704
 7705  021540  000002                    DORTI:  RTI                     ;RETURN FROM INTERRUPT
 7706
 7707                                    ;*******************************************************************
 7708                                    .SBTTL  STRTMR - START THE TIMERS
```

```
7709                                    ;*THIS ROUTINE WILL INITIALIZE THE TIMERS
7710                                    ;*USED BY THE "TIMING ROUTI ES
7711                                    ;*CALL
7712                                    ;*      JSR     PC,@#STRTMR
7713                                    ;*      RETURN
7714
7715  021542  104416                    STRTMR: SAVREG                  ;SAVE R0-R5
7716  021544  012700  001260                    MOV     #TIM.UP,R0      ;START AT TIM.UP (MINIMUM)
7717  021550  012701  001314                    MOV     #TIM.PT,R1      ;STOP AT TIM.PT
7718  021554  005020                    1$:     CLR     (R0)+           ;CLEAR
7719  021556  020001                            CMP     R0,R1           ;DONE?
7720  021560  103775                            BLO     1$              ;NO--BRANCH
7721  021562  012710  040524                    MOV     #BUFFER,(R0)    ;SETUP POINTER
7722  021566  012737  077777  001260            MOV     #^CBIT15,@#TIM.UP ;SET MINIMUM TIME TO MAXIMUM
7723  021574  012737  077777  001276            MOV     #^CBIT15,@#TIM.DN ;POSITIVE NUMBER
7724  021602  104420                            RESREG                  ;RESTORE R0-R5
7725  021604  000207                            RTS     PC              ;RETURN
7726
7727                                    ;******************************************************************
7728                                    .SBTTL  COUNT - THIS ROUTINE COUNTS THE ELAPSED TIME
7729                                    ;*THIS ROUTINE WILL ADD THE ELAPSED TIME TO THE AVERAGE COUNTER AND
7730                                    ;*MAINTAIN THE MINIMUM AND MAXIMUM TIMES.
7731                                    ;*NOTE: THIS ROUTINE DESTROYS R2
7732                                    ;*CALL
7733                                    ;*      MOV     #TP,R3          ;PARAMETER POINTER
7734                                    ;*      MOV     FLAG,R5         ;FLAG=0=COUNT UP
7735                                    ;*                              ;FLAG=-1=COUNT DOWN
7736                                    ;*      JSR     PC,@#COUNT
7737                                    ;*      RETURN
7738
7739  021606  012702  001260            COUNT:  MOV     #TIM.UP,R2      ;PICKUP THE "UP" POINTER
7740  021612  005705                            TST     R5              ;USE IT?
7741  021614  001402                            BEQ     1$              ;YES--BRANCH
7742  021616  012702  001276                    MOV     #TIM.DN,R2      ;NO--PICKUP "DOWN" POINTER
7743  021622  027722  157516            1$:     CMP     @PKC,(R2)+      ;LESS THAN PREVIOUS LOW?
7744  021626  002003                            BGE     2$              ;NO--BRANCH
7745  021630  017762  157510  177776            MOV     @PKC,-2(R2)     ;YES--SAVE IT
7746  021636  027763  157502  000004    2$:     CMP     @PKC,4(R3)      ;LESS THAN THE LOW LIMIT?
7747  021644  002001                            BGE     3$              ;NO--BRANCH
7748  021646  005212                            INC     (R2)            ;YES--COUNT IT
7749  021650  005722                    3$:     TST     (R2)+           ;ADVANCE THE POINTER
7750  021652  027722  157466                    CMP     @PKC,(R2)+      ;GREATER THAN PREVIOUS HIGH?
7751  021656  003403                            BLE     4$              ;NO--BRANCH
7752  021660  017762  157460  177776            MOV     @PKC,-2(R2)     ;YES--SAVE IT
7753  021666  027763  157452  000006    4$:     CMP     @PKC,6(R3)      ;GREATER THAN THE HIGH LIMIT?
7754  021674  003401                            BLE     5$              ;NO--BRANCH
7755  021676  005212                            INC     (R2)            ;YES--COUNT IT
7756  021700  005722                    5$:     TST     (R2)+           ;ADVANCE THE POINTER
7757  021702  067722  157436                    ADD     @PKC,(R2)+      ;ADD THIS COUNT TO THE TOTAL
7758  021706  005522                            ADC     (R2)+
7759  021710  005212                            INC     (R2)            ;COUNT THIS READING
7760  021712  022737  043674  001314            CMP     #BUFFER+<4*410.>,@#TIM.PT ;SAVE THIS COUNT?
7761  021720  101406                            BLOS    6$              ;NO--BRANCH
7762  021722  017777  157416  157364            MOV     @PKC,@TIM.PT    ;YES--WELL SAVE IT THEN
```

```
7763  021730  062737  000002  001314          ADD     #2,@#TIM.PT     ;ADVANCE THE POINTER
7764  021736  000207                  6$:     RTS     PC              ;RETURN
7765
7766                                  ;*****************************************************************
7767                                  .SBTTL  TYPTIM - TYPE TIMES
7768                                  ;*THIS ROUTINE IS USED TO TYPE THE MINIMUM,
7769                                  ;*MAXIMUM, AND AVERAGE TIMES FOR TESTS 7,10,11, AND 12.
7770                                  ;*IT WILL ALSO CHECK THE TIMES TO INSURE
7771                                  ;*THEY ARE WITHIN TOLERANCE AND IF NOT FLAG THE BAD TIMES.
7772                                  ;*NOTE: THIS ROUTINE DESTROYS R2-R5
7773                                  ;*CALL
7774                                  ;*      JSR     R0,@#TYPTIM     ;GO REPORT THE TIMES
7775                                  ;*      TABLE                   ;POINT TO THE PROPER TABLE
7776                                  ;*      RETURN
7777                                  ;*
7778                                  ;*TABLE:MSGADR1                 ;ADDRESS OF ASCIZ MESSAGE NUMBER 1
7779                                  ;*      MSGADR2                 ;ADDRESS OF ASCIZ MESSAGE NUMBER 2
7780                                  ;*      MIN.ALLOWED             ;MINIMUM TIME ALLOWED
7781                                  ;*      MAX.ALLOWED             ;MAXIMUM TIME ALLOWED
7782
7783  021740  012002                  TYPTIM: MOV     (R0)+,R2        ;PICKUP THE TABLE POINTER
7784  021742  032737  000100  177570          BIT     #SW06,@#SWR     ;INHIBIT TIME REPORTS?
7785  021750  001145                          BNE     7$              ;YES--BRANCH
7786  021752  012237  021772                  MOV     (R2)+,2$        ;ADDRESS OF MESSAGE NUMBER 1
7787  021756  012205                          MOV     (R2)+,R5        ;ADDRESS OF MESSAGE NUMBER 2
7788  021760  012203                          MOV     (R2)+,R3        ;PICKUP THE LOW LIMIT
7789  021762  011202                          MOV     (R2),R2         ;AND THE HIGH LIMIT
7790  021764  012704  001260                  MOV     #TIM.UP,R4      ;PARAMETER POINTER
7791  021770  104400                  1$:     TYPE                    ;TYPE THE MESSAGE
7792  021772  000000                  2$:     .WORD   0               ;ASCIZ MESSAGE POINTER GOES HERE
7793  021774  005764  000014                  TST     14(R4)          ;DID ANY COUNTS OCCUR?
7794  022000  001527                          BEQ     6$              ;NO--BRANCH
7795  022002  104400  035042                  TYPE    ,MSGMIN         ;"MIN="
7796  022006  012446                          MOV     (R4)+,-(SP)             ;PUT (R4)+ ON THE STACK
 (1)  022010  004737  016460                  JSR     PC,@#$B2D       ;CHANGE (R4)+ TO DECIMAL ASCIZ
 (1)  022014  004737  016704                  JSR     PC,@#$SUPRS     ;TYPE WITHOUT LEADING ZEROS
7797  022020  104400  035067                  TYPE    ,MSGOUS         ;"0 US"
7798  022024  005724                          TST     (R4)+           ;ANY SEEKS BELOW THE LOW LIMIT
7799  022026  001421                          BEQ     3$              ;NO--BRANCH
7800  022030  104400  035202                  TYPE    ,MSG.SP         ;" "
7801  022034  016446  177776                  MOV     -2(R4),-(SP)            ;PUT -2(R4) ON THE STACK
 (1)  022040  004737  016460                  JSR     PC,@#$B2D       ;CHANGE -2(R4) TO DECIMAL ASCIZ
 (1)  022044  004737  016704                  JSR     PC,@#$SUPRS     ;TYPE WITHOUT LEADING ZEROS
7802  022050  104400  035074                  TYPE    ,MBELOW         ;"BELOW THE MINIMUM OF"
7803  022054  010346                          MOV     R3,-(SP)                ;PUT R3 ON THE STACK
 (1)  022056  004737  016460                  JSR     PC,@#$B2D       ;CHANGE R3 TO DECIMAL ASCIZ
 (1)  022062  004737  016704                  JSR     PC,@#$SUPRS     ;TYPE WITHOUT LEADING ZEROS
7804  022066  104400  035067                  TYPE    ,MSGOUS
7805  022072  104400  035051          3$:     TYPE    ,MSGMAX         ;"MAX="
7806  022076  012446                          MOV     (R4)+,-(SP)             ;PUT (R4)+ ON THE STACK
 (1)  022100  004737  016460                  JSR     PC,@#$B2D       ;CHANGE (R4)+ TO DECIMAL ASCIZ
 (1)  022104  004737  016704                  JSR     PC,@#$SUPRS     ;TYPE WITHOUT LEADING ZEROS
7807  022110  104400  035067                  TYPE    ,MSGOUS
7808  022114  005724                          TST     (R4)+           ;ANY SEEKS ABOVE THE HIGH LIMIT
```

```
7809  022116  001421                        BEQ     4$              ;NO--BRANCH
7810  022120  104400  035202                TYPE    ,MSG.SP         ;YES--REPORT HOW MANY
7811  022124  016446  177776                MOV     -2(R4),-(SP)            ;PUT -2(R4) ON THE STACK
 (1)  022130  004737  016460                JSR     PC,@#$B2D       ;CHANGE -2(R4) TO DECIMAL ASCIZ
 (1)  022134  004737  016704                JSR     PC,@#$SUPRS     ;TYPE WITHOUT LEADING ZEROS
7812  022140  104400  035123                TYPE    ,MABOVE         ;"ABOVE THE MAXIMUM OF"
7813  022144  010246                        MOV     R2,-(SP)                ;PUT R2 ON THE STACK
 (1)  022146  004737  016460                JSR     PC,@#$B2D       ;CHANGE R2 TO DECIMAL ASCIZ
 (1)  022152  004737  016704                JSR     PC,@#$SUPRS     ;TYPE WITHOUT LEADING ZEROS
7814  022156  104400  035067                TYPE    ,MSGOUS
7815  022162  104400  035060        4$:     TYPE    ,MSGAVG         ;"AVG="
7816  022166  012446                        MOV     (R4)+,-(SP)     ;FORM THE AVERAGE
7817  022170  012446                        MOV     (R4)+,-(SP)
7818  022172  012446                        MOV     (R4)+,-(SP)
7819  022174  004737  016744                JSR     PC,@#$DIV
7820  022200  006126                        ROL     (SP)+           ;IS THE REMAINDER OVER HALF?
7821  022202  100001                        BPL     5$              ;NO--BRANCH
7822  022204  005216                        INC     (SP)            ;YES--ROUND UP
7823  022206                        5$:
 (1)  022206  004737  016460                JSR     PC,@#$B2D       ;CHANGE  TO DECIMAL ASCIZ
 (1)  022212  004737  016704                JSR     PC,@#$SUPRS     ;TYPE WITHOUT LEADING ZEROS
7824  022216  104400  035067                TYPE    ,MSGOUS
7825  022222  104400  035202                TYPE    ,MSG.SP
7826  022226  016446  177776                MOV     -2(R4),-(SP)            ;PUT -2(R4) ON THE STACK
 (1)  022232  004737  016460                JSR     PC,@#$B2D       ;CHANGE -2(R4) TO DECIMAL ASCIZ
 (1)  022236  004737  016704                JSR     PC,@#$SUPRS     ;TYPE WITHOUT LEADING ZEROS
7827  022242  104400  035152                TYPE    ,MSGNUM         ;"SEEKS TIMED"
7828  022246  010537  021772                MOV     R5,2$           ;NEXT MESSAGE POINTER
7829  022252  001404                        BEQ     7$              ;IF NONE EXIT
7830  022254  005005                        CLR     R5              ;NO MORE THAN 2
7831  022256  000644                        BR      1$
7832  022260  104400  035167        6$:     TYPE    ,MSGNON
7833  022264  000200        7$:     RTS     R0                      ;EXIT
7834
7835                        ;*****************************************************************
7836                        .SBTTL  INCTRK - INCREMENT TRACK NUMBER
7837                        ;*THIS SUBROUTINE WILL INCREMENT THE TRACK
7838                        ;*NUMBER (R2) BY THE AMOUNT SPECIFIED BY IT15.
7839                        ;*CALL
7840                        ;*      JSR     R0,@#INCTRK
7841                        ;*      RETURN1                 ;TRACK NUMBER GREATER THAN LT15
7842                        ;*      RETURN2                 ;TRACK NUMBER INCREMENTED
7843
7844  022266  020237  002306        INCTRK: CMP     R2,@#LT15       ;LAST TRACK COMPLETED?
7845  022272  001410                        BEQ     2$              ;YES--EXIT
7846  022274  063702  002310                ADD     @#IT15,R2       ;NO--UPDATE TRACK
7847  022300  020237  002306                CMP     R2,@#LT15       ;TRACK TO BIG?
7848  022304  003402                        BLE     1$              ;NO--EXIT
7849  022306  013702  002306                MOV     @#LT15,R2       ;YES--SET TRACK TO LAST TRACK
7850  022312  005720        1$:     TST     (R0)+           ;ADJUST FOR RETURN 2
7851  022314  000200        2$:     RTS     R0              ;RETURN
7852
7853
7854                        ;*****************************************************************
```

```
                                        .SBTTL  INCCYL - INCRMENT CYLINDER NUMBER
                                        ;*THIS SUBROUTINE WILL INCREMENT THE CYLINDER
                                        ;*NUMBER (R1) BY THE AMOUNT SPECIFIED BY IC15.
                                        ;*CALL
                                        ;*      JSR     R0,@#INCCYL
                                        ;*      RETURN1                 ;CYLINDER NUMBER GREATER THAN LC15
                                        ;*      RETURN2                 ;CYLINDER NUMBER INCREMENTED
022316  020137  002300          INCCYL: CMP     R1,@#LC15               ;LAST CYLINDER COMPLETED?
022322  001410                          BEQ     2$                      ;YES--EXIT
022324  063701  002302                  ADD     @#IC15,R1               ;NO--UPDATE CYLINDER
022330  020137  002300                  CMP     R1,@#LC15               ;CYLINDER TO BIG?
022334  003402                          BLE     1$                      ;NO--EXIT
022336  013701  002300                  MOV     @#LC15,R1               ;YES--SET CYLINDER TO LAST CYLINDER
022342  005720                  1$:     TST     (R0)+                   ;ADJUST FOR RETURN 2
022344  000200                  2$:     RTS     R0                      ;RETURN

                                        ;*****************************************************************
                                        .SBTTL  FILBUF - FILL BUFFER WITH ADDRESS DATA
                                        ;*THIS SUBROUTINE IS USED TO FILL THE DATA BUFFER
                                        ;*WITH ADDRESSES FROM 0 TO 21 WITH EACH ADDRESS
                                        ;*BEING STORED IN 256 CONSECUTIVE LOCATIONS
                                        ;*CALL
                                        ;*      JSR     PC,@#FILBUF
                                        ;*      RETURN
022346  004037  034334          FILBUF: JSR     R0,@#SAVR05             ;SAVE R0-R5
022352  005000                          CLR     R0                      ;FIRST DISK ADDRESS
022354  012701  040524                  MOV     #BUFFER,R1              ;START FILLING HERE
022360  012702  000400          1$:     MOV     #^D256,R2               ;DO 256 WORDS
022364  010021                  2$:     MOV     R0,(R1)+                ;STORE
022366  005302                          DEC     R2                      ;MORE?
022370  003375                          BGT     2$                      ;YES--BRANCH
022372  005200                          INC     R0                      ;NO--UPDATE DISK ADDRESS
022374  022700  000026                  CMP     #^D22,R0                ;DONE?
022400  003367                          BGT     1$                      ;NO--BRANCH
022402  004037  034360                  JSR     R0,@#GETR05             ;RESTORE R0-R5
022406  000207                          RTS     PC                      ;RETURN

                                        ;*****************************************************************
                                        .SBTTL  CLRBUF - CLEAR BUFFER
                                        ;*THIS ROUTINE WILL CLEAR THE BUFFER BY
                                        ;*SETTING EACH WORD TO "177400".
                                        ;*CALL
                                        ;*      JSR     R0,@#CLRBUF
                                        ;*      RETURN

022410  004037  034334          CLRBUF: JSR     R0,@#SAVR15             ;SAVE R1-R5
022414  012701  177400                  MOV     #177400,R1              ;WORD TO FILL BUFFER WITH
022420  012702  040524                  MOV     #BUFFER,R2              ;FIRST ADDRESS OF BUFFER
022424  012703  066524                  MOV     #BUFFER+<512.*22.>,R3 ;LAST ADDRESS+2 OF BUFFER
022430  010122                  1$:     MOV     R1,(R2)+                ;FILL WORDS 1, 9,...249,...5625
022432  010122                          MOV     R1,(R2)+                ;FILL WORDS 2,10,...250,...5626
022434  010122                          MOV     R1,(R2)+                ;FILL WORDS 3,11,...251,...5627
```

```
 7909  022436  010122                        MOV     R1,(R2)+        ;FILL WORDS 4,12,...252,...5628
 7910  022440  010122                        MOV     R1,(R2)+        ;FILL WORDS 5,13,...253,...5629
 7911  022442  010122                        MOV     R1,(R2)+        ;FILL WORDS 6,14,...254,...5630
 7912  022444  010122                        MOV     R1,(R2)+        ;FILL WORDS 7,15,...255,...5631
 7913  022446  010122                        MOV     R1,(R2)+        ;FILL WORDS 8,16,...256,...5632
 7914  022450  020203                        CMP     R2,R3           ;DONE?
 7915  022452  103766                        BLO     1$              ;NO--BRANCH
 7916  022454  004037  034354                JSR     R0,@#GETR15     ;RESTORE R1-R5
 7917  022460  000200                        RTS     R0              ;RETURN FROM CALL
 7918
 7919                          ;*****************************************************************
 7920                          .SBTTL  CKSCTR - CHECK SECTOR DATA
 7921                          ;*THIS ROUTINE IS USED TO CHECK THE DATA BUFFER
 7922                          ;*FOR ADDRESSES 0 THROUGH 21 WITH EACH ADDRESS
 7923                          ;*BEING STORED IN 256 CONSECUTIVE LOCATIONS
 7924                          ;*CALL
 7925                          ;*      JSR     R0,@#CKSCTR
 7926                          ;*      RETURN
 7927
 7928  022462  004037  034334  CKSCTR: JSR     R0,@#SAVR15     ;SAVE R1-R5
 7929  022466  162706  000004          SUB     #4,SP           ;RESERVE TEMP. STORAGE AREA
 7930  022472  005001                   CLR     R1              ;FIRST SECTOR
 7931  022474  012716  040524          MOV     #BUFFER,(SP)    ;FIRST ADDRESS OF DATA BUFFER
 7932  022500  005066  000002          CLR     2(SP)           ;NO ERRORS
 7933  022504  012702  000020  1$:     MOV     #^D16,R2        ;LOOP COUNT (16*16=256)
 7934  022510  011603                   MOV     (SP),R3         ;GET 1ST ADDRESS OF THIS SECTORS DATA
 7935  022512                  2$:
 7938  022512  020123                   CMP     R1,(R3)+        ;WORD 1
  (1)  022514  001064                   BNE     7$              ;BRANCH IF BAD
  (1)  022516  020123                   CMP     R1,(R3)+        ;WORD 2
  (1)  022520  001062                   BNE     7$              ;BRANCH IF BAD
  (1)  022522  020123                   CMP     R1,(R3)+        ;WORD 3
  (1)  022524  001060                   BNE     7$              ;BRANCH IF BAD
  (1)  022526  020123                   CMP     R1,(R3)+        ;WORD 4
  (1)  022530  001056                   BNE     7$              ;BRANCH IF BAD
  (1)  022532  020123                   CMP     R1,(R3)+        ;WORD 5
  (1)  022534  001054                   BNE     7$              ;BRANCH IF BAD
  (1)  022536  020123                   CMP     R1,(R3)+        ;WORD 6
  (1)  022540  001052                   BNE     7$              ;BRANCH IF BAD
  (1)  022542  020123                   CMP     R1,(R3)+        ;WORD 7
  (1)  022544  001050                   BNE     7$              ;BRANCH IF BAD
  (1)  022546  020123                   CMP     R1,(R3)+        ;WORD 8
  (1)  022550  001046                   BNE     7$              ;BRANCH IF BAD
  (1)  022552  020123                   CMP     R1,(R3)+        ;WORD 9
  (1)  022554  001044                   BNE     7$              ;BRANCH IF BAD
  (1)  022556  020123                   CMP     R1,(R3)+        ;WORD 10
  (1)  022560  001042                   BNE     7$              ;BRANCH IF BAD
  (1)  022562  020123                   CMP     R1,(R3)+        ;WORD 11
  (1)  022564  001040                   BNE     7$              ;BRANCH IF BAD
  (1)  022566  020123                   CMP     R1,(R3)+        ;WORD 12
  (1)  022570  001036                   BNE     7$              ;BRANCH IF BAD
  (1)  022572  020123                   CMP     R1,(R3)+        ;WORD 13
  (1)  022574  001034                   BNE     7$              ;BRANCH IF BAD
  (1)  022576  020123                   CMP     R1,(R3)+        ;WORD 14
```

```
 (1)  022600  001032                        BNE     7$              ;BRANCH IF BAD
 (1)  022602  020123                        CMP     R1,(R3)+        ;WORD 15
 (1)  022604  001030                        BNE     7$              ;BRANCH IF BAD
 (1)  022606  020123                        CMP     R1,(R3)+        ;WORD 16
 (1)  022610  001026                        BNE     7$              ;BRANCH IF BAD
7939  022612  005302                        DEC     R2              ;FINISHED WITH THIS SECTORS DATA?
7940  022614  001336                        BNE     2$              ;NO--BRANCH
7941  022616  062716  001000          3$:   ADD     #^D512,(SP)     ;YES--FIRST ADDRESS OF NEXT SECTOR
7942  022622  005201                        INC     R1              ;MOVE TO NEXT SECTOR
7943  022624  022701  000026                CMP     #^D22,R1        ;DONE?
7944  022630  003325                        BGT     1$              ;NO--BRANCH
7945  022632  005766  000002          4$:   TST     2(SP)           ;ERROR OCCUR?
7946  022636  001406                        BEQ     6$              ;NO--BRANCH
7947  022640  123737  001372  001103        CMPB    @#ERR.CT,@#$ERFLG ;MAX. ERROR OCCURRED?
7948  022646  101002                        BHI     6$              ;NO--BRANCH
7949  022650  013700  001234          5$:   MOV     @#BYPASS,R0     ;TAKE ERROR EXIT
7950  022654  062706  000004          6$:   ADD     #4,SP           ;FREE TEMP. AREA
7951  022660  004037  034354                JSR     R0,@#GETR15     ;RESTORE R1-R5
7952  022664  000200                        RTS     R0              ;RETURN FROM CALL
7953  022666  010304                  7$:   MOV     R3,R4           ;FORM WORD NUMBER AND
7954  022670  161604                        SUB     (SP),R4         ;ADDRESS TO CONTINUE FROM
7955  022672  010405                        MOV     R4,R5
7956  022674  006204                        ASR     R4              ;WORD NUMBER
7957  022676  042705  177740                BIC     #^C37,R5
7958  022702  001002                        BNE     8$              ;BRANCH IF NOT A MULTIPLE OF 16
7959  022704  012705  000040                MOV     #40,R5          ;SET TO WORD 16
7960  022710  006305                  8$:   ASL     R5
7961  022712  062705  022512                ADD     #2$,R5          ;ADDRESS
7962  022716  016337  177776  001126        MOV     -2(R3),@#$BDDAT ;SAVE BAD DATA
7963  022724  005766  000002                TST     2(SP)           ;FIRST ERROR?
7964  022730  001015                        BNE     10$             ;NO--BRANCH
7965  022732  013737  003532  001252        MOV     @#DTADPB+12,@#CYL.DS ;CYLINDER NUMBER
7966  022740  113737  003531  001256        MOVB    @#DTADPB+11,@#TRK.DS ;TRACK NUMBER
7967  022746  012737  022756  001200        MOV     #9$,$ESCAPE     ;;ESCAPE TO 9$ ON ERROR
7968  022754  104021                        ERROR   21              ;REPORT THE ERROR
7969  022756  105166  000002          9$:   COMB    2(SP)           ;SET ERROR SWITCH
7970  022762  000404                        BR      11$
7971  022764                          10$:
 (1)  022764  012737  022774  001200        MOV     #11$,$ESCAPE    ;;ESCAPE TO 11$ ON ERROR
7972  022772  104022                        ERROR   22              ;REPORT THE ERROR
7973  022774  032737  001000  177570  11$:  BIT     #SW09,@#SWR     ;LOOP ON ERROR?
7974  023002  001322                        BNE     5$              ;YES
7975  023004  032737  000002  177570        BIT     #SW01,@#SWR     ;STOP DATA COMPARE?
7976  023012  001307                        BNE     4$              ;YES--BRANCH
7977  023014  123737  001372  001103        CMPB    @#ERR.CT,@#$ERFLG ;MAX. ERRORS?
7978  023022  101712                        BLOS    5$              ;YES--BRANCH
7979  023024  032737  000040  177570        BIT     #SW05,@#SWR     ;REPORT ONLY 1ST ERROR PER SECTOR?
7980  023032  001271                        BNE     3$              ;YES--BRANCH
7981  023034  000115                        JMP     (R5)
7982
7983                          ;*****************************************************************
7984                          .SBTTL  SETBUF - SET BUFFER TO DATA PATTERN
7985                          ;*THIS ROUTINE WILL MOVE THE 16 WORDS OF THE
7986                          ;*DESIRED PATTERN INTO THE DATA BUFFER.
```

```
                                        ;*CALL
                                        ;*      MOV     #NX,R0          ;PATTERN NUMBER INDEX TO R0
                                        ;*      JSR     PC,@#SETBUF
                                        
023036  004037  034334          SETBUF: JSR     R0,@#SAVR15     ;SAVE R1-R5
023042  012701  040524                  MOV     #BUFFER,R1      ;FIRST ADDRESS
023046  013702  003524                  MOV     @#DTADPB+4,R2   ;WORD COUNT
023052  016003  002400          1$:     MOV     PAT.PT(R0),R3   ;PICKUP PATTERN POINTER
023056  012321                          MOV     (R3)+,(R1)+     ;MOVE WORD 1 INTO DATA BUFFER
(1) 023060  012321                      MOV     (R3)+,(R1)+     ;MOVE WORD 2 INTO DATA BUFFER
(1) 023062  012321                      MOV     (R3)+,(R1)+     ;MOVE WORD 3 INTO DATA BUFFER
(1) 023064  012321                      MOV     (R3)+,(R1)+     ;MOVE WORD 4 INTO DATA BUFFER
(1) 023066  012321                      MOV     (R3)+,(R1)+     ;MOVE WORD 5 INTO DATA BUFFER
(1) 023070  012321                      MOV     (R3)+,(R1)+     ;MOVE WORD 6 INTO DATA BUFFER
(1) 023072  012321                      MOV     (R3)+,(R1)+     ;MOVE WORD 7 INTO DATA BUFFER
(1) 023074  012321                      MOV     (R3)+,(R1)+     ;MOVE WORD 8 INTO DATA BUFFER
(1) 023076  012321                      MOV     (R3)+,(R1)+     ;MOVE WORD 9 INTO DATA BUFFER
(1) 023100  012321                      MOV     (R3)+,(R1)+     ;MOVE WORD 10 INTO DATA BUFFER
(1) 023102  012321                      MOV     (R3)+,(R1)+     ;MOVE WORD 11 INTO DATA BUFFER
(1) 023104  012321                      MOV     (R3)+,(R1)+     ;MOVE WORD 12 INTO DATA BUFFER
(1) 023106  012321                      MOV     (R3)+,(R1)+     ;MOVE WORD 13 INTO DATA BUFFER
(1) 023110  012321                      MOV     (R3)+,(R1)+     ;MOVE WORD 14 INTO DATA BUFFER
(1) 023112  012321                      MOV     (R3)+,(R1)+     ;MOVE WORD 15 INTO DATA BUFFER
(1) 023114  012321                      MOV     (R3)+,(R1)+     ;MOVE WORD 16 INTO DATA BUFFER
7998    023116  062702  000020          ADD     #^D16,R2        ;DONE?
7999    023122  001353                  BNE     1$              ;NO--BRANCH
8000    023124  004037  034354          JSR     R0,@#GETR15     ;YES--RESTORE R1-R5
8001    023130  000207                  RTS     PC              ;RETURN
8002
8003                                    ;******************************************************************
8004                                    .SBTTL  DATCMP - DATA COMPARE
8005                                    ;*THIS ROUTINE COMPARES A 16 WORD DATA PATTERN
8006                                    ;*AGAINST THE DATA BUFFER
8007                                    ;*CALL
8008                                    ;*      MOV     #NX,R0          ;PATTERN NUMBER INDEX TO R0
8009                                    ;*      JSR     PC,@#DATCMP
8010                                    ;*      RETURN
8011
8012    023132  004037  034334  DATCMP: JSR     R0,@#SAVR05     ;SAVE R0-R5
8013    023136  012701  040524          MOV     #BUFFER,R1      ;FIRST ADDRESS OF BUFFER
8014    023142  013702  003524          MOV     @#DTADPB+4,R2   ;WORD COUNT
8015    023146  005046                  CLR     -(SP)           ;NO ERROR
8016    023150  016003  002400  1$:     MOV     PAT.PT(R0),R3   ;PATTERN POINTER
8017    023154                  2$:
8020    023154  162321                  SUB     (R3)+,(R1)+     ;CHECK WORD 1
(1) 023156  001045                      BNE     4$              ;BRANCH IF DIFFERENT
(1) 023160  162321                      SUB     (R3)+,(R1)+     ;CHECK WORD 2
(1) 023162  001043                      BNE     4$              ;BRANCH IF DIFFERENT
(1) 023164  162321                      SUB     (R3)+,(R1)+     ;CHECK WORD 3
(1) 023166  001041                      BNE     4$              ;BRANCH IF DIFFERENT
(1) 023170  162321                      SUB     (R3)+,(R1)+     ;CHECK WORD 4
(1) 023172  001037                      BNE     4$              ;BRANCH IF DIFFERENT
(1) 023174  162321                      SUB     (R3)+,(R1)+     ;CHECK WORD 5
(1) 023176  001035                      BNE     4$              ;BRANCH IF DIFFERENT
```

```
    (1)  023200  162321                    SUB     (R3)+,(R1)+     ;CHECK WORD 6
    (1)  023202  001033                    BNE     4$              ;BRANCH IF DIFFERENT
    (1)  023204  162321                    SUB     (R3)+,(R1)+     ;CHECK WORD 7
    (1)  023206  001031                    BNE     4$              ;BRANCH IF DIFFERENT
    (1)  023210  162321                    SUB     (R3)+,(R1)+     ;CHECK WORD 8
    (1)  023212  001027                    BNE     4$              ;BRANCH IF DIFFERENT
    (1)  023214  162321                    SUB     (R3)+,(R1)+     ;CHECK WORD 9
    (1)  023216  001025                    BNE     4$              ;BRANCH IF DIFFERENT
    (1)  023220  162321                    SUB     (R3)+,(R1)+     ;CHECK WORD 10
    (1)  023222  001023                    BNE     4$              ;BRANCH IF DIFFERENT
    (1)  023224  162321                    SUB     (R3)+,(R1)+     ;CHECK WORD 11
    (1)  023226  001021                    BNE     4$              ;BRANCH IF DIFFERENT
    (1)  023230  162321                    SUB     (R3)+,(R1)+     ;CHECK WORD 12
    (1)  023232  001017                    BNE     4$              ;BRANCH IF DIFFERENT
    (1)  023234  162321                    SUB     (R3)+,(R1)+     ;CHECK WORD 13
    (1)  023236  001015                    BNE     4$              ;BRANCH IF DIFFERENT
    (1)  023240  162321                    SUB     (R3)+,(R1)+     ;CHECK WORD 14
    (1)  023242  001013                    BNE     4$              ;BRANCH IF DIFFERENT
    (1)  023244  162321                    SUB     (R3)+,(R1)+     ;CHECK WORD 15
    (1)  023246  001011                    BNE     4$              ;BRANCH IF DIFFERENT
    (1)  023250  162321                    SUB     (R3)+,(R1)+     ;CHECK WORD 16
    (1)  023252  001007                    BNE     4$              ;BRANCH IF DIFFERENT
 8021    023254  062702  000020            ADD     #^D16,R2        ;DONE ?
 8022    023260  001333                    BNE     1$              ;NO--BRANCH
 8023    023262  005726              3$:   TST     (SP)+           ;YES -- CLEAN UP STACK
 8024    023264  004037  034360            JSR     R0,@#GETRDS     ;RESTORE R0-R5
 8025    023270  000207                    RTS     PC
 8026    023272  010104              4$:   MOV     R1,R4           ;FORM THE WORD NUMBER
 8027    023274  162704  040524            SUB     #BUFFER,R4
 8028    023300  006204                    ASR     R4              ;WORD NUMBER
 8029    023302  010305                    MOV     R3,R5           ;FORM ADDRESS TO CONTINUE FROM
 8030    023304  166005  002400            SUB     PAT.PT(R0),R5
 8031    023310  006305                    ASL     R5
 8032    023312  062705  023154            ADD     #2$,R5          ;ADDRESS
 8033    023316  064341                    ADD     -(R3),-(R1)     ;RECONSTRUCT THE BAD WORD
 8034    023320  010137  001122            MOV     R1,@#$BDADR     ;SAVE THE ERROR INFORMATION
 8035    023324  010337  001120            MOV     R3,@#$GDADR     ;
 8036    023330  012137  001126            MOV     (R1)+,@#$BDDAT  ;
 8037    023334  012337  001124            MOV     (R3)+,@#$GDDAT  ;
 8038    023340  005716                    TST     (SP)            ;1ST DATA COMPARE ERROR?
 8039    023342  001023                    BNE     6$              ;NO--BRANCH
 8040    023344  013737  003532  001252    MOV     @#DTADPB+12,@#CYL.DS ;CYLINDER
 8041    023352  113737  003531  001256    MOVB    @#DTADPB+11,@#TRK.DS ;TRACK
 8042    023360  113737  003530  001254    MOVB    @#DTADPB+10,@#SEC.DS ;SECTOR
 8043    023365  016600  000016            MOV     16(SP),R0       ;GET TEST PC+4
 8044    023372  012737  023402  001200    MOV     #5$,$ESCAPE     ;;ESCAPE TO 5$ ON ERROR
 8045    023400  104013                    ERROR   13              ;REPORT THE ERROR
 8046    023402  016600  000014      5$:   MOV     14(SP),R0       ;PATTERN NUMBER INDEX
 8047    023406  105116                    COMB    (SP)            ;SET THE ERROR SWITCH
 8048    023410  000404                    BR      7$
 8049    023412                      6$:
    (1)  023412  012737  023422  001200    MOV     #7$,$ESCAPE     ;;ESCAPE TO 7$ ON ERROR
 8050    023420  104014                    ERROR   14              ;REPORT THE ERROR
 8051    023422  032737  000002  177570 7$:BIT     #SW01,@#SWR     ;STOP DATA COMPARE?
```

```
 8052  023430  001314                        BNE     3$                ;YES--EXIT
 8053  023432  123737  001372  001103        CMPB    @#ERR.CT,@#$ERFLG ;MAX. ERRORS?
 8054  023440  101004                        BHI     8$                ;NO--BRANCH
 8055  023442  013766  001234  000016        MOV     @#BYPASS,16(SP)   ;YES--ERROR EXIT
 8056  023450  000704                        BR      3$
 8057  023452  000115                8$:     JMP     (R5)              ;NO--CONTINUE AT NEXT WORD
 8058
 8059                                ;*****************************************************************
 8060                                .SBTTL  FILRAN - FILL DATA BUFFER WITH RANDOM PATTERN
 8061                                ;*THIS ROUTINE WILL FILL THE DATA BUFFER (256*22 WORDS) WITH
 8062                                ;*A RANDOM PATTERN.  THE FIRST TWO WORDS OF EVERY 256 WILL
 8063                                ;*BE THE BASE OF THE RANDOM NUMBER GENERATOR FOR THE
 8064                                ;*NEXT 254 WORDS.
 8065                                ;*NOTE: THIS ROUTINE DESTROYS R1 AND R2
 8066                                ;*CALL
 8067                                ;*      JSR     R0,@#FILRAN
 8068                                ;*      RETURN
 8069
 8070  023454  012701  040524        FILRAN: MOV     #BUFFER,R1
 8071  023460  012702  000026                MOV     #^D22,R2
 8072  023464  004037  023674        1$:     JSR     R0,@#RANPAT
 8073  023470  005302                        DEC     R2
 8074  023472  003374                        BGT     1$
 8075  023474  000200                        RTS     R0
 8076
 8077                                ;*****************************************************************
 8078                                .SBTTL  RANCK - RANDOM PATTERN BUFFER CHECK
 8079                                ;*THIS ROUTINE USES THE FIRST TWO WORDS OF THE
 8080                                ;*READ BUFFER TO GENERATED A RANDOM PATTERN.  THEN
 8081                                ;*THE READ BUFFER IS COMPARED TO THE PATTERN GENERATED.
 8082                                ;*NOTE: THIS ROUTINE DESTROYS R1-R4
 8083                                ;*CALL
 8084                                ;*      JSR     R0,@#RANCK
 8085                                ;*      RETURN
 8086
 8087  023476  013746  017312        RANCK:  MOV     @#$HINUM,-(SP)  ;SAVE THE PRESENT RANDOM NUMBER
 8088  023502  013746  017314                MOV     @#$LONUM,-(SP)
 8089  023506  012702  041524                MOV     #BUFFER+512.,R2 ;READ BUFFER ADDRESS
 8090  023512  012701  042524                MOV     #BUFFER+1024.,R1 ;RANDOM PATTERN ADDRESS
 8091  023516  010103                        MOV     R1,R3           ;COPY IT INTO R3 FOR LATER USE
 8092  023520  011237  017314                MOV     (R2),@#$LONUM   ;PRIME THE RANDOM NUMBER GENERATOR
 8093  023524  016237  000002  017312        MOV     2(R2),@#$HINUM
 8094  023532  004037  023674                JSR     R0,@#RANPAT     ;GENERATE A RANDOM PATTERN
 8095  023536  012637  017314                MOV     (SP)+,@#$LONUM  ;RESTORE PRESENT RANDOM NUMBER
 8096  023542  012637  017312                MOV     (SP)+,@#$HINUM
 8097  023546  005046                        CLR     -(SP)           ;NO ERRORS
 8098  023550  162322                1$:     SUB     (R3)+,(R2)+     ;ARE THESE TWO WORDS DIFFERENT?
 8099  023552  001441                        BEQ     4$              ;NO--BRANCH
 8100  023554  012737  023626  001200        MOV     #3$,$ESCAPE     ;;ESCAPE TO 3$ ON ERROR
 8101  023562  064342                        ADD     -(R3),-(R2)     ;RECREATE THE BAD WORD
 8102  023564  010237  001122                MOV     R2,@#$BDADR     ;ADDRESS OF BAD DATA
 8103  023570  010337  001120                MOV     R3,@#$GDADR     ;ADDRESS OF GOOD DATA
 8104  023574  012237  001126                MOV     (R2)+,@#$BDDAT  ;BAD DATA
 8105  023600  012337  001124                MOV     (R3)+,@#$GDDAT  ;GOOD DATA
```

```
8106  023604  010204                          MOV     R2,R4           ;FORM WORD NUMBER (1 TO 256)
8107  023606  162704  041524                  SUB     #BUFFER+512.,R4
8108  023612  006204                          ASR     R4
8109  023614  005716                          TST     (SP)            ;FIRST ERROR
8110  023616  001002                          BNE     2$              ;NO--BRANCH
8111  023620  105116                          COMB    (SP)            ;YES--SET ERROR SWITCH
8112  023622  104015                          ERROR   15              ;REPORT THE ERROR
8113  023624  104016                  2$:     ERROR   16              ;REPORT THE ERROR
8114  023626  032737  001000  177570  3$:     BIT     #SWD9,@#SWR     ;LOOP ON ERROR?
8115  023634  001012                          BNE     5$              ;YES--BRANCH
8116  023636  123737  001372  001103          CMPB    @#ERR.CT,@#$ERFLG ;MAX. ERRORS OCCURRED?
8117  023644  101406                          BLOS    5$              ;YES--BRANCH
8118  023646  032737  000002  177570          BIT     #SW01,@#SWR     ;STOP COMPARING?
8119  023654  001002                          BNE     5$              ;YES--BRANCH
8120  023656  020103                  4$:     CMP     R1,R3           ;ALL DATA BEEN COMPARED?
8121  023660  101333                          BHI     1$              ;NO--BRANCH
8122  023662  005726                  5$:     TST     (SP)+           ;ERROR OCCUR?
8123  023664  001402                          BEQ     6$              ;NO--BRANCH
8124  023666  013700  001234                  MOV     @#BYPASS,R0     ;TAKE ERROR EXIT
8125  023672  000200                  6$:     RTS     R0              ;EXIT
8126
8127                                  ;*****************************************************************
8128                                  .SBTTL  RANPAT - RANDOM PATTERN GENERATOR
8129                                  ;*THIS ROUTINE FILLS A 256 WORD BUFFER WITH A RANDOM
8130                                  ;*PATTERN OF WHICH THE FIRST TWO WORDS ARE THE BASE
8131                                  ;*OF THE PATTERN.
8132                                  ;*CALL
8133                                  ;*      MOV     #ADR,R1         ;ADDRESS OF THE BUFFER
8134                                  ;*      JSR     R0,@#RANPAT
8135                                  ;*      RETURN
8136
8137  023674  010246                  RANPAT: MOV     R2,-(SP)        ;SAVE R2
8138  023676  012702  000200                  MOV     #^D256/2,R2     ;GENERATE 256 WORDS
8139  023702  000402                          BR      2$
8140  023704  004737  017166          1$:     JSR     PC,@#$RAND      ;GENERATE A RANDOM NUMBER
8141  023710  013721  017314          2$:     MOV     @#$LONUM,(R1)+  ;PUT LOW WORD IN BUFFER
8142  023714  013721  017312                  MOV     @#$HINUM,(R1)+  ;PUT HIGH WORD IN BUFFER
8143  023720  005302                          DEC     R2              ;DONE?
8144  023722  003370                          BGT     1$              ;NO--BRANCH
8145  023724  012602                          MOV     (SP)+,R2        ;RESTORE R2
8146  023726  000200                          RTS     R0              ;EXIT
8147
8148                                  ;*****************************************************************
8149                                  .SBTTL  RANADR - RANDOM ADDRESS GENERATOR
8150                                  ;*THIS ROUTINE GENERATES RANDOM CYLINDER, TRACK, AND SECTOR
8151                                  ;*ADDRESSES AND SAVES THEN IN THE DPB (DTADPB+10 AND DTADPB+12).
8152                                  ;*NOTE: THIS ROUTINE DESTROYS R1-R3
8153                                  ;*CALL
8154                                  ;*      JSR     R0,@#RANADR
8155                                  ;*      RETURN
8156
8157  023730  004737  017166          RANADR: JSR     PC,@#$RAND      ;GENERATE A RANDOM NUMBER
8158  023734  113701  017314                  MOVB    @#$LONUM,R1     ;FORM SECTOR IN R1
8159  023740  113702  017315                  MOVB    @#$LONUM+1,R2   ;FORM TRACK IN R2
```

F10


```
8160  023744  013703  017312              MOV     @#$HINUM,R3     ;FORM CYLINDER IN R3
8161  023750  105701                      TSTB    R1              ;INSURE THE SECTOR IS BETWEEN 0 AND 21
8162  023752  002403                      BLT     2$
8163  023754  122701  000026      1$:     CMPB    #^D22,R1
8164  023760  003003                      BGT     3$
8165  023762  000241              2$:     CLC
8166  023764  106001                      RORB    R1
8167  023766  000772                      BR      1$
8168  023770  105702              3$:     TSTB    R2              ;INSURE THE TRACK IS BETWEEN 0 AND 18
8169  023772  002403                      BLT     5$
8170  023774  122702  000023      4$:     CMPB    #^D19,R2
8171  024000  003003                      BGT     6$
8172  024002  000241              5$:     CLC
8173  024004  106002                      RORB    R2
8174  024006  000772                      BR      4$
8175  024010  023703  002324      6$:     CMP     @#FC16,R3       ;INSURE THE CYLINDER IS BETWEEN FC AND LC
8176  024014  003413                      BLE     7$
8177  024016  000241                      CLC
8178  024020  006003                      ROR     R3
8179  024022  005503                      ADC     R3
8180  024024  001371                      BNE     6$
8181  024026  010103                      MOV     R1,R3
8182  024030  000303                      SWAB    R3
8183  024032  060203                      ADD     R2,R3
8184  024034  005203                      INC     R3
8185  024036  003364                      BGT     6$
8186  024040  005403                      NEG     R3
8187  024042  000762                      BR      6$
8188  024044  023703  002326      7$:     CMP     @#LC16,R3
8189  024050  002003                      BGE     8$
8190  024052  000241                      CLC
8191  024054  006003                      ROR     R3
8192  024056  000772                      BR      7$
8193  024060  023703  002324      8$:     CMP     @#FC16,R3
8194  024064  003403                      BLE     9$
8195  024066  005203                      INC     R3
8196  024070  000303                      SWAB    R3
8197  024072  000764                      BR      7$
8198  024074  110137  003530      9$:     MOVB    R1,@#DTADPB+10  ;SAVE SECTOR ADDRESS
8199  024100  110237  003531              MOVB    R2,@#DTADPB+11  ;SAVE TRACK ADDRESS
8200  024104  010337  003532              MOV     R3,@#DTADPB+12  ;SAVE CYLINDER ADDRESS
8201  024110  000200                      RTS     R0              ;RETURN
8202
8203                                  ;******************************************************************
8204                                  .SBTTL  GETSWR - GET THE CONTROL SWITCH SETTINGS
8205                                  ;*THIS ROUTINE IS USED TO INPUT THE "CONTROL SWITCHES".
8206                                  ;*IF SWR<07>=1 THE PRESENT SETTING WILL BE TYPED AND THE NEW
8207                                  ;*SETTING IS READ AND STORED.
8208                                  ;*NOTE: THIS ROUTINE DESTROYS R3 AND R4
8209                                  ;*CALL
8210                                  ;*      JSR     PC,@#GETSWR
8211                                  ;*      RETURN                  ;(C.SWR)=DESIRED CONTROL SWITCHES
8212
8213  024112  032737  000200  177570  GETSWR: BIT     #SW07,@#SWR     ;READ CONTROL SWITCHES?
```

```
 8214  024120  001426                        BEQ     2$              ;NO--BRANCH
 8215  024122  104400  024130                TYPE    .+4             ;;TYPE ASCIZ STRING
  (1)  024126  000410                        BR      64$             ;;GET OVER THE ASCIZ
  (1)                                        ;;.ASCIZ      <15><12>/SET SWR<07>=0/
  (1)  024150                             64$:
 8216  024150  012703  034436              1$:     MOV     #MSG.CS,R3      ;"CONTROL SWITCHES="
 8217  024154  013704  001212                      MOV     @#C.SWR,R4      ;PRESENT CONTROL SWITCH SETTINGS
 8218  024160  004037  024420                      JSR     R0,@#GETNUM     ;GET THE NEW SWITCH SETTINGS
 8219  024164  000771                        BR      1$              ; COMMA
 8220  024166  000770                        BR      1$              ; PERIOD
 8221  024170  010437  001212                      MOV     R4,@#C.SWR      ; DOUBLE PERIOD-SAVE NEW SWITCH SETTING
 8222  024174  000746                        BR      GETSWR          ;LOOP
 8223  024176  000207                     2$:     RTS     PC              ;RETURN FROM CALL
 8224
 8225                                     ;*****************************************************************
 8226                                     .SBTTL  GETADR - GET BUS ADDRESS AND VECTOR ADDRESS
 8227                                     ;*THIS ROUTINE IS USED TO INSURE THE BUS ADDRESS
 8228                                     ;*OF THE RH11/RP04 IS SETUP TO READ THE PROPER VALUE.
 8229                                     ;*IT WILL ALSO READ THE ADDRESS FROM THE TTY IF
 8230                                     ;*REQUIRED.
 8231                                     ;*NOTE: THIS ROUTINE DESTROYS R0-R4
 8232                                     ;*CALL
 8233                                     ;*
 8234                                     ;*      JSR     PC,@#GETADR
 8235                                     ;*      RETURN
 8236
 8237  024200  005737  001216             GETADR: TST     @#BUSADR        ;INPUT FROM TTY REQUESTED?
 8238  024204  001447                        BEQ     7$              ;NO--BRANCH
 8239  024206  005037  001216                CLR     @#BUSADR        ;YES--CLEAR THE REQUEST FLAG
 8240  024212  012700  001326              1$:     MOV     #RH.ADR,R0      ;FIRST ADDRESS
 8241  024216  012703  034462                      MOV     #MRHCS1,R3      ;"RHCS1="
 8242  024222  011004                        MOV     (R0),R4         ;PRESENT RHCS1 ADDRESS
 8243  024224  004037  024420                MOV     R0,@#GETNUM     ;GET NEW RHCS1
 8244  024230  000402                        BR      2$              ;COMMA
 8245  024232  000767                        BR      1$              ;PERIOD
 8246  024234  000430                        BR      5$              ;DOUBLE PERIOD
 8247  024236  010420                     2$:     MOV     R4,(R0)+        ;SAVE NEW RHCS1
 8248  024240  012703  034473                      MOV     #MRHVEC,R3      ;"RHVEC="
 8249  024244  011004                        MOV     (R0),R4         ;PRESENT RH11 VECTOR ADDRESS
 8250  024246  004037  024420                MOV     R0,@#GETNUM     ;GET NEW RHVEC
 8251  024252  000402                        BR      3$              ;COMMA
 8252  024254  000756                        BR      1$              ;PERIOD
 8253  024256  000417                        BR      5$              ;DOUBLE PERIOD
 8254  024260  010420                     3$:     MOV     R4,(R0)+        ;SAVE NEW RHVEC
 8255  024262  012703  034504                      MOV     #MRHPRI,R3      ;"RHPRIO="
 8256  024266  011004                        MOV     (R0),R4         ;PRESENT RH11 PRIORITY LEVEL
 8257  024270  006304                        ASL     R4              ;POSITION FOR TYPEOUT
 8258  024272  006304                        ASL     R4
 8259  024274  006304                        ASL     R4
 8260  024276  000304                        SWAB    R4
 8261  024300  004037  024420                JSR     R0,@#GETNUM     ;GET NEW RHPRIO
 8262  024304  000402                        BR      4$              ;COMMA
 8263  024306  000401                        BR      4$              ;PERIOD
 8264  024310  000404                        BR      6$              ;DOUBLE PERIOD
```

```
8265  024312  010210                    4$:   MOV   R2,(R0)          ;SAVE NEW RHPRIO
8266  024314  000736                          BR    1$               ;LOOP
8267  024316  010410                    5$:   MOV   R4,(R0)          ;SAVE INPUT
8268  024320  000401                          BR    7$
8269  024322  010210                    6$:   MOV   R2,(R0)          ;SAVE PRIORITY
8270  024324  013701  000004            7$:   MOV   @#ERRVEC,R1      ;SAVE THE ERROR VECTOR
8271  024330  012737  024366  000004          MOV   #8$,@#ERRVEC     ;SETUP FOR TRAP
8272  024336  005777  154764                  TST   @RH.ADR          ;CHECK FOR RH11/RP04
8273  024342  010137  000004                  MOV   R1,@#ERRVEC      ;RESTORE ERROR VECTOR
8274  024346  012700  001326                  MOV   #RH.ADR,R0       ;FIRST ADDRESS OF NEW PARAMETERS
8275  024352  012701  027166                  MOV   #RPADR,R1        ;FIRST ADDRESS OF WHERE TO PUT THEM
8276  024356  012021                          MOV   (R0)+,(R1)+      ;BUS ADDRESS
8277  024360  012021                          MOV   (R0)+,(R1)+      ;VECTOR ADDRESS
8278  024362  012021                          MOV   (R0)+,(R1)+      ;PRIORITY LEVEL
8279  024364  000207                          RTS   PC               ;RETURN
8280  024366  010137  000004            8$:   MOV   R1,@#ERRVEC      ;RESTORE ERROR VECTOR
8281  024372  022626                          CMP   (SP)+,(SP)+      ;CLEAN OFF THE STACK
8282  024374  104010                          ERROR 10               ;REPORT THE ERROR
8283  024376  005737  000042                  TST   @#42             ;IS THERE A MONITOR?
8284  024402  001703                          BEQ   1$               ;NO--GO ASK FOR ADDRESS
8285  024404  005037  001222                  CLR   @#DRVSEL         ;YES--NO DRIVES SELECTED
8286  024410  005037  013652                  CLR   @#SEOPCT         ;NO PASSES
8287  024414  000137  013476                  JMP   @#SEOP           ;GO TO END OF PROGRAM
8288
8289                                ;*****************************************************************
8290                                .SBTTL  GETNUM - ROUTINE TO GET A NUMBER
8291                                ;*THIS ROUTINE WILL TYPE AN ASCIZ MESSAGE AND THEN
8292                                ;*INPUT AN ASCIZ STRING AND CHANGE THE STRING TO OCTAL
8293                                ;*IF REQUIRED.
8294                                ;*NOTE: THIS ROUTINE DESTROYS R1
8295                                ;*CALL
8296                                ;*      MOV     #ADR,R3         ;ADDRESS OF ASCIZ MESSAGE
8297                                ;*      MOV     #NUM,R4         ;OCTAL NUMBER
8298                                ;*      JSR     R0,@#GETNUM
8299                                ;*      RETURN1                 ;INPUT TERMINATED WITH A COMMA
8300                                ;*      RETURN2                 ;WITH A PERIOD
8301                                ;*      RETURN3                 ;WITH A DOUBLE PERIOD
8302                                ;*                              ;R4=INPUT NUMBER AND
8303                                ;*                              ;R2=R4*32 FOR ALL
8304                                ;*                              ;THREE RETURNS
8305
8306  024420  010337  024426    GETNUM: MOV   R3,2$             ;SAVE MESSAGE POINTER
8307  024424  104400            1$:     TYPE                    ;TYPE THE MESSAGE
8308  024426  000000            2$:     .WORD 0                 ;MESSAGE POINTER GOES HERE
8309  024430  010446                    MOV   R4,-(SP)          ;;SAVE R4 FOR TYPEOUT
 (1)  024432  104402                    TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
8310  024434  104414                    RDLIN                   ;READ AN ASCIZ STRING
8311  024436  012601                    MOV   (SP)+,R1          ;ADDRESS OF FIRST CHARACTER
8312  024440  004037  026454            JSR   R0,@#CK.CHR       ;CHECK ONE CHARACTER
 (1)  024444  024424                    1$                      ;ILLEGAL CHARACTER
 (1)  024446  024424                    1$                      ;CARRIAGE RETURN
 (1)  024450  024460                    3$                      ;"/"
 (1)  024452  024504                    7$                      ;"."
 (1)  024454  024512                    8$                      ;"'"
```

```
  (1)  024456  024424                          1$                      ;DIGIT 0-9
8313   024460                        3$:
  (1)  024460  004037  026710                  JSR     R0,@#CK.NUM     ;CHECK THE NUMBER
  (1)  024464  024424                          1$                      ;ILLEGAL INPUT
  (1)  024466  024500                          6$                      ;TERMINATED WITH A ","
  (1)  024470  024476                          5$                      ;TERMINATED WITH A "."
  (1)  024472  024474                          4$                      ;TERMINATED WITH A ".."
8314   024474  005720                4$:       TST     (R0)+           ;DOUBLE PERIOD
8315   024476  005720                5$:       TST     (R0)+           ;SINGLE PERIOD
8316   024500  010204                6$:       MOV     R2,R4           ;COMMA--SAVE INPUT NUMBER
8317   024502  000414                          BR      10$             ;GO TO EXIT
8318   024504  105711                7$:       TSTB    (R1)            ;TERMINATOR AFTER A COMMA?
8319   024506  001346                          BNE     1$              ;NO--LOOP
8320   024510  000411                          BR      10$             ;YES--EXIT
8321   024512  105711                8$:       TSTB    (R1)            ;TERMINATOR AFTER A PERIOD?
8322   024514  001406                          BEQ     9$              ;YES--EXIT
8323   024516  122721  000056                  CMPB    #'.,(R1)+       ;NO--DOUBLE PERIOD?
8324   024522  001340                          BNE     1$              ;NO--LOOP
8325   024524  105711                          TSTB    (R1)            ;YES--TERMINATOR?
8326   024526  001336                          BNE     1$              ;NO--LOOP
8327   024530  005720                          TST     (R0)+           ;DOUBLE PERIOD
8328   024532  005720                9$:       TST     (R0)+           ;PERIOD
8329   024534  010402                10$:      MOV     R4,R2           ;COMMA--POSITION THE
8330   024536  000302                          SWAB    R2              ;NUMBER IN CASE IT
8331   024540  006202                          ASR     R2              ;IS THE PRIORITY LEVEL
8332   024542  006202                          ASR     R2
8333   024544  006202                          ASR     R2
8334   024546  000200                          RTS     R0              ;EXIT
8335
8336                                ;*****************************************************************
8337                                .SBTTL  GT.PRM - GET PARAMETERS
8338                                ;*THIS ROUTINE IS USED TO CHANGE OR MODIFY
8339                                ;*THE TEST PARAMETERS.  IT GIVES THE OPERATOR
8340                                ;*THE CAPABILITY OF SPECIFYING WHICH DRIVES TO TEST, WHICH
8341                                ;*TESTS TO RUN AND HOW MANY TIMES TO
8342                                ;*REPEAT EACH TEST
8343   024550  004037  034334       GT.PRM: JSR     R0,@#SAVR05     ;SAVE R0-R5
8344   024554  005037  001222       GT.PR1: CLR     DRVSEL          ;NO DRIVE SELECTED
8345   024560  104400  024566               TYPE    ,.+4            ;;TYPE ASCIZ STRING
  (1)  024564  000406                       BR      64$             ;;GET OVER THE ASCIZ
  (1)                               ;;.ASCIZ      <15><12>/DRIVE(S)=/
  (1)  024602                       64$:
8346   024602  104414                       RDLIN                   ;READ TTY
8347   024604  012601                       MOV     (SP)+,R1        ;ADDRESS OF ASCIZ STRING
8348   024606  004037  026454               JSR     R0,@#CK.CHR     ;CHECK ONE CHARACTER
  (1)  024612  024554                       GT.PR1                  ;ILLEGAL CHARACTER
  (1)  024614  024554                       GT.PR1                  ;CARRIAGE RETURN
  (1)  024616  024554                       GT.PR1                  ;"/"
  (1)  024620  024554                       GT.PR1                  ;","
  (1)  024622  024554                       GT.PR1                  ;"."
  (1)  024624  024626                       1$                      ;DIGIT 0-9
8349   024626  005301               1$:     DEC     R1
8350   024630                       2$:
  (1)  024630  012702  000007               MOV     #7,R2   ;UPPER LIMIT OF INPUT
```

```
  (1)  024634  004037  026530                JSR     R0,@#CK.DIG     ;CHECK THE DIGIT(S)
  (1)  024640  024554                        GT.PR1                  ;ILLEGAL INPUT
  (1)  024642  024554                        GT.PR1                  ;INPUT TO LARGE
  (1)  024644  024652                        3$                      ;TERMINATED WITH A "."
  (1)  024646  024672                        4$                      ;TERMINATED WITH A ":"
  (1)  024650  024672                        4$                      ;TERMINATED WITH A ".."
 8351  024652  156237  027154  001222  3$:   BISB    ATABIT(R2),DRVSEL ;SET THE DRIVE SELECTED BIT
 8352  024660  105741                        TSTB    -(R1)           ;WAS THE LINE TERMINATED?
 8353  024662  001362                        BNE     2$              ;NO-GET THE NEXT DRIVE
 8354  024664  005037  001224                CLR     @#TSTNMS        ;DESELECT ALL TESTS
 8355  024670  000406                        BR      GTTST1          ;YES--SELECT TEST
 8356  024672  156237  027154  001222  4$:   BISB    ATABIT(R2),DRVSEL ;SET THE SELECTED DRIVE BITS
 8357  024700  004037  034360        GT.PR2: JSR     R0,@#GETR0S     ;RESTORE R0-R5
 8358  024704  000207                        RTS     PC              ;EXIT
 8359
 8360  024706                        GTTST1:
  (1)  024706  104400  024714                TYPE    .+4             ;;TYPE ASCIZ STRING
  (1)  024712  000403                        BR      64$             ;;GET OVER THE ASCIZ
  (1)                                        ;;.ASCIZ        /TEST=/
  (1)  024722                        64$:
 8361  024722  104414                        RDLIN                   ;READ AN ASCIZ STRING
 8362  024724  012601                        MOV     (SP)+,R1        ;POINTER TO R1
 8363  024726  122711  000056                CMPB    #'.,(R1)        ;DOUBLE PERIOD?
 8364  024732  001007                        BNE     1$              ;NO--BRANCH
 8365  024734  122761  000056  000001        CMPB    #'.,1(R1)
 8366  024742  001003                        BNE     1$
 8367  024744  105761  000002                TSTB    2(R1)           ;"CR"?
 8368  024750  001753                        BEQ     GT.PR2          ;YES--EXIT
 8369  024752  005037  001224        1$:     CLR     TSTNMS          ;NO TEST SELECTED
 8370  024756  005003                        CLR     R3              ;NO TEST TO BE OPENED
 8371  024760  005004                GTTST2: CLR     R4              ;NO TEST BITS SET
 8372  024762  121127  000123                CMPB    (R1),#'S        ;ALL SEEK TESTS?
 8373  024766  001003                        BNE     1$              ;NO--BRANCH
 8374  024770  012704  000177                MOV     #177,R4         ;YES--SELECT TESTS 0-6
 8375  024774  000512                        BR      GTTST3
 8376  024776  121127  000124        1$:     CMPB    (R1),#'T        ;ALL TIMING TESTS?
 8377  025002  001003                        BNE     2$              ;NO--BRANCH
 8378  025004  012704  003600                MOV     #3600,R4        ;YES--SELECT TESTS 7-12
 8379  025010  000504                        BR      GTTST3
 8380  025012  121127  000101        2$:     CMPB    (R1),#'A        ;ALL ADDRESSING TESTS?
 8381  025016  001003                        BNE     3$              ;NO--BRANCH
 8382  025020  012704  014000                MOV     #14000,R4       ;YES--SELECT TESTS 13 & 14
 8383  025024  000476                        BR      GTTST3
 8384  025026  121127  000104        3$:     CMPB    (R1),#'D        ;DATA TEST?
 8385  025032  001003                        BNE     4$              ;NO--BRANCH
 8386  025034  012704  020000                MOV     #20000,R4       ;YES--SELECT TEST 15
 8387  025040  000470                        BR      GTTST3
 8388  025042  121127  000105        4$:     CMPB    (R1),#'E        ;EXERCISER TEST?
 8389  025046  001003                        BNE     5$              ;NO--BRANCH
 8390  025050  012704  040000                MOV     #40000,R4       ;YES--SELECT TEST 16
 8391  025054  000462                        BR      GTTST3
 8392  025056  004037  026400        5$:     JSR     R0,@#CK.OCT     ;OCTAL DIGIT?
 8393  025062  000460                        BR      GTTST4          ;NO--BRANCH
 8394  025064  010205                        MOV     R2,R5           ;YES--SAVE IT
```

```
8395  025066  005201                        INC     R1              ;MOVE TO NEXT CHARACTER
8396  025070  004037  026400                JSR     R0,@#CK.OCT     ;OCTAL DIGIT
8397  025074  000405                        BR      6$              ;NO--BRANCH
8398  025076  005201                        INC     R1              ;MOVE TO NEXT CHARACTER
8399  025100  006305                        ASL     R5              ;SCALE HIGH DIGIT
8400  025102  006305                        ASL     R5
8401  025104  006305                        ASL     R5
8402  025106  060502                        ADD     R5,R2           ;COMBINE HIGH & LOW DIGITS
8403  025110  020227  000016        6$:     CMP     R2,#16          ;LEGAL TEST NUMBER?
8404  025114  003274                        BGT     GTTST1          ;NO--BRANCH
8405  025116  006302                        ASL     R2
8406  025120  016204  001374                MOV     BITS(R2),R4     ;SELECT TEST
8407  025124  121127  000055                CMPB    (R1),#'-        ;TEST STRING?
8408  025130  001035                        BNE     GTTST4          ;NO--BRANCH
8409  025132  005201                        INC     R1              ;YES--MOVE TO NEXT CHARACTER
8410  025134  004037  026400                JSR     R0,@#CK.OCT     ;OCTAL DIGIT?
8411  025140  000662                        BR      GTTST1          ;NO--BRANCH
8412  025142  010205                        MOV     R2,R5           ;YES--SAVE IT
8413  025144  005201                        INC     R1              ;MOVE TO NEXT CHARACTER
8414  025146  004037  026400                JSR     R0,@#CK.OCT     ;OCTAL DIGIT?
8415  025152  000405                        BR      7$              ;NO--BRANCH
8416  025154  005201                        INC     R1              ;YES--MOVE TO NEXT CHARACTER
8417  025156  006305                        ASL     R5              ;SCALE HIGH DIGIT
8418  025160  006305                        ASL     R5
8419  025162  006305                        ASL     R5
8420  025164  060502                        ADD     R5,R2           ;COMBINE HIGH & LOW DIGIT
8421  025166  020227  000016        7$:     CMP     R2,#16          ;LEGAL TEST NUMBER?
8422  025172  003245                        BGT     GTTST1          ;NO--BRANCH
8423  025174  006302                        ASL     R2
8424  025176  020462  001374                CMP     R4,BITS(R2)     ;IS THE FIRST NUMBER OF THE
8425                                                                ;STRING SMALLER THAN THE LAST?
8426  025202  002241                        BGE     GTTST1          ;NO--BRANCH
8427  025204  056204  001374        8$:     BIS     BITS(R2),R4     ;YES--SET TEST SELECT BIT FOR SELECTED TESTS
8428  025210  005742                        TST     -(R2)           ;ADJUST POINTER
8429  025212  036204  001374                BIT     BITS(R2),R4     ;DONE?
8430  025216  001772                        BEQ     8$              ;NO--BRANCH
8431  025220  000401                        BR      GTTST4          ;SKIP THE INCREMENT
8432  025222  005201        GTTST3:         INC     R1              ;MOVE TO NEXT CHARACTER
8433  025224  050437  001224        GTTST4: BIS     R4,TSTNMS       ;SET SELECTED TEST BITS INTO
8434                                                                ;TEST SELECT WORD
8435  025230  121127  000056                CMPB    (R1),#'.        ;"PERIOD"?
8436  025234  001420                        BEQ     GTTST5          ;YES--BRANCH
8437  025236  005704                        TST     R4              ;ANY TEST SELECTED THIS CYCLE?
8438  025240  001622                        BEQ     GTTST1          ;NO--BRANCH
8439  025242  121127  000057                CMPB    (R1),#'/        ;"OPEN"?
8440  025246  001002                        BNE     1$              ;NO--BRANCH
8441  025250  050403                        BIS     R4,R3           ;YES--SET BITS FOR TEST TO OPEN
8442  025252  000403                        BR      2$
8443  025254  121127  000054        1$:     CMPB    (R1),#',        ;"COMMA"?
8444  025260  001212                        BNE     GTTST1          ;NO--BRANCH
8445  025262  005201        2$:             INC     R1              ;MOVE TO NEXT CHARACTER
8446  025264  105711                        TSTB    (R1)            ;"CR"?
8447  025266  001234                        BNE     GTTST2          ;NO--GO GET NEXT CHARACTER
8448  025270  005703                        TST     R3              ;ANY TESTS TO OPEN?
```

```
8449  025272  001026                        BNE     OPNTST          ;YES--BRANCH
8450  025274  000604                        BR      GTTST1          ;NO--START AGAIN
8451  025276  005201                GTTST5: INC     R1              ;MOVE TO NEXT CHARACTER
8452  025300  121127  000056              CMPB    (R1),#'.        ;"PERIOD"?
8453  025304  001410                        BEQ     GTTST6          ;YES--BRANCH
8454  025306  105711                        TSTB    (R1)            ;"CR"?
8455  025310  001402                        BEQ     1$              ;YES--BRANCH
8456  025312  000137  024706              JMP     GTTST1
8457  025316  005703                1$:     TST     R3              ;ANY TEST TO BE OPENED?
8458  025320  001013                        BNE     OPNTST          ;YES--BRANCH
8459  025322  000137  024700              JMP     GT.PR2          ;NO--GO START TESTING
8460  025326  005201                GTTST6: INC     R1              ;MOVE TO NEXT CHARACTER
8461  025330  105711                        TSTB    (R1)            ;"CR"?
8462  025332  001402                        BEQ     1$              ;YES--BRANCH
8463  025334  000137  024706              JMP     GTTST1          ;NO--GO ASK FOR TEST
8464  025340  005703                1$:     TST     R3              ;ANY TEST TO BE OPENED?
8465  025342  001002                        BNE     OPNTST          ;YES--BRANCH
8466  025344  000137  024700              JMP     GT.PR2          ;NO--GO START TESTING
8467                          ;OPEN THE TEST FOR CHANGES
8468  025350  004037  034334        OPNTST: JSR     R0,@#SAVR05
8469  025354  005027                        CLR     (PC)+           ;START WITH TEST 0
8470  025356  000000                OPN.CT: .WORD   0               ;COUNT STORED HERE
8471  025360  000412                        BR      OPN.2           ;SKIP THE INCREMENT
8472  025362  005237  025356        OPN.1:  INC     OPN.CT          ;MOVE TO THE NEXT TEST
8473  025366  022737  000016  025356      CMP     #16,OPN.CT      ;TEST NUMBER TO BIG?
8474  025374  002004                        BGE     OPN.2           ;NO--OPEN THE NEXT TEST
8475  025376  004037  034360              JSR     R0,@#GETR05
8476  025402  000137  024706              JMP     GTTST1          ;YES--GO ASK FOR MORE TESTS
8477  025406  013705  025356        OPN.2:  MOV     OPN.CT,R5       ;SETUP TO USE THE
8478  025412  006305                        ASL     R5              ;TEST NUMBER AS AN INDEX
8479  025414  036503  001374              BIT     BITS(R5),R3     ;OPEN THIS TEST?
8480  025420  001760                        BEQ     OPN.1           ;NO--MOVE TO NEXT TEST
8481  025422  104400  025430              TYPE    ,.+4            ;;TYPE ASCIZ STRING
 (1)  025426  000404                        BR      64$             ;;GET OVER THE ASCIZ
 (1)                                        ;;.ASCIZ        / TEST /
 (1)  025440                        64$:
8482  025440  013746  025356              MOV     OPN.CT,-(SP)    ;;SAVE OPN.CT FOR TYPEOUT
 (1)                                                                ;;TEST NUMBER
 (1)  025444  104404                        TYPOS                   ;;GO TYPE--OCTAL ASCII
 (1)  025446     002                        .BYTE   2               ;;TYPE 2 DIGIT(S)
 (1)  025447     000                        .BYTE   0               ;;SUPPRESS LEADING ZEROS
8483  025450  104400  001207              TYPE    ,$CRLF          ;TYPE "CR" & "LF"
8484  025454  016500  001462              MOV     PRMPT(R5),R0    ;PICKUP PARAMETER POINTER
8485  025460  011046                        MOV     (R0),-(SP)      ;SAVE THE VARIABLE INDICATOR
8486  025462  012702  001434              MOV     #PRM,R2         ;FIRST ADDRESS OF TABLE
8487  025466  000405                        BR      2$
8488  025470  006216                1$:     ASR     (SP)            ;CHECK FOR A VARIABLE
8489  025472  103403                        BCS     2$              ;GO MOVE THIS ONE
8490  025474  001404                        BEQ     OPNPRM          ;DONE
8491  025476  005722                        TST     (R2)+           ;BUMP THE POINTER
8492  025500  000773                        BR      1$
8493  025502  012022                2$:     MOV     (R0)+,(R2)+     ;MOVE THIS VARIABLE INTO THE
8494  025504  000771                        BR      1$              ;COMMON AREA
8495  025506  013716  001434        OPNPRM: MOV     @#PRM,(SP)      ;GET THE VARIABLE INDICATOR
```

```
8496  025512  005004                          CLR     R4              ;ZERO THE INDEX
8497  025514  006216                  1$:     ASR     (SP)            ;CHECK FOR A VARIABLE
8498  025516  103403                          BCS     3$              ;GO GET IT
8499  025520  001772                          BEQ     OPNPRM          ;OUT OF VARIABLES
8500  025522  005724                  2$:     TST     (R4)+           ;UPDATE THE INDEX
8501  025524  000773                          BR      1$
8502  025526  005764  001520          3$:     TST     PRMLMT(R4)      ;IS THE MAX. MAGNITUDE NEG?
8503  025532  100463                          BMI     OPNPAT          ;YES--THEN IT IS THE PATTERN
8504  025534  104400  035202                  TYPE    ,MSG.SP         ;TYPE SPACES
8505  025540  016437  001544  025550          MOV     PRMMSG(R4),4$   ;TYPE THE NAME OF THIS VARIABLE
8506  025546  104400                          TYPE
8507  025550  000000                  4$:     .WORD   0
8508  025552  104400  034434                  TYPE    ,MSG.EQ         ;TYPE "="
8509  025556  016446  001436                  MOV     RPT(R4),-(SP)           ;PUT RPT(R4) ON THE STACK
 (1)  025562  004737  016460                  JSR     PC,@#$B2D       ;CHANGE RPT(R4) TO DECIMAL ASCIZ
 (1)  025566  004737  016704                  JSR     PC,@#$SUPRS     ;TYPE WITHOUT LEADING ZEROS
8510  025572  104414                          RDLIN
8511  025574  012601                          MOV     (SP)+,R1        ;READ AN ASCIZ STRING
8512  025576  004037  026454                  JSR     R0,@#CK.CHR     ;CHECK ONE CHARACTER
 (1)  025602  025526                          3$                      ;ILLEGAL CHARACTER
 (1)  025604  025526                          3$                      ;CARRIAGE RETURN
 (1)  025606  025652                          7$                      ;"/"
 (1)  025610  025616                          5$                      ;"."
 (1)  025612  025624                          6$                      ;"'"
 (1)  025614  025526                          3$                      ;DIGIT 0-9
8513  025616  105711                  5$:     TSTB    (R1)            ;"CR"?
8514  025620  001342                          BNE     3$              ;NO--STAY ON THIS VARIABLE
8515  025622  000737                          BR      2$              ;YES--MOVE TO NEXT VARIABLE
8516  025624  105711                  6$:     TSTB    (R1)            ;IS THERE A "CR" AFTER THE PERIOD?
8517  025626  001002                          BNE     64$             ;NO
8518  025630  000137  026240                  JMP     OPN.N2          ;YES--GO CLOSE THIS TEST
8519  025634  122721  000056          64$:    CMPB    #'.,(R1)+       ;DOUBLE PERIOD?
8520  025640  001332                          BNE     3$              ;NO--GO ASK FOR THIS VARIABLE
8521  025642  105711                          TSTB    (R1)            ;YES--IS A "CR" AFTER THE DOUBLE PERIOD?
8522  025644  001330                          BNE     3$              ;NO--ASK FOR THIS VARIABLE AGAIN
8523  025646  000137  026256                  JMP     OPN.X2          ;YES--CLOSE ALL TEST
8524  025652                          7$:
 (1)  025652  016402  001520                  MOV     PRMLMT(R4),R2   ;UPPER LIMIT OF INPUT
 (1)  025656  004037  026530                  JSR     R0,@#CK.DIG     ;CHECK THE DIGIT(S)
 (1)  025662  025526                          3$                      ;ILLEGAL INPUT
 (1)  025664  025526                          3$                      ;INPUT TO LARGE
 (1)  025666  025674                          8$                      ;TERMINATED WITH A ","
 (1)  025670  026234                          OPN.N1                  ;TERMINATED WITH A "."
 (1)  025672  026252                          OPN.X1                  ;TERMINATED WITH A ".."
8525  025674  010264  001436          8$:     MOV     R2,RPT(R4)      ;SAVE THIS VARIABLE
8526  025700  000710                          BR      2$              ;MOVE TO NEXT VARIABLE
8527  025702  104400  035202          OPNPAT: TYPE    ,MSG.SP         ;TYPE SPACES
8528  025706  104400  034430                  TYPE    ,MSG.PAT        ;TYPE "PAT"
8529  025712  104400  034434                  TYPE    ,MSG.EQ         ;TYPE "="
8530  025716  016446  001436                  MOV     RPT(R4),-(SP)   ;;SAVE RPT(R4) FOR TYPEOUT
 (1)  025722  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
8531  025724  104414                          RDLIN                   ;READ ASCIZ STRING
8532  025726  012601                          MOV     (SP)+,R1        ;PICKUP POINTER
8533  025730  004037  026454                  JSR     R0,@#CK.CHR     ;CHECK ONE CHARACTER
```

```
  (1)  025734  025702                        OPNPAT                  ;ILLEGAL CHARACTER
  (1)  025736  025702                        OPNPAT                  ;CARRIAGE RETURN
  (1)  025740  025770                        2$                      ;"/"
  (1)  025742  025506                        OPNPRM                  ;" "
  (1)  025744  025750                        1$                      ;"'"
  (1)  025746  025702                        OPNPAT                  ;DIGIT 0-9
8534   025750  105711              1$:       TSTB    (R1)            ;"CR" AFTER THE PERIOD?
8535   025752  001532                        BEQ     OPN.N2          ;YES--GO CLOSE THIS TEST
8536   025754  122721  000056                CMPB    #'.,(R1)+       ;NO--PERIOD?
8537   025760  001350                        BNE     OPNPAT          ;NO--LOOP
8538   025762  105711                        TSTB    (R1)            ;"CR" AFTER A DOUBLE PERIOD?
8539   025764  001534                        BEQ     OPN.X2          ;YES--GO START TESTING
8540   025766  000745                        BR      OPNPAT          ;NO--LOOP
8541   025770                      2$:
  (1)  025770  004037  026710                JSR     R0,@#CK.NUM     ;CHECK THE NUMBER
  (1)  025774  025702                        OPNPAT                  ;ILLEGAL INPUT
  (1)  025776  026004                        3$                      ;TERMINATED WITH A ","
  (1)  026000  026234                        OPN.N1                  ;TERMINATED WITH A "'"
  (1)  026002  026252                        OPN.X1                  ;TERMINATED WITH A "."
8542   026004  010264  001436        3$:       MOV     R2,RPT(R4)      ;SAVE THE INPUT NUMBER
8543   026010  006002                        ROR     R2              ;OPEN PATTERN 0?
8544   026012  103235                        BCC     OPNPRM          ;NO--START AT BEGINNING OF PARAMETER TABLE
8545   026014  004037  034334                JSR     R0,@#SAVR05     ;SAVE R1-R5
8546   026020  005000              OPNWDS:   CLR     R0              ;START WITH WORD 0
8547   026022  012704  002440                MOV     #PAT0,R4
8548   026026                      1$:
  (1)  026026  104400  026034                TYPE    ,.+4            ;;TYPE ASCIZ STRING
  (1)  026032  000403                        BR      64$             ;;GET OVER THE ASCIZ
  (1)                                        ;;.ASCIZ        /   WD/
  (1)  026042                      64$:
8549   026042  010046                        MOV     R0,-(SP)                ;PUT R0 ON THE STACK
  (1)  026044  004737  016460                JSR     PC,@#$SB2D      ;CHANGE R0 TO DECIMAL ASCIZ
  (1)  026050  004737  016704                JSR     PC,@#$SUPRS     ;TYPE WITHOUT LEADING ZEROS
8550   026054  104400  034434                TYPE    ,MSG.EQ         ;TYPE "="
8551   026060  011446                        MOV     (R4),-(SP)      ;;SAVE (R4) FOR TYPEOUT
  (1)  026062  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
8552   026064  104414                        RDLIN                   ;READ ASCIZ STRING
8553   026066  012601                        MOV     (SP)+,R1        ;PICKUP THE POINTER
8554   026070  004037  026454                JSR     R0,@#CK.CHR     ;CHECK ONE CHARACTER
  (1)  026074  026026                        1$                      ;ILLEGAL CHARACTER
  (1)  026076  026026                        1$                      ;CARRIAGE RETURN
  (1)  026100  026110                        2$                      ;"/"
  (1)  026102  026126                        4$                      ;" "
  (1)  026104  026142                        5$                      ;"'"
  (1)  026106  026026                        1$                      ;DIGIT 0-9
8555   026110                      2$:
  (1)  026110  004037  026710                JSR     R0,@#CK.NUM     ;CHECK THE NUMBER
  (1)  026114  026026                        1$                      ;ILLEGAL INPUT
  (1)  026116  026124                        3$                      ;TERMINATED WITH A ","
  (1)  026120  026162                        6$                      ;TERMINATED WITH A "'"
  (1)  026122  026176                        8$                      ;TERMINATED WITH A "."
8556   026124  010214        3$:       MOV     R2,(R4)         ;SAVE THE INPUT
8557   026126  005724        4$:       TST     (R4)+           ;MOVE TO NEXT WORD
8558   026130  005200                  INC     R0              ;INCREMENT THE COUNT
```

```
 8559  026132  022700  000020                CMP     #16.,R0         ;COUNT TO LARGE?
 8560  026136  003333                        BGT     1$              ;NO--BRANCH
 8561  026140  000727                        BR      OPNWDS          ;YES--BRANCH
 8562  026142  105711                5$:     TSTB    (R1)            ;"CR" AFTER THE PERIOD?
 8563  026144  001407                        BEQ     7$              ;YES--GO CLOSE THIS TEST
 8564  026146  122721  000056                CMPB    #'.,(R1)+       ;NO--PERIOD?
 8565  026152  001325                        BNE     1$              ;NO--BRANCH ILLEGAL INPUT STRING
 8566  026154  105711                        TSTB    (R1)            ;"CR" AFTER THE "PERIOD-PERIOD"?
 8567  026156  001410                        BEQ     9$              ;YES--GO START TESTING
 8568  026160  000722                        BR      1$              ;NO--LOOP
 8569  026162  010224                6$:     MOV     R2,(R4)+        ;SAVE THE INPUT
 8570  026164  004737  026212        7$:     JSR     PC,@#CLSWDS     ;CLOSE THE DATA PATTERN
 8571  026170  004037  034360                JSR     R0,@#GETR05     ;GET R0-R5
 8572  026174  000421                        BR      OPN.N2          ;MOVE TO NEXT TEST
 8573  026176  010224                8$:     MOV     R2,(R4)+        ;SAVE THE INPUT
 8574  026200  004737  026212        9$:     JSR     PC,@#CLSWDS     ;CLOSE THE DATA PATTERN
 8575  026204  004037  034360                JSR     R0,@#GETR05     ;GET R0-R5
 8576  026210  000422                        BR      OPN.X2          ;START TESTING
 8577  026212  012701  002440        CLSWDS: MOV     #PAT0,R1        ;FIRST ADDRESS OF DATA PATTERN
 8578  026216  005200                1$:     INC     R0              ;COUNT THE LAST WORD THAT WAS STORED
 8579  026220  022700  000017                CMP     #15.,R0         ;END OF TABLE
 8580  026224  002402                        BLT     2$              ;YES--EXIT
 8581  026226  012124                        MOV     (R1)+,(R4)+     ;COPY
 8582  026230  000772                        BR      1$              ;LOOP
 8583  026232  000207                2$:     RTS     PC              ;RETURN
 8584  026234  010264  001436        OPN.N1: MOV     R2,RPT(R4)      ;SAVE THIS VARIABLE
 8585  026240  005726                OPN.N2: TST     (SP)+           ;CLEAN OFF THE STACK
 8586  026242  004737  026314                JSR     PC,CLOSE        ;CLOSE THIS TEST
 8587  026246  000137  025362                JMP     OPN.1           ;GO OPEN THE NEXT TEST
 8588  026252  010264  001436        OPN.X1: MOV     R2,RPT(R4)      ;SAVE THIS VARIABLE
 8589  026256  005726                OPN.X2: TST     (SP)+           ;CLEAN OFF THE STACK
 8590  026260  004737  026314        1$:     JSR     PC,CLOSE        ;CLOSE THIS TEST
 8591  026264  005725                2$:     TST     (R5)+           ;UPDATE THE INDEX
 8592  026266  020527  000034                CMP     R5,#16*2        ;INDEX TO BIG?
 8593  026272  002404                        BLT     3$              ;NO--BRANCH
 8594  026274  004037  034360                JSR     R0,@#GETR05     ;YES--RESTORE R0-R5
 8595  026300  000137  024700                JMP     GT.PR2          ;GO TO EXIT
 8596  026304  036503  001374        3$:     BIT     BITS(R5),R3     ;IS THIS TEST OPEN FOR CHANGE?
 8597  026310  001363                        BNE     1$              ;YES--GO CLOSE IT
 8598  026312  000764                        BR      2$              ;NO--MOVE TO NEXT TEST
 8599  026314  004037  034334        CLOSE:  JSR     R0,@#SAVR05     ;SAVE R0-R5
 8600  026320  012700  001434                MOV     #PRM,R0         ;"FROM" ADDRESS
 8601  026324  016501  001462                MOV     PRMPT(R5),R1    ;"TO" ADDRESS
 8602  026330  012002                        MOV     (R0)+,R2        ;"FROM" INDICATOR
 8603  026332  012103                        MOV     (R1)+,R3        ;"TO" INDICATOR
 8604  026334  012704  000001                MOV     #1,R4           ;TEST BIT START A "RPT"
 8605  026340  030402                1$:     BIT     R4,R2           ;PARAMETER TO BE MOVED?
 8606  026342  001403                        BEQ     2$              ;NO--BRANCH
 8607  026344  030403                        BIT     R4,R3           ;A PLACE TO PUT IT?
 8608  026346  001404                        BEQ     3$              ;NO--BRANCH
 8609  026350  011011                        MOV     (R0),(R1)       ;YES--MOVE "FROM" TO "TO"
 8610  026352  030403                2$:     BIT     R4,R3           ;"TO" PARAMETER?
 8611  026354  001401                        BEQ     3$              ;NO--BRANCH
 8612  026356  005721                        TST     (R1)+           ;YES--UPDATE THE POINTER
```

```
                                3$:     TST     (R0)+           ;UPDATE FROM POINTER
                                        ASL     R4              ;POSITION THE TEST BIT
                                        BIT     #BIT10,R4       ;DONE?
                                        BEQ     1$              ;NO--BRANCH
                                        JSR     R0,@#GETR0S     ;YES--RESTORE R0-R5
                                        RTS     PC              ;RETURN
                               ;*****************************************************************
                               .SBTTL  CK.OCT -- CHECK FOR OCTAL CHARACTER
                               ;*THIS RCUTINE IS USED TO CHECK IF AN
                               ;*ASCII CHARACTER IS A DIGIT BETWEEN 0 AND 7.
                               ;*CALL
                               ;*      MOV     #ADR,R1         ;ADDRESS OF ASCII CHARACTER
                               ;*      JSR     R0,@#CK.OCT     ;CHECK THE CHARACTER
                               ;*      RETURN1                 ;CHARACTER IS NOT BETWEEN 0-7
                               ;*      RETURN2                 ;CHARACTER IS IN R2 AS A
                               ;*                              ;OCTAL DIGIT

                                CK.OCT: CMPB    (R1),#'0        ;LESS THAN ZERO?
                                        BLO     1$              ;YES -- BRANCH
                                        CMPB    (R1),#'7        ;GREATER THAN SEVEN?
                                        BHI     1$              ;YES -- BRANCH
                                        MOVB    (R1),R2         ;GET THE CHARACTER
                                        BIC     #^C7,R2         ;STRIP AWAY THE ASCII
                                        TST     (R0)+           ;ADJUST FOR RETURN
                                1$:     RTS     R0              ;RETURN

                               ;*****************************************************************
                               .SBTTL  CK.DEC -- CHECK FOR DECIMAL CHARACTER
                               ;*THIS ROUTINE IS USED TO CHECK AN ASCII CHARACTER
                               ;*AND DETERMINE IF IT IS A DIGIT BETWEEN 0 AND 9.
                               ;*CALL
                               ;*      MOV     #ADR,R1         ;ADDRESS OF ASCII CHARACTER
                               ;*      JSR     R0,@#CK.DEC     ;CHECK THE CHARACTER
                               ;*      RETURN1                 ;NOT BETWEEN 0 AND 9
                               ;*      RETURN2                 ;BETWEEN 0 AND 9
                               ;*                              ;R2 = DIGIT

                                CK.DEC: CMPB    (R1),#'0        ;LESS THAN ZERO?
                                        BLO     1$              ;YES -- BRANCH
                                        CMPB    (R1),#'9                ;GREATER THAN NINE?
                                        BHI     1$              ;YES -- BRANCH
                                        MOVB    (R1),R2         ;GET THE CHARACTER
                                        BIC     #'0,R2          ;STRIP AWAY THE ASCII
                                        TST     (R0)+           ;ADJUST FOR RETURN
                                1$:     RTS     R0              ;RETURN

                               ;*****************************************************************
                               .SBTTL  CK.CHR -- CHECK CHARACTER
                               ;*THIS ROUTINE WILL CHECK AN ASCII CHARACTER TO
                               ;*DETERMINE WHAT IT IS.
                               ;*CALL
                               ;*      MOV     #ADR,R1         ;ADDRESS OF ASCII CHARACTER
                               ;*      JSR     R0,@#CK.CHR     ;CHECK CHARACTER
                               ;*      RETURN  ADR1            ;UNKNOWN CHARACTER
```

```
                                    ;*      RETURN  ADR2            ;CARRIAGE RETURN * (R1)=ADR+1
                                    ;*      RETURN  ADR3            ;SLASH * (R1)=ADR+1
                                    ;*      RETURN  ADR4            ;COMMA * (R1)=ADR+1
                                    ;*      RETURN  ADR5            ;PERIOD * (R1)=ADR+1
                                    ;*      RETURN  ADR6            ;DIGIT BETWEEN 0 AND 9.
                                    ;*                              ;R2 = DIGIT * (R1)=ADR+1

026454  105711                     CK.CHR: TSTB    (R1)            ;"CARRIAGE RETURN"?
026456  001420                             BEQ     4$              ;YES -- BRANCH
026460  121127  000057                     CMPB    (R1),#'/        ;"SLASH"?
026464  001414                             BEQ     3$              ;YES -- BRANCH
026466  121127  000054                     CMPB    (R1),#',        ;"COMMA"?
026472  001410                             BEQ     2$              ;YES -- BRANCH
026474  121127  000056                     CMPB    (R1),#'.        ;"PERIOD"?
026500  001404                             BEQ     1$              ;YES -- BRANCH
026502  004037  026426                     JSR     R0,@#CK.DEC     ;"DIGIT"?
026506  000406                             BR      5$              ;NO -- BRANCH
026510  005720                             TST     (R0)+           ;DIGIT BETWEEN 0-9
026512  005720                     1$:     TST     (R0)+           ;PERIOD
026514  005720                     2$:     TST     (R0)+           ;COMMA
026516  005720                     3$:     TST     (R0)+           ;SLASH
026520  005720                     4$:     TST     (R0)+           ;CARRIAGE RETURN
026522  005201                             INC     R1              ;MOVE POINTER TO NEXT CHARACTER
026524  011000                     5$:     MOV     (R0),R0         ;UNKNOWN CHARACTER
026526  000200                             RTS     R0              ;RETURN

                                    ;******************************************************************
                                    .SBTTL  CK.DIG - CHECK DIGIT
                                    ;*THIS ROUTINE CHECKS AN ASCII STRING FOR LEGAL
                                    ;*CHARACTERS AND FORMS A DECIMAL VALUE BINARY NUMBER IN R2.
                                    ;*CALL
                                    ;*      MOV     #ADR,R1         ;ADDRESS OF ASCIZ STRING
                                    ;*      MOV     #NUM,R2         ;MAX. MAGNITUDE OF INPUT NUMBER
                                    ;*      JSR     R0,@#CK.DIG     ;CHECK DIGITS
                                    ;*      RETURN  ADR1            ;ILLEGAL CHARACTER -- R2=?
                                    ;*      RETURN  ADR2            ;INPUT NUMBER TO LARGE -- R2=?
                                    ;*      RETURN  ADR3            ;"COMMA" -- R2 = NUMBER
                                    ;*      RETURN  ADR4            ;"PERIOD" -- R2 = NUMBER
                                    ;*      RETURN  ADR5            ;"PERIOD-PERIOD" -- R2 = NUMBER

026530  010446                     CK.DIG: MOV     R4,-(SP)        ;SAVE R4
026532  010346                             MOV     R3,-(SP)        ;SAVE R3
026534  010246                             MOV     R2,-(SP)        ;SAVE THE MAX. SIZE ON THE STACK
026536  005002                             CLR     R2              ;START WITH 0
026540  005003                             CLR     R3
026542  005004                             CLR     R4
026544  004037  026454                     JSR     R0,@#CK.CHR     ;CHECK ONE CHARACTER
  (1)   026550  026674                     8$                      ;ILLEGAL CHARACTER
  (1)   026552  026674                     8$                      ;CARRIAGE RETURN
  (1)   026554  026674                     8$                      ;"/"
  (1)   026556  026674                     8$                      ;","
  (1)   026560  026674                     8$                      ;"."
  (1)   026562  026564                     1$                      ;DIGIT 0-9
026564  006303                     1$:     ASL     R3              ;*2
```

```
 8715  026566  010346                        MOV     R3,-(SP)        ;SAVE #2
 8716  026570  006303                        ASL     R3              ;*4
 8717  026572  006303                        ASL     R3              ;*8
 8718  026574  062603                        ADD     (SP)+,R3        ;(*8)+(*2)=*10.
 8719  026576  060203                        ADD     R2,R3           ;UPDATE THE INPUT NUMBER
 8720  026600  004037  026454                JSR     R0,@#CK.CHR     ;CHECK ONE CHARACTER
   (1) 026604  026674                        8$                      ;ILLEGAL CHARACTER
   (1) 026606  026674                        8$                      ;CARRIAGE RETURN
   (1) 026610  026674                        8$                      ;"/"
   (1) 026612  026622                        3$                      ;"."
   (1) 026614  026620                        2$                      ;","
   (1) 026616  026564                        1$                      ;DIGIT 0-9
 8721  026620  005724                2$:     TST     (R4)+           ;"PERIOD"
 8722  026622  005724                3$:     TST     (R4)+           ;"COMMA"
 8723  026624  004037  026454                JSR     R0,@#CK.CHR     ;CHECK ONE CHARACTER
   (1) 026630  026674                        8$                      ;ILLEGAL CHARACTER
   (1) 026632  026664                        6$                      ;CARRIAGE RETURN
   (1) 026634  026674                        9$                      ;"/"
   (1) 026636  026674                        9$                      ;"."
   (1) 026640  026644                        4$                      ;","
   (1) 026642  026654                        5$                      ;DIGIT 0-9
 8724  026644  005724                4$:     TST     (R4)+           ;"PERIOD-PERIOD"
 8725  026646  105711                        TSTB    (R1)            ;"CR"?
 8726  026650  001405                        BEQ     6$              ;YES--BRANCH
 8727  026652  000410                        BR      8$
 8728  026654  126127  177776  000054  5$:   CMPB    -2(R1),#',      ;WAS CHARACTER BEFORE THE DIGIT A COMMA?
 8729  026662  001004                        BNE     8$              ;NO--EXIT
 8730  026664  020316                6$:     CMP     R3,(SP)         ;INPUT TO LARGE?
 8731  026666  101001                        BHI     7$              ;YES -- BRANCH
 8732  026670  060400                        ADD     R4,R0           ;ADJUST RETURN ADDRESS
 8733  026672  005720                7$:     TST     (R0)+
 8734  026674  010302                8$:     MOV     R3,R2           ;NUMBER TO R2
 8735  026676  005726                        TST     (SP)+           ;CLEAN MAX. SIZE OFF OF STACK
 8736  026700  012603                        MOV     (SP)+,R3        ;RESTORE R3
 8737  026702  012604                        MOV     (SP)+,R4        ;RESTORE R4
 8738  026704  011000                        MOV     (R0),R0         ;GET RETURN ADDRESS
 8739  026706  000200                        RTS     R0              ;RETURN
 8740
 8741                          ;*****************************************************************
 8742                          .SBTTL  CK.NUM - CHECK NUMBER (OCTAL)
 8743                          ;*THIS ROUTINE CHECKS AN ASCIZ STRING FOR LEGAL CHARACTERS
 8744                          ;*AND FORMS AN OCTAL NUMBER IN R2
 8745                          ;*CALL:
 8746                          ;*      MOV     #ADR,R1         ;ADDRESS OF ASCIZ STRING
 8747                          ;*      JSR     R0,@#CK.NUM     ;GO FORM THE NUMBER
 8748                          ;*      RETURN  ADR1            ;ILLEGAL CHARACTER IN THE INPUT STRING
 8749                          ;*      RETURN  ADR2            ;"COMMA"--R2=NUMBER
 8750                          ;*      RETURN  ADR3            ;"PERIOD"--R2=NUMBER
 8751                          ;*      RETURN  ADR4            ;"PERIOD-PERIOD"--R2=NUMBER
 8752
 8753  026710  010346                CK.NUM: MOV     R3,-(SP)        ;SAVE R3
 8754  026712  005003                        CLR     R3              ;START NUMBER AT ZERO
 8755  026714  004037  026400                JSR     R0,@#CK.OCT     ;OCTAL DIGIT?
 8756  026720  000440                        BR      6$              ;NO--BRANCH
```

```
 8757  026722  005201                1$:     INC     R1              ;MOVE TO NEXT CHARACTER
 8758  026724  006303                        ASL     R3              ;FOR THE OCTAL NUMBER IN R3
 8759  026726  103435                        BCS     6$              ;DON'T LET IT GET TO BIG
 8760  026730  006303                        ASL     R3
 8761  026732  103433                        BCS     6$
 8762  026734  006303                        ASL     R3
 8763  026736  103431                        BCS     6$
 8764  026740  060203                        ADD     R2,R3
 8765  026742  004037  026400                JSR     R0,@#CK.OCT     ;IS THIS AN OCTAL DIGIT?
 8766  026746  000401                        BR      2$              ;NO--FIND OUT WHAT IT IS
 8767  026750  000764                        BR      1$              ;YES--MAKE IT PART OF THE NUMBER
 8768  026752  010302                2$:     MOV     R3,R2           ;SAVE THE OCTAL NUMBER
 8769  026754  005003                        CLR     R3              ;START WITH ZERO INDEX
 8770  026756  004037  026454                JSR     R0,@#CK.CHR     ;CHECK ONE CHARACTER
   (1) 026762  027022                        6$                      ;ILLEGAL CHARACTER
   (1) 026764  027022                        6$                      ;CARRIAGE RETURN
   (1) 026766  027022                        6$                      ;"/"
   (1) 026770  027012                        5$                      ;":"
   (1) 026772  026776                        3$                      ;"'"
   (1) 026774  027022                        6$                      ;DIGIT 0-9
 8771  026776  005723                3$:     TST     (R3)+           ;"PERIOD"
 8772  027000  121127  000056                CMPB    (R1),#'.        ;"PERIOD-PERIOD"?
 8773  027004  001002                        BNE     5$              ;NO--BRANCH
 8774  027006  005201                        INC     R1              ;YES--ADVANCE THE POINTER
 8775  027010  005723                4$:     TST     (R3)+           ;"PERIOD-PERIOD"
 8776  027012  005723                5$:     TST     (R3)+           ;"COMMA"
 8777  027014  105711                        TSTB    (R1)            ;"CR"?
 8778  027016  001001                        BNE     6$              ;NO--BRANCH
 8779  027020  060300                        ADD     R3,R0           ;YES--SAVE THE OCTAL NUMBER
 8780  027022  012603                6$:     MOV     (SP)+,R3        ;RESTORE R3
 8781  027024  011000                        MOV     (R0),R0         ;PICKUP EXIT ADDRESS
 8782  027026  000200                        RTS     R0              ;RETURN
```

```
 8784                                ;*****************************************************************
 8785
  (1)                                .SBTTL  RH11/RP04 DRIVER (REV. 0.9)
  (1)
  (1)                                ;*COPYRIGHT (C) 1974
  (1)                                ;*DIGITAL EQUIPMENT CORP.
  (1)                                ;*MAYNARD, MA 01754
  (1)                                ;*AUTHOR: JIM LACEY
  (1)
  (1)
  (1)                                ;*STORAGE FOR RHDS1, RHER1, RHER2, AND RHER3 ON AN ERROR "2" OR "5"
  (1)                                        ;*RPERRS   = RHDS1
  (1)                                        ;*RPERRS+2 = RHER1
  (1)                                        ;*RPERRS+4 = RHER2
  (1)                                        ;*RPERRS+6 = RHER3
  (1)
  (1)  027030  000000                RPERRS: .WORD   0
  (1)  027032  000000                        .WORD   0
  (1)  027034  000000                        .WORD   0
  (1)  027036  000000                        .WORD   0
  (1)
  (1)                                ;*TABLE OF DRIVE ACTIVE INDICATORS (DRVACT=8 BYTES)
  (1)                                        ;*DRVACT=0 IMPLIES DRIVE IS IDLE
  (1)                                        ;*DRVACT>0 IMPLIES DRIVE IS ACTIVE WITH A COMMAND
  (1)                                        ;*DRVACT<0 IMPLIES DRIVE IS ACTIVE WITH AN ERROR RECOVERY OPERATION
  (1)
  (1)  027040     000                DRVACT: .BYTE   0               ;DRIVE 0
  (1)  027041     000                        .BYTE   0               ;DRIVE 1
  (1)  027042     000                        .BYTE   0               ;DRIVE 2
  (1)  027043     000                        .BYTE   0               ;DRIVE 3
  (1)  027044     000                        .BYTE   0               ;DRIVE 4
  (1)  027045     000                        .BYTE   0               ;DRIVE 5
  (1)  027046     000                        .BYTE   0               ;DRIVE 6
  (1)  027047     000                        .BYTE   0               ;DRIVE 7
  (1)
  (1)                                ;*TABLE OF DRIVE STATUS INDICATORS (DRVSTA=8 BYTES)
  (1)                                        ;*DRVSTA=0 IMPLIES DRIVE IS OFFLINE
  (1)                                        ;*DRVSTA>0 IMPLIES DRIVE IS ONLINE
  (1)                                        ;*DRVSTA<0 IMPLIES DRIVE IS UNSAFE OR NONEXISTENT
  (1)
  (1)  027050     000                DRVSTA: .BYTE   0               ;DRIVE 0
  (1)  027051     000                        .BYTE   0               ;DRIVE 1
  (1)  027052     000                        .BYTE   0               ;DRIVE 2
  (1)  027053     000                        .BYTE   0               ;DRIVE 3
  (1)  027054     000                        .BYTE   0               ;DRIVE 4
  (1)  027055     000                        .BYTE   0               ;DRIVE 5
  (1)  027056     000                        .BYTE   0               ;DRIVE 6
  (1)  027057     000                        .BYTE   0               ;DRIVE 7
  (1)
  (1)                                ;*TABLE OF DRIVE TYPES (DRVTYP=8 WORDS)
  (1)                                        ;*DRVTYP WILL CONTAIN THE DRIVE TYPE OF ALL ONLINE, OFFLINE, AND
  (1)                                        ;*UNSAFE DRIVES. IF A DRIVE IS NONEXISTENT DRVTYP WILL BE ZERO.
  (1)
  (1)  027060  000000                DRVTYP: .WORD   0               ;DRIVE 0
```

```
(1)  027062  000000                    .WORD   0               ;DRIVE 1
(1)  027064  000000                    .WORD   0               ;DRIVE 2
(1)  027066  000000                    .WORD   0               ;DRIVE 3
(1)  027070  000000                    .WORD   0               ;DRIVE 4
(1)  027072  000000                    .WORD   0               ;DRIVE 5
(1)  027074  000000                    .WORD   0               ;DRIVE 6
(1)  027076  000000                    .WORD   0               ;DRIVE 7
(1)
(1)                             ;*TRANSFER WAIT FLAG (TRNSWT=1 WORD)
(1)                                     ;*THIS IS A ONE WORD QUEUE. IT WILL CONTAIN THE ADDRESS OF
(1)                                     ;*"DPB" OF THE I/O OPERATION.
(1)
(1)  027100  000000             TRNSWT: .WORD   0
(1)
(1)                             ;*SEARCH WAIT KEYS (SRCHWT=1 WORD)
(1)                                     ;*THIS IS A ONE WORD QUEUE THAT WILL CONTAIN A KEY FOR EACH OF
(1)                                     ;*THE DRIVES THAT ARE PERFORMING A SEARCH COMMAND FOR THE I/O
(1)                                     ;*REQUEST THAT IS AT THE TOP OF THEIR REQUEST QUEUE.
(1)                                     ;*EACH DRIVE IS ASSIGNED ONE BIT, STARTING AT BIT00 FOR DRIVE 0.
(1)
(1)  027102  000000             SRCHWT: .WORD   0
(1)
(1)                             ;*RP04 DRIVER ACTIVE FLAG (ACTDRV=1 BYTE)
(1)                                     ;*ACTDRV=0 IMPLIES DRIVER IS INACTIVE
(1)                                     ;*ACTDRV>0 IMPLIES DRIVER IS ACTIVE
(1)
(1)  027104     000             ACTDRV: .BYTE   0
(1)
(1)                             ;*SOFTWARE TIMER ROUTINE ACTIVE FLAG (ACTSTR=1 BYTE)
(1)                                     ;*ACTSTR=0 IMPLIES SOFTWARE TIMER ROUTINE IS INACTIVE
(1)                                     ;*ACTSTR>0 IMPLIES SOFTWARE TIMER ROUTINE IS ACTIVE
(1)
(1)  027105     000             ACTSTR: .BYTE   0
(1)
(1)                             ;*UNLOAD FLAG (ULDFLG=8 BYTES)
(1)                                     ;*ULDFLG=0 IMPLIES NO UNLOAD COMMAND
(1)                                     ;*ULDFLG>0 IMPLIES UNLOAD COMMAND IN PROGRESS
(1)                                     ;*ULDFLG<0 IMPLIES UNLOAD COMMAND IN WAIT QUEUE
(1)
(1)  027106     000             ULDFLG: .BYTE   0               ;DRIVE 0
(1)  027107     000                     .BYTE   0               ;DRIVE 1
(1)  027110     000                     .BYTE   0               ;DRIVE 2
(1)  027111     000                     .BYTE   0               ;DRIVE 3
(1)  027112     000                     .BYTE   0               ;DRIVE 4
(1)  027113     000                     .BYTE   0               ;DRIVE 5
(1)  027114     000                     .BYTE   0               ;DRIVE 6
(1)  027115     000                     .BYTE   0               ;DRIVE 7
(1)
(1)                             ;*LOOK AHEAD COUNT (LACNT=8 BYTES)
(1)                                     ;*LACNT WILL INDICATE THE NUMBER OF LOOK AHEADS PERFORMED
(1)
(1)  027116     000             LACNT:  .BYTE   0               ;DRIVE 0
(1)  027117     000                     .BYTE   0               ;DRIVE 1
(1)  027120     000                     .BYTE   0               ;DRIVE 2
```

```
(1)  027121  000                        .BYTE   0               ;DRIVE 3
(1)  027122  000                        .BYTE   0               ;DRIVE 4
(1)  027123  000                        .BYTE   0               ;DRIVE 5
(1)  027124  000                        .BYTE   0               ;DRIVE 6
(1)  027125  000                        .BYTE   0               ;DRIVE 7
(1)
(1)                                ;*SAVE REGISTERS FLAG (SAVEFG =1 WORD)
(1)                                        ;*SAVEFG <0 IMPLIES SAVE THE RH11/RPO4 REGISTERS WHEN THE
(1)                                        ;*OPERATION IS COMPLETED AS PER (DPB+14).
(1)                                        ;*SAVEFG=0 IMPLIES SAVE THE RH11/RPO4 REGISTERS, AS PER
(1)                                        ;*(DPB+14), AFTER AN ERROR.
(1)
(1)  027126  000000                SAVEFG: .WORD   0
(1)
(1)                                ;*SEEK FLAG (SEEKFG=1 WORD)
(1)                                        ;*SEEKFG=0 IMPLIES WHEN THE DISK ADDRESS ISN'T IN THE WINDOW
(1)                                        ;*FOR A DATA TRANSFER START A SEARCH COMMAND
(1)                                        ;*SEEKFG<0 IMPLIES DATA TRANSFER WILL DO IMPLIED SEEKS,
(1)                                        ;*DISREGARD THE WINDOW
(1)
(1)  027130  000000                SEEKFG: .WORD   0
(1)
(1)                                ;*TIMEOUT TABLE (TIMER=8 WORDS)
(1)                                        ;*THIS TABLE CONTAINS THE TIME ALLOWED FOR AN OPERATION
(1)
(1)  027132  177777                TIMER:  .WORD   -1      ;DRIVE 0
(1)  027134  177777                        .WORD   -1      ;DRIVE 1
(1)  027136  177777                        .WORD   -1      ;DRIVE 2
(1)  027140  177777                        .WORD   -1      ;DRIVE 3
(1)  027142  177777                        .WORD   -1      ;DRIVE 4
(1)  027144  177777                        .WORD   -1      ;DRIVE 5
(1)  027146  177777                        .WORD   -1      ;DRIVE 6
(1)  027150  177777                        .WORD   -1      ;DRIVE 7
(1)
(1)                                ;*DATA TRANSFER UNDERWAY INDICATOR (DTUW=1 WORD)
(1)                                        ;*DTUW<0 IMPLIES NO DATA TRANSFER UNDERWAY
(1)                                        ;*DTUW=+N (WHERE N=0 TO 7) IMPLIES DATA TRANSFER UNDERWAY ON DRIVE N
(1)
(1)  027152  177777                DTUW:   .WORD   -1
(1)
(1)                                ;*ATTENTION BITS TABLE (ATABIT=8 BYTES)
(1)                                        ;*THIS TABLE CONTAINS THE CORRESPONDING BIT TO EACH DRIVES
(1)                                        ;*ATTENTION BIT
(1)
(1)  027154  001                   ATABIT: .BYTE   1       ;DRIVE 0
(1)  027155  002                           .BYTE   2       ;DRIVE 1
(1)  027156  004                           .BYTE   4       ;DRIVE 2
(1)  027157  010                           .BYTE   10      ;DRIVE 3
(1)  027160  020                           .BYTE   20      ;DRIVE 4
(1)  027161  040                           .BYTE   40      ;DRIVE 5
(1)  027162  100                           .BYTE   100     ;DRIVE 6
(1)  027163  200                           .BYTE   200     ;DRIVE 7
(1)
(1)                                ;*RPO4 TO RH11 "MASS CONTROL BUS PARITY ERRORS" (MCPE) ALLOWED BEFORE
```

```
(1)                                     ;*CALLING IT FATAL (MCPEMX=1 WORD)
(1)
(1)  027164  000003                     MCPEMX: .WORD   3
(1)
(1)                                     ;*STORAGE FOR RPADR (THE FIRST ADDRESS (776700) OF THE RH11/RPO4),
(1)                                     ;*RPVEC (THE VECTOR ADDRESS (254)), AND RPVEC+2 (THE BR LEVEL (5)).
(1)  027166  176700                     RPADR:  .WORD   176700
(1)  027170  000254  000240             RPVEC:  .WORD   254,5*32.
(1)
(1)                                     ;*MAXIMUM NUMBER OF LOOK AHEADS ALLOWED IS 4 (MXLACT=1 WORD)
(1)  027174  000004                     MXLACT: .WORD   4
(1)                                     ;*MAXIMUM DELTA DELAY IS 8 SECTORS (MXDLTA=1 WORD)
(1)  027176  001000                     MXDLTA: .WORD   8.*64.
(1)                                     ;*MINIMUM DELTA DELAY IS 2 SECTORS (MNDLTA=1 WORD)
(1)  027200  000200                     MNDLTA: .WORD   2*64.
(1)                                     ;*MAXIMUM SEARCH FOR I/O WINDOW IS 5 SECTORS (MXWNDW=1 WORD)
(1)  027202  000005                     MXWNDW: .WORD   5
(1)
(1)                                     ;*DEFINITIONS OF THE RH11/RPO4 ADDRESS INDEXES
(1)
(1)          000000                     RHCS1=0                 ;CONTROL AND STATUS REGISTER #1 (DRIVE REG. 00)
(1)          000002                     RHWC=2                  ;WORD COUNT REGISTER (NOT A DRIVE REG)
(1)          000004                     RHBA=4                  ;UNIBUS ADDRESS REGISTER (NOT A DRIVE REG)
(1)          000006                     RHDA=6                  ;DESIRED SECTOR/TRACK ADDRESS REGISTER (DRIVE REG. 05)
(1)          000010                     RHCS2=10                ;CONTROL AND STATUS REGISTER #2 (NOT A DRIVE REG)
(1)          000012                     RHDS1=12                ;DRIVE STATUS REGISTER (DRIVE REG 01)
(1)          000014                     RHER1=14                ;ERROR REGISTER #1 (DRIVE REG. 02)
(1)          000016                     RHAS=16                 ;ATTENTION SUMMARY PSEUDO REGISTER (DRIVE REG. 04)
(1)          000020                     RHLA=20                 ;LOOK AHEAD REGISTER (DRIVE REG. 07)
(1)          000022                     RHDB=22                 ;DATA BUFFER REGISTER (NOT A DRIVE REG.)
(1)          000024                     RHMR=24                 ;MAINTAINABILITY REGISTER (DRIVE REG. 03)
(1)          000026                     RHDT=26                 ;DRIVE TYPE REGISTER (DRIVE REG. 06)
(1)          000030                     RHSN=30                 ;SERIAL NUMBER REGISTER (DRIVE REG. 10)
(1)          000032                     RHOF=32                 ;OFFSET REGISTER (DRIVE REG. 11)
(1)          000034                     RHCA=34                 ;DESIRED CYLINDER ADDRESS REGISTER (DRIVE REG. 12)
(1)          000036                     RHCC=36                 ;CURRENT CYLINDER ADDRESS REGISTER (DRIVE REG. 13)
(1)          000040                     RHER2=40                ;ERROR REGISTER #2 (DRIVE REG. 14)
(1)          000042                     RHER3=42                ;ERROR REGISTER #3 (DRIVE REG. 15)
(1)          000044                     RHEC1=44                ;ECC POSITION REGISTER (DRIVE REG. 16)
(1)          000046                     RHEC2=46                ;ECC PATTERN REGISTER (DRIVE REG.  17)
(1)
(1)                                     ;*RH11/RPO4 DRIVER INIT. CODE
(1)                                             ;*THIS ROUTINE WILL DETERMINE WHICH RPO4 DRIVES ARE
(1)                                             ;*AVAILABLE FOR TESTING AND SET THE DRVSTA INDICATOR
(1)                                             ;*TO THE PROPER STATE FOR EACH DRIVE.
(1)                                             ;*NOTE: THIS ROUTINE CALLS DRVINT
(1)                                     ;
(1)                                     ;*CALL
(1)                                     ;
(1)                                     ;*      JSR     PC,RPINIT
(1)                                     ;*      RETURN
(1)                                     ;
(1)  027204  013746  177776             RPINIT: MOV     @#PS,-(SP)      ;SAVE PROCESSOR STATUS
(1)  027210  013737  027172  177776             MOV     RPVEC+2,@#PS    ;SET PROCESSOR STATUS TO RH11/RPO4 LEVEL
```

```
(1)  027216  004037  034334                    JSR     R0,SAVR15       ;GO SAVE R1-R5
(1)  027222  004737  034072                    JSR     PC,CLRQUE       ;CLEAR ALL REQUEST QUEUES
(1)  027226  012701  027030                    MOV     #RPERRS,R1      ;FIRST ADDRESS TO BE CLEARED
(1)  027232  012702  027130                    MOV     #SEEKFG,R2      ;LAST ADDRESS TO BE CLEARED
(1)  027236  005021                    1$:     CLR     (R1)+           ;CLEAR
(1)  027240  020102                            CMP     R1,R2           ;ARE WE DONE?
(1)  027242  101775                            BLOS    1$              ;BRANCH IF NO
(1)  027244  012702  027152                    MOV     #DTUW,R2        ;LAST ADDRESS
(1)  027250  012721  177777            2$:     MOV     #-1,(R1)+       ;INITIALIZE
(1)  027254  020102                            CMP     R1,R2           ;DONE?
(1)  027256  101774                            BLOS    2$              ;LOOP IF NO
(1)  027260  005001                            CLR     R1              ;DRIVE NUMBER WILL BE IN R1
(1)  027262  013704  027166                    MOV     RPADR,R4        ;FIRST ADDRESS OF RH11/RPO4
(1)  027266  012764  000040  000010            MOV     #BIT05,RHCS2(R4)        ;MASSBUS INIT.
(1)  027274  004037  027346            3$:     JSR     R0,DRVINT       ;GO INIT. A DRIVE
(1)  027300  000417                            BR      7$
(1)  027302  005201                    4$:     INC     R1              ;MOVE TO THE NEXT DRIVE
(1)  027304  042701  177770                    BIC     #^C7,R1         ;DON'T LET THE DRIVE NUMBER GET TO BIG
(1)  027310  001371                            BNE     3$              ;BRANCH IF MORE DRIVES TO INIT.
(1)  027312  013703  027170                    MOV     RPVEC,R3        ;SETUP THE RH11/RPO4 VECTOR
(1)  027316  012723  031356                    MOV     #ISR,(R3)+
(1)  027322  013713  027172                    MOV     RPVEC+2,(R3)
(1)  027326  004037  034354                    JSR     R0,GETR15       ;RESTORE R1-R5
(1)  027332  012637  177776                    MOV     (SP)+,@#PS      ;RESTORE THE PROCESSOR STATUS
(1)  027336  000207                            RTS     PC              ;BYE-BYE
(1)  027340  105061  027050            7$:     CLRB    DRVSTA(R1)      ;SET THE DRIVE STATUS TO OFFLINE
(1)  027344  000756                            BR      4$              ;GO DO THE NEXT DRIVE
(1)
(1)                                    ;*DRIVE INIT. ROUTINE
(1)                                    ;*THIS ROUTINE DETERMINES IF A DRIVE EXIST AND IF IT IS
(1)                                    ;*AN RPO4. IF IT IS, A "READ-IN PRESET" IS ISSUED AND FMT22
(1)                                    ;*IS SET TO A "1". THEN MOL, DPR, DRY, AND VV ARE CHECKED TO
(1)                                    ;*INSURE THEY ARE ALL ON A "1". AND DEPENDING ON THEIR STATE,
(1)                                    ;*DRVSTA IS SET TO THE PROPER CONDITION.
(1)                                    ;*CALL
(1)                                    ;*      MOV     #DRVNUM,R1      ;DRIVE NUMBER TO R1
(1)                                    ;*      MOV     RPADR,R4        ;UNIBUS ADDRESS OF RH11/RPO4 (RHCS1)
(1)                                    ;*      JSR     R0,DRVINT       ;CALLED BY A JSR
(1)                                    ;*      RETURN1                 ;ERROR OCCURRED (PARITY)
(1)                                    ;*      RETURN2                 ;NORMAL RETURN
(1)                                    ;
(1)  027346  004037  034334            DRVINT: JSR     R0,SAVR15       ;SAVE R1-R5
(1)  027352  010164  000010                    MOV     R1,RHCS2(R4)    ;SELECT A DRIVE
(1)  027356  010103                            MOV     R1,R3           ;COPY THE DRIVE NUMBER INTO
(1)  027360  006303                            ASL     R3              ;R3 AND POSITION IT FOR TABLE INDEXING
(1)  027362  112761  177777  027050            MOVB    #-1,DRVSTA(R1)  ;START DRIVE STATUS AS NONEXISTENT
(1)  027370  005063  027060                    CLR     DRVTYP(R3)      ;CLEAR THE DRIVE TYPE INDICATOR
(1)  027374  112777  000111  177564            MOVB    #111,@RPADR     ;DO A "DRIVE CLEAR" COMMAND
(1)  027402  032764  010000  000010            BIT     #BIT12,RHCS2(R4) ;NONEXISTENT DRIVE?
(1)  027410  001403                            BEQ     1$              ;NO---BRANCH
(1)  027412  004737  033526                    JSR     PC,SET.IE       ;GO SET "IE" WITHOUT A "TRE"
(1)  027416  000462                            BR      6$              ;LEAVE THIS ROUTINE
(1)  027420  004037  033134            1$:     JSR     R0,RD.RP        ;READ THE DRIVE TYPE REG.
(1)  027424  000026                            RHDT
```

```
(1)  027426  027566                          7$              ;ERROR RETURN ADDRESS
(1)  027430  012605                          MOV     (SP)+,R5        ;PUT DRIVE TYPE IN R5
(1)  027432  010563  027060                  MOV     R5,DRVTYP(R3)   ;SAVE THE DRIVE TYPE
(1)  027436  022705  020020                  CMP     #20020,R5       ;IS IT A SINGLE PORT RP04?
(1)  027442  001403                          BEQ     2$              ;BRANCH IF YES
(1)  027444  022705  024020                  CMP     #24020,R5       ;IS IT A DUAL PORT RP04?
(1)  027450  001043                          BNE     5$              ;BRANCH IF NO
(1)  027452  012746  000121          2$:     MOV     #121,-(SP)      ;DO A "READ-IN PRESET"
(1)  027456  004037  033272                  JSR     R0,WRT.RP
(1)  027462  000000                          RHCS1
(1)  027464  027566                          7$
(1)  027466  012746  010000                  MOV     #BIT12,-(SP)    ;SET FMT22=1
(1)  027472  004037  033272                  JSR     R0,WRT.RP
(1)  027476  000032                          RHOF
(1)  027500  027566                          7$
(1)  027502  004037  033134                  JSR     R0,RD.RP        ;READ RHDS1
(1)  027506  000012                          RHDS1
(1)  027510  027566                          7$
(1)  027512  012605                          MOV     (SP)+,R5        ;AND SAVE IT IN R5
(1)  027514  100011                          BPL     4$              ;BRANCH IF ATA=0
(1)  027516  116164  027154  000016          MOVB    ATABIT(R1),RHAS(R4) ;CLEAR ATTENTION BIT
(1)  027524  004037  033134                  JSR     R0,RD.RP        ;FIND OUT WHY ATA=1
(1)  027530  000014                          RHER1
(1)  027532  027566                          7$
(1)  027534  006126                          ROL     (SP)+           ;IS IT UNSAFE?
(1)  027536  100412                          BMI     6$              ;BRANCH IF YES
(1)  027540  005105                  4$:     COM     R5              ;CHECK MOL, DPF, DRY, AND VV
(1)  027542  042705  167077                  BIC     #^C<BIT12!BIT08!BIT07!BIT06>,R5
(1)  027546  001004                          BNE     5$              ;BRANCH IF MOL, DPR, DRY, OR VV IS CLEAR
(1)  027550  112761  000001  027050          MOVB    #1,DRVSTA(R1)   ;SET DRIVE STATUS TO ONLINE
(1)  027556  000402                          BR      6$
(1)  027560  105061  027050          5$:     CLRB    DRVSTA(R1)      ;DRIVE STATUS = OFFLINE
(1)  027564  005720                  6$:     TST     (R0)+           ;STEP OVER THE ERROR RETURN
(1)  027566  004037  034354          7$:     JSR     R0,GETR15       ;RESTORE R1-R5
(1)  027572  000200                          RTS     R0              ;RETURN
(1)
(1)                                  ;*REQUEST PRE-PROCESSOR-HANDLES SUBSYSTEM REQUEST
(1)                                  ;
(1)                                  ;*CALL
(1)                                  ;
(1)                                  ;*      JSR     R0,@#RP04       ;CALL THE RP04 DRIVER
(1)                                  ;*      PNTADR                  ;ADDRESS OF POINTER OF DRIVES PARAMETER BLOCK
(1)                                  ;*      RETURN1                 ;RETURN HERE IF QUEUE IS FULL
(1)                                  ;*      RETURN2                 ;RETURN HERE IF REQUEST IS IN QUEUE OR THERE
(1)                                                                  ;IS AN ERROR CONDITION
(1)
(1)  027574  013746  177776          RP04:   MOV     @#PS,-(SP)      ;SAVE THE CALLING STATUS
(1)  027600  013737  027172  177776          MOV     RPVEC+2,@#PS    ;DON'T ALLOW ANY RP04 INTERRUPTS
(1)  027606  112737  000001  027104          MOVB    #1,ACTDRV       ;SET "ACTIVE DRIVER" FLAG
(1)  027614  004037  034334                  JSR     R0,SAVR15       ;SAVE R1-R5
(1)  027620  012002                          MOV     (R0)+,R2        ;PICKUP THE DRIVE PARAMETER BLOCK POINTER
(1)  027622  005062  000016                  CLR     16(R2)          ;CLEAR THE STATUS/ERROR INDICATOR
(1)  027626  111201                          MOVB    (R2),R1         ;PICKUP THE DRIVE NUMBER
(1)  027630  105761  027050                  TSTB    DRVSTA(R1)      ;CHECK DRIVES STATUS
```

```
(1)  027634  003017                          BGT     1$              ;BRANCH IF ONLINE
(1)  027636  105761  027106                 TSTB    ULDFLG(R1)      ;UNLOAD COMMAND IN QUEUE?
(1)  027642  001034                          BNE     3$              ;BRANCH IF YES
(1)  027644  013704  027166                 MOV     RPADR,R4        ;UNIBUS ADDRESS OF RHCS1
(1)  027650  004037  027346                 JSR     R0,DRVINT       ;GO INIT. THE DRIVE
(1)  027654  000442                          BR      6$              ;ERROR RETURN
(1)  027656  105761  027050                 TSTB    DRVSTA(R1)      ;IS DRIVE STATUS ONLINE?
(1)  027662  003004                          BGT     1$              ;BRANCH IF YES
(1)  027664  052762  140000  000016         BIS     #BIT15!BIT14,16(R2) ;ERROR--DRIVE IS OFFLINE OR NONEXISTENT
(1)  027672  000423                          BR      4$              ;GO TO EXIT
(1)  027674  004037  034174          1$:     JSR     R0,DRVQUE       ;PUT THIS REQUEST IN QUEUE
(1)  027700  000421                          BR      5$              ;QUEUE IS FULL
(1)  027702  122762  000103  000002         CMPB    #103,2(R2)      ;IS THIS REQ. FOR AN UNLOAD?
(1)  027710  001003                          BNE     2$              ;BR IF NO
(1)  027712  112761  177777  027106         MOVB    #-1,ULDFLG(R1)  ;SET THE "UNLOAD IN QUEUE" FLAG
(1)  027720  105761  027040          2$:     TSTB    DRVACT(R1)      ;IS THIS DRIVE ACTIVE?
(1)  027724  001006                          BNE     4$              ;BR IS YES
(1)  027726  004737  027770                 JSR     PC,OPT          ;CALL THE OPTIMIZER
(1)  027732  000403                          BR      4$
(1)  027734  052762  120000  000016  3$:     BIS     #BIT15!BIT13,16(R2)     ;SET THE "UNLOAD IN QUEUE" ERROR FLAG
(1)  027742  005720                  4$:     TST     (R0)+
(1)  027744  004037  034354          5$:     JSR     R0,GETR15       ;RESTORE R1-R5
(1)  027750  105037  027104                 CLRB    ACTDRV          ;CLEAR "ACTIVE DRIVER" FLAG
(1)  027754  012637  177776                 MOV     (SP)+,@#PS      ;RETURN "PS" TO USER LEVEL
(1)  027760  000200                          RTS     R0              ;RETURN TO CALLER
(1)  027762  004737  030772          6$:     JSR     PC,CI7          ;GO HANDLE THE PARITY ERROR
(1)  027766  000765                          BR      4$
(1)
(1)                                  ;*OPTIMIZER-CALLED FOR A PARTICULAR DRIVE
(1)                                  ;
(1)                                  ;*CALL
(1)                                  ;*      MOV     #DRVNUM,R1      ;DRIVE NUMBER TO R1
(1)                                  ;*      JSR     PC,OPT          ;SETUP A COMMAND
(1)                                  ;
(1)  027770  004037  034334          OPT:    JSR     R0,SAVR15       ;SAVE R1-R5
(1)  027774  013746  177776                 MOV     @#PS,-(SP)      ;SAVE PROC. STATUS
(1)  030000  146137  027154  027102         BICB    ATABIT(R1),SRCHWT       ;CLEAR "SEARCH WAIT" KEY
(1)  030006  004737  034250                 JSR     PC,GETREQ       ;GET "DPB" POINTER OF REQUEST
(1)  030012  005702                          TST     R2              ;IS THERE A REQUEST IN QUEUE?
(1)  030014  001444                          BEQ     6$              ;NO--BRANCH TO EXIT
(1)  030016  105761  027050                 TSTB    DRVSTA(R1)      ;IS DRIVE ONLINE?
(1)  030022  003006                          BGT     1$              ;YES--BRANCH
(1)  030024  004737  034272                 JSR     PC,POPQUE       ;NO--REMOVE REQUEST FROM QUEUE
(1)  030030  052762  140000  000016         BIS     #BIT15!BIT14,16(R2)     ;SET ERROR BIT OF STATUS/ERROR INDICATOR
(1)  030036  000426                          BR      5$              ;BRANCH TO EXIT
(1)  030040  122762  000150  000002  1$:     CMPB    #150,2(R2)      ;IS THE REQUEST FOR I/O?
(1)  030046  002403                          BLT     2$              ;YES--BRANCH
(1)  030050  004737  030350                 JSR     PC,CI4          ;CALL THE COMMAND INITIATOR
(1)  030054  000417                          BR      5$              ;BRANCH TO EXIT
(1)  030056  005737  027152          2$:     TST     DTUW            ;DATA TRANSFER UNDERWAY?
(1)  030062  002012                          BGE     4$              ;YES--GO START A SEARCH
(1)  030064  005737  027130                 TST     SEEKFG          ;DO IMPLIED SEEKS?
(1)  030070  100404                          BMI     3$              ;YES---BRANCH
(1)  030072  004037  031220                 JSR     R0,LA           ;NO--DO LOOK AHEAD
```

```
(1)  030076  000406                        BR      5$              ;RETURN HERE ON A PARITY ERROR
(1)  030100  000403                        BR      4$              ;GO START A SEARCH
(1)  030102  004737  030134          3$:    JSR     PC,CI1          ;START A DATA TRANSFER
(1)  030106  000402                        BR      5$
(1)  030110  004737  030242          4$:    JSR     PC,CI3          ;START A SEARCH
(1)  030114  012637  177776          5$:    MOV     (SP)+,@#PS      ;RESTORE PROC. STATUS
(1)  030120  004037  034354                 JSR     R0,GETR15       ;RESTORE R1-R5
(1)  030124  000207                        RTS     PC
(1)  030126  004737  033526          6$:    JSR     PC,SET.IE       ;SET "IE" WITHOUT A "TRE"
(1)  030132  000770                        BR      5$              ;EXIT THE ROUTINE
(1)
(1)                                  ;*COMMAND INITIATOR
(1)                                  ;
(1)                                  ;*CALL
(1)                                  ;*      MOV     #DRVNUM,R1      ;DRIVE NUMBER
(1)                                  ;*      MOV     #DPB,R2         ;ADDRESS OF DPB
(1)                                  ;*      JSR     PC,CI?          ;CI?= CI1,CI3, OR CI4
(1)                                  ;*                              ;WHERE:
(1)                                  ;*                              ;CI1=DATA TRANSFER
(1)                                  ;*                              ;CI2=SEARCH REQUESTED BY DATA XFER
(1)                                  ;*                              ;CI4=NOT DATA TRANSFER
(1)
(1)  030134  004737  034272          CI1:   JSR     PC,POPQUE       ;REMOVE REQUEST FROM "DRIVES WAIT" QUEUE
(1)  030140  010237  027100                 MOV     R2,TRNSWT       ;PUT REQ. IN TRANSFER WAIT QUEUE
(1)  030144  010203                  CI2:   MOV     R2,R3           ;DPB ADDRESS TO R3
(1)  030146  013704  027166                 MOV     RPADR,R4        ;RHCS1 ADDRESS
(1)  030152  010164  000010                 MOV     R1,RHCS2(R4)    ;SELECT DRIVE
(1)  030156  062703  000004                 ADD     #4,R3           ;DESIRED WORD COUNT
(1)  030162  062704  000002                 ADD     #2,R4           ;RHWC ADDRESS
(1)  030166  012324                         MOV     (R3)+,(R4)+     ;LOAD WORD COUNT
(1)  030170  012324                         MOV     (R3)+,(R4)+     ;LOAD BUFFER ADDRESS
(1)  030172  012346                         MOV     (R3)+,-(SP)     ;LOAD SECTOR AND TRACK
(1)  030174  004037  033272                 JSR     R0,WRT.RP       ;CALL THE LOAD(WRITE) ROUTINE
(1)  030200  000006                         RHDA                    ;INDEX OF REGISTER TO LOAD
(1)  030202  030772                         CI7                     ;ERROR RETURN ADDRESS
(1)  030204  012346                         MOV     (R3)+,-(SP)     ;LOAD CYLINDER ADDRESS
(1)  030206  004037  033272                 JSR     R0,WRT.RP
(1)  030212  000034                         RHCA
(1)  030214  030772                         CI7
(1)  030216  016246  000002                 MOV     2(R2),-(SP)     ;LOAD "COMMAND+GO", "A17&A16", AND "PSEL"
(1)  030222  004037  033272                 JSR     R0,WRT.RP
(1)  030226  000000                         RHCS1
(1)  030230  030772                         CI7
(1)  030232  010137  027152                 MOV     R1,DTUW         ;SET "DATA TRANSFER UNDERWAY"
(1)  030236  000137  030726                 JMP     CI$
(1)  030242  013704  027166          CI3:   MOV     RPADR,R4        ;RHCS1 ADDRESS
(1)  030246  010164  000010                 MOV     R1,RHCS2(R4)    ;SELECT DRIVE
(1)  030252  016246  000012                 MOV     12(R2),-(SP)    ;DESIRED CYLINDER ADDRESS
(1)  030256  004037  033272                 JSR     R0,WRT.RP
(1)  030262  000034                         RHCA
(1)  030264  030772                         CI7
(1)  030266  116203  000010                 MOVB    10(R2),R3       ;PICKUP SECTOR ADDRESS
(1)  030272  163703  027202                 SUB     MXWNDW,R3       ;BACKUP BY MAX. SEARCH FOR I/O WINDOW
(1)  030276  002002                         BGE     1$
```

```
(1)  030300  062703  000026                    ADD     #22.,R3
(1)  030304  010346                    1$:     MOV     R3,-(SP)        ;COMBINE THE ADJUSTED SECTOR WITH
(1)  030306  116266  000011  000001            MOVB    11(R2),1(SP)    ;THE DESIRED TRACK
(1)  030314  004037  033272                    JSR     R0,WRT.RP       ;LOAD DESIRED TRACK & SECTOR
(1)  030320  000006                            RHDA
(1)  030322  030772                            CI7
(1)  030324  012746  000131                    MOV     #131,-(SP)      ;START A SEARCH
(1)  030330  004037  033272                    JSR     R0,WRT.RP
(1)  030334  000000                            RHCS1
(1)  030336  030772                            CI7
(1)  030340  156137  027154  027102            BISB    ATABIT(R1),SRCHWT       ;SET "SEARCH WAIT" KEY
(1)  030346  000567                            BR      CI5
(1)  030350  013704  027166            CI4:    MOV     RPADR,R4        ;RHCS1 ADDRESS
(1)  030354  010164  000010                    MOV     R1,RHCS2(R4)    ;SELECT DRIVE
(1)  030360  116203  000002                    MOVB    2(R2),R3        ;PICKUP THE REQUESTED COMMAND
(1)  030364  122703  000131                    CMPB    #131,R3         ;IS IT A SEARCH COMMAND?
(1)  030370  001007                            BNE     1$              ;BRANCH IF NO
(1)  030372  016246  000010                    MOV     10(R2),-(SP)    ;LOAD DESIRED TRACK & SECTOR
(1)  030376  004037  033272                    JSR     R0,WRT.RP
(1)  030402  000006                            RHDA
(1)  030404  030772                            CI7
(1)  030406  000403                            BR      2$              ;GO LOAD CYLINDER
(1)  030410  122703  000105            1$:     CMPB    #105,R3         ;IS IT A SEEK COMMAND
(1)  030414  001007                            BNE     3$              ;BRANCH IF NO
(1)  030416  016246  000012            2$:     MOV     12(R2),-(SP)    ;LOAD DESIRED CYLINDER
(1)  030422  004037  033272                    JSR     R0,WRT.RP
(1)  030426  000034                            RHCA
(1)  030430  030772                            CI7
(1)  030432  000546                            BR      CI6
(1)  030434  122703  000115            3$:     CMPB    #115,R3         ;IS IT AN "OFFSET" COMMAND?
(1)  030440  001013                            BNE     4$              ;BR IF NO
(1)  030442  004037  033134                    JSR     R0,RD.RP        ;MERGE THE OFFSET VALUE INTO RHOF
(1)  030446  000032                            RHOF                    ;BUT DON'T CHANGE THE UPPER
(1)  030450  030772                            CI7
(1)  030452  116216  000001                    MOVB    1(R2),(SP)      ;BYTE WHEN LOADING THE
(1)  030456  004037  033272                    JSR     R0,WRT.RP       ;REGISTER (RHOF)
(1)  030462  000032                            RHOF
(1)  030464  030772                            CI7
(1)  030466  000530                            BR      CI6             ;GO START THE COMMAND
(1)  030470  122703  000107            4$:     CMPB    #107,R3         ;IS IT A "RECALIBRATE" COMMAND?
(1)  030474  001525                            BEQ     CI6             ;BRANCH IF YES
(1)  030476  122703  000117                    CMPB    #117,R3         ;IS IT A RETURN TO CENTER?
(1)  030502  001522                            BEQ     CI6             ;BRANCH IF YES
(1)  030504  122703  000103                    CMPB    #103,R3         ;IS IT AN "UNLOAD" COMMAND?
(1)  030510  001016                            BNE     5$              ;BRANCH IF NO
(1)  030512  112761  000001  027040            MOVB    #1,DRVACT(R1)   ;SET THE DRIVE ACTIVE INDICATOR
(1)  030520  105061  027050                    CLRB    DRVSTA(R1)      ;PUT DRIVE STATUS TO OFFLINE
(1)  030524  112761  000001  027106            MOVB    #1,ULDFLG(R1)   ;SET "UNLOAD IN PROGRESS" FLAG
(1)  030532  010346                            MOV     R3,-(SP)        ;START THE "UNLOAD" COMMAND
(1)  030534  004037  033272                    JSR     R0,WRT.RP
(1)  030540  000000                            RHCS1
(1)  030542  030772                            CI7
(1)  030544  000207                            RTS     PC              ;RETURN TO USER
(1)  030546  122703  000143            5$:     CMPB    #143,R3         ;IS IT A "SET FORMAT" COMMAND?
```

```
(1)  030552  001014                          BNE     6$              ;BRANCH IF NO
(1)  030554  004037  033134                  JSR     R0,RD.RP        ;READ THE OFFSET REGISTER
(1)  030560  000032                          RHOF
(1)  030562  030772                          CI7
(1)  030564  116266  000001  000001          MOVB    1(R2),1(SP)     ;COMBINE "FMT22","ECI", AND "HCI"
(1)  030572  004037  033272                  JSR     R0,WRT.RP       ;LOAD "FMT22", "ECI", AND/OR "HCI".
(1)  030576  000032                          RHOF
(1)  030600  030772                          CI7
(1)  030602  000436                          BR      12$
(1)  030604  122703  000141          6$:     CMPB    #141,R3         ;IS IT A "GET REGISTER" COMMAND?
(1)  030610  001023                          BNE     10$             ;BRANCH IF NO
(1)  030612  016203  000006          7$:     MOV     6(R2),R3        ;POINTS TO 1ST ADDRESS OF WHERE
(1)                                                                  ;TO PUT THE REGISTER(S)
(1)  030616  116237  000010  030634          MOVB    10(R2),9$       ;INIT. THE INDEX FOR THE FIRST REG.
(1)  030624  116205  000011                  MOVB    11(R2),R5       ;INDEX OF LAST REG. TO MOVE
(1)  030630  004037  033134          8$:     JSR     R0,RD.RP        ;READ RPO4 REGISTER
(1)  030634  000000          9$:     RHCS1                           ;INDEX OF REG. TO READ
(1)  030636  030772                          CI7
(1)  030640  012623                          MOV     (SP)+,(R3)+     ;GET THE CONTENTS OF RH11/RPO4 REG.
(1)  030642  023705  030634                  CMP     9$,R5           ;LAST REG. BEEN READ?
(1)  030646  001414                          BEQ     12$             ;GET OUT IF YES
(1)  030650  062737  000002  030634          ADD     #2,9$           ;INCREASE THE INDEX BY 2
(1)  030656  000764                          BR      8$              ;LOOP--MORE TO READ
(1)  030660  122703  000145          10$:    CMPB    #145,R3         ;IS IT A "SELECT DRIVE" COMMAND?
(1)  030664  001405                          BEQ     12$             ;BRANCH IF YES
(1)  030666  010346          11$:    MOV     R3,-(SP)        ;LOAD THE COMMAND
(1)  030670  004037  033272                  JSR     R0,WRT.RP
(1)  030674  000000                          RHCS1
(1)  030676  030772                          CI7
(1)  030700  004737  034272          12$:    JSR     PC,POPQUE       ;REMOVE REQ. FROM QUEUE
(1)  030704  052762  000200  000016          BIS     #BIT07,16(R2)   ;SET THE "DONE" BIT
(1)  030712  005737  027126                  TST     SAVEFG          ;SAVE THE RH11/RPO4 REGISTERS?
(1)  030716  100002                          BPL     13$             ;BRANCH IF NO
(1)  030720  004737  033434                  JSR     PC,SVRH11       ;YES--GO SAVE THE REGISTERS
(1)  030724  000207          13$:    RTS     PC              ;RETURN TO USER
(1)  030726  006301          CI5:    ASL     R1
(1)  030730  012761  001750  027132          MOV     #1000.,TIMER(R1)        ;SET A ONE SECOND TIMER
(1)  030736  006201                          ASR     R1
(1)  030740  112761  000001  027040          MOVB    #1,DRVACT(R1)   ;SET THE DRIVE ACTIVE
(1)  030746  000207                          RTS     PC              ;RETURN TO THE USER
(1)  030750  010346          CI6:    MOV     R3,-(SP)        ;LOAD THE COMMAND
(1)  030752  004037  033272                  JSR     R0,WRT.RP
(1)  030756  000000                          RHCS1
(1)  030760  030772                          CI7
(1)  030762  000761                          BR      CI5
(1)  030764  105761  027040          CI7A:   TSTB    DRVACT(R1)              ;IS THE DRIVE ACTIVE?
(1)  030770  001405                          BEQ     CI7B                    ;BRANCH IF NO
(1)  030772  012762  104000  000016  CI7:    MOV     #BIT15!BIT11,16(R2)     ;SET "PARITY" ERROR INDICATOR
(1)  031000  004737  033434                  JSR     PC,SVRH11       ;GO SAVE THE RH11/RPO4 REGISTERS
(1)  031004  012746  000111          CI7B:   MOV     #111,-(SP)      ;DO A "DRIVE CLEAR"
(1)  031010  004037  033272                  JSR     R0,WRT.RP
(1)  031014  000000                          RHCS1
(1)  031016  031056                          CI8
(1)  031020  004737  034154                  JSR     PC,EMPTYQ       ;EMPTY THE QUEUE
```

```
(1)  031024  105061  027106                    CLRB    ULDFLG(R1)          ;CLEAR THE UNLOAD IN QUEUE FLAG
(1)  031030  105061  027040                    CLRB    DRVACT(R1)          ;DRIVE IS IDLE
(1)  031034  020137  027152                    CMP     R1,DTUW             ;IF THIS DRIVE HAD AN I/O REQUEST
(1)  031040  001005                            BNE     1$                  ;IN PROGRESS CLEAR ALL OF THE FLAGS
(1)  031042  005037  027100                    CLR     TRNSWT
(1)  031046  012737  177777  027152            MOV     #-1,DTUW
(1)  031054  000207                    1$:     RTS     PC
(1)  031056  004037  034334            CIB:    JSR     R0,SAVR15           ;SAVE R1-R5
(1)  031062  013704  027166                    MOV     RPADR,R4            ;PICKUP THE ADDRESS OF THE FIRST REGISTER
(1)  031066  005001                            CLR     R1
(1)  031070  005003                            CLR     R3
(1)  031072  105761  027040            1$:     TSTB    DRVACT(R1)          ;DRIVE ACTIVE?
(1)  031076  001421                            BEQ     3$                  ;BRANCH IF NO
(1)  031100  013702  027100                    MOV     TRNSWT,R2           ;GET THE "TRANSFER WAIT" QUEUE
(1)  031104  020137  027152                    CMP     R1,DTUW             ;DID THIS DRIVE HAVE AN I/O IN PROGRESS?
(1)  031110  001402                            BEQ     2$                  ;BRANCH IF YES
(1)  031112  004737  034250                    JSR     PC,GETREQ           ;GET THE DPB POINTER
(1)  031116  004737  033434            2$:     JSR     PC,SVRH11           ;SAVE RH11/RP04 REGISTERS
(1)  031122  052762  102000  000016            BIS     #BIT15!BIT10,16(R2) ;SET "NON-CLEARABLE PARITY" ERROR INDICATOR
(1)  031130  012763  177777  027132            MOV     #-1,TIMER(R3)       ;STOP THE TIMER
(1)  031136  105061  027040                    CLRB    DRVACT(R1)          ;SET "DRIVE ACTIVE" TO IDLE
(1)  031142  105061  027106            3$:     CLRB    ULDFLG(R1)          ;CLEAR UNLOAD FLAG
(1)  031146  005201                            INC     R1                  ;MOVE TO THE NEXT DRIVE
(1)  031150  062703  000002                    ADD     #2,R3
(1)  031154  042701  177770                    BIC     #^C7,R1
(1)  031160  001344                            BNE     1$                  ;BRANCH IF MORE DRIVES
(1)  031162  012737  177777  027152            MOV     #-1,DTUW            ;NO DATA TRANSFERS UNDERWAY
(1)  031170  005037  027100                    CLR     TRNSWT              ;CLEAR THE "TRANSFER WAIT" QUEUE
(1)  031174  004737  034072                    JSR     PC,CLRQUE           ;CLEAR ALL OF THE REQUEST QUEUES
(1)  031200  012764  000040  000010            MOV     #BIT05,RHCS2(R4)    ;DO A MASSBUS INIT.
(1)  031206  004737  033526                    JSR     PC,SET.IE           ;SET "IE" WITHOUT "TRE"
(1)  031212  004037  034354                    JSR     R0,GETR15           ;RESTORE THE REGISTERS
(1)  031216  000207                            RTS     PC                  ;RETURN
(1)
(1)                                    ;*LOOK AHEAD ROUTINE
(1)                                    ;
(1)                                    ;*CALL
(1)                                    ;*      MOV     #DRVNUM,R1          ;DRIVE NUMBER
(1)                                    ;*      MOV     #DPB,R2             ;POINT TO DPB
(1)                                    ;*      JSR     R0,LA               ;GO CHECK THE WINDOW
(1)                                    ;*      RETURN1                     ;ERROR RETURN
(1)                                    ;*      RETURN2                     ;START A SEARCH
(1)                                    ;*      RETURN3                     ;START A DATA TRANSFER
(1)
(1)  031220  013704  027166            LA:     MOV     RPADR,R4            ;GET RHCS1'S ADDRESS
(1)  031224  010164  000010                    MOV     R1,RHCS2(R4)        ;SELECT DRIVE
(1)  031230  004037  033134                    JSR     R0,RD.RP            ;READ CURRENT CYLINDER
(1)  031234  000036                            RHCC
(1)  031236  031350                            4$                          ;ERROR RETURN ADDRESS
(1)  031240  022662  000012                    CMP     (SP)+,12(R2)        ;IS CURRENT CYLINDER=DESIRED
(1)                                                                        ;CYLINDER?
(1)  031244  001037                            BNE     3$                  ;EXIT IF NO
(1)  031246  105261  027116                    INCB    LACNT(R1)           ;INCREMENT THE LOOK AHEAD COUNT
(1)  031252  126137  027116  027174            CMPB    LACNT(R1),MXLACT    ;EXCEED MAX?
```

```
(1)  031260  003026                        BGT     2$              ;BRANCH IF YES
(1)  031262  116203  000010                MOVB    10(R2),R3       ;GET DESIRED SECTOR ADDRESS AND
(1)  031266  000303                        SWAB    R3              ;MULT. BY 64--ALIGN WITH
(1)  031270  006203                        ASR     R3              ;LOOK AHEAD REGISTER
(1)  031272  006203                        ASR     R3
(1)  031274  012737  000340  177776        MOV     #340,@#PS       ;PRIORITY LEVEL "7"
(1)  031302  004037  033134                JSR     R0,RD.RP        ;READ LOOK AHEAD REGISTER
(1)  031306  000020                        RHLA
(1)  031310  031350                        4$
(1)  031312  162603                        SUB     (SP)+,R3        ;CALCULATE THE DELTA
(1)  031314  002002                        BGE     1$
(1)  031316  062703  002600                ADD     #<22.*64.>,R3   ;MAKE THE DELTA POSITIVE
(1)  031322  023703  027176          1$:   CMP     MXDLTA,R3       ;CHECK THE DELTA TO SEE
(1)  031326  002406                        BLT     3$              ;IF IT IS WITHIN THE
(1)  031330  023703  027200                CMP     MNDLTA,R3       ;WINDOW---IF YES, ZERO
(1)  031334  002003                        BGE     3$              ;THE LOOK AHEAD COUNT
(1)  031336  105061  027116          2$:   CLRB    LACNT(R1)       ;AND TAKE THE I/O EXIT
(1)  031342  005720                        TST     (R0)+
(1)  031344  005720                3$:     TST     (R0)+           ;ADJUST THE RETURN ADDRESS
(1)  031346  000200                        RTS     R0
(1)  031350  004737  030772          4$:   JSR     PC,CI7          ;PROCESS THE ERROR
(1)  031354  000200                        RTS     R0              ;TAKE ERROR RETURN
(1)
(1)                                 ;*INTERRUPT SERVICE ROUTINE
(1)
(1)  031356  112737  000001  027104  ISR:   MOVB    #1,ACTDRV       ;SET "ACTIVE DRIVER" FLAG
(1)  031364  004037  034334                JSR     R0,SAVR15       ;SAVE R1-R5
(1)  031370  013704  027166                MOV     RPADR,R4        ;ADDRESS OF RHSCS1
(1)  031374  013701  027152                MOV     DTUW,R1         ;GET "DATA TRANSFER UNDERWAY" INDICATOR
(1)  031400  002403                        BLT     1$              ;BRANCH IF NO DATA TRANSFER UNDERWAY
(1)  031402  004737  031426                JSR     PC,TD           ;CALL TRANSFER DONE
(1)  031406  000402                        BR      2$              ;EXIT
(1)  031410  004737  031554          1$:   JSR     PC,SC           ;CALL SPECIAL CONDITIONS
(1)  031414  004037  034354          2$:   JSR     R0,GETR15       ;RESTORE R1-R5
(1)  031420  105037  027104                CLRB    ACTDRV          ;CLEAR "ACTIVE DRIVER" FLAG
(1)  031424  000002                        RTI                     ;RETURN
(1)
(1)                                 ;*TRANSFER DONE ROUTINE
(1)
(1)  031426  105061  027040          TD:   CLRB    DRVACT(R1)      ;SET DRIVE ACTIVE INDICATOR TO IDLE
(1)  031432  012737  177777  027152        MOV     #-1,DTUW        ;NO DATA TRANSFERS UNDERWAY
(1)  031440  006301                        ASL     R1
(1)  031442  012761  177777  027132        MOV     #-1,TIMER(R1)   ;CANCEL TIMEOUT
(1)  031450  006201                        ASR     R1
(1)  031452  013702  027100                MOV     TRNSWT,R2       ;GET "DPB" ADDRESS FROM THE
(1)  031456  005037  027100                CLR     TRNSWT          ;TRANSFER WAIT QUEUE--CLEAR QUEUE
(1)  031462  052762  000200  000016        BIS     #BIT07,16(R2)   ;SET DONE
(1)  031470  010164  000010                MOV     R1,RHCS2(R4)    ;SELECT THE DRIVE
(1)  031474  004037  033134                JSR     R0,RD.RP        ;TRANSFER ERROR(TRE=1)?
(1)  031500  000000                        RHCS1
(1)  031502  030772                        CI7
(1)  031504  006126                        ROL     (SP)+
(1)  031506  100410                        BMI     2$              ;BR IF YES
(1)  031510  005737  027126                TST     SAVEFG          ;SAVE THE RH11/RP04 REGISTERS?
```

```
(1)  031514  100002                        BPL     1$              ;BRANCH IF NO
(1)  031516  004737  033434                JSR     PC,SVRH11       ;YES--SAVE THE REGISTERS
(1)  031522  004737  027770          1$:   JSR     PC,OPT          ;CALL OPTIMIZER
(1)  031526  000456                        BR      SC2             ;SPECIAL CONDITION (ENTRY #2)
(1)  031530  052762  100100  000016  2$:   BIS     #BIT15!BIT06,16(R2) ;SET DATA ERROR FLAG
(1)  031536  004737  034154                JSR     PC,EMPTYQ       ;EMPTY THE "DRIVES WAIT" QUEUE
(1)  031542  004737  033434                JSR     PC,SVRH11       ;SAVE THE RH11/RP04 REGISTERS
(1)  031546  012714  040111                MOV     #40111,(R4)     ;ISSUE A "DRIVE CLEAR"
(1)  031552  000444                        BR      SC2             ;SPECIAL CONDITION (ENTRY #2)
(1)
(1)                                 ;*SPECIAL CONDITION ROUTINE
(1)
(1)  031554  005001                 SC:    CLR     R1              ;START WITH DRIVE 0
(1)  031556  012702  000001                MOV     #1,R2
(1)  031562  105761  027050          1$:   TSTB    DRVSTA(R1)      ;NONEXISTENT?
(1)  031566  002016                        BGE     2$              ;NO--BRANCH
(1)  031570  005201                        INC     R1              ;YES--MOVE TO THE NEXT DRIVE
(1)  031572  106302                        ASLB    R2              ;MORE DRIVES?
(1)  031574  103372                        BCC     1$              ;YES--BRANCH
(1)  031576  013746  027130                MOV     SEEKFG,-(SP)    ;SAVE THE "SEEK FLAG"
(1)  031602  013746  027126                MOV     SAVEFG,-(SP)    ;SAVE THE "SAVE FLAG"
(1)  031606  004737  027204                JSR     PC,RPINIT       ;GO INIT. THE SUBSYSTEM
(1)  031612  012637  027126                MOV     (SP)+,SAVEFG    ;RESTORE THE "SAVE FLAG"
(1)  031616  012637  027130                MOV     (SP)+,SEEKFG    ;RESTORE THE "SEEK FLAG"
(1)  031622  000405                        BR      SC1             ;TAKE ERROR EXIT
(1)  031624  010164  000010          2$:   MOV     R1,RHCS2(R4)    ;SELECT DRIVE
(1)  031630  116405  000001                MOVB    1(R4),R5        ;IS "SC"=1?
(1)  031634  100413                        BMI     SC2             ;BRANCH IF YES
(1)  031636  012701  000010          SC1:  MOV     #8.,R1          ;DO EIGHT DRIVES
(1)  031642  005301                  1$:   DEC     R1              ;NEXT DRIVE
(1)  031644  002743                        BLT     SC              ;BRANCH IF OUT OF DRIVES
(1)  031646  105761  027040                TSTB    DRVACT(R1)      ;IS THIS DRIVE IDLE?
(1)  031652  001373                        BNE     1$              ;BRANCH IF NO
(2)  031654  104001                        ERROR   1               ;REPORT THE ERROR
(1)  031656  004737  033526                JSR     PC,SET.IE       ;GO SET INTERRUPT ENABLE
(1)  031662  000207                        RTS     PC
(1)  031664  012701  000003          SC2:  MOV     #3,R1           ;READ RHAS UP TO THREE TIMES
(1)  031670  116403  000016          1$:   MOVB    RHAS(R4),R3     ;READ "RHAS"
(1)  031674  001011                        BNE     2$              ;BRANCH IF ANY ATA BITS = 1
(1)  031676  005301                        DEC     R1              ;COUNT THIS READ
(1)  031700  003373                        BGT     1$              ;LOOP IF MORE READS ALLOWED
(1)  031702  004037  033134                JSR     R0,RD.RP        ;READ CONTROL AND STATUS REGISTER
(1)  031706  000000                        RHCS1
(1)  031710  031056                        CI8
(1)  031712  106126                        ROLB    (SP)+           ;IS "IE"=1?
(1)  031714  100350                        BPL     SC1             ;NO--TAKE ERROR EXIT
(1)  031716  000207                        RTS     PC              ;YES--RETURN
(1)  031720  005046                  2$:   CLR     -(SP)           ;PROCESS ALL DRIVES THAT HAVE
(1)  031722  110316                        MOVB    R3,(SP)         ;AN "ATA"=1
(1)  031724  012703  000001                MOV     #1,R3
(1)  031730  005001                        CLR     R1
(1)  031732  030316                  SC4:  BIT     R3,(SP)         ;ATA=1?
(1)  031734  001005                        BNE     SC3             ;YES--BRANCH
(1)  031736  005201                  SC3:  INC     R1              ;MOVE TO THE NEXT DRIVE
```

```
(1)  031740  106303                        ASLB    R3
(1)  031742  001373                        BNE     SC4             ;BRANCH IF MORE TO CHECK?
(1)  031744  005726                        TST     (SP)+           ;CLEAN OFF THE STACK
(1)  031746  000207                        RTS     PC              ;RETURN TO USER
(1)  031750  023701  027152          SC5:  CMP     DTUW,R1         ;IS THIS DRIVE SETUP FOR I/O?
(1)  031754  001002                        BNE     1$              ;NO---BRANCH
(1)  031756  005726                        TST     (SP)+           ;YES---CLEAN OFF THE STACK (RHAS)
(1)  031760  000622                        BR      TD              ;BRANCH TO "TRANSFER DONE"
(1)  031762  105761  027050          1$:   TSTB    DRVSTA(R1)      ;CHECK THE DRIVE STATUS
(1)  031766  003030                        BGT     SC6             ;BRANCH IF ONLINE
(1)  031770  105761  027106                TSTB    ULDFLG(R1)      ;UNLOAD IN PROGRESS?
(1)  031774  003420                        BLE     2$              ;BRANCH IF NO
(1)  031776  004737  034250                JSR     PC,GETREQ       ;GET DPB POINTER
(1)  032002  004737  033434                JSR     PC,SVRH11       ;SAVE THE RH11/RP04 REGISTERS
(1)  032006  004737  032464                JSR     PC,SC12         ;SAVE RHDS1, RHER1, RHER2, AND RHER3
(1)                                                                 ;ALSO DO A DRIVE INIT (DRVINT)
(1)  032012  105761  027050                TSTB    DRVSTA(R1)      ;DID DRIVE COME ONLINE?
(1)  032016  003411                        BLE     3$              ;NO---BRANCH
(1)  032020  032737  040000  027030        BIT     #BIT14,RPERRS   ;WAS THERE AN ERROR?
(1)  032026  001565                        BEQ     SC11            ;NO -- BRANCH
(1)  032030  013705  027032                MOV     RPERRS+2,R5     ;YES -- PICKUP RHER1 AND
(1)  032034  000447                        BR      SC6A            ;GO PROCESS THE ERROR
(1)  032036  004737  032464          2$:   JSR     PC,SC12         ;SAVE RHDS1, RHER1, RHER2, AND RHER3
(1)                                                                 ;ALSO DO A DRVINT
(1)  032042  011605                  3$:   MOV     (SP),R5         ;PICKUP (RHAS) BEFORE THE ERROR CALL
(2)  032044  104005                        ERROR   5               ;REPORT THE ERROR
(1)  032046  000733                        BR      SC3             ;GO CHECK FOR MORE ATA'S
(1)  032050  006301                  SC6:  ASL     R1
(1)  032052  012761  177777  027132        MOV     #-1,TIMER(R1)   ;STOP THE TIMER
(1)  032060  006201                        ASR     R1
(1)  032062  004737  034250                JSR     PC,GETREQ       ;GET THE DPB POINTER FROM THE QUEUE
(1)  032066  010364  000016                MOV     R3,RHAS(R4)     ;CLEAR ATTENTION
(1)  032072  010164  000010                MOV     R1,RHCS2(R4)    ;SELECT DRIVE
(1)  032076  004037  033134                JSR     R0,RD.RP        ;READ THE RP04'S STATUS REG.
(1)  032102  000012                        RHDS1
(1)  032104  032322                        SC8
(1)  032106  011605                        MOV     (SP),R5         ;AND PUT IT IN R5
(1)  032110  006126                        ROL     (SP)+           ;WAS THERE AN ERROR?
(1)  032112  100115                        BPL     SC9             ;BR IF NO
(1)  032114  105761  027040                TSTB    DRVACT(R1)      ;CHECK THE DRIVE ACTIVE INDICATOR
(1)  032120  001522                        BEQ     SC10            ;BRANCH IF IDLE
(1)  032122  004037  033134                JSR     R0,RD.RP        ;READ ERROR REGISTER #1
(1)  032126  000014                        RHER1
(1)  032130  032322                        SC8
(1)  032132  012605                        MOV     (SP)+,R5        ;AND SAVE IT IN R5
(1)  032134  004737  033434                JSR     PC,SVRH11       ;SAVE RH11/RP04 REGISTERS
(1)  032140  012746  000111                MOV     #111,-(SP)      ;ISSUE A DRIVE CLEAR
(1)  032144  004037  033272                JSR     R0,WRT.RP
(1)  032150  000000                        RHCS1
(1)  032152  032322                        SC8
(1)  032154  006105                  SC6A: ROL     R5              ;WAS "UNSAFE" CONDITION =1?
(1)  032156  100404                        BMI     1$              ;BRANCH IF YES
(1)  032160  052762  100240  000016        BIS     #BIT15!BIT07!BIT05,16(R2)       ;INFORM USER OF ERROR
(1)  032166  000443                        BR      SC7
```

```
(1)  032170  004037  033134          1$:   JSR     R0,RD.RP        ;READ DRIVE STATUS REG. #1
(1)  032174  000012                       RHDS1
(1)  032176  032322                       SC8
(1)  032200  011605                       MOV     (SP),R5         ;SAVE RHDS1 IN R5
(1)  032202  006126                       ROL     (SP)+           ;"ERR"=1?
(1)  032204  100013                       BPL     2$              ;BR IF NO--UNSAFE CLEARED
(1)  032206  112761  177777  027050       MOVB    #-1,DRVSTA(R1)  ;DRIVE IS UNSAFE
(1)  032214  004737  033434               JSR     PC,SVRH11       ;SAVE RH11/RP04 REGISTERS
(1)  032220  010364  000016               MOV     R3,RHAS(R4)     ;CLEAR ATTENTION
(1)  032224  052762  110000  000016       BIS     #BIT15!BIT12,16(R2)     ;INFORM USER OF UNSAFE ERROR
(1)  032232  000421                       BR      SC7
(1)  032234  105705                2$:    TSTB    R5              ;"DRY" =1?
(1)  032236  100414                       BMI     3$              ;BRANCH IF YES
(1)  032240  112761  177777  027040       MOVB    #-1,DRVACT(R1)  ;ACTIVE ERROR RECOVER
(1)  032246  112761  000001  027050       MOVB    #1,DRVSTA(R1)   ;ONLINE
(1)  032254  006301                       ASL     R1
(1)  032256  012761  072460  027132       MOV     #30000.,TIMER(R1) ;START 30 SECOND TIMER
(1)  032264  006201                       ASR     R1
(1)  032266  000623                       BR      SC3
(1)  032270  052762  100220  000016  3$:  BIS     #BIT15!BIT07!BIT04,16(R2)       ;INFORM USER OF ERROR
(1)  032276  105061  027040        SC7:   CLRB    DRVACT(R1)      ;DRIVE IS IDLE
(1)  032302  004737  034154               JSR     PC,EMPTYQ       ;DUMP THE QUEUE
(1)  032306  105761  027106               TSTB    ULDFLG(R1)      ;UNLOAD IN PROGRESS OR QUEUE?
(1)  032312  001611                       BEQ     SC3             ;BR IF NO
(1)  032314  105061  027106               CLRB    ULDFLG(R1)      ;CLEAR UNLOAD FLAG
(1)  032320  000606                       BR      SC3
(1)  032322  005726                SC8:   TST     (SP)+           ;REMOVE (RHAS) FROM THE STACK
(1)  032324  105761  027040               TSTB    DRVACT(R1)      ;IS DRIVE IDLE?
(1)  032330  001404                       BEQ     1$              ;YES--BRANCH
(1)  032332  004737  034250               JSR     PC,GETREQ       ;GET DPB POINTER
(1)  032336  000137  030772               JMP     CI7             ;PROCESS THE PARITY ERROR
(1)  032342  000137  031004         1$:   JMP     CI7B            ;PROCESS THE PARITY ERROR
(1)  032346  105761  027040        SC9:   TSTB    DRVACT(R1)      ;TEST DRIVE ACTIVE
(1)  032352  003013                       BGT     SC11            ;BRANCH IF DRIVE IS ACTIVE
(1)  032354  001404                       BEQ     SC10            ;BRANCH IF DRIVE IS IDLE
(1)  032356  052762  100210  000016       BIS     #BIT15!BIT07!BIT03,16(R2) ;INFORM USER OF ERROR RECOVER COMPLETION
(1)  032364  000744                       BR      SC7
(1)  032366  011605                SC10:  MOV     (SP),R5 ;PUT (RHAS) IN R5
(1)  032370  004737  032464               JSR     PC,SC12         ;SAVE RHDS1, RHER1, RHER2, AND RHER3
(2)  032374  104002                       ERROR   2               ;REPORT THE ERROR
(1)  032376  000137  031736               JMP     SC3             ;GO CHECK FOR MORE ATA'S
(1)  032402  105761  027106        SC11:  TSTB    ULDFLG(R1)      ;"UNLOAD IN PROGRESS"?
(1)  032406  003402                       BLE     1$              ;BRANCH IF NO
(1)  032410  105061  027106               CLRB    ULDFLG(R1)      ;CLEAR UNLOAD FLAG
(1)  032414  105061  027040         1$:   CLRB    DRVACT(R1)      ;SET DRIVE IDLE
(1)  032420  136137  027154  027102       BITB    ATABIT(R1),SRCHWT       ;DOING A SEARCH OPERATION FOR
(1)                                                                  ;AN I/O COMMAND?
(1)  032426  001012                       BNE     2$              ;BRANCH IF YES
(1)  032430  004737  034272               JSR     PC,POPQUE       ;REMOVE REQUEST FROM QUEUE
(1)  032434  052762  000200  000016       BIS     #BIT07,16(R2)   ;SET "DONE" BIT
(1)  032442  005737  027126               TST     SAVEFG          ;SAVE THE RH11/RP04 REGISTERS?
(1)  032446  100002                       BPL     2$              ;BRANCH IF NO
(1)  032450  004737  033434               JSR     PC,SVRH11       ;YES--SAVE ALL OF THE RH11/RP04 REG'S
(1)  032454  004737  027770         2$:   JSR     PC,OPT          ;START A REQUEST
```

```
(1)  032460  000137  031736                    JMP     SC3
(1)  032464  010164  000010          SC12:     MOV     R1,RHCS2(R4)     ;SELECT DRIVE
(1)  032470  016437  000012  027030            MOV     RHDS1(R4),RPERRS         ;SAVE THE FOUR REGISTERS THAT
(1)  032476  016437  000014  027032            MOV     RHER1(R4),RPERRS+2       ;WILL TELL US SOMETHING
(1)  032504  016437  000040  027034            MOV     RHER2(R4),RPERRS+4
(1)  032512  016437  000042  027036            MOV     RHER3(R4),RPERRS+6
(1)  032520  004037  027346                    JSR     R0,DRVINT        ;INIT. THE STATE OF THE DRIVE
(1)  032524  000401                            BR      1$               ;TAKE ERROR EXIT
(1)  032526  000207                            RTS     PC               ;RETURN
(1)  032530  005726                  1$:       TST     (SP)+            ;POP PC OFF OF THE STACK
(1)  032532  000673                            BR      SC8              ;PROCESS THE PARITY ERROR
(1)
(1)                                  ;*RP04 TIMER ROUTINE
(1)                                  ;*CALL
(1)                                  ;*       MOV     #TIME,-(SP)      ;ELASPED TIME IN MILLISECONDS ON THE STACK
(1)                                  ;*       JSR     R0,RPTMR         ;CALL RP04 TIME ROUTINE
(1)
(1)  032534  005737  027104          RPTMR:    TST     ACTDRV           ;CHECK "ACTDRV & ACTSTR"
(1)  032540  001032                            BNE     4$               ;IF NON ZERO EXIT
(1)  032542  112737  000001  027105            MOVB    #1,ACTSTR        ;SET "ACTSTR"
(1)  032550  004037  034334                    JSR     R0,SAVR15        ;SAVE R1-R5
(1)  032554  005001                            CLR     R1               ;START WITH DRIVE 0
(1)  032556  005003                            CLR     R3
(1)  032560  005763  027132          1$:       TST     TIMER(R3)        ;IS THE TIMER RUNNING?
(1)  032564  002407                            BLT     2$               ;BRANCH IF NO
(1)  032566  166663  000016  027132            SUB     16(SP),TIMER(R3) ;COUNT THE INTERVAL
(1)  032574  003003                            BGT     2$               ;BR IF NO SOFTWARE TIMEOUT
(1)  032576  004037  032632                    JSR     R0,STO           ;CALL SOFTWARE TIMEOUT ROUTINE
(1)  032602  000405                            BR      3$               ;GO TO THE EXIT
(1)  032604  005201                  2$:       INC     R1               ;MOVE TO NEXT DRIVE
(1)  032606  005723                            TST     (R3)+
(1)  032610  022701  000010                    CMP     #8.,R1           ;OUT OF DRIVES?
(1)  032614  003361                            BGT     1$               ;BRANCH IF NO
(1)  032616  004037  034354          3$:       JSR     R0,GETR15        ;RESTORE R1-R5
(1)  032622  105037  027105                    CLRB    ACTSTR           ;ZERO ACTIVE SOFTWARE TIMEOUT ROUTINE FLAG
(1)  032626  012616                  4$:       MOV     (SP)+,(SP)       ;ADJUST THE STACK
(1)  032630  000200                            RTS     R0               ;RETURN
(1)
(1)                                  ;*SOFTWARE TIMEOUT ROUTINE
(1)                                  ;*
(1)                                  ;*CALL: STO
(1)                                  ;*       MOV     #DRVNUM,R1       ;DRIVE NUMBER
(1)                                  ;*       JSR     R0,STO           ;CALL--DRVACT MUST BE NONZERO
(1)                                  ;*       RETURN
(1)
(1)  032632  013746  177776          STO:      MOV     @#PS,-(SP)       ;SAVE THE PROCESSOR STATUS
(1)  032636  013737  027172  177776            MOV     RPVEC+2,@#PS     ;SET THE "PS" TO RP04 BR LEVEL
(1)  032644  004037  034334                    JSR     R0,SAVR15        ;SAVE R1-R5
(1)  032650  013704  027166                    MOV     RPADR,R4         ;GET ADDRESS OF "RHCS1"
(1)  032654  010164  000010                    MOV     R1,RHCS2(R4)     ;SELECT THE DRIVE
(1)  032660  004037  033134                    JSR     R0,RD.RP         ;READ "DRIVE STATUS REG"
(1)  032664  000012                            RHDS1
(1)  032666  033126                            STO5
(1)  032670  105726                            TSTB    (SP)+            ;IS "DRY"=1?
```

```
(1)  032672  100471                          BMI     ST02            ;BR IF YES
(1)  032674  013702  027100          ST01:    MOV     TRNSWT,R2       ;PICKUP TRANSFER WAIT QUEUE
(1)  032700  020137  027152                   CMP     R1,DTUW         ;TRANSFER UNDERWAY ON THIS DRIVE?
(1)  032704  001402                          BEQ     1$              ;BRANCH IF YES
(1)  032706  004737  034250                   JSR     PC,GETREQ       ;GET DPB ADDRESS
(1)  032712  052762  101000  000016  1$:     BIS     #BIT15!BIT09,16(R2) ;SET THE ERROR FLAGS
(1)  032720  004737  033434                   JSR     PC,SVRH11       ;SAVE RH11/RP04 REGISTERS
(1)  032724  012764  000040  000010          MOV     #BIT05,RHCS2(R4) ;"INIT" THE MASS BUS
(1)  032732  105061  027040                   CLRB    DRVACT(R1)      ;DRIVE IS IDLE
(1)  032736  105061  027106                   CLRB    ULDFLG(R1)      ;CLEAR THE UNLOAD FLAG
(1)  032742  005001                          CLR     R1              ;START WITH DRIVE 0
(1)  032744  005003                          CLR     R3
(1)  032746  004037  027346          2$:     JSR     R0,DRVINT       ;INIT. THIS DRIVE
(1)  032752  000465                          BR      ST05            ;PARITY ERROR RETURN
(1)  032754  105761  027040                   TSTB    DRVACT(R1)      ;DRIVE IDLE BEFORE THE INIT.?
(1)  032760  001414                          BEQ     4$              ;YES--BRANCH
(1)  032762  013702  027100                   MOV     TRNSWT,R2       ;GET TRANSFER WAIT QUEUE
(1)  032766  023701  027152                   CMP     DTUW,R1         ;WAS THERE I/O ON THIS DRIVE?
(1)  032772  001402                          BEQ     3$              ;YES--BRANCH
(1)  032774  004737  034250                   JSR     PC,GETREQ       ;GET THE DPB POINTER FROM QUEUE
(1)  033000  052762  100400  000016  3$:     BIS     #BIT15!BIT08,16(R2) ;INFORM USER OF INIT.
(1)  033006  105061  027040                   CLRB    DRVACT(R1)      ;SET DRIVE ACTIVE TO IDLE
(1)  033012  105061  027106          4$:     CLRB    ULDFLG(R1)      ;NO UNLOAD
(1)  033016  012763  177777  027132           MOV     #-1,TIMER(R3)   ;STOP THE TIMER
(1)  033024  005723                           TST     (R3)+           ;UPDATE THE INDEX
(1)  033026  005201                           INC     R1              ;INCREMENT THE DRIVE NUMBER
(1)  033030  022701  000010                   CMP     #8.,R1          ;LAST DRIVE BEEN CHECKED?
(1)  033034  003344                          BGT     2$              ;NO--LOOP
(1)  033036  012737  177777  027152           MOV     #-1,DTUW        ;NO DATA TRANSFERS UNDERWAY
(1)  033044  005037  027100                   CLR     TRNSWT          ;CLEAR TRANSFER WAIT QUEUE
(1)  033050  004737  034072                   JSR     PC,CLRQUE       ;CLEAR ALL REQUEST QUEUES
(1)  033054  000417                          BR      ST04            ;EXIT
(1)  033056  016405  000016          ST02:    MOV     RHAS(R4),R5     ;IS ATTENTION FOR THIS
(1)  033062  136105  027154                   BITB    ATABIT(R1),R5   ;DRIVE UP?
(1)  033066  001011                          BNE     ST03            ;YES--BRANCH
(1)  033070  020137  027152                   CMP     R1,DTUW         ;DATA TRANSFER UNDERWAY FOR THIS DRIVE
(1)  033074  001277                          BNE     ST01            ;BR IF NO
(1)  033076  004037  033134                   JSR     R0,RD.RP        ;YES--CHECK "RDY"
(1)  033102  000000                          RHCS1
(1)  033104  033126                          ST05
(1)  033106  105726                          TSTB    (SP)+
(1)  033110  100271                          BPL     ST01            ;BR IF "RDY"=0
(1)  033112  005720                  ST03:   TST     (R0)+           ;ADJUST FOR THE PROPER RETURN
(1)  033114  004037  034354          ST04:   JSR     R0,GETR15       ;RESTORE R1-R5
(1)  033120  012637  177776                   MOV     (SP)+,@#PS      ;RESTORE PROCESSOR STATUS
(1)  033124  000200                          RTS     R0              ;RETURN
(1)  033126  004737  031056          ST05:   JSR     PC,CI8          ;GO HANDLE THE PARITY ERROR
(1)  033132  000770                          BR      ST04
(1)
(1)                                  ;*ROUTINE TO READ A RH11/RP04 REGISTER
(1)                                  ;
(1)                                  ;*CALL
(1)                                  ;*      JSR     R0,RD.RP        ;GO READ A REGISTER
(1)                                  ;*      INDEX                   ;REG. INDEX FROM BASE
```

```
(1)                                 ;*      ERRADR                  ;ERROR ADDRESS--PROCESS ERROR STARTING
(1)                                 ;*                              ;AT THIS ADDRESS
(1)                                 ;*      RETURN                  ;CONTENTS OF REG. IS ON THE STACK
(1)                                 ;
(1)  033134  013737  027164  033260  RD.RP:  MOV     MCPEMX,RD.RP2   ;MAX. RETRYS ALLOWED
(1)  033142  011646                          MOV     (SP),-(SP)      ;SAVE R0 FOR RETURN
(1)  033144  013737  027166  033160          MOV     RPADR,RD.ADR    ;FORM THE DESIRED ADDRESS
(1)  033152  062037  033160                  ADD     (R0)+,RD.ADR    ;USING THE BASE AND THE INDEX
(1)  033156  013727                  RD.RP1: MOV     @(PC)+,(PC)+    ;READ THE DESIRED REGISTER OF THE RPO4
(1)  033160  000000                  RD.ADR: .WORD   0               ;ADDRESS IS FORMED HERE
(1)  033162  000000                  RD.WRD: .WORD   0               ;REG. CONTENTS PUT HERE
(1)  033164  013766  033162  000002          MOV     RD.WRD,2(SP)    ;RETURN IT TO THE USER
(1)  033172  017746  173770                  MOV     @RPADR,-(SP)    ;READ RHCS1
(1)  033176  032716  020000                  BIT     #BIT13,(SP)     ;DID MCPE SET?
(1)  033202  001002                          BNE     1$              ;BRANCH IF YES
(1)  033204  022620                          CMP     (SP)+,(R0)+     ;ADJUST FOR RETURN
(1)  033206  000200                          RTS     R0              ;RETURN IF NO
(2)  033210                          1$:
(2)  033210  104003                          ERROR   3               ;REPORT THE ERROR
(1)  033212  005737  027152                  TST     DTUW            ;DATA TRANSFER UNDERWAY?
(1)  033216  100405                          BMI     2$              ;NO--BRANCH
(1)  033220  032716  040000                  BIT     #BIT14,(SP)     ;NO--"TRE"=1?
(1)  033224  001402                          BEQ     2$              ;NO--BRANCH
(1)  033226  005726                          TST     (SP)+           ;YES--CLEAN OFF THE STACK AND
(1)  033230  000415                          BR      RD.RP3          ;TAKE THE FATAL ERROR EXIT
(1)  033232  052716  040000          2$:     BIS     #BIT14,(SP)     ;CLEAR "MCPE" BY SENDING A "1" TO "TRE"
(1)  033236  000316                          SWAB    (SP)            ;POSITION BEFORE WRITING
(1)  033240  013737  027166  033254          MOV     RPADR,3$        ;FORM ADDRESS OF HIGH BYTE
(1)  033246  005237  033254                  INC     3$
(1)  033252  112637                          MOVB    (SP)+,@(PC)+    ;WRITE THE HIGH BYTE OF RHCS1
(1)  033254  000000                  3$:     .WORD   0               ;ADDRESS STORAGE
(1)  033256  005327                          DEC     (PC)+           ;EXCEEDED MAX. RETRYS
(1)  033260  000003                  RD.RP2: .WORD   3
(1)  033262  002335                          BGE     RD.RP1          ;BRANCH IF NO
(1)  033264  011000                  RD.RP3: MOV     (R0),R0         ;FATAL ERROR EXIT
(1)  033266  012616                          MOV     (SP)+,(SP)
(1)  033270  000200                          RTS     R0
(1)
(1)                                 ;*ROUTINE TO WRITE A RH11/RPO4 REGISTER
(1)                                 ;
(1)                                 ;*CALL
(1)                                 ;*      MOV     DATA,-(SP)      ;DATA TO BE LOADED ON THE STACK
(1)                                 ;*      JSR     R0,WRT.RP       ;CALL THE ROUTINE TO LOAD(WRITE) THE REG.
(1)                                 ;*      INDEX                   ;INDEX OF THE REGISTER TO BE LOADED
(1)                                 ;*      ERRADR                  ;ADDRESS TO RETURN TO ON AN ERROR
(1)                                 ;*      RETURN                  ;ERROR FREE RETURN
(1)
(1)  033272  016637  000002  033360  WRT.RP: MOV     2(SP),WRT.WD    ;SAVE THE WORD TO WRITE
(1)  033300  012616                          MOV     (SP)+,(SP)      ;ADJUST THE STACK
(1)  033302  012037  033362                  MOV     (R0)+,WRT.AD    ;GET INDEX OF REGISTER TO BE WRITTEN
(1)  033306  001020                          BNE     2$              ;BRANCH IF NOT RHCS1
(1)  033310  122737  000150  033360          CMPB    #150,WRT.WD     ;IS THE COMMAND FOR DATA TRANSFERS?
(1)  033316  002414                          BLT     2$              ;YES--DON'T GET THE OLD A16&A17, & PSEL
(1)  033320  004037  033134                  JSR     R0,RD.RP        ;NO---COMBINE A16&A17, & PSEL WITH
```

```
(1)  033324  000000                          RHCS1                   ;THE COMMAND BEFORE SENDING IT TO
(1)  033326  033344                          1$                      ;THE RH11/RP04
(1)  033330  000316                          SWAB    (SP)
(1)  033332  042716  177770                 BIC     #^C7,(SP)
(1)  033336  112637  033361                 MOVB    (SP)+,WRT.WD+1
(1)  033342  000402                          BR      2$
(1)  033344  011000                  1$:     MOV     (R0),R0         ;TAKE THE ERROR EXIT
(1)  033346  000200                          RTS     R0
(1)  033350  063737  027166  033362  2$:     ADD     RPADR,WRT.AD    ;FORM THE ADDRESS OF THE DISK REG.
(1)  033356  012737                          MOV     (PC)+,@(PC)+    ;LOAD THE DESIRED REG.
(1)  033360  000000                  WRT.WD: .WORD   0               ;WORD TO WRITE GOES HERE
(1)  033362  000000                  WRT.AD: .WORD   0               ;ADDRESS IS FORMED HERE
(1)  033364  004037  033134                  JSR     R0,RD.RP        ;CHECK FOR PARITY ERROR ON WRITE
(1)  033370  000014                          RHER1
(1)  033372  033424                          2$
(1)  033374  032726  000010                  BIT     #BIT03,(SP)+
(1)  033400  001413                          BEQ     3$              ;BRANCH IF "PAR=0"
(1)  033402  016037  177776  033414          MOV     -2(R0),1$       ;PICKUP THE INDEX
(1)  033410  004037  033134                  JSR     R0,RD.RP        ;READ THE REG.
(1)  033414  000000                  1$:     .WORD   0               ;       REG. INDEX
(1)  033416  033424                          2$                      ;RETURN TO THIS ADDRESS ON ERROR
(2)  033420  104004                          ERROR   4               ;REPORT THE ERROR
(1)  033422  005726                          TST     (SP)+           ;CLEAN OFF THE STACK
(1)  033424  011000                  2$:     MOV     (R0),R0         ;TAKE THE "PARITY ON WRITE" ERROR EXIT
(1)  033426  000200                          RTS     R0
(1)  033430  005720                  3$:     TST     (R0)+           ;ADJUST FOR ERROR FREE EXIT
(1)  033432  000200                          RTS     R0
(1)
(1)                                  ;*ROUTINE TO SAVE THE RH11/RP04 REGISTERS AS PER DPB+14
(1)                                  ;
(1)                                  ;*CALL
(1)                                  ;*      MOV     #DPBNUM,R2      ;DPB POINTER TO R2
(1)                                  ;*      JSR     PC,SVRH11       ;SAVE THE DRIVES REG'S
(1)
(1)  033434  004037  034334          SVRH11: JSR     R0,SAVR15       ;SAVE REG.'S R1-R5
(1)  033440  013704  027166                  MOV     RPADR,R4
(1)  033444  111264  000010                  MOVB    (R2),RHCS2(R4)  ;SELECT DRIVE
(1)  033450  016202  000014                  MOV     14(R2),R2       ;GET THE ERROR TABLE POINTER
(1)  033454  001421                          BEQ     4$              ;EXIT IF 0
(1)  033456  005003                          CLR     R3              ;COUNTER & POINTER
(1)  033460  012705  000022                  MOV     #RHDB,R5        ;PROBLEM REGISTER
(1)  033464  020305                  1$:     CMP     R3,R5           ;REACHED RHDB?
(1)  033466  001005                          BNE     2$              ;BR IF NO
(1)  033470  105764  000010                  TSTB    RHCS2(R4)       ;CHECK "OR"
(1)  033474  100402                          BMI     2$              ;BRANCH IF RHDB CAN BE READ
(1)  033476  005022                          CLR     (R2)+           ;ELSE SAVE IT AS 0'S
(1)  033500  000403                          BR      3$
(1)  033502  010446                  2$:     MOV     R4,-(SP)        ;FORM RH11/RP04 ADDRESS THAT IS
(1)  033504  060316                          ADD     R3,(SP)         ;TO BE READ
(1)  033506  013622                          MOV     @(SP)+,(R2)+    ;AND READ IT
(1)  033510  005723                  3$:     TST     (R3)+           ;MOVE TO NEXT REG INDEX
(1)  033512  020327  000046                  CMP     R3,#RHEC2       ;DONE?
(1)  033516  003762                          BLE     1$              ;BRANCH IF NO
(1)  033520  004037  034354          4$:     JSR     R0,GETR15       ;GET REGISTERS R1-R5
```

```
(1)  033524  000207                          RTS     PC              ;RETURN TO USER
(1)
(1)                                  ;*ROUTINE TO SET THE INTERRUPT WITHOUT GETTING A "TRE"
(1)                                  ;*CALL
(1)                                  ;*      MOV     #DRVNUM,R1      ;DRIVE NUMBER TO R1
(1)                                  ;*      JSR     PC,SET.IE       ;SET "IE"
(1)                                  ;*      RETURN
(1)
(1)  033526  010446                  SET.IE: MOV     R4,-(SP)        ;SAVE R4
(1)  033530  013704  027166                  MOV     RPADR,R4        ;PICKUP ADDRESS OF RHCS1
(1)  033534  010164  000010                  MOV     R1,RHCS2(R4)    ;SELECT DRIVE
(1)  033540  011446                          MOV     (R4),-(SP)      ;READ RHCS1
(1)  033542  052716  040000                  BIS     #BIT14,(SP)     ;SET THE "TRE" BIT OF THE WORD READ
(1)  033546  000316                          SWAB    (SP)            ;ADJUST FOR DATO
(1)  033550  112714  000100                  MOVB    #BIT06,(R4)     ;SET "IE"
(1)  033554  032764  010000  000010          BIT     #BIT12,RHCS2(R4) ;IS "NED"=1?
(1)  033562  001002                          BNE     1$              ;YES--CLEAR "TRE"
(1)  033564  005726                          TST     (SP)+           ;CLEAN OFF THE STACK
(1)  033566  000402                          BR      2$
(1)  033570  112664  000001          1$:     MOVB    (SP)+,1(R4)     ;CLEAR "TRE"
(1)  033574  012604                  2$:     MOV     (SP)+,R4        ;RESTORE R4
(1)  033576  000207                          RTS     PC              ;RETURN TO CALLER
(1)
(1)                                  ;*QUEUE COUNT
(1)  033600      000                 QCNT:   .BYTE   0               ;DRIVE 0
(1)  033601      000                         .BYTE   0               ;DRIVE 1
(1)  033602      000                         .BYTE   0               ;DRIVE 2
(1)  033603      000                         .BYTE   0               ;DRIVE 3
(1)  033604      000                         .BYTE   0               ;DRIVE 4
(1)  033605      000                         .BYTE   0               ;DRIVE 5
(1)  033606      000                         .BYTE   0               ;DRIVE 6
(1)  033607      000                         .BYTE   0               ;DRIVE 7
(1)
(1)                                  ;QUEUE INPUT POINTERS
(1)
(1)  033610  033672                  QINPT:  .WORD   QDRV0            ;DRIVE 0
(1)  033612  033712                          .WORD   QDRV1           ;DRIVE 1
(1)  033614  033732                          .WORD   QDRV2           ;DRIVE 2
(1)  033616  033752                          .WORD   QDRV3           ;DRIVE 3
(1)  033620  033772                          .WORD   QDRV4           ;DRIVE 4
(1)  033622  034012                          .WORD   QDRV5           ;DRIVE 5
(1)  033624  034032                          .WORD   QDRV6           ;DRIVE 6
(1)  033626  034052                          .WORD   QDRV7           ;DRIVE 7
(1)
(1)                                  ;QUEUE OUTPUT POINTERS
(1)
(1)  033630  033672                  QOUTPT: .WORD   QDRV0            ;DRIVE 0
(1)  033632  033712                          .WORD   QDRV1           ;DRIVE 1
(1)  033634  033732                          .WORD   QDRV2           ;DRIVE 2
(1)  033636  033752                          .WORD   QDRV3           ;DRIVE 3
(1)  033640  033772                          .WORD   QDRV4           ;DRIVE 4
(1)  033642  034012                          .WORD   QDRV5           ;DRIVE 5
(1)  033644  034032                          .WORD   QDRV6           ;DRIVE 6
(1)  033646  034052                          .WORD   QDRV7           ;DRIVE 7
```

```
(1)
(1)  033650  033672                QSTART: .WORD   QDRV0           ;DRIVE 0 START ADDRESS
(1)  033652  033712                QSTOP:  .WORD   QDRV1           ;DRIVE 0 STOP ADDRESS & DRIVE 1 START ADDRESS
(1)  033654  033732                        .WORD   QDRV2           ;STOP DRIVE 1--START DRIVE 2
(1)  033656  033752                        .WORD   QDRV3           ;STOP DRIVE 2--START DRIVE 3
(1)  033660  033772                        .WORD   QDRV4           ;STOP DRIVE 3--START DRIVE 4
(1)  033662  034012                        .WORD   QDRV5           ;STOP DRIVE 4--START DRIVE 5
(1)  033664  034032                        .WORD   QDRV6           ;STOP DRIVE 5--START DRIVE 6
(1)  033666  034052                        .WORD   QDRV7           ;STOP DRIVE 6--START DRIVE 7
(1)  033670  034072                        .WORD   QTERM           ;STOP DRIVE 7
(1)
(1)                                ;DRIVE REQUEST QUEUES
(1)
(1)  033672  000010                QDRV0:  .BLKW   10
(1)  033712  000010                QDRV1:  .BLKW   10
(1)  033732  000010                QDRV2:  .BLKW   10
(1)  033752  000010                QDRV3:  .BLKW   10
(1)  033772  000010                QDRV4:  .BLKW   10
(1)  034012  000010                QDRV5:  .BLKW   10
(1)  034032  000010                QDRV6:  .BLKW   10
(1)  034052  000010                QDRV7:  .BLKW   10
(1)          034072                QTERM=.
(1)
(1)                                ;*ROUTINE TO CLEAR ALL OF THE REQUEST QUEUES
(1)                                ;
(1)                                ;*CALL
(1)                                ;*      JSR     PC,CLRQUE
(1)
(1)  034072  004037  034334        CLRQUE: JSR     R0,SAVR15       ;SAVE R1-R5
(1)  034076  012702  033600                MOV     #QCNT,R2        ;ZERO THE QUEUE COUNTS
(1)  034102  005022                        CLR     (R2)+           ;DRIVES 0 & 1
(1)  034104  005022                        CLR     (R2)+           ;DRIVES 2 & 3
(1)  034106  005022                        CLR     (R2)+           ;DRIVES 4 & 5
(1)  034110  005022                        CLR     (R2)+           ;DRIVES 6 & 7
(1)  034112  012703  000010                MOV     #8.,R3          ;MOVE THE STARTING
(1)  034116  012701  033650                MOV     #QSTART,R1      ;ADDRESS OF THE QUEUE INTO
(1)  034122  012122                1$:     MOV     (R1)+,(R2)+     ;THE QUEUE INPUT POINTER
(1)  034124  005303                        DEC     R3
(1)  034126  001375                        BNE     1$
(1)  034130  012703  000010                MOV     #8.,R3          ;MOVE THE STARTING ADDRESS
(1)  034134  012701  033650                MOV     #QSTART,R1      ;OF THE QUEUE INTO THE
(1)  034140  012122                2$:     MOV     (R1)+,(R2)+     ;QUEUE OUTPUT POINTER
(1)  034142  005303                        DEC     R3
(1)  034144  001375                        BNE     2$
(1)  034146  004037  034354                JSR     R0,GETR15       ;RESTORE R1-R5
(1)  034152  000207                        RTS     PC
(1)
(1)                                ;*EMPTY THE QUEUE SPECIFIED BY R1
(1)                                ;
(1)                                ;*CALL
(1)                                ;*      MOV     DRVNUM,R1       ;DRIVE NUMBER TO R1
(1)                                ;*      JSR     PC,EMPTYQ
(1)
(1)  034154  105061  033600        EMPTYQ: CLRB    QCNT(R1)        ;CLEAR NUMBER OF ITEMS IN QUEUE
```

```
(1)  034160  006301                          ASL     R1
(1)  034162  016161  033610  033630          MOV     QINPT(R1),QOUTPT(R1) ;SET OUTPUT QUEUE POINTER=INPUT POINTER
(1)  034170  006201                          ASR     R1
(1)  034172  000207                          RTS     PC
(1)
(1)                                ;*ROUTINE TO PUT A REQUEST IN QUEUE
(1)                                ;
(1)                                ;*CALL
(1)                                ;*      MOV     #DRVNUM,R1      ;DRIVE NUMBER
(1)                                ;*      MOV     #DPB,R2         ;ADDRESS OF PARAMETER BLOCK
(1)                                ;*      JSR     R0,DRVQUE       ;GO PUT REQUEST IN QUEUE
(1)                                ;*      RETURN1                 ;RETURN HERE IF QUEUE IS FULL
(1)                                ;*      RETURN2                 ;RETURN HERE IF REQUEST IS IN QUEUE
(1)
(1)  034174  122761  000010  033600  DRVQUE: CMPB    #10,QCNT(R1)    ;IS QUEUE FULL?
(1)  034202  001421                          BEQ     2$              ;BR IF YES-TAKE RETURN1
(1)  034204  105261  033600                  INCB    QCNT(R1)        ;INCREMENT QUEUE COUNT
(1)  034210  006301                          ASL     R1
(1)  034212  010271  033610                  MOV     R2,@QINPT(R1)   ;PUT THIS REQUEST IN QUEUE
(1)  034216  062761  000002  033610          ADD     #2,QINPT(R1)    ;UPDATE THE QUEUE POINTER
(1)  034224  026161  033610  033652          CMP     QINPT(R1),QSTOP(R1) ;TIME TO RESET THE POINTER
(1)  034232  001003                          BNE     1$              ;BRANCH IF NO
(1)  034234  016161  033650  033610          MOV     QSTART(R1),QINPT(R1) ;YES--RESET POINTER
(1)  034242  006201                  1$:     ASR     R1
(1)  034244  005720                          TST     (R0)+           ;TAKE RETURN 2
(1)  034246  000200                  2$:     RTS     R0              ;RETURN TO USER
(1)
(1)                                ;*ROUTINE TO GET THE "DPB" ADDRESS OF NEXT REQUEST IN QUEUE
(1)                                ;
(1)                                ;*CALL
(1)                                ;*      MOV     #DRVNUM,R1      ;DRIVE NUMBER TO R1
(1)                                ;*      JSR     PC,GETREQ       ;GO GET THE REQUEST
(1)                                ;*      RETURN                  ;R2="DPB" ADDRESS OF THE REQUEST
(1)                                ;*                              ;R2=0 IF NO REQUEST IN QUEUE
(1)
(1)  034250  005002                  GETREQ: CLR     R2
(1)  034252  105761  033600                  TSTB    QCNT(R1)        ;IS THERE ANY REQUEST IN QUEUE?
(1)  034256  001404                          BEQ     2$              ;NO---BRANCH
(1)  034260  006301                  1$:     ASL     R1
(1)  034262  017102  033630                  MOV     @QOUTPT(R1),R2  ;PICKUP "DPB" POINTER FOR THIS DRIVE
(1)  034266  006201                          ASR     R1
(1)  034270  000207                  2$:     RTS     PC              ;RETURN TO USER
(1)
(1)                                ;*ROUTINE TO "POP" THE REQUEST FROM QUEUE
(1)                                ;
(1)                                ;*CALL
(1)                                ;*      MOV     #DRVNUM,R1      ;DRIVE NUMBER TO R1
(1)                                ;*      JSR     PC,POPQUE       ;CALL TO REMOVE REQUEST
(1)                                ;*      RETURN                  ;R2=ADDRESS OF DPB REMOVED
(1)
(1)  034272  105361  033600          POPQUE: DECB    QCNT(R1)        ;DECREMENT QUEUE COUNT
(1)  034276  006301                          ASL     R1
(1)  034300  017102  033630                  MOV     @QOUTPT(R1),R2  ;GET THE "DPB" POINTER
(1)  034304  062761  000002  033630          ADD     #2,QOUTPT(R1)   ;UPDATE THE QUEUE POINTER
```

```
(1)  034312  026161  033630  033652          CMP     QOUTPT(R1),QSTOP(R1) ;TIME TO RESET THE POINTER?
(1)  034320  001003                           BNE     1$                   ;NO--BRANCH TO EXIT
(1)  034322  016161  033650  033630          MOV     QSTART(R1),QOUTPT(R1) ;YES--RESET THE POINTER
(1)  034330  006201                    1$:    ASR     R1
(1)  034332  000207                           RTS     PC                   ;RETURN TO USER
(1)
(1)                                     ;*ROUTINES TO SAVE R0-R5 AND R1-R5
(1)                                     ;*
(1)                                     ;*CALL: SAVR05
(1)                                     ;*      JSR     R0,SAVR05
(1)                                     ;*      RETURN                       ;R0-R5 IS ON THE STACK
(1)                                     ;*
(1)                                     ;*CALL: SAVR15
(1)                                     ;*      JSR     R0,SAVR15
(1)                                     ;*      RETURN                       ;R1-R5 IS ON THE STACK
(1)                                     ;*
(1)                                     ;*UPON RETURN FROM SAVR05 AND SAVR15 THE STACK WILL LOOK LIKE:
(1)                                     ;*
(1)                                     ;*+12   R0
(1)                                     ;*+10   R1
(1)                                     ;*+06   R2
(1)                                     ;*+04   R3
(1)                                     ;*+02   R4
(1)                                     ;*TOP   R5
(1)
(1)  034334                             SAVR05:                              ;SAVE R0 BY THE JSR INST.
(1)  034334  010146                     SAVR15: MOV     R1,-(SP)             ;SAVE R1
(1)  034336  010246                             MOV     R2,-(SP)             ;SAVE R2
(1)  034340  010346                             MOV     R3,-(SP)             ;SAVE R3
(1)  034342  010446                             MOV     R4,-(SP)             ;SAVE R4
(1)  034344  010546                             MOV     R5,-(SP)             ;SAVE R5
(1)  034346  016646  000012                     MOV     12(SP),-(SP)         ;GET R0
(1)  034352  000200                             RTS     R0
(1)
(1)                                     ;*ROUTINES TO RESTORE R0-R5 AND R1-R5
(1)                                     ;*
(1)                                     ;*CALL: GETR05
(1)                                     ;*      JSR     R0,GETR05
(1)                                     ;*      RETURN                       ;R0-R5 HAVE BEEN RESTORED
(1)                                     ;*
(1)                                     ;*CALL: GETR15
(1)                                     ;*      JSR     R0,GETR15
(1)                                     ;*      RETURN                       ;R1-R5 HAVE BEEN RESTORED
(1)                                     ;
(1)  034354  011666  000014             GETR15: MOV     (SP),14(SP)          ;POSITION R0
(1)  034360  005726                     GETR05: TST     (SP)+                ;POP R0 OF THE JSR OFF OF THE STACK
(1)  034362  012605                             MOV     (SP)+,R5             ;RESTORE R5
(1)  034364  012604                             MOV     (SP)+,R4             ;RESTORE R4
(1)  034366  012603                             MOV     (SP)+,R3             ;RESTORE R3
(1)  034370  012602                             MOV     (SP)+,R2             ;RESTORE R2
(1)  034372  012601                             MOV     (SP)+,R1             ;RESTORE R1
(1)  034374  000200                             RTS     R0                   ;RESTORE R0
(1)
8786
```

```
                                        .SBTTL  ASCIZ MESSAGES

034376 000122                   MSG.R:  .ASCIZ  /R/
034400 041506    000            MSG.FC: .ASCIZ  /FC/
034403    114 000103            MSG.LC: .ASCIZ  /LC/
034406 041511    000            MSG.IC: .ASCIZ  /IC/
034411    106 000124            MSG.FT: .ASCIZ  /FT/
034414 052114    000            MSG.LT: .ASCIZ  /LT/
034417    111 000124            MSG.IT: .ASCIZ  /IT/
034422 051506    000            MSG.FS: .ASCIZ  /FS/
034425    114 000123            MSG.LS: .ASCIZ  /LS/
034430 040520 000124            MSG.PAT:        .ASCIZ  /PAT/
034434 000075                   MSG.EQ: .ASCIZ  /=/
034436 005015 047503 052116     MSG.CS: .ASCIZ  <15><12>/CONTROL SWITCHES=/
034444 047522 020114 053523
034452 052111 044103 051505
034460 000075
034462 005015 044122 051503     MRHCS1: .ASCIZ  <15><12>/RHCS1=/
034470 036461    000
034473    015 051012 053110     MRHVEC: .ASCIZ  <15><12>/RHVEC=/
034500 041505 000075
034504 005015 044122 051120     MRHPRI: .ASCIZ  <15><12>/RHPRIC=/
034512 047511 000075

034516 005015 051012 052117     MSG7:   .ASCIZ  <15><12><12>/ROTATIONAL SPEED TIMES/
034524 052101 047511 040516
034532 020114 050123 042505
034540 020104 044524 042515
034546 000123
034550 005015 047412 042516     MSG10A: .ASCIZ  <15><12><12>/ONE CYLINDER SEEK TIMES/<15><12>/** 0, 1, 2,...410 **/
034556 041440 046131 047111
034564 042504 020122 042523
034572 045505 052040 046511
034600 051505 005015 025052
034606 030040 020054 026061
034614 031040 027054 027056
034622 030464 020060 025052
034630    000
034631    015 025012 020052     MSG10B: .ASCIZ  <15><12>/** 410, 409, 408,...0 **/
034636 030464 026060 032040
034644 034460 020054 030064
034652 026070 027056 030056
034660 025040 000052
034664 005015 040412 042526     MSG11A: .ASCIZ  <15><12><12>/AVERAGE SEEK TIMES/<15><12>/** 0 TO 136 **/
034672 040522 042507 051440
034700 042505 020113 044524
034706 042515 006523 025012
034714 020052 020060 047524
034722 030440 033063 025040
034730 000052
034732 005015 025052 030440     MSG11B: .ASCIZ  <15><12>/** 136 TO 0 **/
034740 033063 052040 020117
034746 020060 025052    000
034753    015 005012 040515     MSG12A: .ASCIZ  <15><12><12>/MAXIMUM SEEK TIMES/<15><12>/** 0 TO 410 **/
```

```
                035760  044530  052515  020115
                034766  042523  045505  052040
                034774  046511  051505  005015
                035002  025052  030040  052040
                035010  020117  030464  020060
                035016  025052     000
 8811   035021     015  025012  020052  MSG12B: .ASCIZ  <15><12>/** 410 TO 0 **/
        035026  030464  020060  047524
        035034  030040  025040  000052
 8812
 8813   035042  005015  044515  036516  MSGMIN: .ASCIZ  <15><12>/MIN=/
        035050     000
 8814   035051     015  046412  054101  MSGMAX: .ASCIZ  <15><12>/MAX=/
        035056  000075
 8815   035060  005015  053101  036507  MSGAVG: .ASCIZ  <15><12>/AVG=/
        035066     000
 8816   035067     060  052440  000123  MSG0US: .ASCIZ  /0 US/
 8817   035074  041040  046105  053517  MBELOW: .ASCIZ  / BELOW THE MINIMUM OF /
        035102  052040  042510  046440
        035110  047111  046511  046525
        035116  047440  020106     000
 8818   035123     040  041101  053117  MABOVE: .ASCIZ  / ABOVE THE MAXIMUM OF /
        035130  020105  044124  020105
        035136  040515  044530  052515
        035144  020115  043117  000040
 8819   035152  051440  042505  051513  MSGNUM: .ASCIZ  / SEEKS TIMED/
        035160  052040  046511  042105
        035166     000
 8820   035167     040  047516  020124  MSGNON: .ASCIZ  / NOT TIMED/
        035174  044524  042515  000104
 8821   035202  020040     000          MSG.SP: .ASCIZ  / /             ;TWO (2) SPACES
 8822
 8823                                    .SBTTL  ERROR HEADER (EM) MESSAGES
 8824
 8825   035205     111  046114  043505  EM1:    .ASCIZ  /ILLEGAL RH11(SC=0),INTERRUPT OCCURRED/
        035212  046101  051040  030510
        035220  024061  041523  030075
        035226  026051  047111  042524
        035234  051122  050125  020124
        035242  041517  052503  051122
        035250  042105     000
 8826   035253     111  046114  043505  EM2:    .ASCIZ  /ILLEGAL RP04 INTERRUPT OCCURRED/
        035260  046101  051040  030120
        035266  020064  047111  042524
        035274  051122  050125  020124
        035302  041517  052503  051122
        035310  042105     000
 8827   035313     115  051501  041123  EM3:    .ASCIZ  /MASSBUS PARITY ERROR(MCPE=1)/
        035320  051525  050040  051101
        035326  052111  020131  051105
        035334  047522  024122  041515
        035342  042520  030475  000051
 8828   035350  040515  051523  052502  EM4:    .ASCIZ  /MASSBUS PARITY ERROR(PAR=1)/
        035356  020123  040520  044522
```

```
                035364  054524  042440  051122
                035372  051117  050050  051101
                035400  030475  000051
 8829   035404  043117  046106  047111  EM5:    .ASCIZ  /OFFLINE, NON-EXISTENT, OR UNSAFE DRIVE INTERRUPTED/
        035412  026105  047040  047117
        035420  042455  044530  052123
        035426  047105  026124  047440
        035434  020122  047125  040523
        035442  042506  042040  044522
        035450  042526  044440  052116
        035456  051105  052522  052120
        035464  042105     000
 8830   035467     122  030510  027461  EM10:   .ASCIZ  "RH11/RP04 FAILED TO RESPOND TO ADDRESSING"
        035474  050122  032060  043040
        035502  044501  042514  020104
        035510  047524  051040  051505
        035516  047520  042115  052040
        035524  020117  042101  051104
        035532  051505  044523  043516
        035540     000
 8831   035541     104  044522  042526  EM11:   .ASCIZ  /DRIVE SELECTED IS NOT ONLINE/
        035546  051440  046105  041505
        035554  042524  020104  051511
        035562  047040  052117  047440
        035570  046116  047111  000105
 8832   035576  046511  051120  050117  EM12:   .ASCIZ  /IMPROPER HEADER DATA/
        035604  051105  044040  040505
        035612  042504  020122  040504
        035620  040524     000
 8833   035623     104  052101  020101  EM13:   .ASCIZ  /DATA COMPARE FAILURE/
        035630  047503  050115  051101
        035636  020105  040506  046111
        035644  051125  000105
 8834   035650  044504  045523  042440  EM17:   .ASCIZ  /DISK ERROR IN TIMING TEST/
        035656  051122  051117  044440
        035664  020116  044524  044515
        035672  043516  052040  051505
        035700  000124
 8835   035702  046103  041517  020113  EM20:   .ASCIZ  /CLOCK (KW11-P) OVERFLOW IN TIMING TEST/
        035710  045450  030527  026461
        035716  024520  047440  042526
        035724  043122  047514  020127
        035732  047111  052040  046511
        035740  047111  020107  042524
        035746  052123     000
 8836   035751     122  030510  027461  EM41:   .ASCIZ  "RH11/RP04 ERROR"
        035756  050122  032060  042440
        035764  051122  051117     000
 8837   035771     106  052101  046101  EM46:   .ASCIZ  /FATAL WRITE CHECK ERROR/
        035776  053440  044522  042524
        036004  041440  042510  045503
        036012  042440  051122  051117
        036020     000
 8838
```

```
 8839                                   .SBTTL  STATUS/ERROR INDICATOR MESSAGES
 8840
 8841  036021   117  043106  044514  MSGB14: .ASCIZ  /OFFLINE OR UNSAFE DRIVE REQUESTED/
       036026 042516  047440  020122
       036034 047125  040523  042506
       036042 042040  044522  042526
       036050 051040  050505  042525
       036056 052123  042105     000
 8842  036063   125  046116  040517  MSGB13: .ASCIZ  /UNLOADED DRIVE REQUESTED/
       036070 042504  020104  051104
       036076 053111  020105  042522
       036104 052521  051505  042524
       036112 000104
 8843  036114 042520  051522  051511  MSGB12: .ASCIZ  /PERSISTENT UNSAFE/
       036122 042524  052116  052440
       036130 051516  043101  000105
 8844  036136 040520  044522  054524  MSGB11: .ASCIZ  /PARITY ERROR OCCURRED/
       036144 042440  051122  051117
       036152 047440  041503  051125
       036160 042522  000104
 8845  036164 040506  040524  020114  MSGB10: .ASCIZ  /FATAL PARITY ERROR/
       036172 040520  044522  054524
       036200 042440  051122  051117
       036206    000
 8846  036207   123  043117  053524  MSGB09: .ASCIZ  /SOFTWARE TIMEOUT ON THIS DRIVE/
       036214 051101  020105  044524
       036222 042515  052517  020124
       036230 047117  052040  044510
       036236 020123  051104  053111
       036244 000105
 8847  036246 047523  052106  040527  MSGB08: .ASCIZ  /SOFTWARE TIMEOUT ON ANOTHER DRIVE/
       036254 042522  052040  046511
       036262 047505  052125  047440
       036270 020116  047101  052117
       036276 042510  020122  051104
       036304 053111  000105
 8848  036310 051105  047522  020122  MSGB06: .ASCIZ  "ERROR OCCURRED DURING I/O OPERATION"
       036316 041517  052503  051122
       036324 042105  042040  051125
       036332 047111  020107  027511
       036340 020117  050117  051105
       036346 052101  047511  000116
 8849  036354 051105  047522  020122  MSGB05: .ASCIZ  "ERROR OCCURRED DURING NON-I/O OPERATION"
       036362 041517  052503  051122
       036370 042105  042040  051125
       036376 047111  020107  047516
       036404 026516  027511  020117
       036412 050117  051105  052101
       036420 047511  000116
 8850  036424 047125  040523  042506  MSGB04: .ASCIZ  /UNSAFE OCCURRED/
       036432 047440  041503  051125
       036440 042522  000104
 8851  036444 052501  047524  040515  MSGB03: .ASCIZ  /AUTOMATIC RECALIBRATE SEQUENCE OCCURRED/
       036452 044524  020103  042522
```

```
        036460  040503  044514  051102
        036466  052101  020105  042523
        036474  052521  047105  042503
        036502  047440  041503  051125
        036510  042522  000104
 8852
 8853                                   .SBTTL  DATA HEADER (DT) MESSAGES
 8854
 8855   036514  042524  052123  020040  DH1:    .ASCIZ  /TEST    ERR PC/
        036522  020040  051105  020122
        036530  041520     000
 8856   036533     124  051505  020124  DH2:    .ASCIZ  /TEST    ERR PC  DRIVE   RHAS    RHDS    RHER1   RHER2   RHER3/
        036540  020040  042440  051122
        036546  050040  020103  042040
        036554  044522  042526  020040
        036562  051040  040510  020123
        036570  020040  051040  042110
        036576  020123  020040  051040
        036604  042510  030522  020040
        036612  051040  042510  031122
        036620  020040  051040  042510
        036626  031522     000
 8857   036631     124  051505  020124  DH3:    .ASCIZ  /TEST    ERR PC  ADDRESS DATA/
        036636  020040  042440  051122
        036644  050040  020103  040440
        036652  042104  042522  051523
        036660  042040  052101  000101
 8858   036666  042524  052123  020040  DH4:    .ASCIZ  /TEST    ERR PC  ADDRESS GDDATA  BDDATA/
        036674  020040  051105  020122
        036702  041520  020040  042101
        036710  051104  051505  020123
        036716  042107  040504  040524
        036724  020040  042102  040504
        036732  040524     000
 8859   036735     122  041510  030523  DH10:   .ASCIZ  /RHCS1   ERR PC/
        036742  020040  042440  051122
        036750  050040  000103
 8860   036754  051104  053111  020105  DH11:   .ASCIZ  /DRIVE   ERR PC/
        036762  020040  051105  020122
        036770  041520     000
 8861   036773     124  051505  020124  DH12:   .ASCIZ  /TEST    ERR PC  TST PC  DRIVE   CYLNDR  TRACK   SECTOR/
        037000  020040  042440  051122
        037006  050040  020103  052040
        037014  052123  050040  020103
        037022  042040  044522  042526
        037030  020040  041440  046131
        037036  042116  020122  052040
        037044  040522  045503  020040
        037052  051440  041505  047524
        037060  000122
 8862   037062  042107  054503  020114  DH12A:  .ASCIZ  /GDCYL   GDTRK   GDSCTR  BDCYL   BDTRK   BDSCTR/
        037070  020040  042107  051124
        037076  020113  020040  042107
        037104  041523  051124  020040
```

```
                037112  042102  054503  020114
                037120  020040  042102  051124
                037126  020113  020040  042102
                037134  041523  051124     000
 8863           037141     107  042104  052101  DH13A:  .ASCIZ  /GDDAT   BDDAT   WRDCNT  GDADR   BDADR/
                037146  020040  041040  042104
                037154  052101  020040  053440
                037162  042122  047103  020124
                037170  043440  040504  051104
                037176  020040  041040  040504
                037204  051104     000
 8864           037207     124  051505  020124  DH17:   .ASCIZ  /TEST    ERR PC  DRIVE   RHCS1   RHDS    RHER1   RHER2   RHER3/
                037214  020040  042440  051122
                037222  050040  020103  042040
                037230  044522  042526  020040
                037236  051040  041510  030523
                037244  020040  051040  042110
                037252  020123  020040  051040
                037260  042510  030522  020040
                037266  051040  042510  031122
                037274  020040  051040  042510
                037302  031522     000
 8865           037305     124  051505  020124  DH21:   .ASCIZ  /TEST    ERR PC  TST PC  DRIVE   CYLNDR  TRACK/
                037312  020040  042440  051122
                037320  050040  020103  052040
                037326  052123  050040  020103
                037334  042040  044522  042526
                037342  020040  041440  046131
                037350  042116  020122  052040
                037356  040522  045503     000
 8866           037363     107  042104  052101  DH21A:  .ASCIZ  /GDDAT   BDDAT   WRDCNT  SECTOR/
                037370  020040  041040  042104
                037376  052101  020040  053440
                037404  042122  047103  020124
                037412  051440  041505  047524
                037420  000122
 8867           037422  042524  052123  020040  DH41:   .ASCIZ  /TEST    ERR PC  TST PC  DRIVE/
                037430  020040  051105  020122
                037436  041520  020040  051524
                037444  020124  041520  020040
                037452  051104  053111  000105
 8868           037460  042524  052123  020040  DH42:   .ASCIZ  /TEST    ERR PC  TST PC  DRIVE   RHCS1   RHCS2   RHDS/
                037466  020040  051105  020122
                037474  041520  020040  051524
                037502  020124  041520  020040
                037510  051104  053111  020105
                037516  020040  044122  051503
                037524  020061  020040  044122
                037532  051503  020062  020040
                037540  044122  051504     000
 8869           037545     122  042510  030522  DH43A:  .ASCIZ  /RHER1   RHER2   RHER3/
                037552  020040  051040  042510
                037560  031122  020040  051040
                037566  042510  031522     000
```

```
 8870  037573    122  041510  030523  DH44A:  .ASCIZ  /RHCS1   RHCS2   RHDS    RHCC    RHCA    RHDA/
       037600 020040  051040  041510
       037606 031123  020040  051040
       037614 042110  020123  020040
       037622 051040  041510  020103
       037630 020040  051040  041510
       037636 020101  020040  051040
       037644 042110  000101
 8871  037650 044122  051105  020061  DH44B:  .ASCIZ  /RHER1   RHER2   RHER3/
       037656 020040  044122  051105
       037664 020062  020040  044122
       037672 051105  000063
 8872  037676 044122  051105  020061  DH45A:  .ASCIZ  /RHER1   RHER2   RHER3   RHWC    RHBA    RHDB/
       037704 020040  044122  051105
       037712 020062  020040  044122
       037720 051105  020063  020040
       037726 044122  041527  020040
       037734 020040  044122  040502
       037742 020040  020040  044122
       037750 041104    000
 8873
 8874
 8875        037754                  .EVEN
 8876
 8877                                .SBTTL  DATA TABLE (DT)
 8878
 8879  037754 001170  001116          DT1:    .WORD   $TMPO,$ERRPC
 8880  037760 001170  001116  001156  DT2:    .WORD   $TMPO,$ERRPC,$REG1,$REG5,RPERRS,RPERRS+2,RPERRS+4,RPERRS+6
       037766 001166  027030  027032
       037774 027034  027036
 8881  040000 001170  001116  033160  DT3:    .WORD   $TMPO,$ERRPC,RD.ADR,RD.WRD
       040006 033162
 8882  040010 001170  001116  033362  DT4:    .WORD   $TMPO,$ERRPC,WRT.ADR,WRT.WD,RD.WRD
       040016 033360  033162
 8883  040022 001170  001116  001156  DT5:    .WORD   $TMPO,$ERRPC,$REG1,$REG5,RPERRS,RPERRS+2,RPERRS+4,RPERRS+6
       040030 001166  027030  027032
       040036 027034  027036
 8884  040042 001326  001116          DT10:   .WORD   RH.ADR,$ERRPC
 8885  040046 001160  001116          DT11:   .WORD   $REG2,$ERRPC
 8886  040052 001170  001116  001154  DT12:   .WORD   $TMPO,$ERRPC,$REG0,CHKDRV,CYL.DS,TRK.DS,SEC.DS
       040060 001236  001252  001256
       040066 001254
 8887  040070 001252  001256  001254  DT12A:  .WORD   CYL.DS,TRK.DS,SEC.DS,CYL.RD,TRK.RD,SEC.RD
       040076 001244  001246  001250
 8888  040104 001170  001116  001154  DT13:   .WORD   $TMPO,$ERRPC,$REG0,CHKDRV,CYL.DS,TRK.DS,SEC.DS
       040112 001236  001252  001256
       040120 001254
 8889  040122 001124  001126  001164  DT13A:  .WORD   $GDDAT,$BDDAT,$REG4,$GDADR,$BDADR
       040130 001120  001122
 8890  040134 001170  001116  001236  DT17:   .WORD   $TMPO,$ERRPC,CHKDRV,RP.REG,RP.REG+12,RP.REG+14,RP.REG+40,RP.REG+42
       040142 003540  003552  003554
       040150 003600  003602
 8891  040154 001170  001116  001154  DT21:   .WORD   $TMPO,$ERRPC,$REG0,CHKDRV,CYL.DS,TRK.DS
       040162 001236  001252  001256
```

```
 8892  040170  001156  001126  001164  DT21A:  .WORD   $REG1,$BDDAT,$REG4,$REG1
       040176  001156
 8893  040200  001170  001116  001154  DT41:   .WORD   $TMP0,$ERRPC,$REG0,CHKDRV
       040206  001236
 8894  040210  001170  001116  001154  DT42:   .WORD   $TMP0,$ERRPC,$REG0,CHKDRV,RP.REG,RP.REG+10,RP.REG+12
       040216  001236  003540  003550
       040224  003552
 8895  040226  001170  001116  001154  DT43:   .WORD   $TMP0,$ERRPC,$REG0,CHKDRV,RP.REG,RP.REG+10,RP.REG+12
       040234  001236  003540  003550
       040242  003552
 8896  040244  003554  003600  003602  DT43A:  .WORD   RP.REG+14,RP.REG+40,RP.REG+42
 8897  040252  001170  001116  001154  DT44:   .WORD   $TMP0,$ERRPC,$REG0,CHKDRV,CYL.DS,TRK.DS,SEC.DS
       040260  001236  001252  001256
       040266  001254
 8898  040270  003540  003550  003552  DT44A:  .WORD   RP.REG,RP.REG+10,RP.REG+12,RP.REG+36,RP.REG+34,RP.REG+06
       040276  003576  003574  003546
 8899  040304  003554  003600  003602  DT44B:  .WORD   RP.REG+14,RP.REG+40,RP.REG+42
 8900  040312  001170  001116  001154  DT45:   .WORD   $TMP0,$ERRPC,$REG0,CHKDRV,CYL.DS,TRK.DS,SEC.DS
       040320  001236  001252  001256
       040326  001254
 8901  040330  003540  003550  003552  DT45A:  .WORD   RP.REG,RP.REG+10,RP.REG+12,RP.REG+36,RP.REG+34,RP.REG+06
       040336  003576  003574  003546
 8902  040344  003554  003600  003602  DT45B:  .WORD   RP.REG+14,RP.REG+40,RP.REG+42,RP.REG+2,RP.REG+4,RP.REG+22
       040352  003542  003544  003562
 8903
 8904                                  .SBTTL  DATA FORMAT (DF) TABLE
 8905
 8906  040360  000001                  DF1:    .WORD   1               ;NUMBER OF DATA HEADERS
 8907  040362     002                          .BYTE   2               ;NUMBER OF WORDS IN DATA TABLE
 8908  040363     000                          .BYTE   0               ;BOTH NUMBERS ARE OCTAL
 8909
 8910  040364  000001                  DF2:    .WORD   1
 8911  040366     010                          .BYTE   ^D8
 8912  040367     000                          .BYTE   0
 8913
 8914  040370  000001                  DF3:    .WORD   1
 8915  040372     004                          .BYTE   4
 8916  040373     000                          .BYTE   0
 8917
 8918  040374  000001                  DF4:    .WORD   1
 8919  040376     005                          .BYTE   5
 8920  040377     000                          .BYTE   0
 8921
 8922  040400  000001                  DF10:   .WORD   1
 8923  040402     002                          .BYTE   2
 8924  040403     000                          .BYTE   0
 8925
 8926  040404  000001                  DF11:   .WORD   1
 8927  040406     002                          .BYTE   2
 8928  040407     000                          .BYTE   0
 8929
 8930  040410  000002                  DF12:   .WORD   2               ;2 DH'S TO BE TYPED
 8931  040412     007                          .BYTE   7               ;7 DATA WORDS FOLLOW THE 1ST DH
 8932  040413     160                          .BYTE   160             ;WORDS 1-4 ARE OCTAL 5-7 ARE DECIMAL
```

```
8933  040414  037062                      .WORD   DH12A          ;ADDRESS OF 2ND DH
8934  040416     006                      .BYTE   6              ;6 DATA WORDS FOLLOW THE 2ND DH
8935  040417     000                      .BYTE   0              ;ALL WORDS ARE OCTAL
8936
8937  040420  000002          DF13:       .WORD   2
8938  040422     007                      .BYTE   7
8939  040423     160                      .BYTE   160
8940  040424  037141                      .WORD   DH13A
8941  040426     005                      .BYTE   5
8942  040427     004                      .BYTE   4              ;WORD 3 IS DECIMAL
8943
8944  040430  000000          DF14:       .WORD   0
8945  040432     005                      .BYTE   5
8946  040433     004                      .BYTE   4              ;WORD 3 IS DECIMAL
8947
8948  040434  000001          DF17:       .WORD   1
8949  040436     010                      .BYTE   ↑D8
8950  040437     000                      .BYTE   0
8951
8952  040440  000002          DF21:       .WORD   2
8953  040442     006                      .BYTE   6
8954  040443     060                      .BYTE   60
8955  040444  037363                      .WORD   DH21A
8956  040446     004                      .BYTE   4
8957  040447     014                      .BYTE   14
8958
8959  040450  000000          DF22:       .WORD   0
8960  040452     004                      .BYTE   4
8961  040453     014                      .BYTE   14
8962
8963  040454  000001          DF41:       .WORD   1
8964  040456     004                      .BYTE   4
8965  040457     000                      .BYTE   0
8966
8967  040460  000001          DF42:       .WORD   1
8968  040462     007                      .BYTE   7
8969  040463     000                      .BYTE   0
8970
8971  040464  000002          DF43:       .WORD   2
8972  040466     007                      .BYTE   7
8973  040467     000                      .BYTE   0
8974  040470  037545                      .WORD   DH43A
8975  040472     003                      .BYTE   3
8976  040473     000                      .BYTE   0
8977
8978  040474  000003          DF44:       .WORD   3
8979  040476     007                      .BYTE   7
8980  040477     160                      .BYTE   160
8981  040500  037573                      .WORD   DH44A
8982  040502     006                      .BYTE   6
8983  040503     000                      .BYTE   0
8984  040504  037650                      .WORD   DH44B
8985  040506     003                      .BYTE   3
8986  040507     000                      .BYTE   0
```

```
8987
8988   040510  000003                  DF45:   .WORD   3
8989   040512     007                          .BYTE   7
8990   040513     160                          .BYTE   160
8991   040514  037573                          .WORD   DH44A
8992   040516     006                          .BYTE   6
8993   040517     000                          .BYTE   0
8994   040520  037676                          .WORD   DH45A
8995   040522     006                          .BYTE   6
8996   040523     000                          .BYTE   0
8997
8998                                   .EVEN
8999           040524                  BUFFER=.
9000
9001           000001                  .END
```

```
ACTDRV  027104          8785**
ACTSTR  027105          8785**
ATABIT  027154          6203    8351    8356    8785*
BITS    001374          5715*   6335    6357    6382    6409    6433    6497    6526    6578    6651    6712    6777    6855
                        6903    7012    7046    7112    8406    8424    8427    8429    8479    8596
                        4968*
BIT0  = 000001          4968*   5715
BIT00 = 000001          4968*   5715
BIT01 = 000002          4968*   5715
BIT02 = 000004          4968*   5715
BIT03 = 000010          4968*   5715    7569    8785
BIT04 = 000020          4968*   5715    7566    8785
BIT05 = 000040          4968*   5715    6607    6616    6626    6629    6674    6693    6696    6734    6747    6759    6762
                        6799    6812    6824    6827    7569    7682    8785
BIT06 = 000100          4968*   5715    7272    7572    8785
BIT07 = 000200          4968*   5715    8785
BIT08 = 000400          4968*   5715    7560    8785
BIT09 = 001000          4968*   5715    7169    7273    7575    8785
BIT1  = 000002          4968*
BIT10 = 002000          4968*   5715    7169    7563    8615    8785
BIT11 = 004000          4968*   5715    7273    7563    8785
BIT12 = 010000          4968*   5715    7566    8785
BIT13 = 020000          4968*   5715    7169    7560    8785
BIT14 = 040000          4968*   5715    6605    6614    6672    6732    6745    6797    6810    7496    7560    8785
BIT15 = 100000          4968*   5715    7722    7723    8785
BIT2  = 000004          4968*
BIT3  = 000010          4968*
BIT4  = 000020          4968*
BIT5  = 000040          4968*
BIT6  = 000100          4968*
BIT7  = 000200          4968*
BIT8  = 000400          4968*
BIT9  = 001000          4968*
BPTVEC= 000014          4968*
BUFFER= 040524          5715    6266    6864    6872    6881    6911    6919    6931    7035    7118    7139    7145    7151
                        7155    7161    7650*   7651    7653    7655    7656    7657    7721    7760    7883    7904    7905
                        7931    7992    8013    8027    8070    8089    8090    8107    8999*
BUSADR  001216          5715*   6155*   6157*   6172*   6174*   8237    8239*
BYPASS  001234          5715*   6252*   6340*   6364*   6385*   6412*   6436*   6500*   6529*   6856*   6904*   7036*   7113*
                        7408    7430    7459    7530    7664    7949    8055    8124
CALL.A  017702          6254    6256    6260    6342    7399*   7402
CALL.B  020010          6343    6367    6389    6395    6415    6418    6445    6451    6465    6471    6505    6507    6537
                        6539    6541    6543    6545    6547    7421*   7424
CALL.C  020154          6366    7450*   7453
CHKDRV  001236          5715*   6233*   6237*   6242    7165    8886    8888    8890    8891    8893    8894    8895    8897
                        8900
CI1     030134          8785*
CI2     030144          8785*
CI3     030242          8785*
CI4     030350          8785*
CI5     030726          8785*
CI6     030750          8785*
CI7     030772          8785*
CI7A    030764          8785*
CI7B    031004          8785*
```

DZRPKB.P11 CROSS REFERENCE TABLE

```
CI8     031056  8785#
CKSCTR  022462  6871    7928#
CK.CHR  026454  8312    8348    8512    8533    8554    8674#   8713    8720    8723    8770
CK.DEC  026426  8650#   8682
CK.DIG  026530  8350    8524    8707#
CK.NUM  026710  8313    8541    8555    8753#
CK.OCT  026400  8392    8396    8410    8414    8630#   8755    8765
CLKSTA  001226  5715#   6228    6579    6652    6713    6778    7326*   7328*   7334*
CLOSE   026314  8586    8590    8599#
CLRBUF  022410  6868    7902#
CLRQUE  034072  8785#
CLSWDS  026212  8570    8574    8577#
CNTRLC  001214  5715#   6159*   6175*   6184    6194
COUNT   021606  6618    6676    6737    6750    6802    6815    7739#
CYL.DS  001252  5715#   7407*   7429*   7458*   7491*   7509*   7525*   7660*   7694*   7965*   8040*   8886    8887
                8888    8891    8897    8900
CYL.RD  001244  5715#   7655*   8887
C.SWR   001212  5715#   6208    6438    6592    6632    7120    7341    7351    7361    7381    7436    7465    7531
                7601    8217    8221*
DATCMP  023132  7080    8012#
DECSK   006054  6394#   6398
DFLT    001614  5715#   6160
DF1     040360  5732    8906#
DF10    040400  5796    8922#
DF11    040404  5806    8926#
DF12    040410  5819    8930#
DF13    040420  5832    5853    8937#
DF14    040430  5840    5861    8944#
DF17    040434  5871    5881    8948#
DF2     040364  5742    5772    8910#
DF21    040440  5894    8952#
DF22    040450  5902    8959#
DF3     040370  5752    8914#
DF4     040374  5762    8918#
DF41    040454  5932    8963#
DF42    040460  5942    8967#
DF43    040464  5954    8971#
DF44    040474  5969    8978#
DF45    040510  5984    5999    8988#
DH1     036514  5730    8855#
DH10    036735  5794    8859#
DH11    036754  5804    8860#
DH12    036773  5817    5830    5851    5967    5982    5997    8861#
DH12A   037062  8862#   8933
DH13A   037141  8863#   8940
DH17    037207  5869    5879    8864#
DH2     036533  5740    5770    8856#
DH21    037305  5892    8865#
DH21A   037363  8866#   8955
DH3     036631  5750    8857#
DH4     036666  5760    8858#
DH41    037422  5930    8867#
DH42    037460  5940    5952    8868#
DH43A   037545  8869#   8974
```

```
DH44A   037573          8870#   8981    8991
DH44B   037650          8871#   8984
DH45A   037676          8872#   8994
DISPLA= 177570          4968#   6335*   6357*   6382*   6409*   6433*   6497*   6526*   6578*   6651*   6712*   6777*   6855*
                        6903*   7012*   7112*   7169*   7273*
DORTI   021540          6599    6665    6725    6790    7705#
DPB.A   003440          5715#   6247*   6253*   6255*   6257*   6258*   6259*   6336*   7401    7403    7407
DPB.B   003460          5715#   6248*   6337*   6338*   6339*   6358*   6360*   6362*   6383*   6384*   6387*   6390*   6391
                        6393*   6396*   6397    6410*   6411*   6414*   6417*   6434*   6435*   6444*   6450*   6464*   6470*
                        6498*   6499*   6504*   6506*   6527*   6528*   6536*   6538*   6540*   6542*   6544*   6546*   7383*
                        7386*   7423    7425    7429    7432    7435
DPB.C   003500          5715#   6249*   6359*   6361*   6363*   7384*   7387*   7452    7454    7458    7461    7464
DRVACT  027040          8785#*
DRVCAL  020320          6865    6867    6870    6878    6883    6914    6923    6937    6941    7065    7069    7077    7124
                        7142    7146    7152    7156    7162    7482#   7486
DRVCLR= 000111          4975#
DRVINT  027346          8785#
DRVMSK  001240          5715#   6234*   6235    6238*
DRVOK   005130          6236    6242#
DRVQUE  034174          8785#
DRVSEL  001222          5715#   6198*   6203*   5214    6235    7165    8285*   8344*   8351*   8356*
DRVSTA  027050          6201    6243    8785#*
DRVTYP  027060          8785#*
DTADPB  003520          5715#   6250*   6607    6616    6626    6629    6661*   6667    6674    6680*   6681    6685*   6686
                        6693    6696    6734    6747    6759    6762    6799    6812    6824    6827    6859*   6860*   6861*
                        6862*   6863*   6864*   6866*   6869*   6874*   6875*   6876*   6877*   6879*   6880*   6891*   6882*
                        6907*   6908*   6909*   6910*   6911*   6913*   6915*   6919*   6921*   6922*   6924*   6929*   6931*
                        6934*   6936*   6938*   6939*   6940*   6942    7034*   7035*   7049*   7050*   7051*   7056*   7064*
                        7068*   7076*   7116*   7117*   7118*   7119*   7123*   7125*   7126    7128*   7129    7131*   7136
                        7137    7138*   7140*   7144*   7145*   7148*   7149*   7150*   7151*   7154*   7155*   7158*   7159*
                        7160*   7161*   7485    7487    7491    7494    7500*   7503    7505    7509    7510*   7512    7514
                        7518    7520    7525    7683*   7684*   7685*   7688    7690    7694    7965    7966    7993    8014
                        8040    8041    8042    8198*   8199*   8200*
DTUW    027152          8785#*
DT1     037754          5731    8879#
DT10    040042          5795    8884#
DT11    040046          5805    8885#
DT12    040052          5818    8886#
DT12A   040070          8887#
DT13    040104          5831    5852    8888#
DT13A   040122          5839    5860    8889#
DT17    040134          5870    5880    8890#
DT2     037760          5741    5771    8880#
DT21    040154          5893    8891#
DT21A   040170          5901    8892#
DT3     040000          5751    8881#
DT4     040010          5761    8882#
DT41    040200          5931    8893#
DT42    040210          5941    8894#
DT43    040226          5953    8895#
DT43A   040244          8896#
DT44    040252          5968    8897#
DT44A   040270          8898#
DT44B   040304          8899#
```

```
DT45    040312          5983    5998    8900#
DT45A   040330          8901#
DT45B   040344          8902#
DT5     040022          8883#
EMPTYQ  034154          8785#
EMTVEC= 000030          4968#   6177*
EM1     035205          5729    8825#
EM10    035467          5793    8830#
EM11    035541          5803    8831#
EM12    035576          5816    8832#
EM13    035623          5829    5850    5831    8833#
EM17    035650          5868    8834#
EM2     035253          5739    8826#
EM20    035702          5878    8835#
EM3     035313          5749    8827#
EM4     035350          5759    8828#
EM41    035751          5929    5939    5951    5966    5981    8836#
EM46    035771          5996    8837#
EM5     035404          5769    8829#
ERINDX  020766          7407    7429    7458    7491    7509    7525    7554#   7694
ERR   = ****** U        8785
ERRVEC= 000004          4968#   7273*   7293*   7294*   7299*   7321    8270    8271*   8273*   8280*
ERR.CT  001372          5715#   7528    7947    7977    8053    8116
EXIT0   005552          6340    6344#
EXIT1   005714          6364    6368#
EXIT10  010276          6657    6659    6688    6695#
EXIT11  010776          6718    6720    6754    6761#
EXIT12  011476          6783    6785    6819    6826#
EXIT13  012032          6856    6888#
EXIT14  012366          6904    6947#
EXIT15  013026          7036    7044    7084#
EXIT16  013474          7113    7163#
EXIT2   006076          6385    6399#
EXIT3   006236          6412    6422#
EXIT4   006570          6436    6483#
EXIT5   006746          6500    6511    6514#
EXIT6   007170          6529    6535    6550#
EXIT7   007634          6584    6589    6621    6628#
FC      001440          5715#
FC1     002100          5715#   6362
FC13    002256          5715#   6860
FC14    002266          5715#   6908
FC15    002276          5715#   7038
FC16    002324          5715#   7116    8175    8193
FC2     002122          5715#   6387    6397
FC3     002140          5715#   6414
FC4     002156          5715#   6442    6450    6481
FC5     002174          5715#   6502    6512
FC6     002212          5715#   6531
FC7     002230          5715#   6585
FILBUF  022346          6858    6906    7881#
FILRAN  023454          7122    8070#
FMT22 = 010000          4996#   6253    7650
FS      001454          5715#
```

```
FS0     002072          5715#   6337
FS1     002112          5715#   6358
FS14    002270          5715#   6909
FS15    002312          5715#   7016    7029    7034
FS2     002132          5715#   6383
FS3     002150          5715#   6410
FS4     002166          5715#   6434
FS5     002204          5715#   6498
FS6     002222          5715#   6527
FS7     002234          5715#   6586
FT      001446          5715#
FT0     002070          5715#   6338
FT1     002106          5715#   6360
FT13    002260          5715#   6861
FT15    002304          5715#   7048
FT2     002130          5715#   6384
FT3     002146          5715#   6411
FT4     002164          5715#   6435
FT5     002202          5715#   6499
FT6     002220          5715#   6528
FT7     002232          5715#   6587
GETADR  024200          6190    8237#
GETNUM  024420          8218    8243    8250    8261    8306#
GETREG= 000141          4988#   6259
GETREQ  034250          8785#
GETR0S  034360          7264    7891    8024    8357    8475    8571    8575    8594    8617    8785#
GETR15  034354          7916    7951    8000    8785#
GETSWR  024112          6191    8213#   8222
GNS   = ****** U        4955    6186    6211    6216    6224    6232    6261    6264    7165    7211    7225    7275    8215
                        8345    8360    8481    8548
GTTST1  024706          8355    8360#   8404    8411    8422    8426    8438    8444    8450    8456    8463    8476
GTTST2  024760          8371#   8447
GTTST3  025222          8375    8379    8383    8387    8391    8432#
GTTST4  025224          8393    8408    8431    8433#
GTTST5  025276          8436    8451#
GTTST6  025326          8453    8460#
GT.PRM  024550          6196    8343#
GT.PR1  024554          8344#   8348    8350
GT.PR2  024700          8357#   8368    8459    8466    8595
IC      001444          5715#
IC1     002104          5715#
IC15    002302          5715#   7865
IC2     002126          5715#   6390    6396
IC3     002144          5715#   6416
IC4     002162          5715#   6460    6480
IC5     002200          5715#   6508    6509
IC6     002216          5715#   6548
INCCYL  022316          7043    7863#
INCSK   006024          6388#   6392
INCTRK  022266          7081    7844#
IOTVEC= 000020          4968#   6177*
ISR     031356          6631    6698    6764    6829    7680    8785#
IT      001452          5715#
ITEM41  004030          5922#   7197
```

DZRPKB.P11 CROSS REFERENCE TABLE

```
IT15    002310          5715#   7846
LA      031220          8785#
LACNT   027116          8785#*
LC      001442          5715#
LC0     002066          5715#   6339
LC1     002102          5715#   6363
LC15    002300          5715#   7863    7866    7868
LC16    002326          5715#   7129    8188
LC2     002124          5715#   6391    6393
LC3     002142          5715#   6420
LC4     002160          5715#   6461    6463    6470
LC5     002176          5715#   6503    6510
LC6     002214          5715#   6532
LDCMD   017636          6251    7381#
LKS     001352          5715#   7333    7365*
LKV     001346          5715#   7363*   7364*
LPB     001362          5715#   7169
LPS     001360          5715#   7169    7295
LPTAVL  001220          5715#   7169    7292*   7296*
LP.AVL  017316          6187    7292#
LS      001456          5715#
LS1     002114          5715#   6359
LS15    002314          5715#   7015
LT      001450          5715#
LT1     002110          5715#   6361
LT15    002306          5715#   7844    7847    7849
MABOVE  035123          7812    8818#
MBELOW  035074          7802    8817#
MCPEMX  027164          8785#
MNDLTA  027200          8785#
MRHCS1  034462          8241    8801#
MRHPRI  034504          8255    8803#
MRHVEC  034473          8248    8802#
MSGAVG  035060          7815    8815#
MSGB03  036444          5715    8851#
MSGB04  036424          5715    8850#
MSGB05  036354          5715    8849#
MSGB06  036310          5715    8848#
MSGB08  036246          5715    8847#
MSGB09  036207          5715    8846#
MSGB10  036164          5715    8845#
MSGB11  036136          5715    8844#
MSGB12  036114          5715    8843#
MSGB13  036063          5715    8842#
MSGB14  036021          5715    8841#
MSGMAX  035051          7805    8814#
MSGMIN  035042          7795    8813#
MSGNON  035167          7832    8820#
MSGNUM  035152          7827    8819#
MSG.CS  034436          8216    8800#
MSG.EQ  034434          8508    8529    8550    8799#
MSG.FC  034400          5715    8790#
MSG.FS  034422          5715    8796#
MSG.FT  034411          5715    8793#
```

```
MSG.IC  034406          5715    8792#
MSG.IT  034417          5715    8795#
MSG.LC  034403          5715    8791#
MSG.LS  034425          5715    8797#
MSG.LT  034414          5715    8794#
MSG.PA  034430          8528    8798#
MSG.R   034376          5715    8789#
MSG.SP  035202          6263    7222    7248    7255    7262    7800    7810    7825    8504    8527    8821#
MSGOUS  035067          7797    7804    7807    7814    7824    8816#
MSG10A  034550          5715    8806#
MSG10B  034531          5715    8807#
MSG11A  034564          5715    8808#
MSG11B  034732          5715    8809#
MSG12A  034753          5715    8810#
MSG12B  035021          5715    8811#
MSG7    034516          5715    8805#
MXDLTA  027176          8785#
MXLACT  027174          8785#
MXSTAL  001370          5715#   6457    6477
MXWNDW  027202          8785#
NOOP  = 000101          4971#
OFFSET= 000115          4977#
OPNPAT  025702          8503    8527#   8533    8537    8540    8541
OPNPRM  025506          8490    8495#   8499    8533    8544
OPNTST  025350          8449    8458    8465    8468#
OPNWDS  026020          8546#   8561
OPN.CT  025356          8470#   8472*   8473    8477    8482
OPN.N1  026234          8524    8541    8584#
OPN.N2  026240          8518    8535    8572    8585#
OPN.X1  026252          8524    8541    8588#
OPN.X2  026256          8523    8539    8576    8589#
OPN.1   025362          8472#   8480    8587
OPN.2   025406          8471    8474    8477#
OPT     027770          8785#
PACK  = 000123          4980#
PAT     001460          5715#
PAT.PT  002400          5715#   7994    8016    8030
PAT0    002440          5715#   6167    8547    8577
PAT1    002500          5715#
PAT10   003140          5715#
PAT11   003200          5715#
PAT12   003240          5715#
PAT13   003300          5715#
PAT14   003340          5715#
PAT15   003400          5715#
PAT2    002540          5715#
PAT3    002600          5715#
PAT4    002640          5715#
PAT5    002700          5715#
PAT6    002740          5715#
PAT7    003000          5715#
PAT8    003040          5715#   6166
PAT9    003100          5715#   6169
PC    =%000007          4968#   6187*   6188*   6190*   6191*   6192*   6193*   6196*   6251*   6277*   6597*   6607*   6616*
```

DZRPKB.P11 CROSS REFERENCE TABLE

```
                        6618*  6626*  6630*  6663*  6674*  6676*  6677*  6693*  6697*  6723*  6734*  6737*  6738*
                        6747*  6750*  6751*  6759*  6763*  6788*  6799*  6802*  6803*  6812*  6815*  6816*  6824*
                        6828*  6858*  6906*  7061*  7080*  7165*  7169*  7265*  7269*  7272*  7276*  7277*  7278*
                        7280*  7300*  7329*  7335*  7348*  7358*  7366*  7388*  7407*  7429*  7458*  7491*  7509*
                        7525*  7559   7562   7565   7568   7571   7574   7587*  7603*  7638*  7694*  7725*  7764*
                        7796*  7801*  7803*  7806*  7811*  7813*  7819*  7823*  7826*  7892*  8001*  8025*  8140*
                        8157*  8223*  8279*  8358*  8469*  8509*  8549*  8570*  8574*  8583*  8586*  8590*  8618*
                        8785*
PIRQ  = 177772          4968*
PIRQVE= 000240          4968*
PKB     001342          5715*  6600*  6609*  6666*  6726*  6739*  6791*  6804*  7355*  7632*
PKC     001344          5715*  7743   7745   7746   7750   7752   7753   7757   7762
PKCS    001340          5715*  6601*  6604*  6611*  6613*  6622*  6669*  6671*  6689*  6729*  6731*  6742*  6744*
                        6755*  6794*  6796*  6807*  6809*  6820*  7327   7356*  7633*
PKV     001334          5715*  6598*  6664*  6724*  6789*  7353*  7354*  7630   7631*  7636*
POPQUE  034272          8785*
PRM     001434          5715*  8486   8495   9600
PRMLMT  001520          5715*  8502   8524
PRMMSG  001544          5715*  8505
PRMPT   001462          5715*  8484   8601
PRM0    002062          5715*  6161
PRM1    002074          5715*
PRM10   002236          5715*
PRM11   002242          5715*
PRM12   002246          5715*
PRM13   002252          5715*
PRM14   002262          5715*
PRM15   002272          5715*
PRM16   002320          5715*
PRM2    002116          5715*
PRM3    002134          5715*
PRM4    002152          5715*
PRM5    002170          5715*
PRM6    002206          5715*
PRM7    002224          5715*
PR0   = 000000          4968*
PR1   = 000040          4968*
PR2   = 000100          4968*
PR3   = 000140          4968*
PR4   = 000200          4968*
PR5   = 000240          4968*
PR7   = 000340          4968*
PS    = 177776          4968*  6177*  6189*  6623*  6690*  6756*  6821*  7272*  7279   7628   7629*  7637*  7679*
                        8785*
PSW   = 177776          4968*
PTR%15  002316          5715*  7046
PWRVEC= 000024          4968*
QCNT    033600          8785*
QDRV0   033672          8785*
QDRV1   033712          8785*
QDRV2   033732          8785*
QDRV3   033752          8785*
QDRV4   033772          8785*
QDRV5   034012          8785*
```

DZRPKB.P11 CROSS REFERENCE TABLE

```
QDRV6    034032          8785#
QDRV7    034052          8785#
QINPT    033610          8785#*
QOUTPT   033630          8785#*
QSTART   033650          8785#
QSTOP    033652          8785#
QTERM =  034072          8785#
RANADR   023730          7135    7143    7153    8157#
RANCK    023476          7147    7157    8087#
RANPAT   023674          7141    8072    8094    8137#
RDCHR =  104412          7272    7275#
RDLIN =  104414          7275#   8310    8346    8361    8510    8531    8552
RD.ADR   033160          8785#*  8881
RD.RP    033134          8785#
RD.RP1   033156          8785#
RD.RP2   033260          8785#*
RD.RP3   033264          8785#
RD.WRD   033162          8785#   8881    8882
READ  =  000171          4986#   6869    7076    7144    7154
READHD=  000173          4987#   7383    7384    7432    7461
READIN=  000121          4979#
RECAL =  000107          4974#   6255    6336    7685
RELEAS=  000113          4976#
RESREG=  104420          6607    6616    6626    6674    6693    6734    6747    6759    6799    6812    6824    7275#   7277
                         7724
RESTAR   005114          6237#   6246    7165
RESVEC=  000010          4968#
RHAS  =  000016          8785#*
RHBA  =  000004          8785#
RHCA  =  000034          6585*   6667*   6727*   6740*   6792*   6805*   7698*   8785#
RHCC  =  000036          8785#
RHCS1 =  000000          6728*   6741*   6793*   6806*   8785#
RHCS2 =  000010          6607*   6616*   6626*   6629*   6674*   6693*   6696*   6734*   6747*   6759*   6762*   6799*   6812*
                         6824*   6827*   7682*   8785#*
RHDA  =  000006          6588*   7697*   8785#
RHDB  =  000022          8785#
RHDS1 =  000012          6605    6614    6672    6732    6745    6797    6810    8785#
RHDT  =  000026          8785#
RHEC1 =  000044          8785#
RHEC2 =  000046          8785#
RHER1 =  000014          8785#
RHER2 =  000040          8785#
RHER3 =  000042          8785#
RHLA  =  000020          8785#
RHMR  =  000024          8785#
RHOF  =  000032          8785#
RHSN  =  000030          6257    6258    8785#
RHVEC    001330          5715#
RHWC  =  000002          8785#
RH.ADR   001326          5715#   8240    8272    8274    8884
RPADR    027166          7681    8275    8785#*
RPERRS   027030          8785#*  8880    8883
RPINIT   027204          6192    8785#
RPT      001436          5715#   8509    8525*   8530    8542*   8584*   8588*
```

DZRPKB.P11 CROSS REFERENCE TABLE

```
RPTMR   032534          7369    8785#
RPTO    002064          5715#   6335
RPT1    002076          5715#   6357
RPT10   002240          5715#   6651
RPT11   002244          5715#   6712
RPT12   002250          5715#   6777
RPT13   002254          5715#   6855
RPT14   002264          5715#   6903
RPT15   002274          5715#   7012
RPT16   002322          5715#   7112
RPT2    002120          5715#   6382
RPT3    002136          5715#   6409
RPT4    002154          5715#   6433
RPT5    002172          5715#   6497
RPT6    002210          5715#   6526
RPT7    002226          5715#   6578
RPVEC   027170          6599*   6631*   6665*   6698*   6725*   6764*   6790*   6829*   7680*   8785#
RP.REG  003540          5715#   7496    8890    8894    8895    8896    8898    8899    8901    8902
RPO4    027574          7400    7422    7451    7484    7502    7511    7517    7687    8785#
RP6   = 000300          4968#
RSTRT1  005072          6229    6231    6233#   6239
RSTRT2  005104          6235#   6240
RSTRT3  005112          6236#
RTC   = 000117          4978#
R0    =%000000          4968#   6160*   6163    6164    6166*   6168    6169    6178*   6179*   6180    6199*   6201    6203
                        6204*   6213*   6221    6225*   6254*   6256*   6260*   6265*   6280*   6342*   6343*   6366*   6367*
                        6389*   6395*   6415*   6418*   6445*   6448*   6451*   6454*   6465*   6468*   6471*   6474*   6505*
                        6507*   6537*   6539*   6541*   6543*   6545*   6547*   6582*   6624*   6634*   6636*   6655*   6691*
                        6699*   6716*   6757*   6765*   6781*   6822*   6830*   6865*   6867*   6868*   6870*   6871*   6872*
                        6876    6878*   6883*   6884*   6912*   6913    6914*   6916*   6917    6920*   6922    6923*   6925*
                        6926    6928*   6929    6930    6937*   6941*   6944*   6945    7013*   7019*   7025    7040    7041
                        7043*   7045*   7046    7065*   7069*   7077*   7081*   7122*   7124*   7135*   7141*   7142*   7143*
                        7146*   7147*   7152*   7153*   7156*   7157*   7162*   7165*   7182*   7183*   7184    7190*   7191
                        7192*   7193*   7194*   7195*   7197*   7198*   7199    7226    7231    7234*   7235    7239    7240
                        7256    7264*   7269*   7271*   7274*   7275*   7277*   7278*   7279*   7280*   7369*   7400*   7408*
                        7409*   7422*   7430*   7434*   7438*   7440*   7451*   7459*   7463*   7467*   7469*   7484*   7502*
                        7511*   7517*   7530*   7533*   7535*   7554    7556*   7557    7559    7562    7565    7568    7571
                        7574    7577    7586*   7600    7610    7615*   7649    7662    7664*   7666*   7687*   7695    7699*
                        7716*   7718*   7719    7721*   7783    7833*   7850    7851*   7869    7870*   7881*   7882*   7885
                        7888*   7889    7891*   7902*   7916*   7917*   7928*   7949*   7951*   7952*   7991*   7994    8000*
                        8012*   8016    8024*   8030    8043*   8046*   8072*   8075*   8094*   8124*   8125*   8146*   8201*
                        8218*   8240*   8242    8243*   8247*   8249    8250*   8254*   8256    8261*   8265*   8267*   8269*
                        8274*   8276    8277    8278    8312*   8313*   8314    8315    8327    8328    8334*   8343*   8348*
                        8350*   8357*   8392*   8396*   8410*   8414*   8468*   8475*   8484*   8485    8493    8512*   8524*
                        8533*   8541*   8545*   8546*   8549    8554*   8555*   8558*   8559    8571*   8575*   8578*   8579
                        8594*   8599*   8600*   8602    8609    8613    8617*   8636    8637*   8656    8657*   8682*   8684
                        8685    8686    8687    8688    8690*   8691*   8713*   8720*   8723*   8732*   8733    8738*   8739*
                        8755*   8765*   8770*   8779*   8781*   8782*   8785#
R1    =%000001          4968#   6161*   6162    6163*   6167*   6168*   6200*   6205*   6214*   6218*   6222    6266*   6268*
                        6270*   6272*   6274*   6416*   6417    6419*   6420    6442*   6444    6460*   6461    6463*   6464
                        6480*   6481    6502*   6504    6508*   6510    6531*   6534    6536    6548*   6596*   6619*   6722*
                        6752*   6787*   6817*   6873*   6877    6885*   6886    6930*   6932*   7015*   7016*   7018*   7021*
                        7028*   7029*   7031*   7038*   7049    7052    7139*   7140    7185    7191*   7204*   7206*   7207*
                        7208*   7213*   7220    7231*   7243    7245    7271*   7274*   7277*   7279*   7280*   7320    7321*
```

DZRPKB.P11 CROSS REFERENCE TABLE

```
                7322    7323*   7324    7325*   7332*   7338*   7339*   7340*   7555    7558*   7579*   7580*   7581*
                7582*   7583*   7584    7585*   7648    7649*   7651    7653    7658    7659    7660    7665*   7678*
                7717*   7719    7863    7865*   7866    7868*   7883*   7885*   7903*   7906    7907    7908    7909
                7910    7911    7912    7913    7930*   7938    7942*   7943    7992*   7997*   8013*   8020*   8026
                8033*   8034    8036    8070*   8090*   8091    8120    8141*   8142*   8158*   8161    8163    8166*
                8181    8198    8270*   8273    8275*   8276*   8277*   8278*   8280    8311*   8318    8321    8323
                8325    8347*   8349*   8352    8362*   8363    8365    8367    8372    8376    8380    8384    8388
                8395*   8398*   8407    8409*   8413*   8416*   8432*   8435    8439    8443    8445*   8446    8451*
                8452    8454    8460*   8461    8511*   8513    8516    8519    8521    8532*   8534    8536    8538
                8553*   8562    8564    8566    8577*   8581    8601*   8603    8609*   8612    8630    8632    8634
                8650    8652    8654    8674    8676    8678    8680    8689*   8725    8728    8757*   8772    8774*
                8777    8785*
R2      =%000002  4968#  6162*  6164    6242*   6243    6247    6248    6249    6250    6262    6267*   6269*   6271*
                6273*   6275*   6276*   6277    6437*   6440*   6446    6452    6466    6472    6503*   6506    6509*
                6512    6532*   6533*   6534    6607*   6616*   6626*   6674*   6693*   6734*   6747*   6759*   6799*
                6812*   6824*   7048*   7050    7054    7186    7209*   7212    7235*   7250*   7271*   7274*   7277*
                7279*   7280*   7739*   7742*   7743    7745*   7748*   7749    7750    7752*   7755*   7756    7757*
                7758*   7759*   7783*   7786    7787    7788    7789*   7813    7844    7846*   7847    7849*   7884*
                7886*   7904*   7906*   7907*   7908*   7909*   7910*   7911*   7912*   7913*   7914    7933*   7939*
                7993*   7998*   8014*   8021*   8071*   8073*   8089*   8092    8093    8098*   8101*   8102    8104
                8106    8137    8138*   8143*   8145*   8159*   8168    8170    8173*   8183    8199    8265    8269
                8316    8329*   8330*   8331*   8332*   8333*   8350*   8351    8356    8394    8402*   8403    8405*
                8406    8412    8420*   8421    8423*   8424    8427    8428    8429    8486*   8491    8493*   8524*
                8525    8542    8543*   8556    8569    8573    8584    8588    8602*   8605    8634*   8635*   8654*
                8655*   8709    8710*   8719    8734*   8764    8768*   8785*
R3      =%000003  4968#  6591*  6594*   6658*   6719*   6784*   7057*   7058*   7059    7187    7210*   7215*   7239*
                7246*   7270*   7271*   7272*   7274*   7277*   7279*   7280*   7746    7753    7788*   7803    7905*
                7914    7934*   7938    7953    7962    7994*   7997    8016*   8020    8029    8033    8035    8037
                8091*   8098    8101    8103    8105    8120    8160*   8175    8178*   8179*   8181*   8182*   8183*
                8184*   8186*   8188    8191*   8193    8195*   8196*   8200    8216*   8241*   8248*   8255*   8306
                8370*   8441*   8448    8457    8464    8479    8596    8603*   8607    8610    8708    8711*   8714*
                8715    8716*   8717*   8718*   8719*   8730    8734    8736*   8753    8754*   8758*   8760*   8762*
                8764*   8768    8769*   8771    8775    8776    8779    8780*   8785*
R4      =%000004  4968#  6585*  6588*   6602*   6605    6607*   6610*   6614    6616*   6626*   6629*   6667*   6668*
                6672    6674*   6693*   6696*   6727*   6728*   6732    6734*   6740*   6741*   6745    6747*   6759*
                6762*   6792*   6793*   6797    6799*   6805*   6806*   6810    6812*   6824*   6827*   7014*   7020*
                7023*   7024    7051    7188    7240*   7241*   7270*   7274*   7277*   7681*   7682*   7697*   7698*
                7790*   7793    7796    7798    7801    7806    7808    7811    7816    7817    7818    7826    7953*
                7954*   7955    7956*   8026*   8027*   8028*   8106*   8107*   8108*   8217*   8221    8242*   8247
                8249*   8254    8256*   8257*   8258*   8259*   8260*   8267    8309    8316*   8329    8371*   8374*
                8378*   8382*   8386*   8390*   8406*   8424    8427*   8429    8433    8437    8441    8496*   8500
                8502    8505    8509    8524    8525*   8530    8542*   8547*   8551    8556*   8557    8569*   8573*
                8581*   8584*   8588*   8604*   8605    8607    8610    8614*   8615    8707    8712*   8721    8722
                8724    8732    8737*   8785*
R5      =%000005  4968#  6590*  6662*   6678    6683*   6736*   6749*   6801*   6814*   7024*   7027*   7030*   7033*
                7056    7189    7233*   7237*   7253*   7260    7270*   7271*   7274*   7277*   7740    7787*   7828
                7830*   7955*   7957*   7959*   7960*   7961*   7981    8029*   8030*   8031*   8032*   8057    8394*
                8399*   8400*   8401*   8402    8412*   8417*   8418*   8419*   8420    8477*   8478*   8479    8484
                8591    8592    8596    8601    8785*
R6      =%000006  4968#  6177*
R7      =%000007  4968#
SAVEFG  027126  8785#*
SAVREG= 104416  6607    6616    6626    6674    6693    6734    6747    6759    6799    6812    6924    7275#   7277
                7715
```

```
SAVR05  034334          7182    7881    8012    8343    8468    8545    8599    8785#
SAVR15  034334          7902    7928    7991    8785#
SC      031554          8785#
SCTRWC= 177400          4995#   6875    6910    7131
SC1     031636          8785#
SC10    032366          8785#
SC11    032402          8785#
SC12    032464          8785#
SC2     031664          8785#
SC3     031736          8785#
SC4     031732          8785#
SC5     031750          8785#
SC6     032050          8785#
SC6A    032154          8785#
SC7     032276          8785#
SC8     032322          8785#
SC9     032346          8785#
SEARCH= 000131          4981#   6602    6610
SEC.DS  001254          5715#   7407*   7429*   7458*   7491*   7509*   7525*   7658*   7694*   8042*   8886    8887    8889
                        8897    8900
SEC.RD  001250          5715#   7657*   8887
SEEK  = 000105          4973#   6668    6728    6741    6793    6806    7386    7387
SEEKFG  027130          6207*   6210*   8785#*
SELDRV= 000145          4990#
SETBUF  023036          7061    7991#
SETFOR= 000143          4989#   6253    6255    6259    6336
SET.IE  033526          8785#
SP    =%000006          4968#   6177*   6221*   6262*   6341*   6365*   6386*   6413*   6441*   6501*   6530*   6586*   6587*
                        6588    6595*   6624    6625    6660*   6691    6692    6721*   6757    6758    6786*   6822    6823
                        6857*   6905*   7134*   7165*   7169*   7243*   7245*   7269*   7270*   7271*   7272*   7273*   7274*
                        7275*   7276*   7277*   7278*   7279*   7280*   7298    7320*   7322*   7324*   7331    7337    7339
                        7339    7340    7368*   7407*   7429*   7458*   7491*   7509*   7525*   7554*   7555*   7556    7584*
                        7585    7586    7600*   7604*   7605*   7606*   7607*   7609*   7611*   7614    7628*   7630*   7635*
                        7636    7637    7648*   7665    7694*   7796*   7801*   7803*   7806*   7811*   7813*   7816*   7817*
                        7818*   7820*   7822*   7826*   7929*   7931*   7932*   7934    7941*   7945    7950*   7954    7963
                        7969*   8015*   8023    8038    8043    8046    8047*   8055*   8087*   8088*   8095    8096    8097*
                        8109    8111*   8122    8137*   8145    8281    8309*   8311    8347    8362    8482*   8485*   8488*
                        8495*   8497*   8509*   8511    8530*   8532    8549*   8551*   8553    8585    8589    8707*   8708*
                        8709*   8715*   8718    8730    8735    8736    8737    8753*   8780    8785*
SRCHWT  027102          8785#*
SRCHXX  021356          6582    6655    6716    6781    7678#
SRTINT  004476          6185    6187#
SRVCLK  017624          7353    7363    7368#
STACK = 001100          4968#   6177    6341    6365    6386    6413    6441    6501    6530    6595    6660    6721    6786
                        6857    6905    7134
STALL   021104          6448    6454    6468    6474    7438    7467    7533    7600#
STALL0  001320          5715#   6443*   6449    6455    6456*   6457    6459*   6469    6475    6476*   6477    6479*
STALL1  001364          5715#   7439    7468
STALL2  001366          5715#   7534
START   004222          6171    6176#
START1  004120          4960    6157#
START2  004210          4962    6174#   7272
START3  004110          4964    6155#
START4  004200          4966    6172#
```

DZRPKB.P11 CROSS REFERENCE TABLE

```
STATBL  001566          5715#   7209
STKLMT= 177774          4968#
STO     032632          8785#
STO1    032674          8785#
STO2    033056          8785#
STO3    033112          8785#
STO4    033114          8785#
STO5    033126          8785#
STRTMR  021542          6597    6663    6723    6788    7715#
STRT1A  004124          6156    6158#
STRT2A  004214          6173    6175#
ST.CLK  017360          6193    6630    6697    6763    6828    7320#
ST.LCL  017570          7335    7360#
ST.PCL  017526          7329    7350#
SVADR   001322          5715#   7136*   7137*   7148    7149    7158    7159
SVRH11  033434          6607    6616    6626    6674    6693    6734    6747    6759    6799    6812    6824    8785#
SVSTAT  001242          5715#   7206    7557*
SWR     = 177570        4968#   7057    7062    7066    7072    7074    7078    7169    7273*   7498    7526    7784    7973
                        7975    7979    8051    8114    8118    8213
SW0     = 000001        4968#
SW00    = 000001        4968#   7072
SW01    = 000002        4968#   7078    7975    8051    8118
SW02    = 000004        4968#   7074
SW03    = 000010        4968#   7059    7066
SW04    = 000020        4968#   7059    7062    7498
SW05    = 000040        4968#   7351    7361    7979
SW06    = 000100        4968#   6592    6632    7341    7784
SW07    = 000200        4968#   7381    8213
SW08    = 000400        4968#   6208    7169
SW09    = 001000        4968#   7526    7973    8114
SW1     = 000002        4968#
SW10    = 002000        4968#
SW11    = 004000        4968#
SW12    = 010000        4968#   6438
SW13    = 020000        4968#   7601
SW14    = 040000        4968#   7436    7465    7531
SW15    = 100000        4968#   7120
SW2     = 000004        4968#
SW3     = 000010        4968#
SW4     = 000020        4968#
SW5     = 000040        4968#
SW6     = 000100        4968#
SW7     = 000200        4968#
SW8     = 000400        4968#
SW9     = 001000        4968#
TBITVE= 000014          4968#
TD      031426          8785#
TEST0   005536          6335    6341#
TEST1   005700          6357    6365#
TEST10  010020          6660#
TEST11  010442          6721#
TEST12  011142          6786#
TEST13  011614          6855    6857#
TEST14  012114          6903    6905#
```

```
TEST15  012554          7012    7037*
TEST16  013252          7132    7133    7134*
TEST2   006012          6382    6386*
TEST3   006174          6409    6413*
TEST4   006350          6433    6439    6441*
TEST5   006666          6497    6501*
TEST6   007044          6526    6530*
TEST7   007342          6593    6595*
TICKMS  001230          5715*   7343*   7346*   7368
TICKUS  001232          5715*   7344*   7347*
TIMER   027132          8785**
TIM.DN  001276          5715*   7723*   7742
TIM.PT  001314          5715*   7717    7760    7762*   7763*
TIM.UP  001260          5715*   7716    7722*   7739    7790
TKVEC = 000060          4968*   7272*
TPB     001356          5715*   6183    7169
TPS     001354          5715*   6182    7169
TPVEC = 000064          4968*
TRAPVE= 000034          4968*   6177*
TRCKWC= 165000          4994*   6863    6880    7117
TRK.DS  001256          5715*   7407*   7429*   7458*   7491*   7509*   7525*   7659*   7694*   7966*   8041*   8886    8887
                        8888    8891    8897    8900
TRK.RD  001246          5715*   7656*   8887
TRNSWT  027100          8785**
TRTVEC= 000014          4968*
TSTNMS  001224          5715*   6158*   6230    6335    6357    6382    6409    6433    6497    6526    6578    6651    6712
                        6777    6855    6903    7012    7112    8354*   8369*   8433*
TST0    005426          6335*
TST1    005554          6335    6357*
TST10   007712          6578    6581    6635    6651*
TST11   010334          6651    6654    6712*
TST12   011034          6712    6715    6777*
TST13   011534          6777    6780    6855*
TST14   012034          6855    6903*
TST15   012370          6903    7012*
TST16   013030          7012    7112*
TST2    005716          6357    6382*
TST3    006100          6382    6409*
TST4    006240          6409    6433*
TST5    006572          6433    6497*
TST6    006750          6497    6526*
TST7    007172          6526    6578*
TWOMS   021166          6677    6738    6751    6803    6816    7628*
TYPDS = 104410          7245    7275*
TYPE  = 104400          6186    6211    6216    6224    6232    6261    6263    6264    6279    6282    7165    7169    7201
                        7202    7211    7217    7218    7222    7225    7228    7229    7238    7248    7252    7255    7257
                        7259    7262    7270    7271    7272    7275*   7278    7791    7795    7797    7800    7802    7804
                        7805    7807    7810    7812    7814    7815    7824    7825    7827    7832    8215    8307    8345
                        8360    8481    8483    8504    8506    8508    8527    8528    8529    8548    8550
TYPERR  014164          7169    7181*
TYPOC = 104402          7165    7243    7275*   8309    8530    8551
TYPON = 104406          7275*
TYPOS = 104404          6221    6262    7165    7275*   8482
TYPTIM  021740          6634    6636    6699    6765    6830    7783*
```

```
T10     002350          5715#   6658    6699
T11     002360          5715#   6719    6765
T12     002370          5715#   6784    6830
T7A     002330          5715#   6591    6634
T7B     002340          5715#   6594    6636
ULDFLG  027106          8785#*
UNLOAD= 000103          4972#
VERIFY  021246          7434    7463    7648#
WCEFLG  001316          5715#   7037*   7070    7483*   7523*
WRCKD = 000151          4982#   6866    6879    6921    6940    7068    7150    7160    7494    7510
WRCKHD= 000153          4983#
WRITE = 000161          4984#   6859    6874    6907    6936    7064    7119    7138    7500
WRTHD = 000163          4985#
WRT.AD  033362          8785#*  8882
WRT.RP  033272          8785#
WRT.WD  033360          8785#*  8882
$BDADR  001122          5715#   8034*   8102*   8889
$BDDAT  001126          5715#   7962*   8036*   8104*   8889    8892
$BELL   001202          5715#   6180    7169    7272
$CMTAG  001100          5715#   6177
$CM1  = 000006          5715#
$CM2  = 000014          5715#
$CM3  = 000006          5715#
$CM4  = 000003          5715#
$CNTLC  016075          7272#
$CNTLU  016044          7272#
$CRLF   001207          5715#   6282    7169    7201    7217    7228    7238    7252    7259    7272    8483
$DBLK   015326          7271#
$DB2D   016510          7276    7277#
$DECVL  016670          7277#
$DIV    016744          7279#   7819
$DOAGN  013752          7165#
$DTBL   015316          7271#
$ENDAD  013742          5715    7165#
$ENDCT  013660          6212*   6220*   6227    7165#
$ENULL  013756          7165#
$EOP    013476          6217    6252    7112    7165#   8287
$EOPCT  013652          6227*   7165#   8286*
$ERFLG  001103          5715#   7169*   7273*   7528    7947    7977    8053    8116
$ERMAX  001115          5715#   6177*   7273*
$ERROR  013762          6177    7169#
$ERRPC  001116          5715#   7169*   8879    8880    8881    8882    8883    8884    8885    8896    8888    8890    8891
                        8893    8894    8895    8897    8900
$ERRTB  003610          5715#   7197    7198
$ERTTL  001112          5715#   7165*   7169*
$ESCAP  001200          5715#   6177*   6589*   6659*   6720*   6785*   7169    7273*   7399*   7406*   7421*   7428*   7450*
                        7457*   7482*   7490*   7501*   7508*   7524*   7661*   7686*   7693*   7967*   7971*   8044*   8049*
                        8100#
$FILLC  001150          5715#   7269
$FILLS  001147          5715#   7269
$GDADR  001120          5715#   8035*   8103*   8889
$GDDAT  001124          5715#   8037*   8105*   8889
$GET42  013714          7165#
$HD   = 000000          4921
```

DZRPKB.P11 CROSS REFERENCE TABLE

```
$HINUM  017312          7114*   7280#*  8087    8093*   8096*   8142    8160
$ICNT   001104          5715#   7273*
$ITEMB  001114          5715#   7169*   7190
$LF     001210          5715#   7169    7272
$LONUM  017314          7115*   7280#*  7604    8088    8092*   8095*   8141    8158    8159
$LPADR  001106          5715#   6177*   6335*   6357*   6382*   6409*   6433*   6497*   6526*   6578*   6651*   6712*   6777*
                        6855*   6903*   7012*   7112*   7132*   7273*
$LPERR  001110          5715#   6177*   6335*   6357*   6382*   6409*   6433*   6497*   6526*   6578*   6651*   6712*   6777*
                        6855*   6903*   7012*   7112*   7133*   7169    7273*
$MXCNT  016320          7273#
$NULL   001146          5715#   7269
$NWTST= 000001          6335#   6357#   6382#   6409#   6433#   6497#   6526#   6578#   6651#   6712#   6777#   6855#   6903#
                        7012#   7112#
$OCNT   015106          7270#*
$OMODE  015110          7270#*
$OVER   016304          7273#
$PASS   001100          5715#   7165*
$QUES   001206          5715#   7169    7272
$RAND   017166          7280#   7603    8140    8157
$RDCHR  015524          7272#   7275
$RDDEC= ******  U       7275
$RDLIN  015610          7272#   7275
$RDOCT= ******  U       7275
$RDSZ = 000024          7272#
$REGAD  001152          5715#   6178
$REG0   001154          5715#   7184*   8886    8888    8891    8893    8894    8895    8897    8900
$REG1   001156          5715#   7185*   8880    8883    8892
$REG2   001160          5715#   7186*   8885
$REG3   001162          5715#   7187*
$REG4   001164          5715#   7188*   8889    8892
$REG5   001166          5715#   7189*   8880    8883
$RESRE  016360          7274#   7275
$SAVRE  016322          7274#   7275
$SB2D   016460          7276#   7796    7801    7803    7806    7811    7813    7823    7826    8509    8549
$SCOPE  016102          6177    7273#
$SETUP= 000007          6154#   6177    7165
$STUP = 177777          6154#
$SUPRS  016704          7278#   7796    7801    7803    7806    7811    7813    7823    7826    8509    8549
$SVLAD  016256          7273#
$SWR  = 167000          4914#   4921    4944    5715    6177    6335    6357    6382    6409    6433    6497    6526    6578
                        6651    6712    6777    6855    6903    7012    7112    7165    7169    7273
$SWRMK= 000000          7273
$TIMES  001176          5715#   6177*   6335*   6357*   6382*   6409*   6433*   6497*   6526*   6578*   6651*   6712*   6777*
                        6855*   6903*   7012*   7112*   7165*   7273*
$TKB    001140          5715#   7272
$TKCNT  015336          7272#*
$TKINT  015346          6188    7272#
$TKQEN= 015346          7272#
$TKQIN  015340          7272#*
$TKQOU  015342          7272#*
$TKQSR  015344          7272#
$TKS    001136          5715#   6177    7272*
$TKSRV  015416          7272#
$TMPO   001170          5715#   7181*   8879    8880    8881    8882    8883    8886    8889    8890    8891    8893    8894
```

DZRPKB.P11 CROSS REFERENCE TABLE

DZRPKB.P11 CROSS REFERENCE TABLE

```
CKCHR   6121#  8312   8348   8512   8533   8554   8713   8720   8723   8770
CKDIG   6130#  8350   8524
CKNUM   6140#  8313   8541   8555
COMMEN   904#  4968#  6287   6316   6555   6835
COMND   6051#  6253   6255   6259   6336
DO      6060#  6254   6256   6260   6342   6343   6366   6367   6389   6395   6415   6418   6445   6451   6465
        6471   6505   6507   6537   6539   6541   6543   6545   6547
DODTA   6067#  6865   6867   6870   6878   6883   6914   6923   6937   6941   7065   7069   7077   7124   7142
        7146   7152   7156   7162
DRV.IN  6010#  6629   6696   6762   6827
ENDCOM   916#  4968#  6312   6324   6565   6841
ENDPAS  6071#  7165
ERRCAL  8785#
ERREND  6114#  7169
ERROR   4968#  6245   6608   6617   6627   6675   6694   6735   6748   6760   6800   6813   6825   7407   7429
        7458   7491   7509   7525   7663   7694   7968   7972   8045   8050   8112   8113   8282   8785
ERRTYP  6105#  7169
ER.NDX  6087#  7407   7429   7458   7491   7509   7525   7694
ESCAPE  1018#  4968#  6589   6659   6720   6785   7406   7428   7457   7490   7508   7524   7661   7693   7967
        7971   8044   8049   8100
MORETA  4998#  5715
MORE.S  6035#  6335   6357   6382   6409   6433   6497   6526   6578   6651   6712   6777   6855   6903   7012
        7112
MSG     6326#  6335   6346#  6357   6370#  6382   6401#  6409   6424#  6433   6485#  6497   6516#  6526   6566#
        6578   6639#  6651   6702#  6712   6767#  6777   6843#  6855   6890#  6903   6951#  7012   7088#  7112
MULT    3201#  4968#
NEWTST   956#  4968#  6335   6357   6382   6409   6433   6497   6526   6578   6651   6712   6777   6855   6903
        7012   7112
POP     1351#  4968#  7271   7274   7279   7280
PUSH    1343#  4968#  7271   7274   7279   7280
REPORT  6020#  6634   6636   6699   6765   6830
RPOY.D  3665#  8785
SAV.RH  6002#  6607   6616   6626   6674   6693   6734   6747   6759   6799   6812   6824
SCOPE   4968#  6344   6368   6399   6422   6483   6514   6550   6628   6695   6761   6826   6888   6947   7084
        7163
SETTRA  7275#
SETUP    738#  4968#  6177
SET.TN  6025#  6335   6357   6382   6409   6433   6497   6526   6578   6651   6712   6777   6855   6903   7012
        7112
SKIP    1052#  4968#
SLASH    856#  4968#
SPACE   4968#
STARS    825#  4968#  5715   5904   6335   6357   6382   6409   6433   6497   6526   6578   6651   6712   6777
        6855   6903   7012   7112   7165   7169   7171   7269   7270   7271   7272   7273   7274   7275   7276
        7277   7278   7279   7280   7282   7302   7372   7390   7411   7442   7472   7537   7589   7618   7640
        7668   7701   7707   7727   7766   7835   7854   7872   7894   7919   7983   8003   8059   8077   8127
        8148   8203   8225   8289   8336   8619   8639   8659   8693   8741   8784
TRMTRP  7275#
TYPBIN  1287#  4968#
TYPB2D  6014#  7796   7801   7803   7806   7811   7813   7823   7826   8509   8549
TYPDEC  1257#  4968#  7245
TYPEND  6076#  7165
TYPNUM  1224#  4968#
TYPOCS  1177#  4968#  6221   6262   7165   8482
```

F15

DZRPKB.P11 CROSS REFERENCE TABLE

```
TYPOCT   1140#  4968#  7165  7243  8309  8530  8551
TYPTXT   1094#  4968#  6186  6211  6216  6224  6232  6261  6264  7165  7211  7225  8215  8345  8360
         8481  8548
$$CMRE   5715#
$$CMTM   5715#
$$ESCA   1031#  4968#
$$NEWT    985#  4968#  6335  6357  6382  6409  6433  6497  6526  6578  6651  6712  6777  6855  6903
         7012  7112
$$SET    7275#
$$SKIP   1065#  4968#
.EQUAT    164#  4916#  4968
.HEADE     39#  4917#  4921
.KT11     300#
.SETUP    684#  4917#  6154
.SWRHI     81#  4916#  4944
.SWRLO   4944#  4945   4946  4947  4948  4949  4950  4951  4952
.$CATC    478#  4916#  4955
.$CMTA    535#  4916#  5715
.$DB2D   3401#  4918#  7277
.$DB2O   3525#
.$DIV    3303#  4918#  7279
.$EOP    1399#  4916#  7165
.$ERRO   1773#  4916#  7169
.$ERRT   1928#
.$MULT   3239#
.$POWE   2964#
.$RAND   3021#  4918#  7280
.$RODE   2660#  4917#
.$ROOC   2568#  4917#
.$READ   2354#  4917#  7272
.$SAVE   2736#  4918#  7274
.$SB2D   3486#  4918#  7276
.$SB2O   3588#
.$SCOP   1575#  4918#  7273
.$SIZE   3083#
.$SUPR   3626#  4918#  7278
.$TRAP   2837#  4916#  7275
.$TYPB   2269#
.$TYPD   2191#  4917#  7271
.$TYPE   2016#  4916#  7269
.$TYPO   2094#  4917#  7270
```

DZRPKB.P11 CROSS REFERENCE TABLE

```
ADC     7277  7280  7758  8179
ADD     6184  6276  6390  6460  6508  6538  6542  6546  6548  6884  6915  6924  6934  6938  7018
        7020  7030  7197  7198  7269  7270  7271  7277  7279  7280  7407  7429  7458  7491  7509
        7525  7635  7694  7757  7763  7846  7865  7941  7950  7961  7998  8021  8032  8033  8101
        8183  8402  8420  8719  8719  8732  8764  9779  8785
ASL     6419  7208  7213  7280  7583  7960  8031  8257  8258  8259  8399  8400  8401  8405  8417
        8418  8419  8423  8478  8614  8714  8716  8717  8758  8760  8762  8785
ASLB    6205  6238  7193  7194  7195  7207  7271  8785
ASR     6218  7956  8028  8108  8331  8332  8333  8488  8497  8785
BCC     6219  7196  7214  7271  7279  8544  8785
BCS     6185  6239  7242  8489  8498  8759  8761  8763
BEQ     6195  6223  6231  6439  6447  6453  6467  6473  6593  6606  6615  6673  6733  6746  6798
        6811  7026  7047  7060  7071  7073  7165  7169  7200  7221  7227  7232  7236  7247  7270
        7272  7273  7278  7279  7342  7404  7426  7437  7455  7466  7488  7497  7506  7515  7521
        7532  7576  7602  7654  7691  7741  7794  7799  7809  7829  7845  7864  7946  8099  8123
        8214  8238  8284  8322  8368  8430  8436  8438  8453  8455  8462  8480  8490  8499  8535
        8539  8563  8567  8606  8608  8611  8616  8675  8677  8679  8681  8726  8785
BGE     6392  6482  6513  7017  7022  7032  7130  7273  7279  7744  7747  8189  8426  8474  8785
BGT     6229  6244  6281  6458  6478  6511  6535  6580  6620  6653  6687  6714  6753  6779  6818
        6887  6918  6927  6943  6946  7042  7053  7055  7127  7165  7270  7271  7279  7578  7887
        7890  7944  8074  8144  8164  8171  8185  8404  8422  8560  8785
BHI     7273  7529  7948  8054  8121  8633  8653  8731
BIC     6604  6613  6622  6671  6689  6731  6744  6755  6796  6809  6820  7165  7270  7272  7279
        7605  7650  7957  8635  8655  8785
BICB    8785
BIS     7270  7271  7277  7279  9427  8433  8441  8785
BISB    6203  8351  8356  8785
BIT     6208  6230  6235  6335  6357  6382  6409  6433  6438  6497  6526  6578  6592  6605  6614
        6632  6651  6672  6712  6732  6745  6777  6797  6810  6855  6903  7012  7046  7059  7062
        7066  7072  7074  7078  7112  7120  7169  7273  7341  7351  7361  7381  7436  7465  7496
        7498  7526  7531  7559  7562  7565  7568  7571  7574  7601  7784  7973  7975  7979  8051
        8114  8118  8213  8429  8479  8596  8605  8607  8610  8615  8785
BITB    8785
BLE     6202  6398  6421  6462  7251  7751  7754  7848  7867  8176  8194  8785
BLO     6165  6170  7272  7608  7720  7915  8631  8651
BLOS    7272  7761  7978  8117  8785
BLT     6682  6933  7269  7270  7271  7277  7279  8162  8169  8580  8593  8785
BMI     7205  7271  7273  7516  8503  8785
BNE     6177  6181  6206  6209  6215  6236  6335  6357  6382  6409  6433  6497  6526  6578  6633
        6651  6679  6712  6777  6855  6903  7012  7063  7067  7075  7079  7112  7121  7165  7169
        7216  7224  7254  7261  7269  7270  7271  7272  7273  7277  7279  7280  7352  7362  7382
        7433  7462  7493  7495  7499  7527  7561  7564  7567  7570  7573  7612  7652  7785  7938
        7940  7958  7964  7974  7976  7980  7999  8020  8022  8039  8052  8110  8115  8119  8180
        8319  8324  8326  8353  8364  8366  8373  8377  8381  8385  8389  8408  8440  8444  8447
        8449  8458  8465  8514  8517  8520  8522  8537  8565  8597  8729  8773  8778  8785
BPL     7169  7269  7270  7271  7279  7405  7427  7456  7489  7507  7522  7692  7821  8785
BR      6156  6171  6173  6186  6197  6211  6216  6224  6226  6232  6240  6246  6261  6264  6549
        6583  6584  6621  6635  6656  6657  6684  6688  6717  6718  6754  6782  6783  6819  6935
        7039  7044  7082  7083  7165  7211  7223  7225  7244  7249  7263  7269  7270  7271  7272
        7273  7277  7279  7297  7330  7336  7345  7385  7402  7424  7431  7453  7460  7486  7613
        7689  7696  7831  7970  8048  8056  8139  8167  8174  8187  8192  8197  8215  8219  8220
        8222  8244  8245  8246  8251  8252  8253  8262  8263  8264  8266  8268  8317  8320  8345
        8355  8360  8375  8379  8383  8387  8391  8393  8397  8411  8415  8431  8442  8450  8471
        8481  8487  8492  8494  8501  8515  8526  8540  8548  8561  8568  8572  8576  8582  8598
```

DZRPKB.P11 CROSS REFERENCE TABLE

```
        8683  8727  8756  8766  8767  8785
CLC     7279  8165  8172  8177  8190
CLR     6157  6159  6174  6177  6179  6189  6198  6199  6207  6212  6213  6233  6267  6437  6443
        6459  6479  6590  6600  6609  6623  6662  6666  6690  6726  6736  6739  6756  6791  6801
        6804  6821  6873  6912  6920  6928  7013  7014  7027  7037  7045  7123  7165  7210  7233
        7270  7271  7272  7273  7277  7279  7280  7292  7294  7323  7326  7399  7421  7450  7482
        7483  7501  7558  7606  7678  7679  7683  7684  7686  7697  7698  7718  7830  7882  7930
        7932  8015  8097  8239  8285  8286  8344  8354  8369  8370  8371  8469  8496  8546  8710
        8711  8712  8754  8769  8785
CLRB    6862  6882  7271  7272  7273  7277  7279  8785
CMP     6164  6169  6177  6180  6391  6397  6420  6457  6461  6477  6481  6510  6512  6534  6681
        6686  6886  6917  6926  6945  7041  7052  7054  7129  7165  7271  7272  7273  7277  7298
        7331  7337  7614  7651  7653  7719  7743  7746  7750  7753  7760  7844  7847  7863  7866
        7889  7914  7938  7943  8120  8175  8188  8193  8281  8403  8421  8424  8473  8559  8579
        8592  8730  8785
CMPB    6942  7126  7269  7272  7273  7278  7432  7461  7492  7494  7528  7577  7947  7977  8053
        8116  8163  8170  8323  8363  8365  8372  8376  8380  8384  8388  8407  8435  8439  8443
        8452  8519  8536  8564  8630  8632  8650  8652  8676  8678  8680  8728  8772  8785
COM     6210  6440  7019  7058  7215  7237  7253  8785
COMB    7969  8047  8111
DEC     6280  6619  6685  6752  6817  6932  7021  7031  7165  7192  7246  7250  7272  7277  7278
        7279  7611  7886  7939  8073  8143  8349  8785
DECB    7269  7270  7279  8785
EMT     4968
HALT    4955  7169  7269
INC     6204  6220  6225  6237  6456  6476  6680  6885  6916  6925  6944  7128  7165  7169  7270
        7271  7272  7273  7277  7279  7280  7296  7579  7580  7581  7582  7748  7755  7759  7822
        7888  7942  8184  8195  8395  8398  8409  8413  8416  8432  8445  8451  8460  8472  8558
        8578  8689  8757  8774  8785
INCB    6939  7125  7169  7273  8785
IOT     4968
JMP     4960  4962  4964  4966  6217  6335  6357  6382  6409  6433  6497  6526  6578  6581  6651
        6654  6712  6715  6777  6780  6855  6903  7012  7112  7165  7272  7981  8057  8287  8456
        8459  8463  8466  8476  8518  8523  8587  8595  8785
JSR     6187  6188  6190  6191  6192  6193  6196  6251  6254  6256  6260  6342  6343  6366  6367
        6389  6395  6415  6418  6445  6448  6451  6454  6465  6468  6471  6474  6505  6507  6537
        6539  6541  6543  6545  6547  6582  6597  6607  6616  6618  6626  6630  6634  6636  6655
        6663  6674  6676  6677  6693  6697  6699  6716  6723  6734  6737  6738  6747  6750  6751
        6759  6763  6765  6781  6788  6799  6802  6803  6812  6815  6816  6824  6828  6830  6858
        6865  6867  6868  6870  6871  6878  6883  6906  6914  6923  6937  6941  7043  7061  7065
        7069  7077  7080  7081  7122  7124  7135  7141  7142  7143  7146  7147  7152  7153  7156
        7157  7162  7165  7169  7182  7264  7269  7272  7276  7329  7335  7369  7400  7407  7422
        7429  7434  7438  7451  7458  7463  7467  7484  7491  7502  7509  7511  7517  7525  7533
        7603  7687  7694  7796  7801  7803  7806  7811  7813  7819  7823  7826  7881  7891  7902
        7916  7928  7951  7991  8000  8012  8024  8072  8094  8140  8157  8218  8243  8250  8261
        8312  8313  8343  8348  8350  8357  8392  8396  8410  8414  8468  8475  8509  8512  8524
        8533  8541  8545  8549  8554  8555  8570  8571  8574  8575  8586  8590  8594  8599  8617
        8682  8713  8720  8723  8755  8765  8770  8785
MOV     6155  6158  6160  6161  6162  6163  6166  6167  6168  6172  6175  6177  6178  6182  6183
        6200  6214  6221  6227  6234  6242  6247  6248  6249  6250  6252  6253  6255  6257  6259
        6262  6265  6266  6277  6335  6336  6339  6340  6341  6357  6362  6363  6364  6365  6382
        6385  6386  6387  6393  6409  6412  6413  6414  6416  6417  6433  6436  6441  6442  6444
        6450  6463  6464  6470  6497  6500  6501  6502  6503  6504  6506  6526  6529  6530  6531
        6532  6536  6578  6585  6586  6588  6589  6591  6594  6595  6596  6598  6599  6601  6602
```

```
        6607    6610    6611    6616    6624    6626    6629    6631    6651    6658    6659    6660    6661    6664    6665
        6667    6668    6669    6674    6683    6691    6693    6696    6698    6712    6719    6720    6721    6722    6724
        6725    6727    6728    6729    6734    6740    6741    6742    6747    6749    6757    6759    6762    6764    6777
        6784    6785    6786    6787    6789    6790    6792    6793    6794    6799    6805    6806    6807    6812    6814
        6822    6824    6827    6829    6855    6856    6857    6859    6860    6863    6864    6866    6869    6872    6874
        6875    6876    6879    6880    6881    6903    6904    6905    6907    6908    6910    6911    6919    6921    6930
        6931    6936    6940    7012    7015    7024    7028    7035    7036    7038    7048    7049    7051    7056    7057
        7064    7068    7076    7112    7113    7114    7115    7116    7117    7118    7119    7131    7132    7133    7134
        7136    7137    7138    7139    7140    7144    7145    7148    7149    7150    7151    7154    7155    7158    7159
        7160    7161    7165    7169    7184    7185    7186    7187    7188    7189    7191    7199    7206    7209    7212
        7226    7231    7234    7235    7243    7245    7256    7269    7270    7271    7272    7273    7274    7275    7276
        7277    7278    7279    7280    7293    7299    7320    7321    7322    7324    7325    7328    7332    7334    7338
        7339    7340    7343    7344    7346    7347    7353    7354    7355    7356    7363    7364    7365    7368    7383
        7384    7386    7387    7406    7407    7408    7428    7429    7430    7457    7458    7459    7490    7491    7508
        7509    7523    7524    7525    7530    7554    7555    7556    7557    7584    7585    7586    7600    7604    7609
        7628    7629    7630    7631    7632    7633    7636    7637    7648    7649    7655    7660    7661    7664    7665
        7680    7681    7682    7685    7693    7694    7716    7717    7721    7722    7723    7739    7742    7745    7752
        7762    7783    7786    7787    7788    7789    7790    7796    7801    7803    7806    7811    7813    7816    7817
        7818    7826    7828    7849    7868    7883    7884    7885    7903    7904    7905    7906    7907    7908    7909
        7910    7911    7912    7913    7931    7933    7934    7949    7953    7955    7959    7962    7965    7967    7971
        7992    7993    7994    7997    8013    8014    8016    8026    8029    8034    8035    8036    8037    8040    8043
        8044    8046    8049    8055    8070    8071    8087    8088    8089    8090    8091    8092    8093    8095    8096
        8100    8102    8103    8104    8105    8106    8124    8137    8138    8141    8142    8145    8160    8181    8200
        8216    8217    8221    8240    8241    8242    8247    8248    8249    8254    8255    8256    8265    8267    8269
        8270    8271    8273    9274    8275    8276    8277    8278    8280    8306    8309    8311    8316    8329    8347
        8350    8362    8374    8378    8382    8386    8390    8394    8406    8412    8477    8482    8484    8485    8486
        8493    8495    8505    8509    8511    8524    8525    8530    8532    8542    8547    8549    8551    8553    8556
        8569    8573    8577    8581    8584    8588    8600    8601    8602    8603    8604    8609    8690    8707    8708
        8709    8715    8734    8736    8737    8738    8753    8768    8780    8781    8785
MOVB    6177    6253    6258    6337    6338    6358    6359    6360    6361    6383    6384    6410    6411    6434    6435
        6498    6499    6527    6528    6587    6861    6877    6909    6913    6922    6929    7034    7050    7169    7181
        7190    7239    7240    7269    7270    7271    7272    7273    7275    7277    7407    7429    7458    7491    7500
        7509    7510    7525    7656    7657    7658    7659    7694    7966    8041    8042    8158    8159    8198    8199
        8634    8654    8785
NEG     7023    7033    7270    7271    7279    8186
NOP     7165    7504    7513    7519
RESET   6176    7165
ROL     6268    6269    6270    6271    6272    6273    6274    6275    7270    7273    7279    7280    7820    8785
ROLB    8785
ROR     7241    8178    8191    8543
RORB    8166    8173
RTI     7169    7269    7270    7271    7272    7273    7274    7279    7370    7705    8785
RTS     7265    7269    7272    7275    7276    7277    7278    7280    7300    7348    7358    7366    7388    7409    7440
        7469    7535    7587    7615    7638    7666    7699    7725    7764    7833    7851    7870    7892    7917    7952
        8001    8025    8075    8125    8146    8201    8223    8279    8334    8358    9583    8618    8637    8657    8691
        8739    8782    8785
SBC     7277    7279
SUB     6396    6480    6509    6533    6540    6544    7016    7029    7169    7183    7204    7271    7277    7607    7929
        7954    8020    8027    8030    8098    8107    8785
SWAB    8182    8196    8260    8330    8785
TRAP    7275
TST     6194    6222    6228    6446    6452    6466    6472    6579    6625    6652    6678    6692    6713    6758    6778
        6823    7025    7040    7070    7165    7169    7220    7260    7269    7270    7271    7272    7273    7275    7279
        7295    7327    7333    7403    7425    7454    7487    7505    7514    7520    7610    7662    7690    7695    7740
```

```
          7749    7756    7793    7798    7808    7850    7869    7945    7963    8023    8038    8109    8122    8237    8272
          8283    8314    8315    8327    8328    8428    8437    8448    8457    8464    8491    8500    8502    8557    8585
          8589    8591    8612    8613    8636    8656    8684    8685    8686    8687    8688    8721    8722    8724    8733
          8735    8771    8775    8776    8785
TSTB      6201    6243    7269    7271    7273    7278    8161    8168    8318    8321    8325    8352    8367    8446    8454
          8461    8513    8516    8521    8534    8538    8562    8566    8674    8725    8777    8785
WAIT      6603    6612    6670    6730    6743    6795    6808    7634
.ASCII    5715
.ASCIZ    5715    6186    6211    6216    6224    6232    6261    6264    7165    7211    7225    7272    8215    8345    8360
          8481    8548    8789    8790    8791    8792    8793    8794    8795    8796    8797    8798    8799    8800    8801
          8802    8803    8805    8806    8807    8808    8809    8810    8811    8813    8814    8815    8816    8817    8818
          8819    8820    8821    8825    8826    8827    8828    8829    8830    8831    8832    8833    8834    8835    8836
          8837    8841    8842    8843    8844    8845    8846    8847    8848    8849    8850    8851    8855    8856    8857
          8858    8859    8860    8861    8862    8863    8864    8865    8866    8867    8868    8869    8870    8871    8872
.BLKB     7272    7277
.BLKW     7271    8785
.BYTE     5715    6221    6262    7165    7270    7272    8482    8785    8907    8908    8911    8912    8915    8916    8919
          8920    8923    8924    8927    8928    8931    8932    8934    8935    8938    8939    8941    8942    8945    8946
          8949    8950    8953    8954    8956    8957    8960    8961    8964    8965    8968    8969    8972    8973    8975
          8976    8979    8980    8982    8983    8985    8986    8989    8990    8992    8993    8995    8996
.ENABL       4    4920
.END      9001
.ENDC     4921    4944    4955    4968    5715    5904    6154    6177    6186    6211    6216    6221    6224    6232    6253
          6255    6259    6261    6262    6264    6287    6312    6316    6324    6335    6336    6357    6382    6409    6433
          6497    6526    6555    6565    6578    6589    6651    6659    6712    6720    6777    6785    6835    6841    6855
          6903    7012    7112    7165    7169    7171    7211    7225    7269    7270    7271    7272    7273    7274    7275
          7276    7277    7278    7279    7280    7282    7302    7372    7390    7406    7407    7411    7428    7429    7442
          7457    7458    7472    7490    7491    7508    7509    7524    7525    7537    7589    7618    7640    7661    7668
          7693    7694    7701    7707    7727    7766    7835    7854    7872    7894    7919    7967    7971    7983    8003
          8044    8049    8059    8077    8100    8127    8148    8203    8215    8225    8289    8336    8345    8360    8481
          8482    8548    8619    8639    8659    8693    8741    8784    8785
.EQUIV    4968
.EVEN     6186    6211    6216    6224    6232    6261    6264    7165    7211    7225    8215    8345    8360    8481    8548
          8875    8998
.IF       4921    4944    4955    4968    5715    5904    6154    6177    6186    6211    6216    6221    6224    6232    6253
          6255    6259    6261    6262    6264    6287    6312    6316    6324    6335    6336    6357    6382    6409    6433
          6497    6526    6555    6565    6578    6589    6651    6659    6712    6720    6777    6785    6835    6841    6855
          6903    7012    7112    7165    7169    7171    7211    7225    7269    7270    7271    7272    7273    7274    7275
          7276    7277    7278    7279    7280    7282    7302    7372    7390    7406    7407    7411    7428    7429    7442
          7457    7458    7472    7490    7491    7508    7509    7524    7525    7537    7589    7618    7640    7661    7668
          7693    7694    7701    7707    7727    7766    7835    7854    7872    7894    7919    7967    7971    7983    8003
          8044    8049    8059    8077    8100    8127    8148    8203    8215    8225    8289    8336    8345    8360    8481
          8482    8548    8619    8639    8659    8693    8741    8784    8785
.IFF      4944    4968    5715    5904    6177    6221    6262    6335    6357    6382    6409    6433    6497    6526    6578
          6589    6651    6659    6712    6720    6777    6785    6855    6903    7012    7112    7165    7169    7171    7269
          7270    7271    7272    7273    7274    7275    7276    7277    7278    7279    7280    7282    7302    7372    7390
          7406    7407    7411    7428    7429    7442    7457    7458    7472    7490    7491    7508    7509    7524    7525
          7537    7589    7618    7640    7661    7668    7693    7694    7701    7707    7727    7766    7835    7854    7872
          7894    7919    7967    7971    7983    8003    8044    8049    8059    8077    8100    8127    8148    8203    8225
          8289    8336    8482    8619    8639    8659    8693    8741    8784
.IFT      6186    6211    6216    6224    6232    6261    6264    7165    7169    7211    7225    7272    7273    8215    8345
          8360    8481    8548
.IFTF     6186    6211    6216    6224    6232    6261    6264    7165    7169    7211    7225    7272    7273    8215    8345
          8360    8481    8548
```

DZRPKB.P11 CROSS REFERENCE TABLE

```
.IIF     4921  4944  4945  4946  4947  4948  4949  4950  4951  4952  4955  5715  6177  6221  6262
         7165  7169  7243  7245  7269  7272  7273  7275  7796  7801  7803  7806  7811  7813  7823
         7826  8309  8482  8509  8530  8549  8551
.IRP     5715  6154  6335  6357  6382  6409  6433  6497  6526  6578  6651  6712  6777  6855  6903
         7012  7112  7165  7169  7271  7274  7279  7280  7935  7995  8017
.LIST       2  3664  4912  4919  4955  4968  5715  5904  6148  6154  6186  6211  6216  6224  6232
         6261  6264  6287  6312  6316  6324  6335  6357  6382  6409  6433  6497  6526  6555  6565
         6578  6651  6712  6777  6835  6841  6855  6903  7012  7112  7165  7211  7225  7272  7273
         7275  8215  8345  8360  8481  8548
.MACRO     39    81   164   300   478   535   684   738   825   856   904   916   956   985  1018
         1031  1052  1065  1094  1140  1177  1224  1257  1287  1343  1351  1399  1575  1773  1928
         2016  2094  2191  2269  2354  2568  2660  2736  2837  2964  3021  3083  3201  3239  3303
         3401  3486  3525  3588  3626  3665  4944  4998  5715  6002  6010  6014  6020  6025  6035
         6051  6060  6067  6071  6076  6087  6105  6114  6121  6130  6140  6326  6346  6370  6401
         6424  6485  6516  6566  6639  6702  6767  6843  6890  6951  7088  7275  8785
.MCALL   4916  4917  4918  4968
.NLIST      1     3  4911  4913  4955  4968  5715  5904  6001  6154  6186  6211  6216  6224  6232
         6261  6264  6287  6312  6316  6324  6335  6357  6382  6409  6433  6497  6526  6555  6565
         6578  6651  6712  6777  6835  6841  6855  6903  7012  7112  7165  7211  7225  7272  7273
         7275  8215  8345  8360  8481  8548
.PAGE    4969  5715  6149  6283  6551  6831  6948  7085  7164  7166  8783
.REPT    4955  5715  5904  6287  6312  6316  6324  6555  6565  6835  6841
.SBTTL   4923  4944  4955  4957  4968  5715  6151  6285  6314  6335  6357  6382  6409  6433  6497
         6525  6553  6578  6651  6712  6777  6833  6855  6903  6949  7012  7086  7112  7165  7168
         7169  7172  7269  7270  7271  7272  7273  7274  7275  7276  7277  7278  7279  7280  7283
         7303  7373  7391  7412  7443  7473  7538  7590  7619  7641  7669  7702  7708  7728  7767
         7836  7855  7873  7895  7920  7984  8004  8060  8078  8128  8149  8204  8226  8290  8337
         8620  8640  8660  8694  8742  8785  8787  8823  8839  8853  8877  8904
.TITLE   4921
.WORD    4955  5715  6278  6449  6455  6469  6475  7165  7203  7219  7230  7258  7270  7272  7276
         7277  7278  7280  7439  7468  7534  7560  7563  7566  7569  7572  7575  7792  8308  8470
         8507  8785  8879  8880  8881  8882  8883  8884  8885  8886  8887  8888  8889  8890  8891
         8892  8893  8894  8895  8896  8897  8898  8899  8900  8901  8902  8906  8910  8914  8918
         8922  8926  8930  8933  8937  8940  8944  8948  8952  8955  8959  8963  8967  8971  8974
         8978  8981  8984  8988  8991  8994
```

ERRORS DETECTED: 0

```
*DZRPKB,DZRPKB/CRF=SYSMAC.SMA,RP04.009,DZRPKB
RUN-TIME:  70 83 9 SECONDS
CORE USED:  39K
```

L15

Spooler runtime 6 Seconds, 80 KCS, 764 disk reads, 0 disk writes, 191 pages printed

[redacted]