RP11E

FUNCT LOGIC&READ/WRITE

MD-11-DZRPY-B

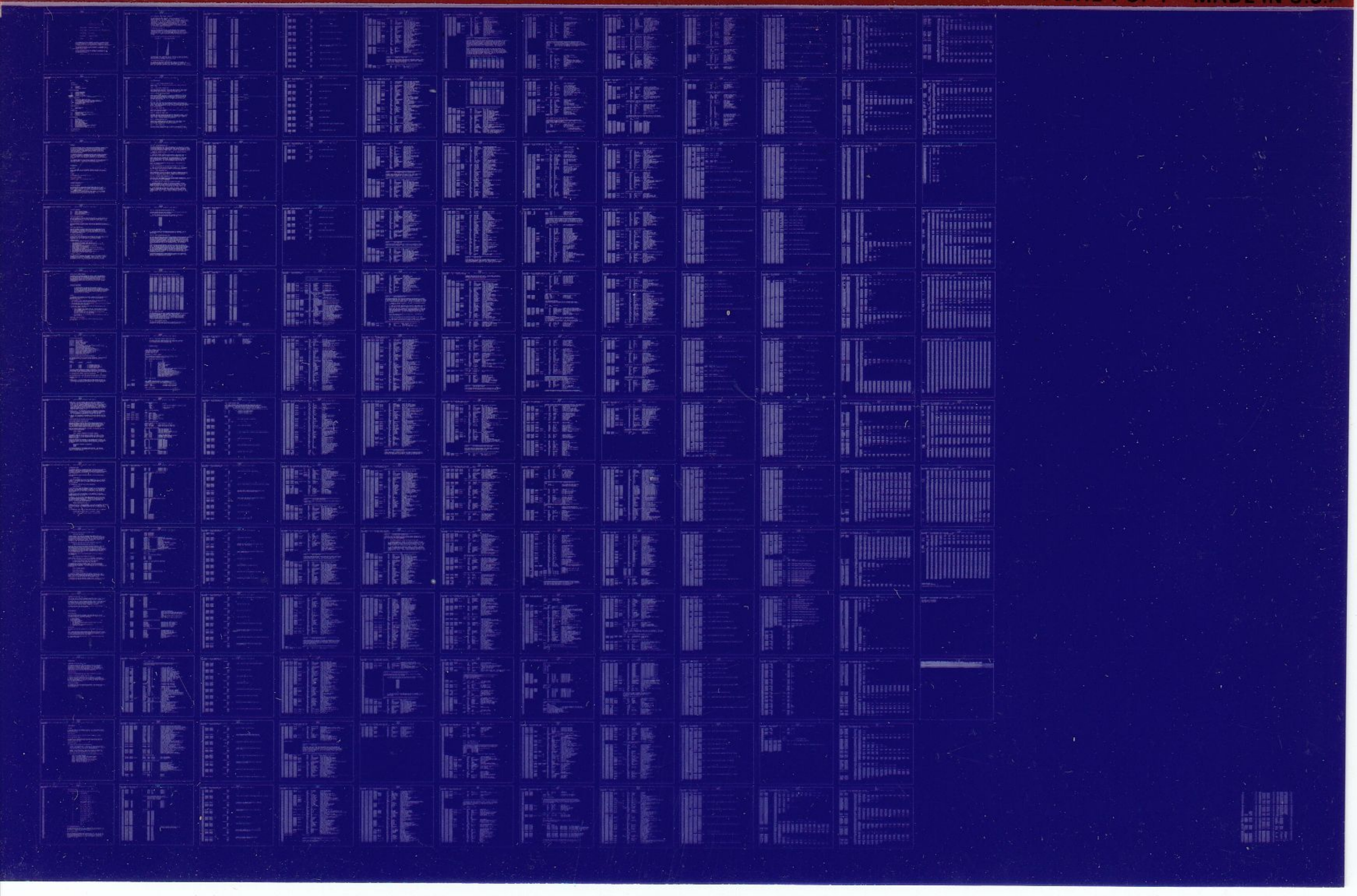
EP-DZRPY-B-DL-A

FICHE 1 OF 1

NOV 1976

digital

MADE IN U.S.A



.REM %

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

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

PRODUCT NAME: RP11E FUNCTIONAL LOGIC & READ/WRITE TEST

DATE CREATED: MAY 21,1976

MAINTAINER: DIAGNOSTIC GROUP

AUTHOR: C. HESS

CONTENTS

1. ABSTRACT

THE RP11E FUNCTIONAL LOGIC & READ/WRITE TEST PERFORMS A CURSORY CHECK OF THE RP11E CONTROLLER LOGIC AND PERFORMS DATA STORAGE/RETRIEVAL TESTING OF THE SUBSYSTEM. INCLUDED IN THE PROGRAM ARE UTILITY TESTS WHICH ASSIST IN PERFORMING HEAD ALIGNMENT, WHICH CHECK SUBSYSTEM POWER FAIL OPERATION, AND WHICH CHECK THE RP11E AND DRIVE CONTROL SWITCHES.

THE PROGRAM CONTAINS A 'CONVERSATION MODE' WHICH ALLOWS THE OPERATOR TO MODIFY THE OPERATION PARAMETERS FOR THE DATA STORAGE/RETREIVAL TEST. THE PROGRAM WILL USE MEMORY MANAGEMENT IN THE OPERATION OF THE DATA TEST IF THE PROGRAM IS BEING RUN ON A MEMORY MANAGEMENT SYSTEM.

THIS PROGRAM COMPLETES THE TESTING STARTED IN THE DISKLESS CONTROLLER TEST (MD-11-DZRPW). TESTING IN THE FUNCTIONAL LOGIC PROGRAM IS PERFORMED WITH 1 - 8 RP02, RPR02, OR RP03 DISK DRIVES.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP-11 COMPUTER WITH 12K OF MEMORY; CONSOLE TELETYPE; RP11E DISK CONTROLLER; 1 TO 8 RP02, RPR02, OR RP03 DISK DRIVES WITH FORMATTED PACKS.

2.2 STORAGE

THIS PROGRAM WILL LOAD AND RUN IN 12K.

2.3 PRELIMINARY PROGRAMS

MAINDEC-11-DZRPW - RP11E DISKLESS LOGIC TEST

3. STARTING PROCEDURES

3.1 LOADING PROCEDURES

THE PROGRAM MAY BE LOADED FROM EITHER PAPER TAPE OR AN 'XXDP' DEVICE. REFER TO EITHER THE STANDARD PROCEDURES FOR LOADING 'ABS' FORMAT PAPER TAPES OR TO THE 'XXDP' SYSTEM REFERENCE DOCUMENT.

THIS PROGRAM MAY BE INCLUDED IN AN 'XXDP' CHAIN. IF THE PROGRAM IS PART OF A CHAIN AND IS LOADED FROM AN RP02, RPR02, OR AN RP03, THE PROGRAM WILL NOT USE DRIVE 0 FOR TESTING.

3.2 STARTING ADDRESSES

200 NORMAL STARTING ADDRESS
204 SELECT OPERATING PARAMETERS
210 SELECT RP11 ADDRESS
214 COMBINATION OF 204 AND 210

3.2.1 START FROM LOCATION 200

WHEN THE PROGRAM IS STARTED FROM LOCATION 200, ALL ONLINE DRIVES WILL BE USED (SEE SECTION 3.1 FOR DRIVE 0 EXCEPTION) USING TESTS 0 - 14. ALL PARAMETERS FOR TEST 14 WILL BE RETURNED TO THEIR DEFAULT VALUES.

3.2.2 START FROM LOCATION 204

WHEN THE PROGRAM IS STARTED FROM LOCATION 204, PRAMETERS FOR TEST 14 ARE RESET TO THEIR DEFAULT VALUES, AND THE PROGRAM ENTERS CONVERSATIONAL MODE.

3.2.3 START FROM LOCATION 210

WHEN THE PROGRAM IS STARTED FROM LOCATION 210, OPERATION IS THE SAME AS THE START FROM 200, EXCEPT THAT THE PROGRAM ASKS FOR AN RP11E ADDRESS. THE OPERATOR MAY ENTER A NEW ADDRESS VALUE OR HE MAY RETAIN THE CURRENT VALUE. THE PARAMETERS FOR TEST 14 ARE RESET TO THEIR DEFAULT VALUES.

3.2.4 START FROM LOCATION 214

PROGRAM START FROM LOCATION 214 IS THE SAME AS THE START FROM LOCATION 204 EXCEPT THAT THE PROGRAM ASKS FOR A NEW RP11E ADDRESS. THE PARAMETERS FOR TEST 14 ARE RESET TO THEIR DEFAULT VALUES.

3.3 OPERATOR ACTION

BEFOR STARTING REFER TO SECTION 4.1

1. LOAD PROGRAM INTO MEMORY (SEE SECTION 3.1)
2. CHANGE TELETYPE KEYBOARD AND PRINTER ADDRESSES AS REQUIRED (SEE SECTION 4.2)
3. PLACE A FORMATTED PACK ON DRIVE(S) TO BE USED
4. BRING DRIVE(S) TO ONLINE STATE.
5. PLACE 'FORMAT' AND 'MAINTENANCE' SWITCHES ON THE RP11E CONTROL PANEL TO 'NORMAL'
6. SET CONTROLLER AND DRIVE WRITE LOCKOUT SWITCHES AS REQUIRED
7. LOAD ADDRESS 200, 204, 210, OR 214.
8. SET CPU SWITCH REGISTER SWITCHES (SEE SECTION 4.1)
9. PRESS START ON THE CPU CONTROL PANEL

3.4 'CONTROL C' OPTION

THE OPERATOR MAY TERMINATE THE CURRENT TEST AT ANY TIME BY TYPING A 'CONTROL C' ON THE TELETYPE KEYBOARD. TYPING A 'CONTROL C' CAUSES THE PROGRAM TO ENTER THE 'CONVERSATION MODE' ROUTINE. ALL THE PARAMETERS FOR TEST 14 WILL BE RETURNED TO THEIR DEFAULT VALUES.

3.5 DEFAULT RP11E UNIBUS ADDRESS

THE DEFAULT RP11E UNIBUS ADDRESS IS 176710. WHEN THE PROGRAM IS STARTED FROM EITHER LOCATION 200 OR LOCATION 204, ADDRESS 176710 WILL BE CHECKED AND IF NO RESPONSE IS RECEIVED FROM THAT ADDRESS THE PROGRAM WILL ENTER THE RP11E ADDRESS CHANGE ROUTINE. ADDRESS CHANGE OPERATION WILL BE IDENTICAL TO THE OPERATION WHEN THE PROGRAM IS STARTED FROM EITHER LOCATION 210 OR 214. (REFER TO SECTION 4.4)

4. OPERATING PROCEDURES

IF THE DIAGNOSTIC IS RUN ON A CPU WITHOUT A SWITCH REGISTER THEN A SOFTWARE SWITCH REGISTER IS USED WHICH ALLOWS THE USER THE SAME SWITCH OPTIONS AS THE HARDWARE SWITCH REGISTER. IF THE HARDWARE SWITCH REGISTER DOES NOT EXIST OR IF ONE DOES AND IT CONTAINS ALL ONES (177777) THEN THE SOFTWARE SWITCH REGISTER (LOC. 176) IS USED.

CONTROL:

THIS PROGRAM ALSO SUPPORTS THE DYNAMIC LOADING OF THE SOFTWARE SWITCH REGISTER (LOC. 176) FROM THE TTY. THIS CAN BE ACCOMPLISHED BY DOING THE FOLLOWING:

1) TYPE CONTROL G (↑G); THIS WILL ALLOW THE TTY TO ENTER DATA INTO LOC. 176 AT SELECTED POINTS WITHIN THE PROGRAM.

2) THE MACHINE WILL THEN TYPE: SWR=XXXXXXNEW= (XXXXXX IS THE OCTAL CONTENTS OF THE SOFTWARE SWITCH REGISTER.)

3) AFTER THE ''NEW='' HAS BEEN TYPED THEN THE OPERATOR CAN DO ONE OF THE FOLLOWING AT THE TTY:

   A) TYPE A NUMBER TO BE LOADED INTO LOC. 176 FOLLOWED BY A (CR). (ONLY NUMBERS BETWEEN 0-7 WILL BE ACCEPTED AND ONLY 6 NUMBERS WILL BE ALLOWED)
   IF A (CR) IS THE FIRST KEY DEPRESSED THE SOFTWARE SWITCH REGISTER CONTENTS WILL NOT BE CHANGED.

   B) IF A CONTROL U (↑U) IS DEPRESSED THEN THE PROGRAM WILL SEND YOU BACK TO STEP 2.

4.1 OPERATIONAL SWITCH SETTINGS

WITH SW(15:0)=1, THE PROGRAM WILL TYPEOUT ERRORS AND CONTINUE WITH THE 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<07>=1...DON'T TYPE DRIVE STATUS WHEN PROGRAM STARTED OR WHEN
          ENTERING CONVERSATION MODE.
          CHANGE HEAD SELECTION IN TEST 16.
SW<06>=1...DON'T USE MEMORY MANAGEMENT IN TEST 14
SW<05>=1...DISPLAY ALL READ RETRY ATTEMPTS IN TEST 14
SW<04>=1...TYPE ALL DATA COMPARE ERRORS IN TESTS 13 & 14
SW<03>=1...INHIBIT DATA COMPARE IN TESTS 13 & 14
SW<02>=1...BYPASS READ & DATA COMPARE IN TEST 14
SW<01>=1...BYPASS WRITE CHECK IN TEST 14
SW<00>=1...BYPASS WRITE IN TEST 14

4.2 TELETYPE KEYBOARD AND PRINTER UNIBUS ADDRESSES

THE PROGRAM ASSUMES THE FOLLOWING UNIBUS ADDRESSES. THESE ADDRESSES MAY BE CHANGED AT THE INDICATED LOCATION BEFORE STARTING THE PROGRAM.

MEMORY LOCATION	CONTENTS	FUNCTION
1136	177560	TTY KEYBOARD STATUS REG
1140	177562	TTY KEYBOARD BUFFER REG
1142	177564	TTY PRINTER STATUS REGISTER
1144	177566	TTY PRINTER BUFFER REG

(THE RP11E UNIBUS ADDRESS IS CHANGED IN RESPONSE TO THE PROGRAM'S TYPED REQUEST IF THE PROGRAM IS STARTED FROM EITHER LOCATION 210 OR 214 OR IF THE THERE IS NO RESPONSE WHEN LOCATION 176710 IS ADDRESSED.)

4.3 TTY KEYBOARD ENTRY CONTROL CODES AND CONVENTIONS

THE FOLLOWING CONTROL CODES AND CONVENTIONS ARE USED BY THE PROGRAM:

CARRIAGE RETURN (CR) - ALL ENTRIES MUST BE TERMINATED BY A CARRIAGE RETURN.

CONTROL U (↑U) - IF THIS CONTROL CODE IS TYPED BEFORE AN ENTRY IS TERMINATED, THE ENTRY IS CLEARED AND ANOTHER VALUE MAY BE ENTERED. A ↑U IS TYPED BY PRESSING THE 'CONTROL' AND 'U' KEYS TOGETHER.

RUBOUT (RO) - IF THIS CONTROL CODE IS TYPED, THE CHARACTER TYPED BEFORE THE 'RO' WILL BE DELETED FROM THE ENTRY. EACH TIME A 'RO' IS TYPED, THE LAST CHARACTER IN THE INPUT ENTRY IS DELETED. WHEN A 'RO' IS TYPED, THE ENTRY ROUTINE WILL TYPE A BACKSLASH ('\') CHARACTER AND THE CHARACTER BEING DELETED. WHEN A CHARACTER OTHER THAN A 'RO' IS TYPED, THE ENTRY ROUTINE WILL TYPE ANOTHER BACKSLASH CHARACTER AND TYPE THE NEW CHARACTER ENTERED. THE 'RO' CANNOT BE USED TO DELETE A 'CR'.

CONTROL C (↑C) - THE OPERATION OF THE '↑C' CHARACTER IS DESCRIBED IN SECTION 3.4. '↑C' MAY ALSO BE USED TO ABORT THE CONVERSATION MODE AND TO RETURN TO THE BEGINNING OF CONVERSATION MODE. '↑C' IS GENERATED BY PRESSING THE 'CONTROL' KEY AND THE 'C' KEY.

IF ONLY A 'CR' IS ENTERED IN RESPONSE TO AN ENTRY REQUEST FROM THE PROGRAM, THE PROGRAM WILL USE THE DEFAULT VALUE FOR THE REQUESTED ENTRY.

LEADING ZEROS ARE NOT REQUIRED ON ANY ENTRY.

4.4 RP11E UNIBUS ADDRESS CHANGE ROUTINE

WHEN THE PROGRAM IS STARTED FROM LOCATION 200 OR LOCATION 204, THE PROGRAM WILL ASSUME THAT THE RP11E IS AT UNIBUS ADDRESS 176710. WHEN THE PROGRAM IS STARTED FROM EITHER LOCATION 210 OR LOCATION 214, THE PROGRAM ENTERS A ROUTINE WHICH ALLOWS THE OPERATOR TO CHANGE THE RP11E UNIBUS ADDRESS TO BE USED BY THE PROGRAM.

WHEN THE ADDRESS CHANGE ROUTINE IS ENTERED, THE PROGRAM TYPES THE PRESENT RP11E UNIBUS ADDRESS:

```
RPADR = AAAAAA/
```

NOTE: 'AAAAAA' REPRESENTS THE PRESENT ADDRESS

THE ADDRESS TYPED OUT MAY BE USED BY TYPING A 'CR' ONLY. A NEW ADDRESS IS ENTERED BY TYPING THE NEW ADDRESS AND TERMINATING THE ENTRY WITH A 'CR'. (THE NEW ADDRESS WILL BE USED UNTIL ANOTHER ADDRESS IS ENTERED.)

AFTER THE OPERATOR HAS RESPONDED TO THE ADDRESS ENTRY REQUEST, THE PROGRAM WILL VERIFY THAT THE RP11E ADDRESS RESPONDS. IF NO RESPONSE IS RECEIVED FROM THE RP11E ADDRESS, THE FOLLOWING ERROR MESSAGE IS TYPED:

```
RP11E DIDN'T RESPOND TO ADDRESSING
RPADR
AAAAA
```

THE PROGRAM RETURNS TO THE ADDRESS ENTRY ROUTINE. THE PROGRAM WILL NOT CONTINUE UNTIL THE OPERATOR ENTERS AN ADDRESS FOR WHICH THERE IS A UNIBUS RESPONSE.

4.5 CONVERSATION MODE

THE PROGRAM CONTAINS A 'CONVERSATION MODE' ROUTINE WHICH ALLOWS THE OPERATOR TO SELECT AND EXECUTE INDIVIDUAL TESTS AND TO SPECIFY THE DRIVE TO BE USED. THE CONVERSATION MODE ALSO ALLOWS THE OPERATOR TO CHANGE THE OPERATING PARAMETERS OF THE 'DATA STORAGE/RETRIEVAL TEST' (TEST 14).

WHEN THE PROGRAM ENTERS CONVERSATION MODE, THE FOLLOWING MESSAGE IS TYPED:

```
ENTER TEST #:
```

A 'CR' MAY BE ENTERED OR A VALID TEST NUMBER (E.G., 0 - 17) MAY BE ENTERED. THE DEFAULT ENTRY ('CR') WILL CAUSE THE PROGRAM TO SELECT TESTS 0 - 14. IF A SINGLE TEST NUMBER IS ENTERED, ONLY THAT TEST WILL BE EXECUTED.

THE PROGRAM WILL THEN ASK FOR A DRIVE ASSIGNMENT.

```
ENTER DRIVE #:
```

A 'CR' ENTRY WILL CAUSE THE PROGRAM TO SELECT ALL ONLINE DRIVES FOR TESTING. IF A VALID DRIVE NUMBER IS ENTERED (0 - 7), THE PROGRAM WILL CHECK THE STATE OF THE DRIVE: IF THE DRIVE IS NOT ONLINE, THE DRIVE'S STATUS WILL BE TYPED AND THE PROGRAM WILL ASK FOR ANOTHER DRIVE ENTRY.

IF TEST 15, 16, OR 17 IS SELECTED, A 'CR' RESPONSE TO THE DRIVE NUMBER REQUEST WILL NOT BE ACCEPTED; A SPECIFIC DRIVE NUMBER MUST BE ENTERED FOR TEST 15, 16, OR 17.

IF TESTS 0 - 13, 15, 16 OR 17 HAVE BEEN SELECTED, OR IF THE ENTIRE TEST SEQUENCE WERE SPECIFIED, THE PROGRAM WILL BEGIN EXECUTING THE THE TEST OR SEQUENCE IMMEDIATELY. IF TEST 14 IS SPECIFIED, THE PROGRAM WILL ASK IF THE OPERATOR WANTS TO CHANGE PARAMETERS FOR TEST 14. THE FOLLOWING MESSAGE IS TYPED.

```
CHANGE PARAMETERS FOR TEST 14 ?
```

EITHER A 'CR' OR A 'Y' MAY BE ENTERED. IF A 'CR' IS ENTERED, THE PROGRAM WILL BYPASS THE TEST 14 PARAMETER ENTRY ROUTINE AND START TEST 14. IF A 'Y' IS ENTERED, THE PROGRAM STARTS THE TEST 14 PARAMETER CHANGE DIALOG.

THE PROGRAM WILL THEN TYPE THE WORD LENGTH FOR ALL TEST 14 TRANSFERS.

```
MAXIMUM WORD COUNT FOR TEST 14 (IN OCTAL): SSSSSS

NOTE: 'SSSSSS' = THE OCTAL VALUE OF THE WORD COUNT
```

THE PROGRAM WILL THEN ASK FOR A NEW WORD COUNT.

ENTER NEW WORD COUNT:

EITHER A VALUE (LESS THAN THE MAXIMUM TYPED ABOVE) OR A 'CR'
MAY BE ENTERED. IF A VALUE IS ENTERED, THE NEW VALUE WILL BE USED
FOR ALL DATA TRANSFER OPERATIONS IN TEST 14. WHEN THE PROGRAM IS
RESTARTED OR A 'CONTROL C' IS TYPED, THE WORD COUNT VALUE WILL BE
RETURNED TO THE SYSTEM DEFAULT VALUE. IF A 'CR' IS TYPED, THE
PROGRAM WILL USE THE DEFAULT WORD COUNT VALUE.

THE PROGRAM WILL THEN ASK FOR A PATTERN SELECTION WORD ENTRY.

ENTER PATTERN SELECTION CODE FOR TEST 14:

ENTER THE OCTAL VALUE OF THE PATTERN SELECTION WORD OR A 'CR' FOR
THE DEFAULT VALUE. REFER TO SECTION 7.9 FOR A DESCRIPTION OF
THE PATTERN SELECTION WORD AND ITS USE IN TEST 14. IF TESTING IS TO
BE AT A SPECIFIC DISK LOCATION, ENTER THE CODE FOR A SINGLE PATTERN.
(IF TESTING IS AT A SPECIFIC ADDRESS, THE PROGRAM WILL NOT ROTATE THE
DATA PATTERNS BUT WILL USE THE LOWEST NUMBERED PATTERN SPECIFIED
BY THE PATTERN SELECT WORD.)

THE PROGRAM WILL ASK IF THE OPERATOR WANTS TO TEST AT A SPECIFIC
DISK ADDRESS.

USE A SPECIFIC DISK ADDRESS FOR TEST 14 ?

ENTER EITHER A 'CR' OR A 'Y'. IF A 'Y' IS ENTERED, THE PROGRAM
WILL TYPE THE FOLLOWING REMINDER MESSAGE ABOUT DATA PATTERN USAGE.

(A SINGLE DATA PATTERN MUST HAVE BEEN SPECIFIED
IF NOT, PATTERN 0 (ALL ZEROS) WILL BE USED)

THE PROGRAM WILL ASK FOR CYLINDER, TRACK, AND SECTOR ADDRESSES.
AN ADDRESS MUST BE ENTERED, THE PROGRAM WILL NOT ACCEPT DEFAULT
ENTRIES FOR THESE ADDRESSES. THE ENTRY REQUESTS FOR THE ADDRESSES ARE
GIVEN BELOW.

ENTER CYLINDER ADDRESS:

ENTER TRACK ADDRESS:

ENTER SECTOR ADDRESS:

IF A SPECIFIC ADDRESS IS ENTERED FOR TESTING, TESTING WILL TAKE
PLACE ONLY AT THAT ADDRESS WITH THE WORD COUNT AND PATTERN ENTERED
ABOVE. THE TEST WILL MAKE ONE PASS AND GO TO END OF TEST. SW<14>
SHOULD BE SET TO INHIBIT THE END OF TEST MESSAGE TYPEOUT.

TO TERMINATE THE TEST OR SEQUENCE, EITHER TYPE A 'CONTROL C' OR HALT
THE PROCESSOR AND RESTART AT THE APPROPRIATE STARTING ADDRESS.

4.6 OPERATION OF TESTS 15, 16, & 17

TEST 15 AND TEST 17 CONTAIN IMBEDDED OPERATING INSTRUCTIONS. SETUP AND OPERATING INSTRUCTIONS ARE TYPED OUT AS THE TEST PROCEEDS.

TO OPERATE TEST 16, 'HEAD ALIGNMENT ROUTINE', REFER TO THE APPLICABLE MAINTENANCE MANUALS FOR THE DRIVE SETUP PROCEDURES. THE PROGRAM WILL TYPEOUT THE CONTROLLER SETUP PROCEDURES. TO CHANGE HEAD SELECTION, SW<07> MUST BE TOGGLED (TURNED ON, THEN OFF).

TO SETUP THE RP11E FOR HEAD ALIGNMENT, CONNECT A JUMPER BETWEEN PINS J13M2 AND J13C2 ON THE BACKPLANE.

5. ERROR HANDLING

5.1 ERROR REPORTING

WHEN THE PROGRAM DETECTS AN ERROR, THE ERROR ROUTINE IS CALLED; IF SW<13> IS NOT SET, THE ERROR MESSAGE FOR THAT ERROR WILL BE TYPED. EACH ERROR TYPEOUT CONTAINS THE FOLLOWING:

A. AN ERROR MESSAGE
B. A DATA HEADER LINE
C. A DATA LINE CONTAINING
    1. THE TEST NUMBER
    2. THE ADDRESS (PROGRAM COUNTER CONTENTS) WHERE THE ERROR CALL WAS MADE
    3. CONTENTS OF THE APPROPRIATE REGISTERS
    4. OTHER SPECIFIC DATA

5.2 ERROR TYPES

THE ERRORS DETECTED ARE DIVIDED INTO 2 CATEGORIES: FATAL AND NON-FATAL.

FATAL ERRORS ARE DRIVE UNSAFE ERRORS WHICH OCCUR ON THE DRIVE BEING USED FOR TESTING OR DRIVES WHICH GO OFFLINE DURING TESTING. IF EITHER OF THESE TWO CONDITIONS OCCUR, THE PROGRAM WILL TERMINATE TESTING USING THE DRIVE WHICH GAVE THE FATAL ERROR.

ALL OTHER ERROR TYPES ARE NON-FATAL, E.G., TESTING CONTINUES AFTER THE ERROR IS REPORTED.

6. RESTRICTIONS

6.1 SUBSYSTEMS WITH MIXED DRIVE TYPES

THE PROGRAM ASSUMES THAT ONLY ONE KIND OF DRIVE IS ATTACHED TO A SYSTEM (E.G. RP02'S & RPR02'S OR RP03'S). IF BOTH RP02'S (AND RPR02'S) OR RP03'S ARE ON THE SAME SYSTEM, THE PROGRAM WILL ASSUME THAT ALL OF THE DRIVES ARE RP03'S. TO TEST DRIVES ON A MIXED SYSTEM, TEST ALL OF THE SAME KIND OF DRIVE AT ONE TIME. THE OTHER DRIVES ON THE CONTROLLER MUST BE POWERED DOWN.

6.2 FORMATTED PACKS

EACH OF THE DRIVES BEING USED MUST HAVE A FORMATTED DISK PACK.

6.3 'CONTROL C' TERMINATION DURING TESTS 5 & 13

IF TESTS 5 OR 13 ARE TERMINATED WITH A 'CONTROL C', CYLINDER 0, TRACK 0, SECTOR 0 MAY BE LEFT WITH EITHER AN INVALID HEADER OR AN INVALID DATA FIELD. IF TEST 5 IS STOPPED, THE HEADER FOR SECTOR 0 MAY STILL BE MISFORMATTED; IF TEST 13 IS STOPPED, SECTOR 0 MAY BE STILL RECORDED IN 10/15 MODE.

6.4 CONTROLLER INTERRUPT

THE PROGRAM DOES NOT USE CONTROLLER INTERRUPT. THE PROGRAM WAITS FOR THE CONTROLLER 'READY' OR DRIVE 'READY' BITS TO SET TO INDICATE ORDER TERMINATION.

7. MISCELLANEOUS

7.1 RUN TIME

TO MAKE ONE PASS OF THE PROGRAM (TESTS 0 - 14) TAKES APPROXIMATELY 20 MINUTES FOR EACH RP02 OR RPR02 USED AND APPROXIMATELY 40 MINUTES FOR EACH RP03 USED.

7.2 STACK POINTER

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

7.3 END OF PASS/END OF TEST

THE PROGRAM WILL PERFORM ONE PASS FOR EACH DRIVE BEING USED FOR TESTING. END OF TEST WILL OCCUR WHEN EACH DRIVE BEING USED FOR TESTING HAS HAD ONE PASS.

7.4 SUBROUTINE CALLS

THE SUBROUTINE CALLS USED BY THE PROGRAM ARE:

A. 'SCOPE' (IOT INSTRUCTION). THIS CALL IS PLACED BETWEEN EACH TEST IN THE PROGRAM. THIS ROUTINE ESTABLISHES THE TEST ITERATION COUNT AND LOOP ON TEST AND LOOP ON ERROR ADDRESSES.

B. 'ERROR' (EMT INSTRUCTION). THIS CALL IS USED TO REPORT ALL ERRORS. A CALL TO THIS ROUTINE IS FOLLOWED BY A NUMBER WHICH IDENTIFIES THE ERROR MESSAGE WHICH WILL BE TYPED.

THE TRAP INSTRUCTION IS USED FOR THE FOLLOWING SUBROUTINE CALLS:

TYPE - TTY TYPEOUT ROUTINE
TYPOC - TYPE OCTAL NUMBER (WITH LEADING ZEROS)
TYPOS - TYPE OCTAL NUMBER (NO LEADING ZEROS)
TYPDS - TYPE DECIMAL NUMBER
RDCHR - READ A CHARACTER FROM THE TTY KEYBOARD
RDLIN - READ A LINE FROM THE TTY KEYBOARD
SAVREG - ROUTINE TO SAVE R0 - R5
RESREG - ROUTINE TO RESTORE R0 - R5

7.5 DISK SURFACE USAGE

THE PROGRAM USES THE FOLLOWING DISK LOCATIONS FOR TESTING.

TEST	DISK ADDRESS
0	CYL 0, TRK 0, SEC 0
1	CYL 0, TRK 0, SEC 0
2	CYL 0, TRK 0, SEC 0
3	CYL 0, TRK 0, SEC 0 MAXIMUM CYL, TRK 19, SEC 9
4	CYL 0, TRK 0, SEC 0
5	CYL 0, TRK 0, SEC 0
6	CYL 192, TRK 0, SEC 0
7	CYL 0, TRK 0, SEC 0
10	CYL 0, TRK 0, SEC 0 - 9
11	CYL 0, TRK 0 - 19, SEC 0
12	CYL 0, TRK 0, SEC 0
13	CYL 0, TRK 0, SEC 0
14	ENTIRE PACK
15	MAXIMUM CYL, TRK 0, SEC 0
16	N/A
17	NONE

7.6 TEST EXECUTION SEQUENCE

THE TEST SEQUENCE IS TESTS 0 - 14. TESTS 15, 16, & 17 ARE ONLY PERFORMED IF DIRECTED BY THE OPERATOR FROM CONVERSATION MODE. TESTS 15, 16, OR 17 CANNOT BE EXECUTED IN SEQUENCE.

7.7 DRIVE STATUS TYPEOUT

WHEN THE PROGRAM IS STARTED FROM LOCATION 200, 204, 210, OR 214 OR ENTERS THE CONVERSATION ROUTINE FROM A 'CONTROL C', THE PROGRAM WILL TYPE THE STATUS OF DRIVES 0 - 7. THIS DRIVE STATUS TYPEOUT CAN BE INHIBITED BY SETTING SW<07> BEFORE THE PROGRAM IS STARTED OR A 'CONTROL C' IS TYPED.

7.8 DATA COMPARSION USING MEMORY MANAGEMENT

IF THE SYSTEM HAS MORE THAN 28K, TEST 14 WILL ROTATE THE READ BUFFERS THROUGH ALL OF MEMORY. IF A DATA COMPARSION ERROR IS DETECTED, THE PROGRAM WILL TYPEOUT THE 'GOOD' AND THE 'BAD' DATA AND THE V I R T U A L ADDRESS OF THE COMPARSION ERROR LOCATION. THE CONTENTS OF THE APR'S IN USE WILL ALSO BE TYPED. IF MEMORY MANAGEMENT IS ENABLED, THE PROGRAM PERFORMS ALL DATA COMPARSIONS FROM VIRTUAL LOCATIONS 60000 - 137776 USING APR'S 3, 4, & 5. THE PHYSICAL ADDRESS OF THE COMPARSION ERROR CAN BE FORMED BY ADDING THE APPLICABLE 'KIPAR' REGISTER CONTENTS (THE REGISTERS ARE DISPLAYED) TO THE VIRTUAL ADDRESS OF THE ERROR.

7.9 PATTERN SELECTION WORD

THE DATA PATTERNS USED BY TEST 14 ARE SELECTED FROM A SELECTION WORD WHICH IS LOADED BY THE PROGRAM OR LOADED BY THE OPERATOR IN CONVERSATION MODE. THE PATTERN SELECTION WORD CONTAINS A BIT FOR EACH PATTERN TO BE USED: BIT00 REPRESENTS PATTERN 0, BIT01 REPRESENTS PATTERN 1, ETC. REFER TO SECTION 8.13 FOR THE CONTENTS OF EACH PATTERN.

THE OCTAL VALUES FOR EACH DATA PATTERN ARE AS FOLLOWS.

PATTERN #	OCTAL SELECTION CODE
0	1
1	2
2	4
3	10
4	20
5	40
6	100
7	200
8	400
9	1000
10	2000
11	4000
12	10000
13	20000
14	40000
15	100000

THE INDIVIDUAL OCTAL CODES ARE 'OR ED' TOGETHER TO FORM THE PATTERN SELECTION WORD. TO SELECT PATTERNS 0, 6, & 13, THE PATTERN SELECTION WORD COULD BE THE FOLLOWING OCTAL VALUE: 20101.

7.10 CONTROLLER/DRIVE OPERATION TIME LIMIT

ALL CONTROLLER AND DRIVE OPERATRIONS ARE TIMED BY THE PROGRAM. IF AN OPERATION EXCEEDS THE TIME LIMIT, THE PROGRAM WILL REPORT THE TIMEOUT, RESET THE CONTROLLER, TERMINATE THE CURRENT TEST, AND GO TO THE NEXT TEST. THE TIME LIMIT IS 150 MS TO 1.5 SEC (DEPENDING ON THE PROCESSOR).

8. TEST DESCRIPTIONS

8.1 TEST 0 - 'CLEAR' COMMAND TERMINATION TEST

VERIFY THAT 'CLEAR' TERMINATES AN OPERATION AND CAUSES CONTROLLER READY TO SET.

8.2 TEST 1 - WRITE CHECK COMMAND TEST

THIS TEST VERIFIES THE WRITE CHECK LOGIC AND TO VERIFY THAT A WRITE CHECK ERROR CAN BE DETECTED. 'FLOATING ONES' AND 'FLOATING ZEROS' PATTERNS ARE USED TO TEST THE WRITE CHECK COMPARE LOGIC.

8.3 TEST 2 - PARTIAL SECTOR WRITE TEST

CHECK THE ABILITY OF THE RP11E TO CLEAR THE REMAINDER OF A SECTOR ON A PARTIAL WRITE OPERATION. A SECTOR OF ALL ONES IS WRITTEN AND THEN A TWO WORD WRITE OPERATION IS PERFORMED. THE SECTOR IS THEN READ BACK AND VERIFIED. THE FIRST TWO WORDS SHOULD BE ONES AND THE REST SHOULD BE ZEROS.

8.4 TEST 3 - 'EOP' TEST

TEST THAT 'EOP' SETS WHEN THE CONTROLLER TRIES TO WRITE BEYOND THE LIMITS OF THE DRIVE. THE FIRST SECTOR OF THE PACK IS WRITTEN WITH ZEROS; THEN A TWO SECTOR WRITE OF ALL ONE'S IS ISSUED FOR THE MAXIMUM BE SET. THE FIRST SECTOR OF THE PACK IS CHECKED TO MAKE SURE THAT IS STILL CONTAINS ZEROS.

8.5 TEST 4 - 'PROG' ERROR TEST

VERIFY THAT THE RP11E GENERATES A 'PROG' ERROR IF A COMMAND IS ISSUED WHILE THE CONTROLLER IS BUSY.

8.6 TEST 5 - 'HEADER NOT FOUND' TEST

MISFORMAT THE FIRST SECTOR ON THE PACK AND THEN READ IT BACK. VERIFY THAT READ AND WRITE HEADER OPERATIONS WILL TRANSFER DATA CORRECTLY. ISSUE A WRITE COMMAND TO SECTOR ZERO; THIS SHOULD CAUSE 'HEADER NOT FOUND' TO SET. THE TEST SECTOR IS REFORMATTED.

8.7 TEST 6 - COMMAND BUFFERING TEST

ISSUE A SEEK COMMAND AND WAIT FOR DONE TO SET. THEN ISSUE A READ COMMAND WHILE THE HEADS ARE STILL MOVING. THE RP11E SHOULD HOLD THE READ COMMAND UNTIL THE SEEK IS COMPLETE.

8.8 TEST 7 - 'NXME' ERROR TEST

THE NON-EXISTENT MEMORY ERROR BIT IS TESTED BY ATTEMPTING TO READ 1 WORD INTO LOCATION 760000. 'NXME', 'NE', & 'ERR' SHOULD ALL BE SET.

8.9 TEST 10 - SECTOR ADDRESSING TEST

THE SECTOR ADDRESSING TEST TESTS THE CAPABILTIY OF THE RP11E TO LOCATE A SECTOR BY USING THE 'SOT' COUNTER (FOR 'HDR' OPERATIONS) AND TO LOCATE A SECTOR BY COMPARING THE HEADER CONTENTS AGAINST THE CYLINDER AND TRACK/SECTOR ADDRESS REGISTERS (NORMAL MODE). THE TEST IS PERFORMED IN 3 PARTS.

1. WRITE ALL HEADERS ON TRACK 0, CYLINDER 0 IN ASCENDING SEQUENCE FROM INDEX (0, 1, 2 ... 9).

2. READ EACH HEADER, BEGINNING WITH SECTOR 0, AND VERIFY THAT THE SECTOR FIELD IN THE HEADER IS CORRECT FOR THE HEADER EXPECTED.

VERIFY THAT THE 'SOT' COUNTER AND THE SECTOR ADDRESS IN 'RPDA' ARE EQUAL AND ARE THE EXPECTED VALUE. THE EXPECTED VALUE IS 1 GREATER THAN THE SECTOR READ. THE SECTOR ADDRESS REGISTER AND THE 'SOT' WILL BE INCREMENTED BY THE TIME THE PROGRAM IS ABLE TO READ THE REGISTER.

WRITE THE SECTOR'S ADDRESS IN THE DATA FIELD. (FOR SECTOR 0, AN OCTAL 12 IS WRITTEN.)

CONTINUE THIS SEQUENCE FOR THE REMAINING SECTORS ON THE TRACK.

3. READ THE DATA FROM EACH SECTOR USING A NORMAL (I.E., NON-HEADER) READ AND VERIFY THAT THE DATA IS CORRECT FOR THE EXPECTED SECTOR.

8.10 TEST 11 - TRACK ADDRESSING TEST

TRACK ADDRESSING IS TESTED BY WRITING THE TRACK'S ADDRESS AS DATA IN SECTOR 0 OF EACH TRACK IN CYLINDER 0. SECTOR 0 IS THEN READ BACK AND THE DATA IS CHECKED TO VERIFY THAT THE PROPER HEAD WAS SELECTED.

8.11 TEST 12 - EXTENDED MEMORY ADDRESS BIT TEST

THIS TEST TESTS THE OPERATION OF THE EXTENDED MEMORY ADDRESS BITS. IF THE SYSTEM DOES NOT HAVE MEMORY MANAGEMENT OR HAS MEMORY MANAGEMENT AND ONLY 32K, THE TEST WILL NOT BE PERFORMED.

1. THE PROGRAM WRITES A 2 WORD TEST SECTOR OF ALL ONES.

2. EXTENDED ADDRESS BIT 'MEX00' IS TESTED BY CLEARING LOCATION 200000 AND READING THE TEST SECTOR INTO LOCATION 177776. LOCATION 200000 IS CHECKED TO VERIFY THAT THE DATA IS CORRECT (ONES). THE PROGRAM VERIFIES THAT 'MEX00' HAS SET AND THAT 'MEX01' IS NOT SET.

3. IF THE SYSTEM HAS AT LEAST 64K, 'MEX01' IS TESTED. LOCATION 400000 IS CLEARED AND THE TEST SECTOR IS READ INTO LOCATION 377776 ('MEX00' IS SET FOR THE READ). LOCATION 400000 IS CHECKED FOR THE PROPER CONTENTS (ONES). 'MEX00' SHOULD HAVE RESET AND 'MEX01' SHOULD BE SET.

8.12 TEST 13 - RP11E DATA BUFFER REGISTER BIT TEST

THE DATA BUFFER REGISTER TEST VERIFIES THAT ALL 36 BITS OF THE DATA BUFFER REGISTER CAN BE SET AND CLEARED.

1. THE TEST WRITES THE FOLLOWING TEST PATTERN IN CYLINDER 0, TRACK 0, SECTOR 0 USING 10/15 MODE ('MODE' BIT SET).

052525
052525
000005
125252
125252
000012
052525
052525
000005
125252
.
.
.
ETC

2. THE TEST SECTOR IS THEN READ AND THE DATA IS COMPARED. AT THE COMPLETION OF THE TEST, THE PROGRAM RESTORES THE TEST SECTOR TO PDP-11 MODE.

8.13 TEST 14 - DATA STORAGE/RETRIEVAL TEST

THE DATA TEST PERFORMS DATA STORAGE AND RETREIVAL TESTING USING THE CONTROLLER AND THE CURRENTLY SELECTED DRIVE. UNLESS ALTERED BY THE OPERATOR, THE TEST WILL WRITE AND WRITE CHECK THE ENTIRE DISK PACK USING ALL 16 DATA PATTERNS IN A ROTATING MANNER. THE BUFFER SIZE USED FOR THE WRITE AND WRITE CHECK ORDERS IS DETERMINED BY THE AVAILABLE MEMORY ON THE SYSTEM (MINUS THE SPACE REQUIRED BY THE PROGRAM LOADER). THE PROGRAM WILL USE THE AVAILABLE MEMORY SIZE OR 8K, WHICHEVER IS LESS, AS THE BUFFER SIZE.

AFTER THE PACK HAS BEEN WRITTEN, THE TEST WILL READ AND COMPARE THE DATA. THE BUFFER SIZE USED FOR THE READ IS THE SAME SIZE AS THAT USED FOR THE WRITE/WRITE CHECK SEGEMENT OF THE TEST. FOR THE READ SEGEMENT OF THE TEST, THE PROGRAM ROTATES THE READ BUFFER THROUGH MEMORY SO THAT ALL OF AVAILABLE MEMORY IS USED BY THE READ AND COMPARE OPERATIONS. THE TEST USES MEMORY MANAGEMENT ON SYSTEMS WITH MORE THAN 28K. (MEMORY MANAGEMENT WILL NOT BE USED IF THE TEST IS RUN ON MEMORY MANAGEMENT SYSTEMS WITH 28K OR LESS.)

IF AN ERROR OCCURS DURING A READ OPERATION (EXCEPT DATA COMPARSION ERRORS), THE PROGRAM WILL RETRY THE READ ORDER UP TO 3 TIMES. WRITE AND WRITE CHECK ORDERS ARE NOT RETRIED.

THE DATA PATTERNS USED BY THE DATA TEST ARE GIVEN BELOW:

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

PAT 8	PAT 9	PAT 10	PAT 11	PAT 12	PAT 13	PAT 14	PAT 15
077577	000001	177776	172666	077777	177777	000000	177777
077577	000002	177775	155555	137777	177777	000000	177777
077577	000004	177773	172666	157777	177777	177777	000000
077577	000010	177767	155555	167777	177777	177777	000000
077577	000020	177757	172666	173777	177777	177777	000000
077577	000040	177737	155555	175777	177777	177777	000000
077577	000100	177677	172666	176777	177777	177777	000000
077577	000200	177577	155555	177377	177777	177777	000000
077577	000400	177377	172666	177577	177777	177777	000000
077577	001000	176777	155555	177677	177777	177777	000000
077577	002000	175777	172666	177737	177777	177777	000000
077577	004000	173777	155555	177757	177777	177777	000000
077577	010000	167777	172666	177767	177777	177777	000000
077577	020000	157777	155555	177773	177777	177777	000000
077577	040000	137777	172666	177775	177777	177777	000000
077577	100000	077777	155555	177776	177777	177777	000000

8.14 TEST 15 - POWER FAIL TEST

TEST THE ABILITY OF THE RP11E TO SENSE POWER FAILURE AND FOR THE DRIVES TO RETRACT THEIR HEADS. WHEN POWER IS TURNED ON AGAIN THE CYLINDER ADDRESS IS TESTED FOR ZERO. AFTER TYPING THE MESSAGE REQUESTING POWER TO BE TURNED OFF, THE PROGRAM GOES INTO A LOOP READING FROM THE SELECTED DISK DRIVE. AFTER POWER IS RESTORED, MEMORY IS CHECKED TO SEE THAT THE DISK DID NOT TRANSFER ANY DATA TO MEMORY WHILE POWER WAS GOING DOWN.

8.15 TEST 16 - HEAD ALIGNMENT ROUTINE

THIS ROUTINE PERFORMS HEAD SELECTION AS DIRECTED FROM THE KEYBOARD; ALIGNMENT OF THE SELECTED HEAD MAY BE CHECKED OR ADJUSTED.

```
                                8.16    TEST 17 - RP11E/DRIVE CONTROL PANEL SWITCH TEST

                                        THIS TEST TESTS THE 'ENABLE/DISABLE' AND 'READ ONLY' SWITCHES
                                        ON THE DRIVE CONTROL PANEL AND TESTS THE 'WRITE LOCKOUT' AND
                                        'LOA' SWITCHES ON THE RP11E CONTROL PANEL.



                                9.      PROGRAM LISTING
                                        ---------------

                                        %
                                .TITLE  MD-11-DZRPY-B, RP11-E FUNCTIONAL LOGIC & R/W TEST
                                ;#COPYRIGHT (C) 1975,1976
                                ;#DIGITAL EQUIPMENT CORP.
                                ;#MAYNARD, MASS. 01754
                                ;#
                                ;#PROGRAM BY C. HESS/F. ROEMER
                                ;#
                                ;#THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;#PACKAGE (MAINDEC-11-DZQAC-C1),MAR 24, 1976.
                                ;#
                                .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
                                ;#          7            DON'T TYPE DRIVE STATUS WHEN PROGRAM STARTED OR WHEN
                                ;#                       ENTERING CONVERSATION MODE
                                ;#                       CHANGE HEAD SELECTION IN TEST 16
                                ;#          6            DON'T USE MEMORY MANAGEMENT IN TEST 14
                                ;#          5            DISPLAY ALL READ RETRY ATTEMPTS IN TEST 14
                                ;#          4            PRINT ALL COMPARE ERRORS IN TESTS 13 & 14
                                ;#          3            INHIBIT DATA COMPARE IN TESTS 13 & 14
                                ;#          2            BYPASS READ AND DATA COMPARE IN TEST 14
                                ;#          1            BYPASS WRITE CHECK IN TEST 14
                                ;#          0            BYPASS WRITE IN TEST 14

                                .SBTTL  TRAP CATCHER

                        000000          .=0
                                ;#ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A "..+2,HALT"
                                ;#SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
                                ;#LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
                        000174          .=174
000174  000000         DISPREG: .WORD  0               ;;SOFTWARE DISPLAY REGISTER
000176  000000         SWREG:   .WORD  0               ;;SOFTWARE SWITCH REGISTER

                                .SBTTL  ACT11 HOOKS

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

```
                                ;HOOKS REQUIRED BY ACT11
                000200                  $SVPC=.                 ;SAVE PC
                000046                  .=46
000046  017040                          $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOP
                000052                  .=52
000052  000000                          .WORD   0               ;;2)SET LOC.52 TO ZERO
                000200                  .=$SVPC                 ;; RESTORE PC

                                .SBTTL  STARTING ADDRESSES

                000200          .=200
                                ;*200 = NORMAL START
000200  000137  003654                  JMP     START1
                                ;*204 = SELECT OPERATING PARAMETERS
000204  000137  003676                  JMP     START2
                                ;*210 = SELECT RP11E ADDRESS
000210  000137  003644                  JMP     START3
                                ;*214 = COMBINATION OF 204 AND 210
000214  000137  003666                  JMP     START4

                                .SBTTL  BASIC DEFINITIONS

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

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

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

                                ;*PRIORITY LEVEL DEFINITIONS
                000000          PR0=    0               ;;PRIORITY LEVEL 0
                000040          PR1=    40              ;;PRIORITY LEVEL 1
                000100          PR2=    100             ;;PRIORITY LEVEL 2
                000140          PR3=    140             ;;PRIORITY LEVEL 3
```

```
1021            000200                  PR4=    200             ;;PRIORITY LEVEL 4
1022            000240                  PR5=    240             ;;PRIORITY LEVEL 5
1023            000300                  PR6=    300             ;;PRIORITY LEVEL 6
1024            000340                  PR7=    340             ;;PRIORITY LEVEL 7
1025
1026                                    ;*"SWITCH REGISTER" SWITCH DEFINITIONS
1027            100000                  SW15=   100000
1028            040000                  SW14=   40000
1029            020000                  SW13=   20000
1030            010000                  SW12=   10000
1031            004000                  SW11=   4000
1032            002000                  SW10=   2000
1033            001000                  SW09=   1000
1034            000400                  SW08=   400
1035            000200                  SW07=   200
1036            000100                  SW06=   100
1037            000040                  SW05=   40
1038            000020                  SW04=   20
1039            000010                  SW03=   10
1040            000004                  SW02=   4
1041            000002                  SW01=   2
1042            000001                  SW00=   1
1043                                    .EQUIV  SW09,SW9
1044                                    .EQUIV  SW08,SW8
1045                                    .EQUIV  SW07,SW7
1046                                    .EQUIV  SW06,SW6
1047                                    .EQUIV  SW05,SW5
1048                                    .EQUIV  SW04,SW4
1049                                    .EQUIV  SW03,SW3
1050                                    .EQUIV  SW02,SW2
1051                                    .EQUIV  SW01,SW1
1052                                    .EQUIV  SW00,SW0
1053
1054                                    ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
1055         100000                     BIT15=  100000
1056         040000                     BIT14=  40000
1057         020000                     BIT13=  20000
1058         010000                     BIT12=  10000
1059         004000                     BIT11=  4000
1060         002000                     BIT10=  2000
1061         001000                     BIT09=  1000
1062         000400                     BIT08=  400
1063         000200                     BIT07=  200
1064         000100                     BIT06=  100
1065         000040                     BIT05=  40
1066         000020                     BIT04=  20
1067         000010                     BIT03=  10
1068         000004                     BIT02=  4
1069         000002                     BIT01=  2
1070         000001                     BIT00=  1
1071                                    .EQUIV  BIT09,BIT9
1072                                    .EQUIV  BIT08,BIT8
1073                                    .EQUIV  BIT07,BIT7
1074                                    .EQUIV  BIT06,BIT6
1075                                    .EQUIV  BIT05,BIT5
1076                                    .EQUIV  BIT04,BIT4
```

```
                        .EQUIV  BIT03,BIT3
                        .EQUIV  BIT02,BIT2
                        .EQUIV  BIT01,BIT1
                        .EQUIV  BIT00,BIT0

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

                        ;*KT11 VECTOR ADDRESS

000250                  MMVEC=  250

                        ;*KT11 STATUS REGISTER ADDRESSES

177572                  SR0=    177572
177574                  SR1=    177574
177576                  SR2=    177576
172516                  SR3=    172516

                        ;*KERNEL "I" PAGE DESCRIPTOR REGISTERS

172300                  KIPDR0= 172300
172302                  KIPDR1= 172302
172304                  KIPDR2= 172304
172306                  KIPDR3= 172306
172310                  KIPDR4= 172310
172312                  KIPDR5= 172312
172314                  KIPDR6= 172314
172316                  KIPDR7= 172316

                        ;*KERNEL "I" PAGE ADDRESS REGISTERS

172340                  KIPAR0= 172340
172342                  KIPAR1= 172342
172344                  KIPAR2= 172344
172346                  KIPAR3= 172346
172350                  KIPAR4= 172350
172352                  KIPAR5= 172352
172354                  KIPAR6= 172354
172356                  KIPAR7= 172356


                        ;RP11E REGISTER INDEX EQUATES

```

```
1133    000000          RPDS=00
1134    000002          RPER=02
1135    000004          RPCS=04
1136    000006          RPWC=06
1137    000010          RPBA=10
1138    000012          RPCA=12
1139    000014          RPDA=14
1140    000016          RPM1=16
1141    000020          RPM2=20
1142    000022          RPM3=22
1143    000024          SUCA=24
1144    000026          SILO=26
1145
1146                    ;RP11 OP CODE DEFINITIONS
1147
1148    000001          CLEAR=1                 ;CLEAR THE CONTROLLER
1149    000003          WRTSEK=3                ;WRITE WITH IMPLIED SEEK AND HEAD SELECTION
1150    000005          RDSEK=5                 ;READ WITH IMPLIED SEEK AND HEAD SELECT
1151    000007          WRTCHK=7                ;WRITE CHECK WITH IMPLIIED SEEK AND HEAD SELECT
1152    000011          SEEK=11                 ;SEEK
1153    000013          WRITE=13                ;WRITE (NO IMPLIED SEEK OR HEAD SELECT)
1154    000015          HOMSEK=15               ;HOME SEEK
1155    000017          READ=17                 ;READ (NO IMPLIED SEEK OR HEAD SELECT)
1156
1157
1158                    ;RPDS REGISTER BIT DEFINITIONS
1159
1160    000400          SUWP=400                ;SELECTED UNIT WRITE PROTECTED
1161    001000          SUFU=1000               ;SELECTED UNIT FILE UNSAFE
1162    004000          SUSI=4000               ;SELECTED UNIT SEEK INCOMPLETE
1163    020000          SURP03=20000            ;SELECTED UNIT IS AN RP03
1164    040000          SUOL=40000              ;SELECTED UNIT IS ONLINE
1165    100000          SURDY=100000            ;SELECTED UNIT IS READY
1166
1167                    ;RPCS REGISTER BIT DEFINITIONS
1168
1169    000020          MEX0=20                 ;EXTENDED MEMORY BIT #1
1170    000040          MEX1=40                 ;EXTENDED MEMORY BIT #2
1171    000100          IDE=100                 ;INTERRUPT ON DONE ENABLE BIT
1172    000200          RDY=200                 ;RP11 CONTROLLER READY BIT
1173    004000          HDR=4000                ;HEADER OPERATION BIT
1174    010000          MODE=10000              ;MODE BIT
1175    020000          AIE=20000               ;ATTENTION INTERRUPT ENABLE BIT
1176    040000          HE=40000                ;HARD ERROR BIT
1177    100000          ERR=100000              ;ERROR BIT
1178
1179
```

```
                                .SBTTL  COMMON TAGS

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

                        001100          .=1100
001100                  $CMTAG:                         ;;START OF COMMON TAGS
001100  000000          $PASS:  .WORD   0               ;;CONTAINS PASS COUNT
001102  000             $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001103  000             $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001104  000000          $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001106  000000          $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001110  000000          $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001112  000000          $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001114  000             $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001115  001             $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001116  000000          $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001120  000000          $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001122  000000          $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001124  000000          $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001126  000000          $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001130  000000                  .WORD   0               ;;RESERVED--NOT TO BE USED
001132  000000                  .WORD   0
001134  000             $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001135  000             $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001136  000000                  .WORD   0
001140  177570          SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001142  177570          DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
001144  177560          $TKS:   177560                  ;;TTY KBD STATUS
001146  177562          $TKB:   177562                  ;;TTY KBD BUFFER
001150  177564          $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
001152  177566          $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001154  000             $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001155  002             $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001156  012             $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001157  000             $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001160  000000          $TMP0:  .WORD   0               ;;USER DEFINED
001162  000000          $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
001164  000000          $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
001166  177607  000377  $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
001172  077             $QUES:  .ASCII  /?/             ;;QUESTION MARK
001173  015             $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
001174  000012          $LF:    .ASCIZ  <12>            ;;LINE FEED
                        ;;**********************************************************************
001176  000000          CHKDRV: .WORD   0               ;ADDRESS OF DRIVE BEING USED
001200  000000          BYPASS: .WORD   0               ;TEST EXIT ADDRESS
001202  000000          FLAG:   .WORD   0               ;INTERNAL PROGRAM FLAG WORD
001204  000000          PATNUM: .WORD   0               ;CURRENTLY SELECTED PATTERN
001206  000000          CNTRLC: .WORD   0               ;CONVERSATION MODE FLAG
001210  000000          MAXCYL: .WORD   0               ;MAXIMUM CYLINDER ADDRESS FOR THE DRIVE IN USE
001212  001750          TIMOUT: .WORD   1000.           ;TIME ALLOWED FOR AN OPERATION
001214  000000          ICNT:   .WORD   0               ;TEST ITERATION COUNT
001216  000000          BUSADR: .WORD   0               ;CHANGE RP11 ADDRESS FLAG
001220  000000          STALLT: .WORD   0               ;STALL VALUE
001222  000000          MMACTV: .WORD   0               ;MEMORY MANAGEMENT IS ACTIVE FLAG
```

```
1236  001224  000000  000000          ACTMEM: .WORD   0,0             ;PHYSICAL ADDRESS OF NEXT READ BUFFER
1237  001230  000000  000000          MAXMEM: .WORD   0,0             ;ADDRESS OF LAST MEMORY BANK (FILLED BY $SIZE)
1238  001234  000000  000000          HIMEM:  .WORD   0,0             ;PHYSICAL ADDRESS OF LAST MEMORY BANK (FOR TEST 14)
1239  001240  000000  000000          LOMEM:  .WORD   0,0             ;PHYSICAL ADDRESS OF FIRST BUFFER LOCATION
1240  001244  000000  000000          PAKSIZ: .WORD   0,0             ;SECTORS ON THE PACK ON THE DRIVE BEING TESTED
1241  001250  000000                  INCSEC: .WORD   0               ;SECTOR INCREMENT VALUE
1242  001252  000000                  INCTRK: .WORD   0               ;TRACK INCREMENT VALUE
1243  001254  000000                  BLKS14: .WORD   0               ;NUMBER SECTORS IN TRANSFER IN TEST 14
1244  001256  000000                  WC14:   .WORD   0               ;WORD COUNT FOR TEST 14
1245  001260  000000                  OPRSEL: .WORD   0               ;OPERATOR SELECTED ADDRESS FLAG
1246  001262  020000                  LMTBK:  .WORD   8192.           ;LARGEST TRANSFER PERMITTED
1247  001264  000000                  MAXWC:  .WORD   0               ;MAXMUM WORD COUNT ALLOWED ON THIS SYSTEM
1248  001266  000000                  SCYL:   .WORD   0               ;OPERATOR SELECTED CYLINDER
1249  001270  000000                  STRK:   .WORD   0               ;OPERATOR SELECTED TRACK
1250  001272  000000                  SSEC:   .WORD   0               ;OPERATOR SELECTED SECTOR
1251  001274  000000                  RDERR:  .WORD   0               ;READ RETRY COUNTER
1252  001276  000000                  TSTNMS: .WORD   0               ;TEST SELECTION WORD.  BIT 0 = TEST 0, ETC.
1253  001300  000000                  PATTRN: .WORD   0               ;PATTERN SELECTION WORD FOR TEST 14
1254  001302  000000                  MEMSIZ: .WORD   0               ;ADDRESS OF HIGHEST NON-MEMORY MANAGEMENT
1255                                                                  ;MEMORY LOCATION
1256  001304  000000                  DRVTYP: .WORD   0               ;CONTAINS A BIT FOR EACH RP03 DRIVE
1257                                                                  ;ON THE SYSTEM.  BIT00 = DRIVE 0, ETC.
1258  001306  000000                  DRVMSK: .WORD   0               ;DRIVE TEST SELECTION MASK
1259  001310  000000                  DRVBAD: .WORD   0               ;CONTAINS A BIT FOR THE DRIVE IF THE DRIVE
1260                                                                  ;GOES OFFLINE OR BECOMES UNSAFE DURING TESTING.
1261                                                                  ;BIT00 = DRIVE 0, ETC.
1262  001312  000000                  DRVSEL: .WORD   0               ;CONTAINS A BIT FOR EACH DRIVE TO BE TESTED
1263  001314  000000                  RETRY:  .WORD   0               ;RETRY COUNTER FOR TEST 14
1264  001316  000003                  LRETRY: .WORD   3               ;RETRY LIMIT FOR TEST 14
1265  001320  000003                  CMP.CT: .WORD   3               ;NUMBER OF COMPARE ERROR TYPEOUTS
1266  001322  000000                  LDRFLG: .WORD   0               ;WHEN =0 LOADERS ARE AT TOP OF MEMORY
1267
1268                                  ;RP11 ADDRESS VALUES
1269
1270  001324  176710                  RPADR:  .WORD   176710          ;RP11 BUS ADDRESS
1271  001326  000254  000256          RPVEC:  .WORD   254,256         ;RP11 VECTOR ADDRESS
1272  001332  000240                  RPPRIO: .WORD   <5*32.>          ;RP11 PRIORITY
1273
1274                                  ;SAVE THE RP11E REGISTERS HERE
1275
1276  001334  000000                  $RPDS:  .WORD   0               ;DRIVE STATUS REGISTER
1277  001336  000000                  $RPER:  .WORD   0               ;ERROR REGISTER
1278  001340  000000                  $RPCS:  .WORD   0               ;COMMAND & STATUS REGISTER
1279  001342  000000                  $RPWC:  .WORD   0               ;WORD COUNT REGISTER
1280  001344  000000                  $RPBA:  .WORD   0               ;BUFFER ADDRESS REGISTER
1281  001346  000000                  $RPCA:  .WORD   0               ;CURRENT CYLINDER ADDRESS REGISTER
1282  001350  000000                  $RPDA:  .WORD   0               ;TRACK-SECTOR ADDRESS REGISTER
1283  001352  000000                  $RPM1:  .WORD   0               ;MAINTENANCE REGISTER #1
1284  001354  000000                  $SUCA:  .WORD   0               ;SELECTED UNIT CYLINDER ADDRESS REGISTER
1285  001356  000000                  $SILO:  .WORD   0               ;SILO REGISTER
1286
1287                                  ;ATTENTION BITS
1288
1289  001360      001                 ATABIT: .BYTE   1               ;DRIVE 0
1290  001361      002                         .BYTE   2               ;DRIVE 1
1291  001362      004                         .BYTE   4               ;DRIVE 2
```

```
1292  001363  010                         .BYTE   10              ;DRIVE 3
1293  001364  020                         .BYTE   20              ;DRIVE 4
1294  001365  040                         .BYTE   40              ;DRIVE 5
1295  001366  100                         .BYTE   100             ;DRIVE 6
1296  001367  200                         .BYTE   200             ;DRIVE 7
1297
1298
1299                                ;DRIVE STATUS INDICATORS (DRVSTA=8 BYTES)
1300                                        ;DRVSTA=0 DRIVE NONEXISTENT OR OFFLINE
1301                                        ;DRVSTA>0 DRIVE IS ONLINE
1302                                        ;DRVSTA<0 DRIVE IS UNSAFE
1303
1304  001370  000                   DRVSTA: .BYTE   0               ;DRIVE 0
1305  001371  000                         .BYTE   0               ;DRIVE 1
1306  001372  000                         .BYTE   0               ;DRIVE 2
1307  001373  000                         .BYTE   0               ;DRIVE 3
1308  001374  000                         .BYTE   0               ;DRIVE 4
1309  001375  000                         .BYTE   0               ;DRIVE 5
1310  001376  000                         .BYTE   0               ;DRIVE 6
1311  001377  000                         .BYTE   0               ;DRIVE 7
1312
1313                                ;BIT TABLE
1314
1315  001400  000001                BITS:   .WORD   BIT00
1316  001402  000002                        .WORD   BIT01
1317  001404  000004                        .WORD   BIT02
1318  001406  000010                        .WORD   BIT03
1319  001410  000020                        .WORD   BIT04
1320  001412  000040                        .WORD   BIT05
1321  001414  000100                        .WORD   BIT06
1322  001416  000200                        .WORD   BIT07
1323  001420  000400                        .WORD   BIT08
1324  001422  001000                        .WORD   BIT09
1325  001424  002000                        .WORD   BIT10
1326  001426  004000                        .WORD   BIT11
1327  001430  010000                        .WORD   BIT12
1328  001432  020000                        .WORD   BIT13
1329  001434  040000                        .WORD   BIT14
1330  001436  100000                        .WORD   BIT15
1331
1332  001440  001500                PAT.PT: .WORD   PAT0            ;TABLE OF POINTERS WHICH POINT TO THE
1333  001442  001540                        .WORD   PAT1            ;PATTERNS USED BY THE DATA TEST
1334  001444  001600                        .WORD   PAT2
1335  001446  001640                        .WORD   PAT3
1336  001450  001700                        .WORD   PAT4
1337  001452  001740                        .WORD   PAT5
1338  001454  002000                        .WORD   PAT6
1339  001456  002040                        .WORD   PAT7
1340  001460  002100                        .WORD   PAT8
1341  001462  002140                        .WORD   PAT9
1342  001464  002200                        .WORD   PAT10
1343  001466  002240                        .WORD   PAT11
1344  001470  002300                        .WORD   PAT12
1345  001472  002340                        .WORD   PAT13
1346  001474  002400                        .WORD   PAT14
1347  001476  002440                        .WORD   PAT15
```

```
1348
1349                                    ;PATTERNS 0 THRU 15
1350
1351  001500  000000    PAT0:   .WORD   000000          ;PATTERN 0
1352  001502  000000            .WORD   000000
1353  001504  000000            .WORD   000000
1354  001506  000000            .WORD   000000
1355  001510  000000            .WORD   000000
1356  001512  000000            .WORD   000000
1357  001514  000000            .WORD   000000
1358  001516  000000            .WORD   000000
1359  001520  000000            .WORD   000000
1360  001522  000000            .WORD   000000
1361  001524  000000            .WORD   000000
1362  001526  000000            .WORD   000000
1363  001530  000000            .WORD   000000
1364  001532  000000            .WORD   000000
1365  001534  000000            .WORD   000000
1366  001536  000000            .WORD   000000
1367
1368  001540  000001    PAT1:   .WORD   000001          ;PATTERN 1
1369  001542  000003            .WORD   000003
1370  001544  000007            .WORD   000007
1371  001546  000017            .WORD   000017
1372  001550  000037            .WORD   000037
1373  001552  000077            .WORD   000077
1374  001554  000177            .WORD   000177
1375  001556  000377            .WORD   000377
1376  001560  000777            .WORD   000777
1377  001562  001777            .WORD   001777
1378  001564  003777            .WORD   003777
1379  001566  007777            .WORD   007777
1380  001570  017777            .WORD   017777
1381  001572  037777            .WORD   037777
1382  001574  077777            .WORD   077777
1383  001576  177777            .WORD   177777
1384
1385  001600  177776    PAT2:   .WORD   177776          ;PATTERN 2
1386  001602  177774            .WORD   177774
1387  001604  177770            .WORD   177770
1388  001606  177760            .WORD   177760
1389  001610  177740            .WORD   177740
1390  001612  177700            .WORD   177700
1391  001614  177600            .WORD   177600
1392  001616  177400            .WORD   177400
1393  001620  177000            .WORD   177000
1394  001622  176000            .WORD   176000
1395  001624  174000            .WORD   174000
1396  001626  170000            .WORD   170000
1397  001630  160000            .WORD   160000
1398  001632  140000            .WORD   140000
1399  001634  100000            .WORD   100000
1400  001636  000000            .WORD   000000
1401
1402  001640  000000    PAT3:   .WORD   000000          ;PATTERN 3
1403  001642  000000            .WORD   000000
```

```
1404  001644  000000                          .WORD   000000
1405  001646  177777                          .WORD   177777
1406  001650  177777                          .WORD   177777
1407  001652  177777                          .WORD   177777
1408  001654  000000                          .WORD   000000
1409  001656  000000                          .WORD   000000
1410  001660  177777                          .WORD   177777
1411  001662  177777                          .WORD   177777
1412  001664  000000                          .WORD   000000
1413  001666  177777                          .WORD   177777
1414  001670  000000                          .WORD   000000
1415  001672  177777                          .WORD   177777
1416  001674  177777                          .WORD   177777
1417  001676  000000                          .WORD   000000
1418
1419  001700  000000                  PAT4:   .WORD   000000          ;PATTERN 4
1420  001702  010421                          .WORD   010421
1421  001704  021042                          .WORD   021042
1422  001706  031463                          .WORD   031463
1423  001710  042104                          .WORD   042104
1424  001712  052525                          .WORD   052525
1425  001714  063146                          .WORD   063146
1426  001716  073567                          .WORD   073567
1427  001720  104210                          .WORD   104210
1428  001722  114631                          .WORD   114631
1429  001724  125252                          .WORD   125252
1430  001726  135673                          .WORD   135673
1431  001730  146314                          .WORD   146314
1432  001732  156735                          .WORD   156735
1433  001734  167356                          .WORD   167356
1434  001736  177777                          .WORD   177777
1435
1436  001740  052525                  PAT5:   .WORD   052525          ;PATTERN 5
1437  001742  052525                          .WORD   052525
1438  001744  052525                          .WORD   052525
1439  001746  125252                          .WORD   125252
1440  001750  125252                          .WORD   125252
1441  001752  125252                          .WORD   125252
1442  001754  052525                          .WORD   052525
1443  001756  052525                          .WORD   052525
1444  001760  125252                          .WORD   125252
1445  001762  125252                          .WORD   125252
1446  001764  052525                          .WORD   052525
1447  001766  125252                          .WORD   125252
1448  001770  052525                          .WORD   052525
1449  001772  125252                          .WORD   125252
1450  001774  125252                          .WORD   125252
1451  001776  052525                          .WORD   052525
1452
1453  002000  007417                  PAT6:   .WORD   007417          ;PATTERN 6
1454  002002  007417                          .WORD   007417
1455  002004  007417                          .WORD   007417
1456  002006  170360                          .WORD   170360
1457  002010  170360                          .WORD   170360
1458  002012  170360                          .WORD   170360
1459  002014  007417                          .WORD   007417
```

```
1460  002016  007417                          .WORD   007417
1461  002020  170360                          .WORD   170360
1462  002022  170360                          .WORD   170360
1463  002024  007417                          .WORD   007417
1464  002026  170360                          .WORD   170360
1465  002030  007417                          .WORD   007417
1466  002032  170360                          .WORD   170360
1467  002034  170360                          .WORD   170360
1468  002036  007417                          .WORD   007417
1469
1470  002040  026455                  PAT7:   .WORD   026455          ;PATTERN 7
1471  002042  026455                          .WORD   026455
1472  002044  026455                          .WORD   026455
1473  002046  151322                          .WORD   151322
1474  002050  151322                          .WORD   151322
1475  002052  151322                          .WORD   151322
1476  002054  026455                          .WORD   026455
1477  002056  026455                          .WORD   026455
1478  002060  151322                          .WORD   151322
1479  002062  151322                          .WORD   151322
1480  002064  026455                          .WORD   026455
1481  002066  151322                          .WORD   151322
1482  002070  026455                          .WORD   026455
1483  002072  151322                          .WORD   151322
1484  002074  151322                          .WORD   151322
1485  002076  026455                          .WORD   026455
1486
1487  002100  077577                  PAT8:   .WORD   077577          ;PATTERN 8 (WORST CASE PATTERN)
1488  002102  077577                          .WORD   077577
1489  002104  077577                          .WORD   077577
1490  002106  077577                          .WORD   077577
1491  002110  077577                          .WORD   077577
1492  002112  077577                          .WORD   077577
1493  002114  077577                          .WORD   077577
1494  002116  077577                          .WORD   077577
1495  002120  077577                          .WORD   077577
1496  002122  077577                          .WORD   077577
1497  002124  077577                          .WORD   077577
1498  002126  077577                          .WORD   077577
1499  002130  077577                          .WORD   077577
1500  002132  077577                          .WORD   077577
1501  002134  077577                          .WORD   077577
1502  002136  077577                          .WORD   077577
1503
1504  002140  000001                  PAT9:   .WORD   000001          ;PATTERN 9
1505  002142  000002                          .WORD   000002
1506  002144  000004                          .WORD   000004
1507  002146  000010                          .WORD   000010
1508  002150  000020                          .WORD   000020
1509  002152  000040                          .WORD   000040
1510  002154  000100                          .WORD   000100
1511  002156  000200                          .WORD   000200
1512  002160  000400                          .WORD   000400
1513  002162  001000                          .WORD   001000
1514  002164  002000                          .WORD   002000
1515  002166  004000                          .WORD   004000
```

```
1516  002170  010000                        .WORD   010000
1517  002172  020000                        .WORD   020000
1518  002174  040000                        .WORD   040000
1519  002176  100000                        .WORD   100000
1520
1521  002200  177776          PAT10:        .WORD   177776          ;PATTERN 10
1522  002202  177775                        .WORD   177775
1523  002204  177773                        .WORD   177773
1524  002206  177767                        .WORD   177767
1525  002210  177757                        .WORD   177757
1526  002212  177737                        .WORD   177737
1527  002214  177677                        .WORD   177677
1528  002216  177577                        .WORD   177577
1529  002220  177377                        .WORD   177377
1530  002222  176777                        .WORD   176777
1531  002224  175777                        .WORD   175777
1532  002226  173777                        .WORD   173777
1533  002230  167777                        .WORD   167777
1534  002232  157777                        .WORD   157777
1535  002234  137777                        .WORD   137777
1536  002236  077777                        .WORD   077777
1537
1538  002240  172666          PAT11:        .WORD   172666          ;PATTERN 11
1539  002242  155555                        .WORD   155555
1540  002244  172666                        .WORD   172666
1541  002246  155555                        .WORD   155555
1542  002250  172666                        .WORD   172666
1543  002252  155555                        .WORD   155555
1544  002254  172666                        .WORD   172666
1545  002256  155555                        .WORD   155555
1546  002260  172666                        .WORD   172666
1547  002262  155555                        .WORD   155555
1548  002264  172666                        .WORD   172666
1549  002266  155555                        .WORD   155555
1550  002270  172666                        .WORD   172666
1551  002272  155555                        .WORD   155555
1552  002274  172666                        .WORD   172666
1553  002276  155555                        .WORD   155555
1554
1555  002300  077777          PAT12:        .WORD   077777          ;PATTERN 12
1556  002302  137777                        .WORD   137777
1557  002304  157777                        .WORD   157777
1558  002306  167777                        .WORD   167777
1559  002310  173777                        .WORD   173777
1560  002312  175777                        .WORD   175777
1561  002314  176777                        .WORD   176777
1562  002316  177377                        .WORD   177377
1563  002320  177577                        .WORD   177577
1564  002322  177677                        .WORD   177677
1565  002324  177737                        .WORD   177737
1566  002326  177757                        .WORD   177757
1567  002330  177767                        .WORD   177767
1568  002332  177773                        .WORD   177773
1569  002334  177775                        .WORD   177775
1570  002336  177776                        .WORD   177776
1571
```

```
1572  002340  177777                PAT13:  .WORD   177777          ;PATTERN 13
1573  002342  177777                        .WORD   177777
1574  002344  177777                        .WORD   177777
1575  002346  177777                        .WORD   177777
1576  002350  177777                        .WORD   177777
1577  002352  177777                        .WORD   177777
1578  002354  177777                        .WORD   177777
1579  002356  177777                        .WORD   177777
1580  002360  177777                        .WORD   177777
1581  002362  177777                        .WORD   177777
1582  002364  177777                        .WORD   177777
1583  002366  177777                        .WORD   177777
1584  002370  177777                        .WORD   177777
1585  002372  177777                        .WORD   177777
1586  002374  177777                        .WORD   177777
1587  002376  177777                        .WORD   177777
1588
1589  002400  000000                PAT14:  .WORD   000000          ;PATTERN 14
1590  002402  000000                        .WORD   000000
1591  002404  177777                        .WORD   177777
1592  002406  177777                        .WORD   177777
1593  002410  177777                        .WORD   177777
1594  002412  177777                        .WORD   177777
1595  002414  177777                        .WORD   177777
1596  002416  177777                        .WORD   177777
1597  002420  177777                        .WORD   177777
1598  002422  177777                        .WORD   177777
1599  002424  177777                        .WORD   177777
1600  002426  177777                        .WORD   177777
1601  002430  177777                        .WORD   177777
1602  002432  177777                        .WORD   177777
1603  002434  177777                        .WORD   177777
1604  002436  177777                        .WORD   177777
1605
1606  002440  177777                PAT15:  .WORD   177777          ;PATTERN 15
1607  002442  177777                        .WORD   177777
1608  002444  000000                        .WORD   000000
1609  002446  000000                        .WORD   000000
1610  002450  000000                        .WORD   000000
1611  002452  000000                        .WORD   000000
1612  002454  000000                        .WORD   000000
1613  002456  000000                        .WORD   000000
1614  002460  000000                        .WORD   000000
1615  002462  000000                        .WORD   000000
1616  002464  000000                        .WORD   000000
1617  002466  000000                        .WORD   000000
1618  002470  000000                        .WORD   000000
1619  002472  000000                        .WORD   000000
1620  002474  000000                        .WORD   000000
1621  002476  000000                        .WORD   000000
1622
1623                          ;DRIVE OPERATION CONTROL BLOCK
1624
1625  002500      000         DPB:    .BYTE   0               ;DRIVE NUMBER
1626  002501      000                 .BYTE   0               ;'MEX' BITS
1627  002502      000                 .BYTE   0               ;OPERATION CODE
```

```
1628  002503  000                     .BYTE   0               ;MODE AND HDR BITS
1629  002504  000000          $WC:    .WORD   0               ;WORD COUNT
1630  002506  000000          $BUF:   .WORD   0               ;BUFFER ADDRESS
1631  002510  000000          $CYL:   .WORD   0               ;CYLINDER ADDRESS
1632  002512  000             $SEC:   .BYTE   0               ;SECTOR ADDRESS
1633  002513  000             $TRK:   .BYTE   0               ;TRACK ADDRESS
1634
```

```
                                .SBTTL  ERROR POINTER TABLE

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

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

002514                          $ERRTB:
                                ;ERROR 1

002514  032117                          EM1             ;RP11 DIDN'T RESPOND TO ADDRESSING
002516  040241                          DH1
002520  042762                          DT1
002522  043442                          DF1

                                ;ERROR 2

002524  032161                          EM2             ;CAN'T START READ COMMAND
002526  040247                          DH2
002530  042764                          DT2
002532  043446                          DF2

                                ;ERROR 3

002534  032212                          EM3             ;'CLEAR' COMMAND DIDN'T TERMINATE DATA TRANSFER OPERATION
002536  040247                          DH2
002540  042764                          DT2
002542  043446                          DF2

                                ;ERROR 4

002544  032303                          EM4             ;ERROR WRITING TEST SECTOR(S)
002546  040247                          DH2
002550  042764                          DT2
002552  043446                          DF2

                                ;ERROR 5

002554  032340                          EM5             ;WRITE CHECK ERROR CHECKING TEST SECTOR(S)
002556  040411                          DH5
002560  043016                          DT5
002562  043456                          DF5

                                ;ERROR 6

002564  032412                          EM6             ;EXPECTED 'WCE' DIDN'T OCCUR
002566  040411                          DH5
002570  043016                          DT5
002572  043456                          DF5
```

```
                                        ;ERROR 7
002574  032446                                EM7             ;ERROR WRITING LESS THAN A FULL SECTOR
002576  040247                                DH2
002600  042764                                DT2
002602  043446                                DF2

                                        ;ERROR 10
002604  000000                                0               ;UNUSED
002606  000000                                0
002610  000000                                0
002612  000000                                0

                                        ;ERROR 11
002614  032514                                EM11            ;ONE OF 2 WORDS WRITTEN IN THE PARTIAL WRITE DIDN'T COMPARE
002616  040563                                DH11
002620  043052                                DT11
002622  043466                                DF11

                                        ;ERROR 12
002624  032607                                EM12            ;NON-ZERO DATA IN PART OF SECTOR WHICH SHOULD HAVE
                                                                ;BEEN ZERO FILLED DURING PARTIAL WRITE
002626  040644                                DH12
002630  043066                                DT12
002632  043466                                DF11

                                        ;ERROR 13
002634  032740                                EM13            ;'EOP' DIDN'T SET DURING A 2 SECTOR WRITE BEGINNING
                                                                ;AT THE LAST SECTOR OF THE PACK
002636  040247                                DH2
002640  042764                                DT2
002642  043446                                DF2

                                        ;ERROR 14
002644  033063                                EM14            ;'ERR' DIDN'T SET WITH 'EOP'
002646  040721                                DH14
002650  043102                                DT14
002652  043466                                DF11

                                        ;ERROR 15
002654  033117                                EM15            ;'EOP' DIDN'T CLEAR RPCA
002656  040247                                DH2
002660  042764                                DT2
002662  043446                                DF2

                                        ;ERROR 16
002664  033147                                EM16            ;SUCA NOT CORRECT AFTER 'EOP'
002666  040776                                DH16
```

```
1747  002670  043116                        DT16
1748  002672  043542                        DF53
1749
1750                                  ;ERROR 17
1751
1752  002674  033204                        EM17        ;ERROR READING TEST SECTOR
1753  002676  040247                        DH2
1754  002700  042764                        DT2
1755  002702  043446                        DF2
1756
1757                                  ;ERROR 20
1758
1759  002704  033236                        EM20        ;CONTENTS OF FIRST SECTOR OF PACK CHANGED AFTER
1760                                                    ;FORCING 'EOP' DURING WRITE
1761  002706  040563                        DH11
1762  002710  043052                        DT11
1763  002712  043466                        DF11
1764
1765                                  ;ERROR 21
1766
1767  002714  033351                        EM21        ;'PROG' ERROR DIDN'T SET WHEN READ COMMAND ISSUED
1768                                                    ;WHILE CONTROLLER BUSY
1769  002716  040721                        DH14
1770  002720  043102                        DT14
1771  002722  043466                        DF11
1772
1773                                  ;ERROR 22
1774
1775  002724  033461                        EM22        ;'ERR' DIDN'T SET WITH 'PROG'
1776  002726  040721                        DH14
1777  002730  043102                        DT14
1778  002732  043466                        DF11
1779
1780                                  ;ERROR 23
1781
1782  002734  033516                        EM23        ;ERROR ATTEMPTING TO MISFORMAT SECTOR 0
1783  002736  040247                        DH2
1784  002740  042764                        DT2
1785  002742  043446                        DF2
1786
1787                                  ;ERROR 24
1788
1789  002744  033565                        EM24        ;ERROR VERIFYING THE MISFORMATTED TEST HEADER
1790  002746  040721                        DH14
1791  002750  043102                        DT14
1792  002752  043466                        DF11
1793
1794                                  ;ERROR 25
1795
1796  002754  033642                        EM25        ;MISFORMATTED TEST HEADER IS NOT CORRECT
1797  002756  041113                        DH25
1798  002760  043130                        DT25
1799  002762  043472                        DF25
1800
1801                                  ;ERROR 26
1802
```

```
1803  002764  033712                          EM26        ;'HNF' DIDN'T SET WHEN SEARCHING FOR MISFORMATTED SECTOR
1804  002766  040247                          DH2
1805  002770  042764                          DT2
1806  002772  043446                          DF2
1807
1808                                  ;ERROR 27
1809
1810  002774  034002                          EM27        ;'DSK' NOT SET WITH 'HNF'
1811  002776  040721                          DH14
1812  003000  043102                          DT14
1813  003002  043466                          DF11
1814
1815                                  ;ERROR 30
1816
1817  003004  034033                          EM30        ;'ME' DIDN'T SET WITH 'HNF'
1818  003006  040721                          DH14
1819  003010  043102                          DT14
1820  003012  043466                          DF11
1821
1822                                  ;ERROR 31
1823
1824  003014  034066                          EM31        ;'ERR' NOT SET WITH 'HNF'
1825  003016  040721                          DH14
1826  003020  043102                          DT14
1827  003022  043466                          DF11
1828
1829                                  ;ERROR 32
1830
1831  003024  034117                          EM32        ;ERROR WHILE RESTORING MISFORMATTED TEST HEADER
1832  003026  040247                          DH2
1833  003030  042764                          DT2
1834  003032  043446                          DF2
1835
1836                                  ;ERROR 33
1837
1838  003034  034176                          EM33        ;CONTROLLER DIDN'T BECOME BUSY WHEN READ COMMAND
1839                                                      ;ISSUED TO SEEKING DRIVE
1840  003036  040247                          DH2
1841  003040  042764                          DT2
1842  003042  043446                          DF2
1843
1844                                  ;ERROR 34
1845
1846  003044  034307                          EM34        ;ERROR DURING READ COMMAND WHICH WAS ISSUED TO SEEKING DRIVE
1847  003046  040247                          DH2
1848  003050  042764                          DT2
1849  003052  043446                          DF2
1850
1851                                  ;ERROR 35
1852
1853  003054  034403                          EM35        ;DATA INCORRECT FROM HEADER READ COMMAND ISSUED
1854                                                      ;TO SEEKING DRIVE
1855  003056  041113                          DH25
1856  003060  043130                          DT25
1857  003062  043472                          DF25
1858
```

```
                                          ;ERROR 36

003064  034505                                    EM36        ;'NXME' DIDN'T SET WHEN LOCATION 760000 ADDRESSED
003066  040247                                    DH2
003070  042764                                    DT2
003072  043446                                    DF2

                                          ;ERROR 37

003074  034566                                    EM37        ;'HE' DIDN'T SET WITH 'NXME'
003076  040721                                    DH14
003100  043102                                    DT14
003102  043466                                    DF11

                                          ;ERROR 40

003104  034622                                    EM40        ;'ERR' DIDN7T SET WITH 'NXME'
003106  040721                                    DH14
003110  043102                                    DT14
003112  043466                                    DF11

                                          ;ERROR 41

003114  034657                                    EM41        ;ERROR FORMATTING TRACK 0
003116  040247                                    DH2
003120  042764                                    DT2
003122  043446                                    DF2

                                          ;ERROR 42

003124  034710                                    EM42        ;ERROR READING THE HEADER FROM THE TEST SECTOR
003126  040247                                    DH2
003130  042764                                    DT2
003132  043446                                    DF2

                                          ;ERROR 43

003134  034766                                    EM43        ;SECTOR FIELD FROM HEADER IS NOT CORRECT
003136  041202                                    DH43
003140  043150                                    DT43
003142  043502                                    DF43

                                          ;ERROR 44

003144  035036                                    EM44        ;ERROR WRITING SECTOR ADDRESS IN DATA FIELD
003146  040247                                    DH2
003150  042764                                    DT2
003152  043446                                    DF2

                                          ;ERROR 45

003154  035111                                    EM45        ;'SOT' OR SECTOR ADDRESS REGISTER IS NOT CORRECT
003156  041426                                    DH45
003160  043206                                    DT45
003162  043516                                    DF45
```

```
1915                                    ;ERROR 46
1916
1917  003164  035171                        EM46        ;DATA NOT CORRECT FOR SECTOR READ
1918  003166  041620                        DH46
1919  003170  043232                        DT46
1920  003172  043466                        DF11
1921
1922                                    ;ERROR 47
1923
1924  003174  035242                        EM47        ;SECTOR CONTENTS WRONG, DATA SHOULD BE THE
1925                                                      ;TRACK NUMBER OF THE CURRENTLY SELECTED HEAD
1926  003176  041755                        DH47
1927  003200  043246                        DT47
1928  003202  043526                        DF47
1929
1930                                    ;ERROR 50
1931
1932  003204  035371                        EM50        ;ERROR AFTER 2 WORD READ STARTING AT LOC 177776
1933  003206  040247                        DH2
1934  003210  042764                        DT2
1935  003212  043446                        DF2
1936
1937                                    ;ERROR 51
1938
1939  003214  035450                        EM51        ;'MEXD' DIDN'T SET AFTER 2 WORD READ STARTING
1940                                                      ;AT LOC 177776
1941  003216  042071                        DH51
1942  003220  043272                        DT51
1943  003222  043536                        DF51
1944
1945                                    ;ERROR 52
1946
1947  003224  035544                        EM52        ;'MEX1' SET AFTER 2 WORD READ STARTING AT LOC 177776
1948  003226  042071                        DH51
1949  003230  043272                        DT51
1950  003232  043536                        DF51
1951
1952                                    ;ERROR 53
1953
1954  003234  035630                        EM53        ;CONTENTS OF LOC 200000 WRONG AFTER 2 WORD READ
1955                                                      ;STARTING AT LOC 177776
1956  003236  041113                        DH25
1957  003240  043130                        DT25
1958  003242  043542                        DF53
1959
1960                                    ;ERROR 54
1961
1962  003244  035737                        EM54        ;ERROR AFTER 2 WORD READ STARTING AT LOC 377776
1963  003246  040247                        DH2
1964  003250  042764                        DT2
1965  003252  043446                        DF2
1966
1967                                    ;ERROR 55
1968
1969
1970  003254  036016                        EM55        ;'MEXD' DIDN'T CLEAR AFTER 2 WORD STARTING AT LOC 377776
```

```
1971  003256  042071                          DH51
1972  003260  043272                          DT51
1973  003262  043536                          DF51
1974
1975                                  ;ERROR 56
1976
1977  003264  036106                          EM56        ;'MEX1' DIDN'T SET AFTER 2 WORD READ STARTING AT LOC 377776
1978  003266  042071                          DH51
1979  003270  043272                          DT51
1980  003272  043536                          DF51
1981
1982                                  ;ERROR 57
1983
1984  003274  036201                          EM57        ;CONTENTS OF LOC 400000 WRONG AFTER 2 WORD READ
1985                                                      ;STARTING AT LOCATION 377776
1986  003276  041113                          DH25
1987  003300  043130                          DT25
1988  003302  043542                          DF53
1989
1990                                  ;ERROR 60
1991
1992  003304  036315                          EM60        ;ERROR RETURNING SECTOR TO PDP-11 MODE USING A
1993                                                      ;2 WORD WRITE
1994  003306  040247                          DH2
1995  003310  042764                          DT2
1996  003312  043446                          DF2
1997
1998                                  ;ERROR 61
1999
2000  003314  036411                          EM61        ;ERROR WRITING TEST SECTOR(S)
2001  003316  040411                          DH5
2002  003320  043016                          DT5
2003  003322  043456                          DF5
2004
2005                                  ;ERROR 62
2006
2007  003324  036446                          EM62        ;ERROR READING TEST SECTOR(S)
2008  003326  040411                          DH5
2009  003330  043016                          DT5
2010  003332  043456                          DF5
2011
2012                                  ;ERROR 63
2013
2014  003334  036503                          EM63        ;DRIVE DIDN'T RETURN TO CYL 0 FROM THE MAXIMUM
2015                                                      ;CYL ON CONTROLLER POWER FAIL
2016  003336  042166                          DH63
2017  003340  043312                          DT63
2018  003342  043546                          DF63
2019
2020                                  ;ERROR 64
2021
2022  003344  036617                          EM64        ;CONTENTS OF MEMORY CHANGED DURING POWER
2023                                                      ;FAIL WHILE READING THE DISK
2024  003346  040644                          DH12
2025  003350  043066                          DT12
2026  003352  043466                          DF11
```

```
                                ;ERROR 65
003354  036724                          EM65            ;'SUOL' SET WITH DRIVE DISABLED
003356  040721                          DH14
003360  043102                          DT14
003362  043466                          DF11

                                ;ERROR 66
003364  036763                          EM66            ;'SUOL' NOT SET WITH DRIVE ENABLED
003366  040721                          DH14
003370  043102                          DT14
003372  043466                          DF11

                                ;ERROR 67
003374  037025                          EM67            ;'SUWP' NOT SET WITH DRIVE SET TO 'READ ONLY'
003376  040721                          DH14
003400  043102                          DT14
003402  043466                          DF11

                                ;ERROR 70
003404  037102                          EM70            ;'SUWP' SET WITH DRIVE SET TO 'READ/WRITE'
003406  040721                          DH14
003410  043102                          DT14
003412  043466                          DF11

                                ;ERROR 71
003414  037154                          EM71            ;'SUWP' SET WITH RP11 'WRITE LOCKOUT' SET, CYLINDER
                                                        ;LOA'S = 0, AND RPCA = 0
003416  042253                          DH71
003420  043330                          DT71
003422  043546                          DF63

                                ;ERROR 72
003424  037270                          EM72            ;'SUWP' SET WITH RPCA = 2 & CYL LOA'S = 0
003426  042253                          DH71
003430  043330                          DT71
003432  043546                          DF63

                                ;ERROR 73
003434  037341                          EM73            ;'SUWP' NOT SET; RPCA = VALUE IN CYL LOA'S
003436  042253                          DH71
003440  043330                          DT71
003442  043546                          DF63

                                ;ERROR 74
003444  037413                          EM74            ;'SUWP' SET WITH RPCA ONE GREATER THAN CYL LOA VALUE
003446  042253                          DH71
003450  043330                          DT71
```

```
2083  003452  043546                        DF63
2084
2085                                ;ERROR 75
2086
2087  003454  037477                        EM75        ;'SUMP' SET WITH DRIVE LOA SWITCHES EQUAL SELECTED DRIVE
2088  003456  040721                        DH14
2089  003460  043102                        DT14
2090  003462  043466                        DF11
2091
2092                                ;ERROR 76
2093
2094  003464  037572                        EM76        ;DRIVE HAS GONE OFFLINE
2095  003466  040247                        DH2
2096  003470  042764                        DT2
2097  003472  043446                        DF2
2098
2099                                ;ERROR 77
2100
2101  003474  037621                        EM77        ;DRIVE IS UNSAFE
2102  003476  040247                        DH2
2103  003500  042764                        DT2
2104  003502  043446                        DF2
2105
2106                                ;ERROR 100
2107
2108  003504  037641                        EM100       ;DRIVE TIMED OUT
2109  003506  040247                        DH2
2110  003510  042764                        DT2
2111  003512  043446                        DF2
2112
2113                                ;ERROR 101
2114
2115  003514  037661                        EM101       ;CONTROLLER TIMED OUT
2116  003516  040247                        DH2
2117  003520  042764                        DT2
2118  003522  043446                        DF2
2119
2120                                ;ERROR 102
2121
2122  003524  037706                        EM102       ;DATA COMPARE ERROR
2123  003526  042340                        DH102
2124  003530  043346                        DT102
2125  003532  043552                        DF102
2126
2127                                ;ERROR 103
2128
2129  003534  037731                        EM103       ;DATA COMPARE ERROR (MEMORY MANAGEMENT ENABLED)
2130  003536  042340                        DH102
2131  003540  043374                        DT103
2132  003542  043562                        DF103
2133
2134                                ;ERROR 104
2135
2136  003544  000000                        0           ;DATA COMPARE ERROR DETAIL LINE
2137  003546  000000                        0
2138  003550  043430                        DT104
```

```
2139  003552  043576                          DF104
2140
2141                                  ;ERROR 105
2142
2143  003554  000000                          0                ;DATA COMPARE ERROR SUMMARY LINE
2144  003556  042643                          DH105
2145  003560  043436                          DT105
2146  003562  043442                          DF1
2147
2148                                  ;ERROR 106
2149
2150  003564  040010                          EM106            ;'ERR' DIDN'T SET WITH 'WCE'
2151  003566  040721                          DH14
2152  003570  043102                          DT14
2153  003572  043466                          DF11
2154
2155                                  ;ERROR 107
2156
```

```
2157  003574  040044                    EM107          ;'HE' DIDN'T SET WITH 'PROG'
2158  003576  040721                    DH14
2159  003600  043102                    DT14
2160  003602  043466                    DF11
2161
2162                                ;ERROR 110
2163
2164  003604  000000                    0              ;UNSUCESSFUL AFTER 3 RETRIES
2165  003606  042671                    DH110
2166  003610  000000                    0
2167  003612  000000                    0
2168
2169                                ;ERROR 111
2170
2171  003614  000000                    0              ;SUCESSFUL AFTER 'N' RETRIES
2172  003616  042724                    DH111
2173  003620  043440                    DT111
2174  003622  043442                    DF1
2175
2176                                ;ERROR 112
2177
2178  003624  040100                    EM112          ;DRIVE UNSAFE AFTER POWER FAIL
2179  003626  040247                    DH2
2180  003630  042764                    DT2
2181  003632  043446                    DF2
2182
2183                                ;ERROR 113
2184
2185  003634  040136                    EM113          ;'SUMP' NOT SET WITH DRIVE LOA SWITCHES GREATER THAN SELECTED DR
2186  003636  040721                    DH14
2187  003640  043102                    DT14
2188  003642  043466                    DF11
2189
```

```
2190
2191
2192                            ;;**********************************************************************
2193
2194                            .SBTTL  START OF PROGRAM
2195
2196                            ;;**********************************************************************
2197
2198
2199  003644  012737  177777  001216  START3: MOV     #-1,BUSADR      ;SET BUSADR FLAG
2200  003652  000402                    BR      STRT1A
2201  003654  005037  001216            START1: CLR     @#BUSADR        ;CLR BUSADR FLAG
2202  003660  005037  001206            STRT1A: CLR     @#CNTRLC        ;NO CONVERSATION MODE
2203  003664  000411                    BR      START           ;
2204  003666  012737  177777  001216    START4: MOV     #-1,BUSADR      ;SET BUSADR FLAG
2205  003674  000402                    BR      STRT2A
2206  003676  005037  001216            START2: CLR     @#BUSADR        ;CLR BUSADR FLAG
2207  003702  012737  177777  001206    STRT2A: MOV     #-1,CNTRLC      ;SET CONVERSATION MODE FLAG
2208  003710  000005                    START:  RESET                   ;CLEAR THE BUS
2209                                    .SBTTL  INITIALIZE THE COMMON TAGS
2210                            ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
2211  003712  012706  001100            MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
2212  003716  005026                    CLR     (R6)+           ;;CLEAR MEMORY LOCATION
2213  003720  022706  001140            CMP     #SWR,R6 ;;DONE?
2214  003724  001374                    BNE     .-6             ;;LOOP BACK IF NO
2215  003726  012706  001100            MOV     #STACK,SP       ;;SETUP THE STACK POINTER
2216                            ;;INITIALIZE A FEW VECTORS
2217  003732  012737  021624  000020    MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
2218  003740  012737  000340  000022    MOV     #340,@#IOTVEC+2 ;;LEVEL 7
2219  003746  012737  017060  000030    MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
2220  003754  012737  000340  000032    MOV     #340,@#EMTVEC+2 ;;LEVEL 7
2221  003762  012737  022452  000034    MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
2222  003770  012737  000340  000036    MOV     #340,@#TRAPVEC+2;LEVEL 7
2223  003776  013737  016774  016766    MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
2224  004004  005037  001162            CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
2225  004010  005037  001164            CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
2226  004014  112737  000001  001115    MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
2227  004022  012737  004022  001106    MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
2228  004030  012737  004030  001110    MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
2229                            ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
2230                            ;;EQUAL TO A "-1"  SETUP FOR A SOFTWARE SWITCH REGISTER.
2231  004036  013746  000004            MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
2232  004042  012737  004076  000004    MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
2233  004050  012737  177570  001140    MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
2234  004056  012737  177570  001142    MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
2235  004064  022777  177777  175046    CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
2236  004072  001012                    BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
2237                                                            ;;AND  THE HARDWARE SWR IS NOT = -1
2238  004074  000403                    BR      65$             ;;BRANCH IF NO TIMEOUT
2239  004076  012716  004104    64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
2240  004102  000002                    RTI
2241  004104  012737  000176  001140    65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
2242  004112  012737  000174  001142    MOV     #DISPREG,DISPLAY
2243  004120  012637  000004            66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
2244
2245  004124  005227  177777            INC     #-1             ;FIRST START ?
```

```
2246  004130  001002                          BNE     START5          ;BR IF NOT
2247  004132  104401  043602                  TYPE    ,TITLE          ;TYPE THE PROGRAM'S MAINDEC NUMBER AND NAME
2248  004136  004737  020374          START5: JSR     PC,$TKINT       ;SETUP THE TTY KEYBOARD
2249                                  .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
2250  004142  005737  000042                  TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
2251  004146  001006                          BNE     64$             ;;BRANCH IF YES
2252  004150  023727  001140  000176          CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
2253  004156  001005                          BNE     65$             ;;BRANCH IF NO
2254  004160  104406                          GTSWR                   ;;GET SOFT-SWR SETTINGS
2255  004162  000403                          BR      65$
2256  004164  112737  000001  001134  64$:   MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
2257  004172                          65$:
2258  004172  004737  023404                  JSR     PC,RESLDR       ;RESTORE THE LOADER
2259  004176  004737  025624                  JSR     PC,SIZMEM       ;SET MEMORY SIZE VARIABLES
2260  004202  005037  001112                  CLR     $ERTTL          ;CLEAR TOTAL ERROR COUNT
2261  004206  005037  001260                  CLR     OPRSEL          ;CLEAR OPERATOR SELECTED ADDRESS FLAG
2262  004212  005037  001266                  CLR     SCYL            ;OPERATOR SPECIFIED CYLINDER
2263  004216  005037  001270                  CLR     STRK            ;OPERATOR SPECIFIED TRACK
2264  004222  005037  001272                  CLR     SSEC            ;OPERATOR SPECIFIED SECTOR
2265  004226  013737  001264  001256          MOV     MAXWC,WC14      ;ASSUME THAT MAXIMUM WORD COUNT LESS THAN 8K
2266  004234  023737  001264  001262          CMP     MAXWC,LMTBK     ;MAXIMUM WORD COUNT GREATER THAN LIMIT
2267  004242  101403                          BLOS    1$              ;BR IF NOT
2268  004244  013737  001262  001256          MOV     LMTBK,WC14      ;USE AN 8K MAXIMUM
2269  004252  012737  177777  001300  1$:     MOV     #177777,PATTRN  ;ENABLE ALL PATTERNS
2270  004260  012737  017777  001276          MOV     #17777,TSTNMS   ;SELECT TESTS 0 - 14
2271  004266  005037  177776                  CLR     PSW             ;ENSURE THE PRIORITY = 0
2272  004272  005037  001310                  CLR     DRVBAD          ;CLEAR OFFLINE/UNSAFE DRIVE BITS
2273  004276  005037  001162                  CLR     $TIMES          ;INITIALIZE NUMBER OF ITERATIONS
2274  004302  005037  001164                  CLR     $ESCAPE         ;CLEAR THE ESCAPE ON ERROR ADDRESS
2275  004306  012737  016606  001200          MOV     #SEOP,BYPASS    ;INITIAL BYPASS ADDRESS
2276  004314  012737  000001  001104          MOV     #1,$ICNT        ;PRESET ITERATION COUNT TO 1
2277  004322  112737  000001  001115          MOVB    #1,$ERMAX       ;ALLOW ONE ERROR PER TEST
2278  004330  012737  004330  001106          MOV     #.,$LPADR       ;INITIALIZE THE LOOP ADDRESS FOR SCOPE
2279  004336  012737  004336  001110          MOV     #.,$LPERR       ;SETUP THE ERROR LOOP ADDRESS
2280  004344  005037  001222                  CLR     MMACTV          ;CLEAR MEMORY MANAGEMENT FLAG
2281  004350  005037  177776          START6: CLR     PS              ;CLEAR THE PROCESSOR PRIORITY
2282  004354  004737  026074                  JSR     PC,GETADR       ;CHECK THE RP11 ADDRESS
2283  004360  004737  022536                  JSR     PC,RPINIT       ;FIND OUT WHICH DRIVES ARE ON SYSTEM
2284  004364  012737  000312  001210          MOV     #202.,MAXCYL    ;ASSUME RP02'S
2285  004372  005737  001304                  TST     DRVTYP          ;WHICH DRIVES ?
2286  004376  001403                          BEQ     2$              ;BR IF THEY REALLY WERE RP02'S
2287  004400  012737  000625  001210          MOV     #405.,MAXCYL    ;SET MAX CYLINDER FOR RP03'S
2288  004406  005227  177777          2$:     INC     #-1             ;FIRST START ?
2289  004412  001404                          BEQ     9$              ;BR IF IT IS
2290  004414  032777  000200  174516          BIT     #SW07,@SWR      ;BYPASS DRIVE STATUS TYPEOUT ?
2291  004422  001044                          BNE     START7          ;BR IF YES
2292  004424  104401  027162          9$:     TYPE    ,DRSTAT         ;'DRIVE STATUS'
2293  004430  005001                          CLR     R1              ;CLEAR TABLE POINTER
2294  004432  012702  000010                  MOV     #8.,R2          ;COUNTER
2295  004436                          3$:
2296  004436  010146                          MOV     R1,-(SP)        ;;SAVE R1 FOR TYPEOUT
2297                                                                  ;;TYPE DRIVE NUMBER
2298  004440  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
2299  004442     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
2300  004443     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
2301  004444  105761  001370                  TSTB    DRVSTA(R1)      ;CHECK DRIVE'S STATUS
```

```
2302  004450  001420                        BEQ     7$              ;DRIVE IS OFFLINE OR NON-EXISTENT
2303  004452  100403                        BMI     4$              ;DRIVE IS UNSAFE
2304  004454  104401  027205                TYPE    ,ONLINE         ;'ONLINE'
2305  004460  000402                        BR      5$              ;
2306  004462  104401  027222        4$:     TYPE    ,UNSAFE         ;'UNSAFE'
2307  004466  136137  001360  001304  5$:   BITB    @TABIT(R1),DRVTYP  ;SEE WHICH DRIVE TYPE
2308  004474  001003                        BNE     6$              ;BR IF RP03
2309  004476  104401  027253                TYPE    ,RP02           ;'RP02'
2310  004502  000405                        BR      8$              ;
2311  004504  104401  027262        6$:     TYPE    ,RP03           ;'RP03'
2312  004510  000402                        BR      8$              ;
2313  004512  104401  027236        7$:     TYPE    ,OFFLIN         ;'OFFLINE'
2314  004516  104401  001173        8$:     TYPE    ,$CRLF          ;CR-LF
2315  004522  005201                        INC     R1              ;INCREMENT TABLE POINTER
2316  004524  005302                        DEC     R2              ;DECREMENT THE COUNTER
2317  004526  001343                        BNE     3$              ;CONTINUE
2318  004530  104401  001173                TYPE    ,$CRLF          ;CR-LF
2319  004534  005737  001206        START7: TST     @#CNTRLC        ;CONVERSATION MODE ?
2320  004540  001403                        BEQ     1$              ;NO--BRANCH
2321  004542  004737  026240                JSR     PC,ENTPRM       ;YES--GET PARAMETERS
2322  004546  000416                        BR      5$              ;GO TYPE DRIVES TO BE TESTED
2323  004550  005037  001312        1$:     CLR     DRVSEL          ;NO DRIVES SELECTED
2324  004554  005000                        CLR     R0              ;DETERMINE THE DRIVES THAT
2325  004556  012701  000001                MOV     #1,R1           ;ARE AVAILABLE FOR TESTING
2326  004562  105760  001370        2$:     TSTB    DRVSTA(R0)      ;DRIVE ONLINE ?
2327  004566  003403                        BLE     3$              ;BR IF NOT
2328  004570  156037  001360  001312        BISB    @TABIT(R0),DRVSEL  ;SET SELECTION BIT FOR DRIVE
2329  004576  005200                3$:     INC     R0              ;INCREMENT DRIVE ADDRESS
2330  004600  106301                        ASLB    R1              ;COUNT
2331  004602  001367                        BNE     2$              ;BR IF NOT ALL DRIVES CHECKED
2332  004604  104401  027271        5$:     TYPE    ,DRVTST         ;'DRIVES TO BE TESTED'
2333  004610  005037  016774                CLR     $ENDCT          ;DETERMINE PASSES TO MAKE AND
2334  004614  005000                        CLR     R0              ;THE DRIVES TO BE TESTED
2335  004616  013701  001312                MOV     DRVSEL,R1       ;ANY DRIVES SELECTED?
2336  004622  001004                        BNE     6$              ;YES--BRANCH
2337  004624  104401  027316                TYPE    ,NONE           ;'NONE'
2338  004630  000137  016606                JMP     @#SEOP          ;GO TO END OF PROGRAM
2339  004634  005737  001206        6$:     TST     @#CNTRLC        ;CONVERSATION MODE START ?
2340  004640  001011                        BNE     7$              ;BR IF NOT
2341  004642  005737  000042                TST     @#42            ;UNDER MONITOR CONTROL ?
2342  004646  001406                        BEQ     7$              ;BR IF NOT
2343  004650  122737  000007  000041        CMPB    #7,@#41         ;LOADED BY RP02/RP03 ?
2344  004656  001002                        BNE     7$              ;BR IF NOT
2345  004660  006201                        ASR     R1              ;EXCLUDE DRIVE 0 FROM TESTING
2346  004662  000413                        BR      8$              ;CHECK ON THE OTHER DRIVES
2347  004664  006201                7$:     ASR     R1              ;REPORT THE DRIVES TO BE TESTED
2348  004666  103011                        BCC     8$              ;BR IF DRIVE IS NOT SELECTED
2349  004670  005237  016774                INC     $ENDCT          ;GIVE THIS DRIVE A PASS
2350  004674  010046                        MOV     R0,-(SP)        ;SAVE R0 FOR TYPEOUT
2351                                                                ;TYPE THE DRIVE NUMBER
2352  004676  104403                TYPOS                           ;GO TYPE--OCTAL ASCII
2353  004700     001                .BYTE   1                       ;TYPE 1 DIGIT(S)
2354  004701     000                .BYTE   0                       ;SUPPRESS LEADING ZEROS
2355  004702  005701                TST     R1              ;MORE DRIVES?
2356  004704  001404                BEQ     9$              ;NO--BRANCH
2357  004706  104401  027323                TYPE    ,COMMA          ;','
```

```
2358  004712  005200                    8$:     INC     R0          ;INCREMENT DRIVE NUMBER
2359  004714  000763                            BR      7$          ;CONTINUE
2360  004716  013737  016774  016766    9$:     MOV     $ENDCT,$EOPCT ;SETUP TEST COUNT
2361  004724  005037  001206            RSTRT1: CLR     CNTRLC      ;CLEAR CONVERSATION MODE FLAG
2362  004730  005037  001176                    CLR     CHKDRV      ;INITIALIZE THE CHECK DRIVE KEY
2363  004734  012737  000001  001306            MOV     #1,DRVMSK   ;START TO CHECK DESIRED DRIVES
2364  004742  033737  001306  001312    RSTRT2: BIT     DRVMSK,DRVSEL ;IS THIS DRIVE SELECTED?
2365  004750  001011                    RSTRT3: BNE     DRVOK       ;YES--GO CHECK IF DRIVE IS READY FOR TESTING
2366  004752  005237  001176            RESTART:INC     CHKDRV      ;MOVE  TO NEXT DRIVE NUMBER
2367  004756  106337  001306                    ASLB    DRVMSK      ;POSITION THE MASK
2368  004762  103367                            BCC     RSTRT2      ;BR IF MORE DRIVES
2369  004764  043737  001310  001312            BIC     DRVBAD,DRVSEL ;CLEAR SELECTION BITS FOR ANY OFFLINE/UNSAFE
2370                                                                ;DRIVES
2371  004772  000756                            BR      RSTRT1      ;CONTINUE WITH CYCLE
2372  004774  113737  001176  002500  DRVOK:  MOVB    CHKDRV,DPB  ;PICKUP THE DRIVE NUMBER
2373  005002  104401  027325                    TYPE    ,TSTING     ;'TESTING WITH DRIVE '
2374  005006  013746  001176                    MOV     CHKDRV,-(SP) ;;SAVE CHKDRV FOR TYPEOUT
2375                                                                ;;TYPE THE DRIVE NUMBER
2376  005012  104403                            TYPOS               ;;GO TYPE--OCTAL ASCII
2377  005014     001                            .BYTE   1           ;;TYPE 1 DIGIT(S)
2378  005015     000                            .BYTE   0           ;;SUPPRESS LEADING ZEROS
2379  005016  104401  001173                    TYPE    ,$CRLF      ;CR-LF
2380  005022  013704  001324                    MOV     RPADR,R4    ;RP11 ADDRESS
2381  005026  004737  025606                    JSR     PC,CLRP     ;CLEAR THE RP11
2382  005032  112737  000015  002502            MOVB    #HOMSEK,DPB+2 ;HOME SEEK COMMAND
2383  005040  004737  022740                    JSR     PC,RP11     ;START THE COMMAND
2384  005044  004737  023500                    JSR     PC,DRVRDY   ;WAIT FOR THE DRIVE TO FINISH
2385  005050  000137  005054                    JMP     TST0        ;START THE PROGRAM
2386
2387                                    .SBTTL  #### TESTS ####
2388
2389
2390                                    ;;*********************************************************************
2391                                    ;*TEST 0        TEST 'CLEAR' TERMINATION
2392
2393                                    ;*VERIFY THAT 'CLEAR' TERMINATES AN OPERATION AND CAUSES CONTROLLER
2394                                    ;*READY TO SET.
2395
2396                                    ;;*********************************************************************
2397  005054                            TST0:
2398  005054  033737  001400  001276            BIT     BITS+<$TN*2>,TSTNMS  ;IS THIS TEST SELECTED
2399  005062  001002                            BNE     .+6                 ;BR IF YES
2400  005064  000137  005300                    JMP     TST1                ;GO TO THE NEXT TEST
2401  005070  012737  005150  001110            MOV     #TEST0,$LPERR       ;SETUP THE ERROR LOOP ADDRESS
2402  005076  012737  005054  001106            MOV     #TST0,$LPADR        ;SETUP THE SCOPE LOOP ADDRESS
2403  005104  012737  000000  001102            MOV     #0,$TSTNM           ;SETUP TEST NUMBER AND
2404                                                                        ;CLEAR THE ERROR FLAG ($ERFLG)
2405  005112  013777  001102  174022            MOV     $TSTNM,@DISPLAY     ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
2406  005120  012737  005276  001200            MOV     #EXIT0,BYPASS       ;SETUP BYPASS ADDRESS
2407  005126  013704  001324                    MOV     RPADR,R4            ;RP11E BUS ADDRESS
2408  005132  105037  002501                    CLRB    DPB+1               ;CLEAR ANY EXTENDED MEMORY BITS
2409  005136  105037  002503                    CLRB    DPB+3               ;CLEAR 'MODE' & 'HDR' BITS
2410  005142  012737  000144  001162            MOV     #100.,$TIMES        ;;DO 100. ITERATIONS
2411  005150  012706  001100            TEST0:  MOV     #STACK,SP           ;SETUP THE STACK POINTER
2412  005154  004737  025606                    JSR     PC,CLRP             ;CLEAR THE RP11
2413  005160  012737  043602  002506            MOV     #BUFFER,$BUF        ;LOAD BUFFER ADDRESS
```

```
2414  005166  012737  000001  002504          MOV     #1,$WC          ;SETUP WORD COUNT FOR 1 WORD
2415  005174  005037  002510                  CLR     $CYL            ;CYLINDER 0
2416  005200  005037  002512                  CLR     $SEC            ;TRACK & SECTOR 0
2417  005204  112737  000005  002502          MOVB    #RDSEK,DPB+2    ;READ COMMAND
2418  005212  004737  022740                  JSR     PC,RP11         ;ISSUE THE COMMAND AND RETURN
2419  005216  032764  000200  000004          BIT     #RDY,RPCS(R4)   ;IS THE CONTROLLER READY ?
2420  005224  001404                          BEQ     1$              ;BR IF NOT
2421  005226  004737  023172                  JSR     PC,SAVRP        ;STORE THE REGISTERS
2422  005232  104002                          ERROR   2               ;CAN'T START READ COMMAND
2423  005234  000420                          BR      EXITO           ;EXIT FROM TEST
2424  005236  012764  000001  000004  1$:     MOV     #CLEAR,RPCS(R4) ;ISSUE THE CONTROLLER CLEAR
2425  005244  012764  000001  000004          MOV     #CLEAR,RPCS(R4)  ;CLEAR THE CONTROLLER AGAIN
2426  005252  004737  023172                  JSR     PC,SAVRP        ;STORE THE REGISTERS
2427  005256  032737  000200  001340          BIT     #RDY,$RPCS      ;IS THE CONTROLLER READY ?
2428  005264  001004                          BNE     EXITO           ;BR IF IT IS
2429  005266  104003                          ERROR   3               ;'READY' DIDN'T SET AFTER ISSUING 'CLEAR'
2430                                                                  ;DURING READ COMMAND
2431  005270  000005                          RESET                   ;FORCE THE CONTROLLER READY
2432  005272  004737  020374                  JSR     PC,$TKINT       ;SETUP THE TTY KEYBOARD
2433  005276  000004                  EXITO:  SCOPE                   ;LOOP ?
2434
2435
2436                                          ;;**********************************************************************
2437                                          ;*TEST 1         WRITE CHECK TEST
2438
2439                                          ;*THIS TEST VERIFIES THE WRITE CHECK LOGIC AND TO VERIFY THAT A WRITE
2440                                          ;*CHECK ERROR CAN BE DETECTED.  'FLOATING ONES' AND 'FLOATING ZEROS'
2441                                          ;*PATTERNS ARE USED TO TEST THE WRITE CHECK COMPARE LOGIC.
2442
2443                                          ;;**********************************************************************
2444  005300                          TST1:
2445  005300  033737  001402  001276          BIT     BITS+<$TN*2>,TSTNMS  ;IS THIS TEST SELECTED
2446  005306  001002                          BNE     .+6             ;BR IF YES
2447  005310  000137  005714                  JMP     TST2            ;GO TO THE NEXT TEST
2448  005314  012737  005374  001110          MOV     #TEST1,$LPERR   ;SETUP THE ERROR LOOP ADDRESS
2449  005322  012737  005300  001106          MOV     #TST1,$LPADR    ;SETUP THE SCOPE LOOP ADDRESS
2450  005330  012737  000001  001102          MOV     #1,$TSTNM       ;SETUP TEST NUMBER AND
2451                                                                  ;CLEAR THE ERROR FLAG ($ERFLG)
2452  005336  013777  001102  173576          MOV     $TSTNM,@DISPLAY  ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
2453  005344  012737  005712  001200          MOV     #EXIT1,BYPASS   ;SETUP BYPASS ADDRESS
2454  005352  013704  001324                  MOV     RPADR,R4        ;RP11E BUS ADDRESS
2455  005356  105037  002501                  CLRB    DPB+1           ;CLEAR ANY EXTENDED MEMORY BITS
2456  005362  105037  002503                  CLRB    DPB+3           ;CLEAR 'MODE' & 'HDR' BITS
2457  005366  012737  000144  001162          MOV     #100.,$TIMES    ;DO 100. ITERATIONS
2458  005374  012706  001100          TEST1:  MOV     #STACK,SP       ;SETUP THE STACK POINTER
2459  005400  005037  002512                  CLR     $SEC            ;CLEAR TRACK & SECTOR ADDRESS
2460  005404  005037  002510                  CLR     $CYL            ;CLEAR CYLINDER ADDRESS
2461  005410  012737  000400  002504          MOV     #256.,$WC       ;SET WORD COUNT TO 1 SECTOR
2462  005416  012737  000011  001204          MOV     #9.,PATNUM      ;STARTING PATTERN NUMBER
2463  005424  004737  024364                  JSR     PC,SETBUF       ;FILL THE BUFFER WITH THE PATTERN
2464  005430  012737  043602  002506          MOV     #BUFFER,$BUF    ;SETUP OUTPUT BUFFER
2465  005436  012737  005436  001110  1$:     MOV     #1$,$LPERR      ;SETUP LOOP ON ERROR ADDRESS
2466  005444  004737  025606                  JSR     PC,CLRP         ;CLEAR THE RP11
2467  005450  012737  000003  002502          MOV     #WRTSEK,DPB+2   ;LOAD COMMAND
2468  005456  004737  022740                  JSR     PC,RP11         ;WRITE THE SECTOR WITH THE TEST PATTERN
2469  005462  004737  023616                  JSR     PC,CONRDY       ;WAIT FOR THE OPERATION TO COMPLETE
```

```
2470  005466  004737  023172                    JSR     PC,SAVRP        ;SAVE THE REGISTERS
2471  005472  032737  100000  001340            BIT     #ERR,$RPCS      ;DID AN ERROR OCCUR ?
2472  005500  001401                            BEQ     3$              ;BR IF NOT
2473  005502  104004                            ERROR   4               ;ERROR OCCURED WRITING WRITE CHECK TEST PATTERN
2474  005504  012737  005504  001110  3$:       MOV     #3$,$LPERR      ;CHANGE LOOP ON ERROR ADDRESS
2475  005512  004737  025606                    JSR     PC,CLRP         ;CLEAR THE RP11
2476  005516  012737  000007  002502            MOV     #WRTCHK,DPB+2   ;LOAD WRITE CHECK COMMAND
2477  005524  004737  022740                    JSR     PC,RP11         ;CHECK THE SECTOR JUST WRITTEN
2478  005530  004737  023616                    JSR     PC,CONRDY       ;WAIT FOR WRITE CHECK TO COMPLETE
2479  005534  004737  023172                    JSR     PC,SAVRP        ;STORE THE REGISTERS
2480  005540  032737  100000  001340            BIT     #ERR,$RPCS      ;CHECK OK ?
2481  005546  001403                            BEQ     4$              ;BR IF OK
2482  005550  104005                            ERROR   5               ;ERROR ATTEMPTING TO CHECK TEST PATTERN
2483  005552  004737  025606                    JSR     PC,CLRP         ;CLEAR THE RP11
2484  005556  005001                  4$:       CLR     R1              ;SETUP TO CLEAR BUFFER
2485  005560  005002                            CLR     R2              ;BUFFER INDEX
2486  005562  023727  001204  000011            CMP     PATNUM,#9.      ;'FLOATING 1' PATTERN ?
2487  005570  001002                            BNE     5$              ;BRANCH IF NOT
2488  005572  012701  177777                    MOV     #-1,R1          ;FILL BUFFER WITH ONES
2489  005576  010162  043602          5$:       MOV     R1,BUFFER(R2)   ;FILL BUFFER WITH PATTERN IN R1
2490  005602  022702  000400                    CMP     #256.,R2        ;SEE IF FINISHED
2491  005606  001403                            BEQ     6$              ;BR IF FINISHED
2492  005610  062702  000002                    ADD     #2,R2           ;INCREMENT THE INDEX
2493  005614  000770                            BR      5$              ;CONTINUE
2494  005616  012737  005616  001110  6$:       MOV     #6$,$LPERR      ;CHANGE THE LOOP ON ERROR ADDRESS
2495  005624  004737  025606                    JSR     PC,CLRP         ;CLEAR THE RP11
2496  005630  004737  022740                    JSR     PC,RP11         ;CHECK THE SECTOR AGAIN, ERROR SHOULD OCCUR
2497  005634  004737  023616                    JSR     PC,CONRDY       ;WAIT FOR THE WRITE CHECK TO COMPLETE
2498  005640  004737  023172                    JSR     PC,SAVRP        ;STORE THE REGISTERS
2499  005644  032737  000010  001336            BIT     #BIT03,$RPER    ;DID WRITE CHECK ERROR SET?
2500  005652  001002                            BNE     7$              ;BRANCH IF YES
2501  005654  104006                            ERROR   6               ;WRITE CHECK ERROR DID NOT SET
2502  005656  000405                            BR      8$              ;BYPASS REST OF TEST
2503  005660  032737  100000  001340  7$:       BIT     #ERR,$RPCS      ;DID 'ERR' SET ?
2504  005666  001001                            BNE     8$              ;BR IF IT SET
2505  005670  104106                            ERROR   106             ;'ERR' DIDN'T SET WITH 'WCE'
2506  005672  023727  001204  000012  8$:       CMP     PATNUM,#10.     ;FLOATING A ZERO ?
2507  005700  001404                            BEQ     EXIT1           ;BR IF ZERO
2508  005702  012737  000012  001204            MOV     #10.,PATNUM     ;'FLOATING ZERO' PATTERN
2509  005710  000652                            BR      1$              ;DO THE TEST AGAIN
2510  005712  000004                  EXIT1:    SCOPE                   ;LOOP ?
2511
2512
2513                                            ;;*****************************************************************
2514                                            ;*TEST 2         PARTIAL SECTOR WRITE TEST
2515
2516                                            ;*CHECK THE ABILITY OF THE RP11E TO CLEAR THE REMAINDER OF A SECTOR
2517                                            ;*ON A PARTIAL WRITE OPERATION. A SECTOR OF ALL ONES IS WRITTEN AND
2518                                            ;*THEN A TWO WORD WRITE OPERATION IS PERFORMED. THE SECTOR IS THEN
2519                                            ;*READ BACK AND VERIFIED. THE FIRST TWO WORDS SHOULD BE ONES AND
2520                                            ;*THE REST SHOULD BE ZEROS.
2521
2522                                            ;;*****************************************************************
2523  005714                                    TST2:
2524  005714  033737  001404  001276                    BIT     BITS+<$TN*2>,TSTNMS  ;IS THIS TEST SELECTED
2525  005722  001002                                    BNE     .+6                  ;BR IF YES
```

```
2526  005724  000137  006372                    JMP     TST3            ;GO TO THE NEXT TEST
2527  005730  012737  006010  001110            MOV     #TEST2,$LPERR   ;SETUP THE ERROR LOOP ADDRESS
2528  005736  012737  005714  001106            MOV     #TST2,$LPADR    ;SETUP THE SCOPE LOOP ADDRESS
2529  005744  012737  000002  001102            MOV     #2,$TSTNM       ;SETUP TEST NUMBER AND
2530                                                                    ;CLEAR THE ERROR FLAG ($ERFLG)
2531  005752  013777  001102  173162            MOV     $TSTNM,@DISPLAY ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
2532  005760  012737  006370  001200            MOV     #EXIT2,BYPASS   ;SETUP BYPASS ADDRESS
2533  005766  013704  001324                    MOV     RPADR,R4        ;RP11E BUS ADDRESS
2534  005772  105037  002501                    CLRB    DPB+1           ;CLEAR ANY EXTENDED MEMORY BITS
2535  005776  105037  002503                    CLRB    DPB+3           ;CLEAR 'MODE' & 'HDR' BITS
2536  006002  012737  000144  001162            MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
2537  006010  012706  001100          TEST2:    MOV     #STACK,SP       ;SETUP THE STACK POINTER
2538  006014  012737  006010  001110            MOV     #TEST2,$LPERR   ;SETUP INITIAL LOOP ON ERROR ADDRESS
2539  006022  004737  025606                    JSR     PC,CLRP         ;CLEAR THE RP11
2540  006026  012737  043602  002506            MOV     #BUFFER,$BUF    ;LOAD BUFFER ADDRESS
2541  006034  012737  000400  002504            MOV     #256.,$WC       ;WORD COUNT - 1 SECTOR
2542  006042  005037  002512                    CLR     $SEC            ;CLEAR SECTOR/TRACK ADDRESS
2543  006046  005037  002510                    CLR     $CYL            ;CLEAR CYLINDER ADDRESS
2544  006052  012737  000015  001204            MOV     #13.,PATNUM     ;ONE'S PATTERN NUMBER
2545  006060  004737  024364                    JSR     PC,SETBUF       ;LOAD THE PATTERN INTO THE BUFFER
2546  006064  012737  000003  002502            MOV     #WRTSEK,DPB+2   ;WRITE COMMAND
2547  006072  004737  022740                    JSR     PC,RP11         ;WRITE THE SECTOR
2548  006076  004737  023616                    JSR     PC,CONRDY       ;WAIT FOR THE WRITE TO COMPLETE
2549  006102  004737  023172                    JSR     PC,SAVRP        ;STORE THE REGISTERS
2550  006106  032737  100000  001340            BIT     #ERR,$RPCS      ;DID AN ERROR OCCUR ?
2551  006114  001402                            BEQ     1$              ;BR IF NOT
2552  006116  104004                            ERROR   4               ;ERROR OCCURED DURING WRITE
2553  006120  000523                            BR      EXIT2           ;BYPASS REST OF TEST
2554  006122  012737  006122  001110  1$:       MOV     #1$,$LPERR      ;CHANGE LOOP ON ERROR ADDRESS
2555  006130  004737  025606                    JSR     PC,CLRP         ;CLEAR THE RP11
2556  006134  004737  024364                    JSR     PC,SETBUF       ;LOAD ONE'S PATTERN INTO THE BUFFER
2557  006140  012737  000002  002504            MOV     #2,$WC          ;2 WORD TRANSFER
2558  006146  112737  000003  002502            MOVB    #WRTSEK,DPB+2   ;CHANGE COMMAND TO WRITE
2559  006154  004737  022740                    JSR     PC,RP11         ;WRITE
2560  006160  004737  023616                    JSR     PC,CONRDY       ;WAIT FOR CONTROLLER READY
2561  006164  004737  023172                    JSR     PC,SAVRP        ;STORE THE REGISTERS
2562  006170  032737  100000  001340            BIT     #ERR,$RPCS      ;ERROR ?
2563  006176  001402                            BEQ     2$              ;BR IF NOT
2564  006200  104007                            ERROR   7               ;ERROR WRITING THE PARTIAL SECTOR
2565  006202  000472                            BR      EXIT2           ;BYPASS THE REST OF THE TEST
2566  006204  012737  006204  001110  2$:       MOV     #2$,$LPERR      ;CHANGE THE ERROR LOOP ADDRESS
2567  006212  004737  025606                    JSR     PC,CLRP         ;CLEAR THE RP11
2568  006216  012737  000400  002504            MOV     #256.,$WC       ;CHANGE THE WORD COUNT
2569  006224  012737  000005  002502            MOV     #RDSEK,DPB+2    ;READ COMMAND
2570  006232  004737  022740                    JSR     PC,RP11         ;READ THE SECTOR
2571  006236  004737  023616                    JSR     PC,CONRDY       ;WAIT FOR CONTROLLER READY
2572  006242  004737  023172                    JSR     PC,SAVRP        ;STORE THE REGISTERS
2573  006246  032737  100000  001340            BIT     #ERR,$RPCS      ;DID AN ERROR OCCUR ?
2574  006254  001402                            BEQ     3$              ;BR IF NOT
2575  006256  104017                            ERROR   17              ;ERROR OCCURED READING SECTOR
2576  006260  000443                            BR      EXIT2           ;BYPASS REST OF TEST
2577  006262  022737  177777  043602  3$:       CMP     #177777,BUFFER  ;COMPARE FIRST WORD SHOULD BE ONES
2578  006270  001004                            BNE     4$              ;BR IF NOT
2579  006272  022737  177777  043604            CMP     #177777,BUFFER+2  ;CHECK THE SECOND WORD
2580  006300  001407                            BEQ     5$              ;BR IF OK
2581  006302  012737  177777  001124  4$:       MOV     #177777,$GDDAT  ;EXPECTED DATA
```

```
2582  006310  013737  043602  001126          MOV     BUFFER,$BDDAT   ;ACTUAL DATA
2583  006316  104011                          ERROR   11              ;DATA COMPARE ERROR IN FIRST 2 WORDS
2584  006320  012700  043606          5$:     MOV     #BUFFER+4,R0    ;STARTING ADDRESS OF LAST PART OF THE BUFFER
2585  006324  012701  000376                  MOV     #254.,R1        ;WORD COUNT
2586  006330  005720                  6$:     TST     (R0)+           ;REMAINDER OF SECTOR SHOULD BE ZEROS
2587  006332  001003                          BNE     7$              ;BRANCH IF NOT
2588  006334  005301                          DEC     R1              ;DECREMENT THE CUNT
2589  006336  001374                          BNE     6$              ;BR IF NOT FINISHED
2590  006340  000413                          BR      EXIT2           ;EXIT
2591  006342  016037  177776  001126  7$:     MOV     -2(R0),$BDDAT   ;INCORRECT DATA
2592  006350  005037  001124                  CLR     $GDDAT          ;EXPECTED DATA (ZEROS)
2593  006354  010037  001122                  MOV     R0,$BDADR       ;ADDRESS OF INCORRECT CHARACTER
2594  006360  162737  000002  001122          SUB     #2,$BDADR       ;DECREMENT THE ADDRESS
2595  006366  104012                          ERROR   12              ;DATA FOUND IN AREA OF SECTOR
2596                                                                  ;WHICH SHOULD HAVE BEEN CLEARED
2597                                                                  ;BY A ONE WORD WRITE
2598  006370  000004                  EXIT2:  SCOPE
2599
2600
2601                                  ;;*****************************************************************
2602                                  ;*TEST 3         'EOP' TEST
2603
2604                                  ;*TEST THAT 'EOP' SETS WHEN THE CONTROLLER TRIES TO WRITE BEYOND THE
2605                                  ;*LIMITS OF THE DRIVE.  THE FIRST SECTOR OF THE PACK IS WRITTEN WITH
2606                                  ;*ZEROS; THEN A TWO SECTOR WRITE OF ALL ONE'S IS ISSUED FOR THE MAXIMUM
2607                                  ;*CYLINDER, HEAD 19, SECTOR 9.  'EOP' AND THE ERROR BITS IN 'RPCS' SHOULD
2608                                  ;*BE SET.  THE FIRST SECTOR OF THE PACK IS CHECKED TO MAKE SURE THAT IS
2609                                  ;*STILL CONTAINS ZEROS.
2610
2611                                  ;;*****************************************************************
2612  006372                  TST3:
2613  006372  033737  001406  001276          BIT     BITS+<$TN*2>,TSTNMS  ;IS THIS TEST SELECTED
2614  006400  001002                          BNE     .+6             ;BR IF YES
2615  006402  000137  007050                  JMP     TST4            ;GO TO THE NEXT TEST
2616  006406  012737  006466  001110          MOV     #TEST3,$LPERR   ;SETUP THE ERROR LOOP ADDRESS
2617  006414  012737  006372  001106          MOV     #TST3,$LPADR    ;SETUP THE SCOPE LOOP ADDRESS
2618  006422  012737  000003  001102          MOV     #3,$TSTNM       ;SETUP TEST NUMBER AND
2619                                                                  ;CLEAR THE ERROR FLAG ($ERFLG)
2620  006430  013777  001102  172504          MOV     $TSTNM,@DISPLAY  ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
2621  006436  012737  007046  001200          MOV     #EXIT3,BYPASS   ;SETUP BYPASS ADDRESS
2622  006444  013704  001324                  MOV     RPADR,R4        ;RP11E BUS ADDRESS
2623  006450  105037  002501                  CLRB    DPB+1           ;CLEAR ANY EXTENDED MEMORY BITS
2624  006454  105037  002503                  CLRB    DPB+3           ;CLEAR 'MODE' & 'HDR' BITS
2625  006460  012737  000031  001162          MOV     #25.,$TIMES     ;;DO 25. ITERATIONS
2626  006466  012706  001100          TEST3:  MOV     #STACK,SP       ;SETUP THE STACK POINTER
2627  006472  012737  006466  001110          MOV     #TEST3,$LPERR   ;INITIAL LOOP ON ERRORR ADDRESS
2628  006500  004737  025606                  JSR     PC,CLRP         ;CLEAR THE RP11
2629  006504  012737  000400  002504          MOV     #256.,$WC       ;SET WORD COUNT TO 1 SECTOR
2630  006512  012737  043602  002506          MOV     #BUFFER,$BUF    ;SETUP OUTPUT BUFFER
2631  006520  005037  001204                  CLR     PATNUM          ;SETUP FOR PATTERN ZERO
2632  006524  004737  024364                  JSR     PC,SETBUF       ;LOAD ZEROS INTO THE BUFFER
2633  006530  005037  002512                  CLR     $SEC            ;SET SECTOR & TRACK TO ZERO
2634  006534  005037  002510                  CLR     $CYL            ;SET CYLINDER TO ZERO
2635  006540  112737  000003  002502          MOVB    #WRTSEK,DPB+2   ;WRITE COMMAND
2636  006546  004737  022740                  JSR     PC,RP11         ;WRITE ZEROS IN CO, TO, SO
2637  006552  004737  023616                  JSR     PC,CONRDY       ;WAIT FOR THE ORDER TO COMPLETE
```

```
2638  006556  004737  023172                    JSR     PC,SAVRP        ;STORE THE REGISTERS
2639  006562  032737  100000  001340            BIT     #ERR,$RPCS      ;DID AN ERROR OCCUR ?
2640  006570  001402                            BEQ     1$              ;BR IF NOT
2641  006572  104004                            ERROR   4               ;ERROR OCCURED DURING SETUP
2642  006574  000524                            BR      EXIT3           ;BYPASS REST OF TEST
2643  006576  012737  006576  001110  1$:       MOV     #1$,$LPERR      ;CHANGE LOOP ON ERROR ADDRESS
2644  006604  004737  025606                    JSR     PC,CLRP         ;CLEAR THE RP11
2645  006610  012737  000015  001204            MOV     #13.,PATNUM     ;ONE'S PATTERN INDEX
2646  006616  004737  024364                    JSR     PC,SETBUF       ;LOAD THE PATTERN INTO THE BUFFER
2647  006622  012737  001000  002504            MOV     #512.,$WC       ;CHANGE WORD COUNT TO 2 SECTORS
2648  006630  013737  001210  002510            MOV     MAXCYL,$CYL     ;SELECT THE DRIVE'S MAXIMUM CYLINDER
2649  006636  012737  000011  002512            MOV     #9.,$SEC        ;SELECT SECTOR 9
2650  006644  112737  000023  002513            MOVB    #19.,$TRK       ;SELECT HEAD 19
2651  006652  004737  022740                    JSR     PC,RP11         ;START THE READ
2652  006656  004737  023616                    JSR     PC,CONRDY       ;WAIT FOR CONTROLLER READY
2653  006662  004737  023172                    JSR     PC,SAVRP        ;STORE THE RP11 REGISTERS
2654  006666  032737  000002  001336            BIT     #BIT01,$RPER    ;DID EOP ERROR SET?
2655  006674  001002                            BNE     2$              ;BRANCH IF SET
2656  006676  104013                            ERROR   13              ;'EOP' DIDN'T SET
2657  006700  000405                            BR      3$              ;CHECK RPCA
2658  006702  032737  100000  001340  2$:       BIT     #ERR,$RPCS      ;DID THE ERROR FLAG SET?
2659  006710  001001                            BNE     3$              ;BRANCH IF SET
2660  006712  104014                            ERROR   14              ;ERROR DID NOT SET AFTER GENERATING 'EOP'
2661  006714  005737  001346            3$:     TST     $RPCA           ;SEE IF THE CYLINDER ADDRESS REGISTER WAS CLEARED
2662  006720  001401                            BEQ     4$              ;BR IF CLEARED
2663  006722  104015                            ERROR   15              ;'RPCA' NOT CLEARED BY 'EOP'
2664  006724  023737  001354  001210  4$:       CMP     $SUCA,MAXCYL    ;SUCA STILL EQUAL TO THE MAXIMUM CLINDER
2665  006732  001401                            BEQ     5$              ;BR IF IT IS
2666  006734  104016                            ERROR   16              ;SUCA NOT CORRECT AFTER EOP ERROR
2667  006736  012737  006736  001110  5$:       MOV     #5$,$LPERR      ;CHANGE LOOP ON ERROR ADDRESS
2668  006744  004737  025606                    JSR     PC,CLRP         ;CLEAR THE RP11
2669  006750  012737  000002  002504            MOV     #2,$WC          ;WORD COUNT
2670  006756  005037  002512                    CLR     $SEC            ;CLEAR SECTOR/TRACK ADDRESS
2671  006762  005037  002510                    CLR     $CYL            ;CLEAR THE CYLINDER ADDRESS
2672  006766  012737  000005  002502            MOV     #RDSEK,DPB+2    ;CHANGE COMMAND TO READ
2673  006774  004737  022740                    JSR     PC,RP11         ;EXECUTE THE COMMAND
2674  007000  004737  023616                    JSR     PC,CONRDY       ;WAIT FOR CONTROLLER READY
2675  007004  004737  023172                    JSR     PC,SAVRP        ;STORE THE REGISTERS
2676  007010  032737  100000  001340            BIT     #ERR,$RPCS      ;WERE THERE ANY ERRORS?
2677  007016  001402                            BEQ     6$              ;BRANCH IF NOT
2678  007020  104017                            ERROR   17              ;ERROR ENCOUNTERED ON 2 WORD READ
2679  007022  000411                            BR      EXIT3           ;OF FIRST SECTOR ON THE PACK
2680  007024  013737  043602  001126  6$:       MOV     BUFFER,$BDDAT   ;GET FIRST WORD OF BUFFER
2681  007032  005737  001126                    TST     $BDDAT          ;DOES 1ST SECTOR STILL CONTAIN ZEROS?
2682  007036  001403                            BEQ     EXIT3           ;BRANCH IF YES
2683  007040  005037  001124                    CLR     $GDDAT          ;ZEROS WERE EXPECTED
2684  007044  104020                            ERROR   20              ;CONTENTS OF THE FIRST SECTOR OF THE
2685                                                                    ;PACK CHANGED AFTER FORCING EOP
2686  007046  000004            EXIT3:  SCOPE
2687
2688
2689                                    ;;*****************************************************************
2690                                    ;*TEST 4         TEST 'PROG' ERROR
2691
2692                                    ;*VERIFY THAT THE RP11E GENERATES A 'PROG' ERROR IF A COMMAND IS ISSUED
2693                                    ;*WHILE THE CONTROLLER IS BUSY.
```

```
                                ;;*****************************************************************
007050                          TST4:
007050 033737 001410 001276             BIT     BITS+<$TN*2>,TSTNMS  ;IS THIS TEST SELECTED
007056 001002                           BNE     .+6                  ;BR IF YES
007060 000137 007300                    JMP     TST5                 ;GO TO THE NEXT TEST
007064 012737 007144 001110             MOV     #TEST4,$LPERR        ;SETUP THE ERROR LOOP ADDRESS
007072 012737 007050 001106             MOV     #TST4,$LPADR         ;SETUP THE SCOPE LOOP ADDRESS
007100 012737 000004 001102             MOV     #4,$TSTNM            ;SETUP TEST NUMBER AND
                                                                     ;CLEAR THE ERROR FLAG ($ERFLG)
007106 013777 001102 172026             MOV     $TSTNM,@DISPLAY      ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
007114 012737 007276 001200             MOV     #EXIT4,BYPASS        ;SETUP BYPASS ADDRESS
007122 013704 001324                    MOV     RPADR,R4             ;RP11E BUS ADDRESS
007126 105037 002501                    CLRB    DPB+1                ;CLEAR ANY EXTENDED MEMORY BITS
007132 105037 002503                    CLRB    DPB+3                ;CLEAR 'MODE' & 'HDR' BITS
007136 012737 000144 001162             MOV     #100.,$TIMES         ;DO 100. ITERATIONS
007144 012706 001100            TEST4:  MOV     #STACK,SP            ;SETUP THE STACK POINTER
007150 004737 025606                    JSR     PC,CLRP              ;CLEAR THE RP11
007154 005037 002512                    CLR     $SEC                 ;CLEAR THE SECTOR - TRACK ADDRESSES
007160 005037 002510                    CLR     $CYL                 ;CLEAR THE CYLINDER ADDRESS
007164 012737 043602 002506             MOV     #BUFFER,$BUF         ;SETUP BUFFER ADDRESS
007172 012737 002000 002504             MOV     #1024.,$WC           ;SETUP WORD COUNT
007200 112737 000005 002502             MOVB    #ROSEK,DPB+2         ;READ COMMAND
007206 004737 022740                    JSR     PC,RP11              ;ISSUE THE COMMAND
007212 012737 000001 001220             MOV     #1,STALLT            ;LOAD STALL VALUE
007220 004737 023134                    JSR     PC,STALL             ;STALL FOR 1 MILLISECONDS
007224 112764 000005 000004             MOVB    #ROSEK,RPCS(R4)      ;ISSUE READ COMMAND WHILE BUSY
007232 004737 023172                    JSR     PC,SAVRP             ;STORE THE REGISTERS
007236 032737 002000 001336             BIT     #BIT10,$RPER         ;DID PROGRAM ERROR SET?
007244 001002                           BNE     1$                   ;BRANCH IF SET
007246 104021                           ERROR   21                   ;PROGRAM ERROR DID NOT SET WHEN A
                                                                     ;READ COMMAND WAS ISSUED WHILE
                                                                     ;THE DEVICE WAS BUSY
007250 000412                           BR      EXIT4                ;BYPASS REST OF TEST
007252 032737 040000 001340     1$:     BIT     #HE,$RPCS            ;'HE' SET ?
007260 001001                           BNE     2$                   ;BR IF NOT
007262 104107                           ERROR   107                  ;'HE' DIDN'T SET WITH 'PROG'
007264 032737 100000 001340     2$:     BIT     #ERR,$RPCS           ;DID 'ERR' SET ?
007272 001001                           BNE     EXIT4                ;BRANCH IF SET
007274 104022                           ERROR   22                   ;'ERR' DID NOT SET WITH PROGRAM ERROR
007276 000004                   EXIT4:  SCOPE

                                ;;*****************************************************************
                                ;*TEST 5         'HEADER NOT FOUND' TEST

                                ;*UNFORMAT THE FIRST SECTOR ON THE PACK AND THEN READ IT BACK.  VERIFY
                                ;*THAT READ AND WRITE HEADER OPERATIONS WILL TRANSFER DATA CORRECTLY.
                                ;*ISSUE A WRITE COMMAND TO SECTOR ZERO; THIS SHOULD CAUSE 'HEADER NOT
                                ;*FOUND' TO SET.  REFORMAT THE SECTOR.

                                ;;*****************************************************************
007300                          TST5:
007300 033737 001412 001276             BIT     BITS+<$TN*2>,TSTNMS  ;IS THIS TEST SELECTED
007306 001002                           BNE     .+6                  ;BR IF YES
007310 000137 010024                    JMP     TST6                 ;GO TO THE NEXT TEST
```

```
2750  007314  012737  007374  001110          MOV     #TEST5,$LPERR   ;SETUP THE ERROR LOOP ADDRESS
2751  007322  012737  007300  001106          MOV     #T5T5,$LPADR    ;SETUP THE SCOPE LOOP ADDRESS
2752  007330  012737  000005  001102          MOV     #5,$TSTNM       ;SETUP TEST NUMBER AND
2753                                                                  ;CLEAR THE ERROR FLAG ($ERFLG)
2754  007336  013777  001102  171576          MOV     $TSTNM,@DISPLAY ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
2755  007344  012737  010022  001200          MOV     #EXIT5,BYPASS   ;SETUP BYPASS ADDRESS
2756  007352  013704  001324                  MOV     RPADR,R4        ;RP11E BUS ADDRESS
2757  007356  105037  002501                  CLRB    DPB+1           ;CLEAR ANY EXTENDED MEMORY BITS
2758  007362  105037  002503                  CLRB    DPB+3           ;CLEAR 'MODE' & 'HDR' BITS
2759  007366  012737  000144  001162          MOV     #100.,$TIMES    ;DO 100. ITERATIONS
2760  007374  012706  001100          TEST5:  MOV     #STACK,SP       ;SETUP THE STACK POINTER
2761  007400  012737  007374  001110          MOV     #TEST5,$LPERR   ;INITIAL LOOP ON ERROR ADDRESS
2762  007406  004737  025606                  JSR     PC,CLRP         ;CLEAR THE RP11
2763  007412  012737  000001  043602          MOV     #1,BUFFER       ;SETUP INVALID HEADER IMAGE
2764  007420  012737  000001  043604          MOV     #1,BUFFER+2     ;CYLINDER/TRACK
2765  007426  012737  000001  043606          MOV     #1,BUFFER+4     ;SECTOR
2766  007434  012737  000003  002504          MOV     #3,$WC          ;LOAD WORD COUNT
2767  007442  005037  002512                  CLR     $SEC            ;CLEAR SECTOR/TRACK ADDRESS
2768  007446  005037  002510                  CLR     $CYL            ;CLEAR THE CYLINDER ADDRESS
2769  007452  012737  014000  002502          MOV     #MODE!HDR,DPB+2 ;'MODE' AND 'HEADER' BITS
2770  007460  112737  000003  002502          MOVB    #WRTSEK,DPB+2   ;WRITE ORDER
2771  007466  004737  022740                  JSR     PC,RP11         ;MISFORMAT SECTOR 0
2772  007472  004737  023616                  JSR     PC,CONRDY       ;WAIT FOR THE OPERATION TO COMPLETE
2773  007476  004737  023172                  JSR     PC,SAVRP        ;STORE THE REGISTERS
2774  007502  032737  100000  001340          BIT     #ERR,$RPCS      ;DID AN ERROR OCCUR ?
2775  007510  001402                  BEQ     1$              ;BR IF NOT
2776  007512  104023                  ERROR   23              ;ERROR OCCURED ATTEMPTING TO MISFORMAT SECTOR 0
2777  007514  000506                  BR      TST5B           ;BYPASS TEST & REFORMAT SECTOR 0
2778  007516  012737  007516  001110  1$:     MOV     #1$,$LPERR      ;CHANGE LOOP ON ERROR ADDRESS
2779  007524  004737  025606                  JSR     PC,CLRP         ;CLEAR THE RP11
2780  007530  112737  000005  002502          MOVB    #RDSEK,DPB+2    ;CHANGE COMMAND
2781  007536  004737  022740                  JSR     PC,RP11         ;EXECUTE THE COMMAND
2782  007542  004737  023616                  JSR     PC,CONRDY       ;WAIT FOR CONTROLLER READY
2783  007546  032737  100000  001340          BIT     #ERR,$RPCS      ;ANY ERRORS?
2784  007554  001401                  BEQ     2$              ;BRANCH IF NOT
2785  007556  104024                  ERROR   24              ;ERROR WHILE READING THE HEADER
2786                                                          ;ON SECTOR ZERO
2787  007560  012737  000001  001124  2$:     MOV     #1,$GDDAT       ;EXPECTED RESULT
2788  007566  005000                  CLR     R0              ;CLEAR THE INDEX
2789  007570  023760  001124  043602  3$:     CMP     $GDDAT,BUFFER(R0)  ;CHECK DATA READ BACK
2790  007576  001006                  BNE     4$              ;BRANCH ON NON COMPARE
2791  007600  062700  000002                  ADD     #2,R0           ;UPDATE MODIFIER
2792  007604  022700  000006                  CMP     #6,R0           ;END OF BUFFER?
2793  007610  001367                  BNE     3$              ;BR IF NOT
2794  007612  000404                  BR      TST5A           ;CONTINUE WITH TEST
2795  007614  016037  043602  001126  4$:     MOV     BUFFER(R0),$BDDAT  ;GET BAD DATA
2796  007622  104025                  ERROR   25              ;DATA DID NOT VERIFY AFTER READING
2797                                                          ;THE HEADER OF SECTOR ZERO
2798  007624  012737  007624  001110  TST5A:  MOV     #TST5A,$LPERR   ;CHANGE LOOP ON ERROR ADDRESS
2799  007632  004737  025606                  JSR     PC,CLRP         ;CLEAR THE RP11
2800  007636  012737  000003  002502          MOV     #WRTSEK,DPB+2   ;CHANGE COMMAND (AND CLEAR 'MODE' & 'HDR')
2801  007644  004737  022740                  JSR     PC,RP11         ;EXECUTE THE COMMAND
2802  007650  004737  023616                  JSR     PC,CONRDY       ;WAIT FOR CONTROLLER READY
2803  007654  004737  023172                  JSR     PC,SAVRP        ;STORE THE REGISTERS
2804  007660  032737  010000  001334  1$:     BIT     #BIT12,$RPDS    ;DID HEADER NOT FOUND SET?
2805  007666  001002                  BNE     2$              ;BRANCH IF YES
```

```
 2806  007670  104026                          ERROR   26          ;'HNF' DID NOT SET
 2807  007672  000417                          BR      TST5B       ;BYPASS REST OF TEST
 2808  007674  032737  000001  001336  2$:     BIT     #BIT00,$RPER ;DID DISK ERROR SET?
 2809  007702  001001                          BNE     3$          ;BRANCH IF YES
 2810  007704  104027                          ERROR   27          ;'DSK' DID NOT SET WITH 'HNF'
 2811  007706  032737  040000  001340  3$:     BIT     #HE,$RPCS   ;DID HARD ERROR SET?
 2812  007714  001001                          BNE     4$          ;BRANCH IF YES
 2813  007716  104030                          ERROR   30          ;HARD ERROR NOT SET AFTER 'HNF'
 2814  007720  032737  100000  001340  4$:     BIT     #ERR,$RPCS  ;DID 'ERR' SET ?
 2815  007726  001001                          BNE     TST5B       ;BRANCH IF YES
 2816  007730  104031                          ERROR   31          ;'ERR' DID NOT SET WITH 'HNF'
 2817  007732  012737  007732  001110  TST5B:  MOV     #TST5B,$LPERR ;CHANGE LOOP ON ERROR ADDRESS
 2818  007740  004737  025606                  JSR     PC,CLRP     ;CLEAR THE RP11
 2819  007744  005037  043602                  CLR     BUFFER      ;SETUP HEADER IMAGE
 2820  007750  005037  043604                  CLR     BUFFER+2    ;CYLINDER/TRACK = 0
 2821  007754  005037  043606                  CLR     BUFFER+4    ;SECTOR = 0
 2822  007760  012737  014000  002502          MOV     #MODE!HDR,DPB+2 ;'MODE' AND 'HDR' BITS
 2823  007766  112737  000003  002502          MOVB    #WRTSEK,DPB+2 ;WRITE COMMAND
 2824  007774  004737  022740                  JSR     PC,RP11     ;RESTORE THE HEADER
 2825  010000  004737  023616                  JSR     PC,CONRDY   ;WAIT FOR THE OPERATION TO COMPLETE
 2826  010004  004737  023172                  JSR     PC,SAVRP    ;STORE THE REGISTERS
 2827  010010  032737  100000  001340          BIT     #ERR,$RPCS  ;DID AN ERROR OCCUR ?
 2828  010016  001557                          BEQ     EXIT6       ;BR IF NOT
 2829  010020  104032                          ERROR   32          ;ERROR ATTEMPTING TO RESTORE HEADER
 2830  010022  000004                  EXIT5:  SCOPE
 2831
 2832
 2833                                  ;;#################################################################
 2834                                  ;#TEST 6        TEST COMMAND ISSUED TO A SEEKING DRIVE
 2835
 2836                                  ;#ISSUE A SEEK COMMAND AND WAIT FOR DONE TO SET. THEN ISSUE A READ
 2837                                  ;#COMMAND WHILE THE HEADS ARE STILL MOVING. THE RP11E SHOULD
 2838                                  ;#HOLD THE WRITE COMMAND TILL THE SEEK IS COMPLETE.
 2839
 2840                                  ;;#################################################################
 2841  010024                          TST6:
 2842  010024  033737  001414  001276          BIT     BITS+<$TN*2>,TSTNMS  ;IS THIS TEST SELECTED
 2843  010032  001002                          BNE     .+6                 ;BR IF YES
 2844  010034  000137  010360                  JMP     TST7                ;GO TO THE NEXT TEST
 2845  010040  012737  010120  001110          MOV     #TEST6,$LPERR       ;SETUP THE ERROR LOOP ADDRESS
 2846  010046  012737  010024  001106          MOV     #TST6,$LPADR        ;SETUP THE SCOPE LOOP ADDRESS
 2847  010054  012737  000006  001102          MOV     #6,$TSTNM           ;SETUP TEST NUMBER AND
 2848                                                                      ;CLEAR THE ERROR FLAG ($ERFLG)
 2849  010062  013777  001102  171052          MOV     $TSTNM,@DISPLAY     ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
 2850  010070  012737  010356  001200          MOV     #EXIT6,BYPASS       ;SETUP BYPASS ADDRESS
 2851  010076  013704  001324                  MOV     RPADR,R4            ;RP11E BUS ADDRESS
 2852  010102  105037  002501                  CLRB    DPB+1               ;CLEAR ANY EXTENDED MEMORY BITS
 2853  010106  105037  002503                  CLRB    DPB+3               ;CLEAR 'MODE' & 'HDR' BITS
 2854  010112  012737  000031  001162          MOV     #25.,$TIMES         ;DO 25. ITERATIONS
 2855  010120  012706  001100          TEST6:  MOV     #STACK,SP           ;SETUP THE STACK POINTER
 2856  010124  004737  025606                  JSR     PC,CLRP             ;CLEAR THE RP11
 2857  010130  005037  002512                  CLR     $SEC                ;SECTOR/TRACK 0
 2858  010134  005037  002510                  CLR     $CYL                ;CYLINDER 0
 2859  010140  112737  000015  002502          MOVB    #HOMSEK,DPB+2       ;HOME SEEK COMMAND
 2860  010146  004737  022740                  JSR     PC,RP11             ;DO THE COMMAND
 2861  010152  004737  023500                  JSR     PC,DRVRDY           ;WAIT FOR DRIVE READY
```

```
 2862  010156  012737  000300  002510          MOV     #192.,$CYL      ;CHANGE CYLINDER TO CYLINDER 192
 2863  010164  112737  000011  002502          MOVB    #SEEK,DPB+2     ;SEEK COMMAND
 2864  010172  004737  022740                  JSR     PC,RP11         ;INITIATE THE SEEK
 2865  010176  012737  000001  001220          MOV     #1,STALLT       ;LOAD STALL VALUE
 2866  010204  004737  023134                  JSR     PC,STALL        ;STALL FOR 1 MILLISECONDS
 2867  010210  012764  177775  000006          MOV     #-3,RPWC(R4)    ;SET WORD COUNT TO THREE WORDS
 2868  010216  012764  043602  000010          MOV     #BUFFER,RPBA(R4)  ;BUFFER ADDRESS
 2869  010224  052737  014000  002502          BIS     #MODE!HDR,DPB+2 ;SETUP TO READ THE HEADER
 2870  010232  112764  000017  000004          MOVB    #READ,RPCS(R4)  ;TRY TO DO A READ
 2871  010240  004737  023500                  JSR     PC,DRVRDY       ;WAIT FOR DRIVE READY
 2872  010244  004737  023172                  JSR     PC,SAVRP        ;STORE THE REGISTERS
 2873  010250  032737  000200  001340          BIT     #RDY,$RPCS      ;DID CONTROLLER BECOME READY ALSO ?
 2874  010256  001402                          BEQ     1$              ;BR IF NOT
 2875  010260  104033                          ERROR   33              ;READ DIDN'T START DURING SEEK
 2876  010262  000435                          BR      EXIT6           ;EXIT
 2877  010264  004737  023616          1$:     JSR     PC,CONRDY       ;WAIT FOR CONTROLLER READY
 2878  010270  004737  023172                  JSR     PC,SAVRP        ;STORE THE REGISTERS
 2879  010274  032737  100000  001340          BIT     #ERR,$RPCS      ;ANY ERRORS?
 2880  010302  001401                          BEQ     2$              ;BRANCH IF NO
 2881  010304  104034                          ERROR   34              ;ERROR OCCURED DURING A READ ISSUED
 2882                                                                  ;WHILE THE DRIVE IS SEEKING
 2883  010306  012737  014000  001124  2$:     MOV     #(192.*32.),$GDDAT  ;EXPECTED CYLINDER VALUE
 2884  010314  013737  043604  001126          MOV     BUFFER+2,$BDDAT ;RECEIVED DATA
 2885  010322  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CORRECT ?
 2886  010330  001366                          BNE     2$              ;BR IF NOT
 2887  010332  005037  001124                  CLR     $GDDAT          ;EXPECTED VALUE FOR SECTOR
 2888  010336  013737  043606  001126          MOV     BUFFER+4,$BDDAT ;RECEIVED SECTOR
 2889  010344  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CORRECT ?
 2890  010352  001401                          BEQ     EXIT6           ;BR IF OK
 2891  010354  104035                          ERROR   35              ;HEADER DATA INCORRECT
 2892  010356  000004                  EXIT6:  SCOPE
 2893
 2894
 2895                                  ;;#####################################################################
 2896                                  ;#TEST 7         TEST 'NXME' BIT
 2897
 2898                                  ;#THE NON-EXISTENT MEMORY ERROR BIT IS TESTED BY ATTEMPTING TO READ 1
 2899                                  ;#WORD INTO LOCATION 760000.  'NXME', 'HE', & 'ERR' SHOULD ALL BE SET.
 2900
 2901                                  ;;#####################################################################
 2902  010360                          TST7:
 2903  010360  033737  001416  001276          BIT     BITS+($TN*2),TSTNMS  ;IS THIS TEST SELECTED
 2904  010366  001002                          BNE     .+6             ;BR IF YES
 2905  010370  000137  010602                  JMP     TST10           ;GO TO THE NEXT TEST
 2906  010374  012737  010454  001110          MOV     #TEST7,$LPERR   ;SETUP THE ERROR LOOP ADDRESS
 2907  010402  012737  010360  001106          MOV     #TST7,$LPADR    ;SETUP THE SCOPE LOOP ADDRESS
 2908  010410  012737  000007  001102          MOV     #7,%TSTNM       ;SETUP TEST NUMBER AND
 2909                                                                  ;CLEAR THE ERROR FLAG ($ERFLG)
 2910  010416  013777  001102  170516          MOV     $TSTNM,@DISPLAY  ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
 2911  010424  012737  010600  001200          MOV     #EXIT7,BYPASS   ;SETUP BYPASS ADDRESS
 2912  010432  013704  001324                  MOV     RPADR,R4        ;RP11E BUS ADDRESS
 2913  010436  105037  002501                  CLRB    DPB+1           ;CLEAR ANY EXTENDED MEMORY BITS
 2914  010442  105037  002503                  CLRB    DPB+3           ;CLEAR 'MODE' & 'HDR' BITS
 2915  010446  012737  000144  001162          MOV     #100.,STIMES    ;;DO 100. ITERATIONS
 2916  010454  012706  001100          TEST7:  MOV     #STACK,SP       ;SETUP THE STACK POINTER
 2917  010460  004737  025606                  JSR     PC,CLRP         ;CLEAR THE RP11
```

```
 2918  010464  012737  000001  002504          MOV     #1,$WC          ;WORD COUNT
 2919  010472  005037  002512                  CLR     $SEC            ;SECTOR/TRACK ZERO
 2920  010476  005037  002510                  CLR     $CYL            ;CYLINDER ZERO
 2921  010502  112737  000060  002501          MOVB    #MEX1!MEX0,DPB+1  ;LOAD BOTH EXTENDED MEMORY BITS
 2922  010510  012737  160000  002506          MOV     #160000,$BUF    ;LOAD ADDR TO FORCE NON EX MEMORY
 2923  010516  112737  000005  002502          MOVB    #RDSEK,DPB+2    ;TRY TO READ INTO THE LOCATION
 2924  010524  004737  022740                  JSR     PC,RP11         ;INITIATE THE OPERATION
 2925  010530  004737  023616                  JSR     PC,CONRDY       ;WAIT FOR CONTROLLER READY
 2926  010534  004737  023172                  JSR     PC,SAVRP        ;STORE THE REGISTERS
 2927  010540  032737  000004  001336          BIT     #BIT02,$RPER    ;DID NON EX MEMORY SET ?
 2928  010546  001002                          BNE     2$              ;BRANCH IF SET
 2929  010550  104036                          ERROR   36              ;'NXME' ERROR DIDN'T SET
 2930  010552  000412                          BR      EXIT7           ;DON'T CHECK THE RPCS BITS
 2931  010554  032737  040000  001340  2$:     BIT     #HE,$RPCS       ;DID 'HE' SET ON NON EX MEMORY
 2932  010562  001001                          BNE     3$              ;BRANCH IF SET
 2933  010564  104037                          ERROR   37              ;'HE' DIDN'T SET
 2934  010566  032737  100000  001340  3$:     BIT     #ERR,$RPCS      ;DID 'ERR' SET WITH NON EX MEMORY
 2935  010574  001001                          BNE     EXIT7           ;BRANCH IF SET
 2936  010576  104040                          ERROR   40              ;'ERR' DIDN'T SET
 2937  010600  000004                  EXIT7:  SCOPE
 2938
 2939
 2940                                  ;;*****************************************************************
 2941                                  ;*TEST 10       SECTOR ADDRESSING TEST
 2942
 2943                                  ;*THE SECTOR ADDRESSING TEST TESTS THE CAPABILTIY OF THE RP11E TO LOCATE
 2944                                  ;*A SECTOR BY USING THE 'SOT' COUNTER (FOR 'HDR' OPERATIONS) AND TO LOCATE
 2945                                  ;*A SECTOR BY COMPARING THE HEADER CONTENTS AGAINST THE CYLINDER AND TRACK/
 2946                                  ;*SECTOR ADDRESS REGISTERS (NORMAL MODE).  THE TEST IS PERFORMED IN 3 PARTS.
 2947                                  ;*
 2948                                  ;*      1.  WRITE ALL HEADERS ON TRACK 0, CYLINDER 0 IN ASCENDING SEQUENCE
 2949                                  ;*      FROM INDEX (0, 1, 2 ... 9).
 2950                                  ;*
 2951                                  ;*      2.  READ EACH HEADER, BEGINNING WITH SECTOR 0, AND VERIFY THAT THE
                                       ;*      SECTOR FIELD IN THE HEADER IS CORRECT FOR THE HEADER EXPECTED.
                                       ;*
                                       ;*      VERIFY THAT THE 'SOT' COUNTER AND THE SECTOR ADDRESS IN 'RPDA' ARE
                                       ;*      EQUAL AND ARE THE EXPECTED VALUE.  THE EXPECTED VALUE IS 1 GREATER
                                       ;*      THAN THE SECTOR READ.  THE SECTOR ADDRESS REGISTER AND THE 'SOT'
                                       ;*      WILL BE INCREMENTED BY THE TIME THE PROGRAM IS ABLE TO READ THE
                                       ;*      REGISTER.
                                       ;*
                                       ;*      WRITE THE SECTOR'S ADDRESS IN THE DATA FIELD.  (FOR SECTOR 0, AN
                                       ;*      OCTAL 12 IS WRITTEN.)
                                       ;*
                                       ;*      CONTINUE THIS SEQUENCE FOR THE REMAINING SECTORS ON THE TRACK.
                                       ;*
                                       ;*      3.  READ THE DATA FROM EACH SECTOR USING A NORMAL (I.E., NON-HEADER)
                                       ;*      READ AND VERIFY THAT THE DATA IS CORRECT FOR THE EXPECTED SECTOR.

                                       ;;*****************************************************************
 2969  010602                          TST10:
 2970  010602  033737  001420  001276          BIT     BITS+<$TN*2>,TSTNMS  ;IS THIS TEST SELECTED
 2971  010610  001002                          BNE     .+6             ;BR IF YES
 2972  010612  000137  011522                  JMP     TST11           ;GO TO THE NEXT TEST
 2973  010616  012737  010676  001110          MOV     #TEST10,$LPERR  ;SETUP THE ERROR LOOP ADDRESS
```

```
2974  010624  012737  010602  001106          MOV     #TST10,$LPADR   ;SETUP THE SCOPE LOOP ADDRESS
2975  010632  012737  000010  001102          MOV     #10,$TSTNM      ;SETUP TEST NUMBER AND
2976                                                                  ;CLEAR THE ERROR FLAG ($ERFLG)
2977  010640  013777  001102  170274          MOV     $TSTNM,@DISPLAY ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
2978  010646  012737  011520  001200          MOV     #EXIT10,BYPASS  ;SETUP BYPASS ADDRESS
2979  010654  013704  001324                  MOV     RPADR,R4        ;RP11E BUS ADDRESS
2980  010660  105037  002501                  CLRB    DPB+1           ;CLEAR ANY EXTENDED MEMORY BITS
2981  010664  105037  002503                  CLRB    DPB+3           ;CLEAR 'MODE' & 'HDR' BITS
2982  010670  012737  000031  001162          MOV     #25.,$TIMES     ;DO 25. ITERATIONS
2983  010676  012706  001100          TEST10: MOV     #STACK,SP       ;RESET THE STACK POINTER
2984  010702  012737  010676  001110          MOV     #TEST10,$LPERR  ;LOOP ON ERROR ADDRESS
2985  010710  004737  025606                  JSR     PC,CLRP         ;CLEAR THE RP11
2986  010714  012737  043602  002506          MOV     #BUFFER,$BUF    ;SETUP BUFFER ADDRESS
2987  010722  012737  000036  002504          MOV     #30.,$WC        ;SETUP WORD COUNT FOR 10 SECTORS
2988  010730  005037  002510                  CLR     $CYL            ;SETUP FOR CYLINDER 0
2989  010734  005037  002512                  CLR     $SEC            ;TRACK/SECTOR 0
2990  010740  004737  024236                  JSR     PC,LOOSEC       ;SETUP FORMAT BUFFER
2991  010744  012737  014000  002502          MOV     #MODE!HDR,DPB+2 ;SETUP THE HEADER CONTROL BITS
2992  010752  112737  000003  002502          MOVB    #WRTSEK,DPB+2   ;THE COMMAND CODE
2993  010760  004737  022740                  JSR     PC,RP11         ;FORMAT THE TRACK
2994  010764  004737  023616                  JSR     PC,CONRDY       ;WAIT FOR THE ORDER TO COMPLETE
2995  010770  004737  023172                  JSR     PC,SAVRP        ;STORE THE REGISTERS
2996  010774  032737  100000  001340          BIT     #ERR,$RPCS      ;ERROR ?
2997  011002  001403                          BEQ     1$              ;BR IF NOT
2998  011004  104041                          ERROR   41              ;ERROR TRYING TO FORMAT THE TRACK
2999  011006  000137  011520                  JMP     EXIT10          ;BYPASS REST OF THE TEST
3000  011012  105037  002512          1$:     CLRB    $SEC            ;START CHECK WITH SECTOR 0
3001  011016  012737  011016  001110  2$:     MOV     #2$,$LPERR      ;CHANGE THE LOOP ON ERROR ADDRESS
3002  011024  004737  025606                  JSR     PC,CLRP         ;CLEAR THE RP11
3003  011030  012737  014000  002502          MOV     #MODE!HDR,DPB+2 ;SET 'MODE' AND 'HDR' BITS
3004  011036  112737  000005  002502          MOVB    #RDSEK,DPB+2    ;CHANGE THE OPERATION TO A READ
3005  011044  012737  000003  002504          MOV     #3,$WC          ;CHANGE THE WORD COUNT TO 3
3006  011052  004737  022740                  JSR     PC,RP11         ;READ THE SECTOR'S HEADER
3007  011056  004737  023616                  JSR     PC,CONRDY       ;WAIT FOR THE ORDER TO COMPLETE
3008  011062  004737  023172                  JSR     PC,SAVRP        ;STORE THE REGISTERS
3009  011066  012737  000000  001220          MOV     #0,STALLT       ;LOAD STALL VALUE
3010  011074  004737  023134                  JSR     PC,STALL        ;STALL FOR 0 MILLISECONDS
3011  011100  016437  000014  001350          MOV     RPDA(R4),$RPDA  ;GET THE CONTENTS OF THE ADDRESS REGISTER
3012  011106  032737  100000  001340          BIT     #ERR,$RPCS      ;ERROR ?
3013  011114  001402                          BEQ     3$              ;BR IF NOT
3014  011116  104042                          ERROR   42              ;ERROR READING THE HEADER
3015  011120  000457                          BR      6$              ;BYPASS DATA CHECK
3016  011122  005037  001124          3$:     CLR     $GDDAT          ;CLEAR FOR EXPECTED DATA
3017  011126  113737  002512  001124          MOVB    $SEC,$GDDAT     ;MOVE SECTOR ADDRESS
3018  011134  023737  001124  043606          CMP     $GDDAT,BUFFER+4 ;CHECK THE SECTOR FIELD
3019  011142  001401                          BEQ     4$              ;BR IF CORRECT
3020  011144  104043                          ERROR   43              ;SECTOR FIELD FROM HEADER NOT CORRECT
3021  011146  005237  001124          4$:     INC     $GDDAT          ;INCREMENT SECTOR ADDRESS FOR 'SOT' CHECK
3022  011152  023727  001124  000012          CMP     $GDDAT,#10.     ;EXCEED THE MAXIMUM ?
3023  011160  103402                          BLO     .+6             ;BR IF NOT
3024  011162  005037  001124                  CLR     $GDDAT          ;RESET COMPARISON VALUE
3025  011166  013737  001350  001126          MOV     $RPDA,$BDDAT    ;CHECK THE 'SOT'
3026  011174  042737  177417  001126          BIC     #^C360,$BDDAT   ;CLEAR THE OTHER BITS
3027  011202  006237  001126                  ASR     $BDDAT          ;SHIFT THE 'SOT'
3028  011206  006237  001126                  ASR     $BDDAT          ;SHIFT THE 'SOT'
3029  011212  006237  001126                  ASR     $BDDAT          ;SHIFT THE 'SOT'
```

```
3030  011216  006237  001126                ASR     $BDDAT          ;SHIFT THE 'SOT'
3031  011222  013737  001350  001122        MOV     $RPDA,$BDADR    ;CHECK THE SECTOR ADDRESS
3032  011230  042737  177760  001122        BIC     #^C17,$BDADR    ;LEAVE THE SECTOR ADDRESS
3033  011236  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;IS THE 'SOT' CORRECT ?
3034  011244  001004                        BNE     5$              ;BR IF NOT
3035  011246  023737  001126  001122        CMP     $BDDAT,$BDADR   ;DOES THE SECTOR ADDRESS EQUAT THE 'SOT' ?
3036  011254  001401                        BEQ     6$              ;BR IF IT DOES
3037  011256  104045                5$:     ERROR   45              ;'SOT' OR SECTOR ADDRESS IS NOT CORRECT
3038  011260  012737  011260  001110  6$:     MOV     #6$,$LPERR      ;CHANGE THE LOOP ON ERROR ADDRESS
3039  011266  004737  025606                JSR     PC,CLRP         ;CLEAR THE RP11
3040  011272  105037  002503                CLRB    DPB+3           ;CLEAR 'MODE' & 'HDR' BITS
3041  011276  004737  024270                JSR     PC,FILSEC       ;SETUP TO WRITE SECTOR ADDRESS AS DATA
3042  011302  112737  000003  002502        MOVB    #WRTSEK,DPB+2   ;CHANGE OPERATION
3043  011310  004737  022740                JSR     PC,RP11         ;WRITE THE SECTOR
3044  011314  004737  023616                JSR     PC,CONRDY       ;WAIT FOR THE ORDER TO COMPLETE
3045  011320  004737  023172                JSR     PC,SAVRP        ;STORE THE REGISTERS
3046  011324  032737  100000  001340        BIT     #ERR,$RPCS      ;ERROR ?
3047  011332  001401                        BEQ     7$              ;BR IF NOT
3048  011334  104044                        ERROR   44              ;ERROR TRYING TO WRITE SECTOR ADDRESS AS DATA
3049  011336  112737  000005  002502  7$:     MOVB    #RDSEK,DPB+2    ;CHANGE THE OPERATION
3050  011344  105237  002512                INCB    $SEC            ;INCREMENT THE SECTOR
3051  011350  123727  002512  000012        CMPB    $SEC,#10.       ;AT MAXIMUM SECTOR ADDRESS ?
3052  011356  001217                        BNE     2$              ;BR IF NOT
3053  011360  012737  011414  001110        MOV     #8$,$LPERR      ;CHANGE THE LOOP ON ERROR ADDRESS
3054  011366  012737  000400  002504        MOV     #256.,$WC       ;CHANGE WORD COUNT
3055  011374  105037  002512                CLRB    $SEC            ;START WITH SECTOR ZERO
3056  011400  012737  000012  001124        MOV     #10.,$GDDAT     ;SECTOR ZERO PATTERN
3057  011406  112737  000005  002502        MOVB    #RDSEK,DPB+2    ;CHANGE OPERATION CODE
3058  011414  004737  025606          8$:   JSR     PC,CLRP         ;CLEAR THE RP11
3059  011420  004737  022740                JSR     PC,RP11         ;READ THE SECTOR
3060  011424  004737  023616                JSR     PC,CONRDY       ;WAIT FOR THE ORDER TO COMPLETE
3061  011430  004737  023172                JSR     PC,SAVRP        ;STORE THE REGISTERS
3062  011434  032737  100000  001340        BIT     #ERR,$RPCS      ;ERROR ?
3063  011442  001402                        BEQ     9$              ;BR IF NOT
3064  011444  104017                        ERROR   17              ;ERROR TRYING TO READ THE SECTOR
3065  011446  000412                        BR      10$             ;BYPASS DATA CHECK
3066  011450  005037  001120          9$:   CLR     $GDADR          ;CLEAR FOR EXPECTED DATA
3067  011454  113737  002512  001120        MOVB    $SEC,$GDADR     ;EXPECTED DATA
3068  011462  023737  001124  043602        CMP     $GDDAT,BUFFER   ;CORRECT CONTENTS ?
3069  011470  001401                        BEQ     10$             ;BR IF CORRECT
3070  011472  104046                        ERROR   46              ;SECTOR PATTERN NOT CORRECT FOR SECTOR ADDRESS
3071  011474  105237  002512          10$:  INCB    $SEC            ;INCREMENT SECTOR ADDRESS
3072  011500  123727  002512  000012        CMPB    $SEC,#10.       ;FINISHED SECTOR NINE ?
3073  011506  001404                        BEQ     EXIT10          ;BR IF COMPLETED
3074  011510  113737  002512  001124        MOVB    $SEC,$GDDAT     ;NEXT SECTOR
3075  011516  000736                        BR      8$              ;CONTINUE CHECKING
3076  011520  000004                EXIT10: SCOPE                   ;LOOP ?
3077
3078
3079                          ;;*****************************************************************
3080                          ;*TEST 11        TRACK ADDRESSING TEST
3081
3082                          ;*TRACK ADDRESSING IS TESTED BY WRITING THE TRACK'S ADDRESS AS DATA IN
3083                          ;*SECTOR 0 OF EACH TRACK IN CYLINDER 0.  SECTOR 0 IS THEN READ BACK AND
3084                          ;*THE DATA IS CHECKED TO VERIFY THAT THE PROPER HEAD WAS SELECTED.
3085
```

```
3086                                    ;;*****************************************************************
3087  011522                            TST11:
3088  011522  033737  001422  001276            BIT     BITS+<$TN*2>,TSTNMS  ;IS THIS TEST SELECTED
3089  011530  001002                            BNE     .+6                  ;BR IF YES
3090  011532  000137  012052                    JMP     TST12                ;GO TO THE NEXT TEST
3091  011536  012737  011642  001110            MOV     #TEST11,$LPERR       ;SETUP THE ERROR LOOP ADDRESS
3092  011544  012737  011522  001106            MOV     #TST11,$LPADR        ;SETUP THE SCOPE LOOP ADDRESS
3093  011552  012737  000011  001102            MOV     #11,$TSTNM           ;SETUP TEST NUMBER AND
3094                                            ;CLEAR THE ERROR FLAG ($ERFLG)
3095  011560  013777  001102  167354            MOV     $TSTNM,@DISPLAY      ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
3096  011566  012737  012050  001200            MOV     #EXIT11,BYPASS       ;SETUP BYPASS ADDRESS
3097  011574  013704  001324                    MOV     RPHDR,R4             ;RP11E BUS ADDRESS
3098  011600  105037  002501                    CLRB    DPB+1                ;CLEAR ANY EXTENDED MEMORY BITS
3099  011604  105037  002503                    CLRB    DPB+3                ;CLEAR 'MODE' & 'HDR' BITS
3100  011610  012737  000031  001162            MOV     #25.,$TIMES          ;DO 25. ITERATIONS
3101  011616  005037  002512                    CLR     $SEC                 ;CLEAR SECTOR/TRACK ADDRESS
3102  011622  005037  002510                    CLR     $CYL                 ;CLEAR CYLINDER ADDRESS
3103  011626  012737  000400  002504            MOV     #256.,$WC            ;SET WORD COUNT TO ONE SECTOR
3104  011634  012737  043602  002506            MOV     #BUFFER,$BUF         ;SETUP THE BUFFER ADDRESS
3105  011642  012706  001100            TEST11: MOV     #STACK,SP            ;SETUP THE STACK POINTER
3106  011646  012737  011646  001110  1$:       MOV     #1$,$LPERR           ;CHANGE LOOP ON ERROR ADDRESS
3107  011654  004737  025606                    JSR     PC,CLRP              ;CLEAR THE RP11
3108  011660  112737  000003  002502            MOVB    #WRTSEK,DPB+2        ;CHANGE COMMAND TO WRITE
3109  011666  004737  024330                    JSR     PC,FILTRK            ;FILL BUFFER WITH THE TRACK ADDRESS
3110  011672  004737  022740                    JSR     PC,RP11              ;WRITE THE TRACK'S ADDRESS
3111  011676  004737  023616                    JSR     PC,CONRDY            ;WAIT FOR CONTROLLER TO FINISH
3112  011702  004737  023172                    JSR     PC,SAVRP             ;STORE THE REGISTERS
3113  011706  032737  100000  001340            BIT     #ERR,$RPCS           ;ERROR ?
3114  011714  001401                            BEQ     2$                   ;BR IF NOT
3115  011716  104004                            ERROR   4                    ;ERROR WRITTING TEST PATTERN
3116  011720  105237  002513            2$:     INCB    $TRK                 ;INCREMENT THE TRACK ADDRESS
3117  011724  123727  002513  000024            CMPB    $TRK,#20.            ;FINISHED ?
3118  011732  103745                            BLO     1$                   ;BR IF NOT FINISHED
3119  011734  005037  001124            3$:     CLR     $GDDAT               ;SETUP EXTPECTED VALUE
3120  011740  005037  002512                    CLR     $SEC                 ;START WITH ZERO
3121  011744  012737  011744  001110  4$:       MOV     #4$,$LPERR           ;CHANGE THE LOOP ON ERROR ADDRESS
3122  011752  004737  025606                    JSR     PC,CLRP              ;CLEAR THE RP11
3123  011756  112737  000005  002502            MOVB    #RDSEK,DPB+2         ;CHANGE COMMAND
3124  011764  004737  022740                    JSR     PC,RP11              ;READ THE SECTOR
3125  011770  004737  023616                    JSR     PC,CONRDY            ;WAIT FOR THE CONTROLLER TO BE READY
3126  011774  004737  023172                    JSR     PC,SAVRP             ;STORE THE REGISTERS
3127  012000  032737  100000  001340            BIT     #ERR,$RPCS           ;ERROR ?
3128  012006  001402                            BEQ     5$                   ;BR IF NOT
3129  012010  104017                            ERROR   17                   ;ERROR READING THE SECTOR
3130  012012  000405                            BR      6$                   ;BYPASS THE DATA CHECK
3131  012014  123737  001125  043603  5$:       CMPB    $GDDAT+1,BUFFER+1    ;CORRECT VALUE ?
3132  012022  001401                            BEQ     6$                   ;BR IF OK
3133  012024  104047                            ERROR   47                   ;CONTENTS WRONG
3134  012026  123727  002513  000023  6$:       CMPB    $TRK,#19.            ;FINISHED ?
3135  012034  001405                            BEQ     EXIT11               ;BR IF FINISHED
3136  012036  105237  002513                    INCB    $TRK                 ;INCREMENT THE TRACK ADDRESS
3137  012042  105237  001125                    INCB    $GDDAT+1             ;COMPARSION WORD
3138  012046  000736                            BR      4$                   ;CONTINUE
3139  012050  000004                    EXIT11: SCOPE                        ;LOOP ?
3140
3141
```

```
                                ;;*****************************************************************
                                ;*TEST 12       EXTENDED MEMORY ADDRESS TEST
                                ;
                                ;*THIS TEST TESTS THE OPERATION OF THE EXTENDED MEMORY ADDRESS BITS.  IF THE
                                ;*SYSTEM DOES NOT HAVE MEMORY MANAGEMENT OR HAS MEMORY MANAGEMENT AND ONLY
                                ;*32K, THE TEST WILL NOT BE PERFORMED.
                                ;*
                                ;*       1.  THE PROGRAM WRITES A 2 WORD TEST SECTOR OF ALL ONES.
                                ;*
                                ;*       2.  EXTENDED ADDRESS BIT 'MEXDO' IS TESTED BY CLEARING LOCATION
                                ;*       200000 AND READING THE TEST SECTOR INTO LOCATION 177776.  LOCATION
                                ;*       200000 IS CHECKED TO VERIFY THAT THE DATA IS CORRECT (ONES).  THE
                                ;*       PROGRAM VERIFIES THAT 'MEXDO' HAS SET AND THAT 'MEXD1' IS NOT SET.
                                ;*
                                ;*       3.  IF THE SYSTEM HAS AT LEAST 64K, 'MEXD1' IS TESTED.  LOCATION
                                ;*       400000 IS CLEARED AND THE TEST SECTOR IS READ INTO LOCATION 377776
                                ;*       ('MEXDO' IS SET FOR THE READ).  LOCATION 400000 IS CHECKED FOR THE
                                ;*       PROPER CONTENTS (ONES).  'MEXDO' SHOULD HAVE RESET AND 'MEXD1' SHOULD
                                ;*       BE SET.
                                ;
                                ;;*****************************************************************
012052                          TST12:
012052  033737  001424  001276          BIT     BITS+<$TN*2>,TSTNMS  ;IS THIS TEST SELECTED
012060  001002                          BNE     .+6                  ;BR IF YES
012062  000137  012710                  JMP     TST13                ;GO TO THE NEXT TEST
012066  012737  012172  001110          MOV     #TEST12,$LPERR       ;SETUP THE ERROR LOOP ADDRESS
012074  012737  012052  001106          MOV     #TST12,$LPADR        ;SETUP THE SCOPE LOOP ADDRESS
012102  012737  000012  001102          MOV     #12,$TSTNM           ;SETUP TEST NUMBER AND
                                                                     ;CLEAR THE ERROR FLAG ($ERFLG)
012110  013777  001102  167024          MOV     $TSTNM,@DISPLAY      ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
012116  012737  012702  001200          MOV     #EXIT12,BYPASS       ;SETUP BYPASS ADDRESS
012124  013704  001324                  MOV     RPADR,R4             ;RP11E BUS ADDRESS
012130  105037  002501                  CLRB    DPB+1                ;CLEAR ANY EXTENDED MEMORY BITS
012134  105037  002503                  CLRB    DPB+3                ;CLEAR 'MODE' & 'HDR' BITS
012140  012737  000144  001162          MOV     #100.,$TIMES         ;DO 100. ITERATIONS
012146  005737  022202                  TST     $KT11                ;MEMORY MANAGEMENT ON THE SYSTEM ?
012152  100402                          BMI     1$                   ;BR IF ON SYSTEM
012154  000137  012710                  JMP     TST13                ;GO TO THE NEXT TEST
012160  005737  001232          1$:     TST     MAXMEM+2             ;ENOUGH MEMORY FOR THIS TEST ?
012164  001002                          BNE     TEST12               ;BR IF YES
012166  000137  012710                  JMP     TST13                ;GO TO THE NEXT TEST
012172  012706  001100          TEST12: MOV     #STACK,SP            ;SETUP THE STACK POINTER
012176  012737  012172  001110          MOV     #TEST12,$LPERR       ;INITIAL ERROR LOOP ADDRESS
012204  004737  025606                  JSR     PC,CLRP              ;CLEAR THE RP11
012210  005037  043602                  CLR     BUFFER               ;TEST PATTERN
012214  012737  177777  043604          MOV     #-1,BUFFER+2         ;SECOND WORD OF PATTERN
012222  012737  043602  002506          MOV     #BUFFER,$BUF         ;SETUP THE BUFFER ADDRESS
012230  012737  000002  002504          MOV     #2,$WC               ;WORD COUNT OF 2
012236  005037  002512                  CLR     $SEC                 ;SECTOR/TRACK ZERO
012242  005037  002510                  CLR     $CYL                 ;CYLINDER ZERO
012246  012737  000003  002502          MOV     #WRTSEK,DPB+2        ;WRITE COMMAND
012254  004737  022740                  JSR     PC,RP11              ;INITIATE THE COMMAND
012260  004737  023616                  JSR     PC,CONRDY            ;WAIT FOR CONTROLLER READY
012264  004737  023172                  JSR     PC,SAVRP             ;STORE THE REGISTERS
012270  032737  100000  001340          BIT     #ERR,$RPCS           ;DID OPERATION TERMINATE WITH AN ERROR ?
012276  001403                          BEQ     1$                   ;BR IF NOT
```

```
3198  012300  104004                        ERROR   4               ;ERROR AFTER 2 WORD WRITE
3199  012302  000137  012702                JMP     EXIT12          ;BYPASS REST OF TEST
3200  012306  012737  012306  001110  1$:   MOV     #1$,$LPERR      ;CHANGE LOOP ON ERROR ADDRESS
3201  012314  004737  025606                JSR     PC,CLRP         ;CLEAR THE RP11
3202  012320  012737  007600  172356        MOV     #7600,KIPAR7    ;OPEN I/O REGISTERS
3203  012326  005037  172340                CLR     KIPAR0          ;ENABLE FIRST 4K
3204  012332  012737  000200  172342        MOV     #200,KIPAR1     ;ENABLE SECOND 4K
3205  012340  012737  002000  172344        MOV     #2000,KIPAR2    ;ENABLE FIRST TEST LOCATION
3206  012346  012737  077406  172300        MOV     #77406,KIPDR0   ;LOAD DESCRIPTOR REGISTER 0
3207  012354  012737  077406  172302        MOV     #77406,KIPDR1   ;LOAD DESCRIPTOR REGISTER 1
3208  012362  012737  077406  172304        MOV     #77406,KIPDR2   ;LOAD DESCRIPTOR REGISTER 2
3209  012370  012737  077406  172316        MOV     #77406,KIPDR7   ;LOAD DESCRIPTOR REGISTER 7
3210  012376  012737  000001  177572        MOV     #1,@#SR0        ;TURN ON MEMORY MANAGEMENT
3211  012404  005037  040000                CLR     @#40000         ;CLEAR LOCATION 200000
3212  012410  012737  177776  002506        MOV     #177776,$BUF    ;SETUP BUS ADDR
3213  012416  112737  000005  002502        MOVB    #RDSEK,DPB+2    ;CHANGE COMMAND TO READ
3214  012424  004737  022740                JSR     PC,RP11         ;START THE COMMAND
3215  012430  004737  023616                JSR     PC,CONRDY       ;WAIT FOR CONTROLLER READY
3216  012434  004737  023172                JSR     PC,SAVRP        ;STORE THE REGISTERS
3217  012440  032737  100000  001340        BIT     #ERR,$RPCS      ;ANY ERRORS?
3218  012446  001401                        BEQ     2$              ;BRANCH IF NOT
3219  012450  104050                        ERROR   50              ;ERROR AFTER READING 2 WORDS
3220  012452  032737  000020  001340  2$:   BIT     #MEX0,$RPCS     ;DID 'MEX0' SET ?
3221  012460  001001                        BNE     3$              ;BRANCH IF IT DID
3222  012462  104051                        ERROR   51              ;'MEX0' DID NOT SET AFTER 2 WORD READ
3223                                                                ;STARTING AT LOCATION 177776
3224  012464  032737  000040  001340  3$:   BIT     #MEX1,$RPCS     ;'MEX1' SHOULD NOT BE SET
3225  012472  001401                        BEQ     4$              ;BRANCH IF NOT SET
3226  012474  104052                        ERROR   52              ;'MEX1' IS SET - SHOULD NOT BE
3227  012476  022737  177777  040000  4$:   CMP     #177777,@#40000 ;WAS DATA READ INTO LOCATION
3228  012504  001407                        BEQ     TST12B          ;200000 CORRECTLY? - BRANCH IF YES
3229  012506  012737  177777  001124        MOV     #177777,$GDDAT  ;EXPECTED CONTENTS
3230  012514  013737  040000  001126        MOV     @#40000,$BDDAT  ;CONTENTS OF LOCATION 200000
3231  012522  104053                        ERROR   53              ;DATA COMPARE ERROR AT LOC 200000
3232  012524  023727  001232  000002  TST12B: CMP    MAXMEM+2,#2     ;ENOUGH MEMORY FOR THE NEXT PART OF THE TEST ?
3233  012532  103463                        BLO     EXIT12          ;BR IF NOT
3234  012534  012737  004000  172344        MOV     #4000,KIPAR2    ;CHANGE TEST LOCATION
3235  012542  012737  012542  001110  1$:   MOV     #1$,$LPERR      ;CHANGE THE LOOP ON ERROR ADDRESS
3236  012550  004737  025606                JSR     PC,CLRP         ;CLEAR THE RP11
3237  012554  005037  040000                CLR     @#40000         ;CLEAR LOCATION 400000
3238  012560  012737  177776  002506        MOV     #177776,$BUF    ;BUFFER ADDRESS
3239  012566  112737  000020  002501        MOVB    #MEX0,DPB+1     ;SETUP EXTENDED MEMORY ADDRESS BIT
3240  012574  112737  000005  002502        MOVB    #RDSEK,DPB+2    ;READ COMMAND
3241  012602  004737  022740                JSR     PC,RP11         ;START THE COMMAND
3242  012606  004737  023616                JSR     PC,CONRDY       ;WAIT FOR CONTROLLER READY
3243  012612  004737  023172                JSR     PC,SAVRP        ;STORE THE REGISTERS
3244  012616  032737  100000  001340        BIT     #ERR,$RPCS      ;ANY ERRORS?
3245  012624  001401                        BEQ     2$              ;BRANCH IF NOT
3246  012626  104054                        ERROR   54              ;ERROR WHILE READING TWO WORDS
3247  012630  032737  000020  001340  2$:   BIT     #MEX0,$RPCS     ;IS 'MEX0' SET ?
3248  012636  001401                        BEQ     3$              ;BRANCH IF NOT
3249  012640  104055                        ERROR   55              ;'MEX0' DID NOT CLEAR AFTER 2 WORD
3250                                                                ;READ STARTING AT LOCATION 377776
3251  012642  032737  000040  001340  3$:   BIT     #MEX1,$RPCS     ;DID 'MEX1' SET ?
3252  012650  001001                        BNE     4$              ;BR IF SET
3253  012652  104056                        ERROR   56              ;'MEX1' DID NOT SET AFTER 2 WORD TRANSFER
```

```
3254
3255  012654  022737  177777  040000  4$:   CMP   #177777,@#40000  ;STARTING AT LOCATION 377776
3256  012662  001407                  BEQ   EXIT12                ;WAS DATA READ INTO LOCATION 400000
3257  012664  012737  177777  001124        MOV   #177777,$GDDAT    ;CORRECTLY ? - BRANCH IF YES
3258  012672  013737  040000  001126        MOV   @#40000,$BDDAT    ;EXPECTED DATA
3259  012700  104057                  ERROR 57                    ;DATA FROM LOCATION 400000
3260  012702  005037  177572  EXIT12: CLR   @#SR0                 ;DATA COMPARE ERROR AT LOC 400000
3261  012706  000004                  SCOPE                       ;TURN OFF MEMORY MANAGEMENT
3262                                                              ;LOOP ?
3263
3264                          ;;*****************************************************************
3265                          ;*TEST 13       DATA BUFFER REGISTER BIT TEST
3266
3267                          ;*THE DATA BUFFER REGISTER TEST VERIFIES THAT ALL 36 BITS OF THE DATA
3268                          ;*BUFFER REGISTER CAN BE SET AND CLEARED.
3269                          ;*
3270                          ;*      1.  THE TEST FIRST WRITES THE FOLLOWING TEST PATTERN IN CYLINDER 0,
3271                          ;*      TRACK 0, SECTOR 0 USING 10/15 MODE ('MODE' BIT SET).
3272                          ;*
3273                          ;*                      052525
3274                          ;*                      052525
3275                          ;*                      000005
3276                          ;*                      125252
3277                          ;*                      125252
3278                          ;*                      000012
3279                          ;*                      052525
3280                          ;*                      052525
3281                          ;*                      000005
3282                          ;*                      125252
3283                          ;*                        .
3284                          ;*                        .
3285                          ;*                        .
3286                          ;*                      ETC
3287                          ;*
3288                          ;*      2.  THE TEST SECTOR IS THEN READ AND THE DATA IS COMPARED.  AT THE
3289                          ;*      COMPLETION OF THE TEST, THE PROGRAM RESTORES THE TEST SECTOR TO
3290                          ;*      PDP-11 MODE.
3291
3292                          ;;*****************************************************************
3293  012710
3294  012710  033737  001426  001276  TST13:  BIT   BITS+<$TN*2>,TSTNMS  ;IS THIS TEST SELECTED
3295  012716  001002                  BNE   .+6                   ;BR IF YES
3296  012720  000137  013564          JMP   TST14                 ;GO TO THE NEXT TEST
3297  012724  012737  013004  001110        MOV   #TEST13,$LPERR    ;SETUP THE ERROR LOOP ADDRESS
3298  012732  012737  012710  001106        MOV   #TST13,$LPADR     ;SETUP THE SCOPE LOOP ADDRESS
3299  012740  012737  000013  001102        MOV   #13,$TSTNM        ;SETUP TEST NUMBER AND
3300                                                              ;CLEAR THE ERROR FLAG ($ERFLG)
3301  012746  013777  001102  166166        MOV   $TSTNM,@DISPLAY   ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
3302  012754  012737  013562  001200        MOV   #EXIT13,BYPASS    ;SETUP BYPASS ADDRESS
3303  012762  013704  001324                MOV   RPADR,R4          ;RP11E BUS ADDRESS
3304  012766  105037  002501                CLRB  DPB+1             ;CLEAR ANY EXTENDED MEMORY BITS
3305  012772  105037  002503                CLRB  DPB+3             ;CLEAR 'MODE' & 'HDR' BITS
3306  012776  012737  000144  001162        MOV   #100.,$TIMES      ;;DO 100. ITERATIONS
3307  013004  012706  001100  TEST13: MOV   #STACK,SP              ;SETUP THE STACK POINTER
3308  013010  012701  043602                MOV   #BUFFER,R1        ;BUFFER ADDRESS
3309  013014  012702  000200                MOV   #128.,R2          ;SIZE
```

```
 3310  013020  012721  052525          1$:   MOV   #52525,(R1)+    ;PATTERN FOR BITS 20 - 35
 3311  013024  012721  052525                MOV   #52525,(R1)+    ;PATTERN FOR BITS 4 - 19
 3312  013030  012721  000005                MOV   #5,(R1)+        ;PATTERN FOR BITS 0 - 3
 3313  013034  005302                        DEC   R2              ;AT THE END ?
 3314  013036  001410                        BEQ   2$              ;BR IF END
 3315  013040  012721  125252                MOV   #125252,(R1)+   ;PATTERN FOR BITS 20 - 35
 3316  013044  012721  125252                MOV   #125252,(R1)+   ;PATTERN FOR BITS 4 - 19
 3317  013050  012721  000012                MOV   #12,(R1)+       ;PATTERN FOR BITS 0 - 3
 3318  013054  005302                        DEC   R2              ;AT THE END ?
 3319  013056  001360                        BNE   1$              ;BR IF NOT
 3320  013060  012737  013060  001110  2$:   MOV   #2$,$LPERR      ;LOOP ON ERROR ADDRESS
 3321  013066  004737  025606                JSR   PC,CLRP         ;CLEAR THE RP11
```

```
3322  013072  005037  002512              CLR     $SEC            ;SECTOR/TRACK ADDRESS OF ZERO
3323  013076  005037  002510              CLR     $CYL            ;CYLINDER ZERO
3324  013102  012737  000600  002504      MOV     #384.,$WC       ;1 SECTOR IN 10/15 MODE
3325  013110  012737  043602  002506      MOV     #BUFFER,$BUF    ;BUFFER ADDRESS
3326  013116  012737  010000  002502      MOV     #MODE,DPB+2     ;SET THE 'MODE' BIT
3327  013124  112737  000003  002502      MOVB    #WRTSEK,DPB+2   ;WRITE COMMAND
3328  013132  004737  022740              JSR     PC,RP11         ;DO THE COMMAND
3329  013136  004737  023616              JSR     PC,CONRDY       ;WAIT FOR THE COMMAND TO COMPLETE
3330  013142  004737  023172              JSR     PC,SAVRP        ;SAVE THE REGISTERS
3331  013146  032737  100000  001340      BIT     #ERR,$RPCS      ;OPERATION GO OK ?
3332  013154  001402                      BEQ     3$              ;BR IF IT DID
3333  013156  104004                      ERROR   4               ;SETUP OPERATION TERMINATED WITH AN ERROR
3334  013160  000541                      BR      TST13A          ;CORRECT THE SECTOR AND BYPASS REST OF TEST
3335  013162  012737  013162  001110  3$: MOV     #3$,$LPERR      ;CHANGE THE LOOP ON ERROR ADDRESS
3336  013170  004737  025606              JSR     PC,CLRP         ;CLEAR THE RP11
3337  013174  012701  043602              MOV     #BUFFER,R1      ;CLEAR THE INPUT BUFFER
3338  013200  063701  002504              ADD     $WC,R1          ;BEGINNING ADDRESS
3339  013204  013702  002504              MOV     $WC,R2          ;SIZE
3340  013210  005001                  4$: CLR     R1              ;CLEAR THE BUFFER
3341  013212  005302                      DEC     R2              ;FINISHED ?
3342  013214  001375                      BNE     4$              ;BR IF NOT
3343  013216  013737  002504  002506      MOV     $WC,$BUF        ;CHANGE BUFFER ADDRESS
3344  013224  063737  002504  002506      ADD     $WC,$BUF        ;DOUBLE THE WORD COUNT
3345  013232  062737  043602  002506      ADD     #BUFFER,$BUF    ;ADD BUFFER BASE ADDRESS
3346  013240  112737  000005  002502      MOVB    #RDSEK,DPB+2    ;CHANGE COMMAND
3347  013246  004737  022740              JSR     PC,RP11         ;READ THE SECTOR
3348  013252  004737  023616              JSR     PC,CONRDY       ;WAIT FOR THE OPERATION TO COMPLETE
3349  013256  004737  023172              JSR     PC,SAVRP        ;STORE THE REGISTERS
3350  013262  032737  100000  001340      BIT     #ERR,$RPCS      ;DID AN ERROR OCCUR ?
3351  013270  001402                      BEQ     5$              ;BR IF NOT
3352  013272  104017                      ERROR   17              ;ERROR OCCURED READING THE TEST SECTOR
3353  013274  000473                      BR      TST13A          ;BYPASS THE COMPARSION
3354  013276  012701  043602          5$: MOV     #BUFFER,R1      ;GOOD DATA
3355  013302  013702  002504              MOV     $WC,R2          ;SIZE
3356  013306  013703  002504              MOV     $WC,R3          ;FORM ADDRESS OF TEST DATA
3357  013312  063703  002504              ADD     $WC,R3          ;DOUBLE THE WORD COUNT
3358  013316  062703  043602              ADD     #BUFFER,R3      ;ADD THE BUFFER START ADDRESS
3359  013322  005037  013462              CLR     10$             ;CLEAR THE COMPARSION ERROR COUNTER
3360  013326  032777  000010  165604  6$: BIT     #SW03,@SWR      ;COMPARE THE DATA ?
3361  013334  001053                      BNE     TST13A          ;BR IF NOT
3362  013336  022123                      CMP     (R1)+,(R3)+     ;COMPARE THE DATA
3363  013340  001445                      BEQ     9$              ;BR IF OK
3364  013342  016137  177776  001124      MOV     -2(R1),$GDDAT   ;GOOD DATA
3365  013350  016337  177776  001126      MOV     -2(R3),$BDDAT   ;BAD DATA
3366  013356  010337  001122              MOV     R3,$BDADR       ;FORM BAD ADDRESS
3367  013362  162737  000002  001122      SUB     #2,$BDADR       ;DECREMENT IT
3368  013370  005737  013462              TST     10$             ;ANY COMPARE ERROR ALREADY ?
3369  013374  001010                      BNE     7$              ;BR IF SOME
3370  013376  012737  013376  001116      MOV     #.,$ERRPC       ;PC FOR ERROR MESSAGE
3371  013404  112737  000102  001114      MOVB    #102,$ITEMB     ;ERROR TABLE INDEX
3372  013412  004737  017232              JSR     PC,TYPERR       ;REPORT COMPARSION ERROR
3373  013416  112737  000104  001114  7$: MOVB    #104,$ITEMB     ;INDEX OF DATA DISPLAY MESSAGE
3374  013424  004737  017232              JSR     PC,TYPERR       ;TYPE THE MESSAGE
3375  013430  005237  013462          8$: INC     10$             ;INCREMENT COMPARSION ERROR COUNT
3376  013434  023737  001320  013462      CMP     CMP.CT,10$      ;COMPARE AGAINST THE LIMIT
3377  013442  103004                      BHIS    9$              ;BR IF NOT GREATER
```

```
3378  013444  032777  000020  165466          BIT     #SWO4,@SWR      ;CONTINUE COMPARING ANYWAY ?
3379  013452  001404                          BEQ     TST13A          ;BR IF NOT
3380  013454  005302                  9$:     DEC     R2              ;FINISHED COMPARING ?
3381  013456  001323                          BNE     6$              ;BR IF NOT
3382  013460  000401                          BR      TST13A          ;CHANGE MODE OF THE SECTOR
3383  013462  000000                  10$:    .WORD   0               ;COMPARSION ERROR COUNTER
3384  013464  012737  013464  001110  TST13A: MOV     #TST13A,$LPERR  ;CHANGE LOOP ON ERROR ADDRESS
3385  013472  004737  025606                  JSR     PC,CLRP         ;CLEAR THE RP11
3386  013476  105037  002503                  CLRB    DPB+3           ;CLEAR THE 'MODE' & 'HDR' BITS
3387  013502  012737  000002  002504          MOV     #2,$WC          ;CHANGE WORD COUNT
3388  013510  112737  000003  002502          MOVB    #WRTSEK,DPB+2   ;CHANGE COMMAND
3389  013516  005037  043602                  CLR     BUFFER          ;SET DATA TO ZERO
3390  013522  005037  043604                  CLR     BUFFER+2        ;OTHER DATA WORD
3391  013526  012737  043602  002506          MOV     #BUFFER,$BUF    ;BUFFER ADDRESS
3392  013534  004737  022740                  JSR     PC,RP11         ;RE-WRITE THE SECTOR
3393  013540  004737  023616                  JSR     PC,CONRDY       ;WAIT FOR THE OPERATION TO COMPLETE
3394  013544  004737  023172                  JSR     PC,SAVRP        ;STORE THE REGISTERS
3395  013550  032737  100000  001340          BIT     #ERR,$RPCS      ;DID THE WRITE FINISH WITHOUT ERROR ?
3396  013556  001401                          BEQ     EXIT13          ;BR IF IT DID
3397  013560  104060                          ERROR   60              ;ERROR RE-WRITING THE SECTOR
3398  013562  000004          EXIT13: SCOPE                           ;LOOP ?
3399
3400                          ;;*****************************************************************
3401                          ;#TEST 14       DATA STORAGE/RETRIEVAL TEST
3402
3403                          ;#THE DATA TEST PERFORMS DATA STORAGE AND RETRIEVAL TESTING USING THE CONTROLLER
3404                          ;#AND THE CURRENTLY SELECTED DRIVE.  UNLESS ALTERED BY THE OPERATOR, THE
3405                          ;#TEST WILL WRITE AND WRITE CHECK THE ENTIRE DISK PACK USING ALL 16 DATA
3406                          ;#PATTERNS IN A ROTATING MANNER.  THE BUFFER SIZE USED FOR THE WRITE AND
3407                          ;#WRITE CHECK ORDERS IS DETERMINED BY THE AVAILABLE MEMORY ON THE SYSTEM
3408                          ;#(MINUS THE SPACE REQUIRED BY THE PROGRAM LOADER).  THE PROGRAM WILL USE
3409                          ;#THE AVAILABLE MEMORY SIZE OR 8K, WHICHEVER IS LESS, AS THE BUFFER
3410                          ;#SIZE.
3411                          ;#
3412                          ;#AFTER THE PACK HAS BEEN WRITTEN, THE TEST WILL READ AND COMPARE THE
3413                          ;#DATA.  THE BUFFER SIZE USED FOR THE READ IS THE SAME SIZE AS THAT USED
3414                          ;#FOR THE WRITE/WRITE CHECK PART OF THE TEST.  FOR THE READ, THE TEST WILL
3415                          ;#ROTATE THE READ BUFFER THROUGH MEMORY SO THAT ALL OF THE AVAILABLE MEMORY
3416                          ;#IS USED BY THE READ AND COMPARE OPERATIONS.  THE TEST USES MEMORY MANAGEMENT
3417                          ;#ON SYSTEMS WITH MORE THAN 28K.  (MEMORY MANAGEMENT WILL NOT BE USED IF
3418                          ;#ON MEMORY MANAGEMENT SYSTEMS WITH 28K OR LESS.)
3419                          ;#
3420                          ;#THE DATA PATTERNS USED BY THE DATA TEST ARE GIVEN BELOW:
3421                          ;#
3422                          ;#      PAT 0   PAT 1   PAT 2   PAT 3   PAT 4   PAT 5   PAT 6   PAT 7
3423                          ;#      ------  ------  ------  ------  ------  ------  ------  ------
3424                          ;#      000000  000001  177776  000000  000000  052525  007417  026455
3425                          ;#      000000  000003  177774  000000  010421  052525  007417  026455
3426                          ;#      000000  000007  177770  000000  021042  052525  007417  026455
3427                          ;#      000000  000017  177760  177777  031463  125252  170360  151322
3428                          ;#      000000  000037  177740  177777  042104  125252  170360  151322
3429                          ;#      000000  000077  177700  177777  052525  125252  170360  151322
3430                          ;#      000000  000177  177600  000000  063146  052525  007417  026455
3431                          ;#      000000  000377  177400  000000  073567  052525  007417  026455
3432                          ;#      000000  000777  177000  177777  104210  125252  170360  151322
3433                          ;#      000000  001777  176000  177777  114631  125252  170360  151322
```

```
3434                                    ;*      000000  003777  174000  000000  125252  052525  007417  026455
3435                                    ;*      000000  007777  170000  177777  135673  125252  170360  151322
3436                                    ;*      000000  017777  160000  000000  146314  052525  007417  026455
3437                                    ;*      000000  037777  140000  177777  156735  125252  170360  151322
3438                                    ;*      000000  077777  100000  177777  167356  125252  170360  151322
3439                                    ;*      000000  177777  000000  000000  177777  052525  007417  026455
3440                                    ;*
3441                                    ;*      PAT 8   PAT 9   PAT 10  PAT 11  PAT 12  PAT 13  PAT 14  PAT 15
3442                                    ;*      ------  ------  ------  ------  ------  ------  ------  ------
3443                                    ;*      077577  000001  177776  172666  077777  177777  000000  177777
3444                                    ;*      077577  000002  177775  155555  137777  177777  000000  177777
3445                                    ;*      077577  000004  177773  172666  157777  177777  177777  000000
3446                                    ;*      077577  000010  177767  155555  167777  177777  177777  000000
3447                                    ;*      077577  000020  177757  172666  173777  177777  177777  000000
3448                                    ;*      077577  000040  177737  155555  175777  177777  177777  000000
3449                                    ;*      077577  000100  177677  172666  176777  177777  177777  000000
3450                                    ;*      077577  000200  177577  155555  177377  177777  177777  000000
3451                                    ;*      077577  000400  177377  172666  177577  177777  177777  000000
3452                                    ;*      077577  001000  176777  155555  177677  177777  177777  000000
3453                                    ;*      077577  002000  175777  172666  177737  177777  177777  000000
3454                                    ;*      077577  004000  173777  155555  177757  177777  177777  000000
3455                                    ;*      077577  010000  167777  172666  177767  177777  177777  000000
3456                                    ;*      077577  020000  157777  155555  177773  177777  177777  000000
3457                                    ;*      077577  040000  137777  172666  177775  177777  177777  000000
3458                                    ;*      077577  100000  077777  155555  177776  177777  177777  000000
3459                                    ;*
3460
3461                                    ;;######################################################################
3462  013564                            TST14:
3463  013564  033737  001430  001276            BIT     BITS+<$TN*2>,TSTNMS  ;IS THIS TEST SELECTED
3464  013572  001002                            BNE     .+6                  ;BR IF YES
3465  013574  000137  014620                    JMP     TST15                ;GO TO THE NEXT TEST
3466  013600  012737  014056  001110            MOV     #TEST14,$LPERR       ;SETUP THE ERROR LOOP ADDRESS
3467  013606  012737  013564  001106            MOV     #TST14,$LPADR        ;SETUP THE SCOPE LOOP ADDRESS
3468  013614  012737  000014  001102            MOV     #14,$TSTNM           ;SETUP TEST NUMBER AND
3469                                            ;CLEAR THE ERROR FLAG (SERFLG)
3470  013622  013777  001102  165312            MOV     $TSTNM,@DISPLAY      ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
3471  013630  012737  014606  001200            MOV     #EXIT14,BYPASS       ;SETUP BYPASS ADDRESS
3472  013636  013704  001324                    MOV     RPADR,R4             ;RP11E BUS ADDRESS
3473  013642  105037  002501                    CLRB    DPB+1                ;CLEAR ANY EXTENDED MEMORY BITS
3474  013646  105037  002503                    CLRB    DPB+3                ;CLEAR 'MODE' & 'MDR' BITS
3475  013652  012706  001100                    MOV     #STACK,SP            ;SETUP THE STACK POINTER
3476  013656  032777  000100  165254            BIT     #SW06,@SWR           ;USE MEMORY MANAGEMENT ?
3477  013664  001021                            BNE     2$                   ;BR IF NOT
3478  013666  005737  022202            1$:     TST     $KT11                ;MEMORY MANAGEMENT AVAILABLE ?
3479  013672  100016                            BPL     2$                   ;BR IF NOT
3480  013674  012737  177777  001222            MOV     #-1,MMACTV           ;SET MEMORY MANAGEMENT ACTIVE INDICATOR
3481  013702  004737  023300                    JSR     PC,MOVLDR            ;MOVE THE LOADER
3482  013706  012637  001240                    MOV     (SP)+,LOMEM          ;BEGINNING ADDRESS OF BUFFER SPACE
3483  013712  013737  001230  001234            MOV     MAXMEM,HIMEM         ;HIGHEST MEMORY ADDRESS USED BY THE TEST
3484  013720  013737  001232  001236            MOV     MAXMEM+2,HIMEM+2     ;UPPER ADDRESS BITS
3485  013726  000412                            BR      3$                   ;CONTINUE
3486  013730  005037  001222            2$:     CLR     MMACTV               ;CLEAR MEMORY MANAGEMENT ACTIVE INDICATOR
3487  013734  005037  001236                    CLR     HIMEM+2              ;CLEAR UPPER 'HIMEM' ADDRESS BITS
3488  013740  013737  001302  001234            MOV     MEMSIZ,HIMEM         ;HIGHEST NON MEMORY MANAGEMENT LOCATION
3489  013746  012737  043602  001240            MOV     #BUFFER,LOMEM        ;DEFAULT BUFFER STARTING ADDRESS
```

```
3490  013754  005037  001242            3$:     CLR     LOMEM+2         ;CLEAR UPPER STORAGE WORD
3491  013760  013737  001240  001224            MOV     LOMEM,ACTMEM    ;STARTING ADDRESS OF FIRST BUFFER
3492  013766  013737  001242  001226            MOV     LOMEM+2,ACTMEM+2  ;UPPER ADDRESS BITS
3493  013774  013737  001256  001254            MOV     WC14,BLKS14     ;CONVERT THE WORD COUNT TO A SECTOR COUNT
3494  014002  000337  001254                    SWAB    BLKS14          ;RIGHT JUSTIFY THE SECTOR COUNT
3495  014006  105037  001255                    CLRB    BLKS14+1        ;CLEAR THE RESIDUE
3496  014012  005237  001254                    INC     BLKS14          ;CHANGE TO ABSOLUTE COUNT
3497  014016  013746  001254                    MOV     BLKS14,-(SP)    ;CONVERT BLOCK COUNT TO SECTOR/TRACK INCREMENTS
3498  014022  005037  001252                    CLR     INCTRK          ;CLEAR TRACK STORAGE
3499  014026  005037  001250                    CLR     INCSEC          ;CLEAR SECTOR STORAGE
3500  014032  162716  000012            4$:     SUB     #10.,(SP)       ;SUBTRACT A FULL SECTOR
3501  014036  100403                    BMI     5$                      ;BR IF MINUS
3502  014040  005237  001252                    INC     INCTRK          ;COUNT THE TRACK
3503  014044  000772                    BR      4$                      ;CONTINUE
3504  014046  062716  000012            5$:     ADD     #10.,(SP)       ;CORRECT THE SECTOR RESIDUE
3505  014052  012637  001250                    MOV     (SP)+,INCSEC    ;SECTOR INCREMENT VALUE
3506  014056  012706  001100            TEST14: MOV     #STACK,SP       ;LOAD THE STACK POINTER
3507  014062  004737  023734                    JSR     PC,SECNBR       ;GET THE NUMBER OF SECTORS ON THE PACK
3508                                                                    ;BEING USED
3509  014066  113737  001272  002512    1$:     MOVB    SSEC,$SEC       ;STARTING SECTOR ADDRESS
3510  014074  113737  001270  002513            MOVB    STRK,$TRK       ;STARTING TRACK ADDRESS
3511  014102  013737  001266  002510            MOV     SCYL,$CYL       ;STARTING CYLINDER ADDRESS
3512  014110  013737  001240  002506            MOV     LOMEM,$BUF      ;BUFFER ADDRESS
3513  014116  013737  001256  002504            MOV     WC14,$WC        ;STARTING WORD COUNT
3514  014124  012737  000017  001204            MOV     #15.,PATNUM     ;PRESET THE PATTERN INDEX
3515  014132  005737  001260                    TST     OPRSEL          ;USING AN OPERATOR SPECIFIED ADDRESS ?
3516  014136  001005                    BNE     2$                      ;BR IF YES
3517  014140  163737  001254  001244            SUB     BLKS14,PAKSIZ   ;DECREMENT THE SECTOR COUNT FOR THE
3518  014146  005637  001246                    SBC     PAKSIZ+2        ;FIRST OPERATION
3519  014152  004737  024002            2$:     JSR     PC,PATSEL       ;SELECT THE PATTERN
3520  014156  012737  014156  001110    3$:     MOV     #3$,$LPERR      ;LOOP ON ERROR ADDRESS
3521  014164  004737  025606                    JSR     PC,CLRP         ;CLEAR THE RP11
3522  014170  017746  164744                    MOV     @$WR,-(SP)      ;INHIBIT WRITE AND
3523  014174  005116                    COM     (SP)                    ;WRITE CHECK ?
3524  014176  032726  000003                    BIT     #$WDO!$WD1,(SP)+
3525  014202  001453                    BEQ     TST14A                  ;YES--BRANCH
3526  014204  004737  024364                    JSR     PC,SETBUF       ;MOVE DATA PATTERN INTO THE BUFFER
3527  014210  032777  000001  164722            BIT     #$WDO,@$WR      ;INHIBIT WRITE ?
3528  014216  001016                    BNE     4$                      ;YES--BRANCH
3529  014220  112737  000003  002502            MOVB    #WRTSEK,DPB+2   ;COMMAND=WRITE
3530  014226  004737  022740                    JSR     PC,RP11         ;DO THE COMMAND
3531  014232  004737  023616                    JSR     PC,CONRDY       ;WAIT FOR THE COMMAND TO COMPLETE
3532  014236  004737  023172                    JSR     PC,SAVRP        ;STORE THE REGISTERS
3533  014242  032737  100000  001340            BIT     #ERR,$RPCS      ;ERROR OCCURED ?
3534  014250  001401                    BEQ     4$                      ;BR IF NOT
3535  014252  104061                    ERROR   61                      ;REPORT THE ERROR
3536  014254  032777  000002  164656    4$:     BIT     #$WD1,@$WR      ;INHIBIT WRITE CHECK ?
3537  014262  001020                    BNE     5$                      ;YES--BRANCH
3538  014264  004737  025606                    JSR     PC,CLRP         ;CLEAR THE RP11
3539  014270  112737  000007  002502            MOVB    #WRTCHK,DPB+2   ;COMMAND=WRITE CHECK
3540  014276  004737  022740                    JSR     PC,RP11         ;DO THE WRITE CHECK
3541  014302  004737  023616                    JSR     PC,CONRDY       ;WAIT FOR THE COMMAND TO FINISH
3542  014306  004737  023172                    JSR     PC,SAVRP        ;STORE THE REGISTERS
3543  014312  032737  100000  001340            BIT     #ERR,$RPCS      ;ERROR ?
3544  014320  001401                    BEQ     5$                      ;BR IF NOT
3545  014322  104005                    ERROR   5                       ;REPORT THE WRITE CHECK ERROR
```

```
3546  014324  004737  024046          5$:     JSR     PC,INCADR       ;INCREMENT THE ADDRESS
3547  014330  000710                  BR      2$                      ;CONTINUE
3548  014332  004737  023734          TST14A: JSR     PC,SECNBR       ;GET NUMBER OF SECTORS ON THE DRIVE
3549                                                                  ;BEING TESTED
3550  014336  113737  001272  002512          MOVB    SSEC,$SEC       ;STARTING SECTOR
3551  014344  113737  001270  002513          MOVB    STRK,$TRK       ;STARTING TRACK
3552  014352  013737  001266  002510          MOV     SCYL,$CYL       ;STARTING CYLINDER
3553  014360  013737  001256  002504          MOV     WC14,$WC        ;INITIAL WORD COUNT
3554  014366  012737  000017  001204          MOV     #15.,PATNUM     ;PRESET PATTERN NUMBER
3555  014374  005737  001260                  TST     OPRSEL          ;USING AN OPERATOR SPECIFIED PATTERN
3556  014400  001005                          BNE     1$              ;BR IF YES
3557  014402  163737  001254  001244          SUB     BLKS14,PAKSIZ   ;DECREMENT THE SECTOR COUNT FOR
3560  014410  005637  001246                  SBC     PAKSIZ+2        ;THE FIRST OPERATION
3561  014414  005037  001314          1$:     CLR     RETRY           ;CLEAR THE RETRY COUNTER
3562  014420  004737  024002                  JSR     PC,PATSEL       ;SELECT A PATTERN
3563  014424  004737  024434                  JSR     PC,GETBUF       ;GENERATE A BUFFER ADDRESS
3564  014430  012737  014430  001110  2$:     MOV     #2$,$LPERR      ;LOOP ON ERROR ADDRESS
3565  014436  004737  025606                  JSR     PC,CLRP         ;CLEAR THE RP11
3566  014442  032777  000004  164470          BIT     #SW02,@SWR      ;INHIBIT READ DATA AND SOFTWARE COMPARE ?
3567  014450  001056                          BNE     EXIT14          ;YES--BRANCH
3570  014452  112737  000005  002502          MOVB    #RDSEK,DPB+2    ;LOAD READ COMMAND
3571  014460  004737  022740                  JSR     PC,RP11         ;START THE READ
3572  014464  004737  023616                  JSR     PC,CONRDY       ;WAIT FOR THE READ TO COMPLETE
3573  014470  004737  023172                  JSR     PC,SAVRP        ;STORE THE REGISTERS
3574  014474  032737  100000  001340          BIT     #ERR,$RPCS      ;ERROR ?
3575  014502  001420                          BEQ     5$              ;BR IF NOT
3576  014504  005737  001314                  TST     RETRY           ;RETRY ATTEMPT ?
3577  014510  001404                          BEQ     3$              ;BR IF NOT
3600  014512  032777  000040  164420          BIT     #SW05,@SWR      ;DISPLAY ALL RETRY ATTEMPTS
3601  014520  001401                          BEQ     4$              ;BR IF NOT
3602  014522  104062                  3$:     ERROR   62              ;REPORT THE ERROR
3603  014524  005237  001314          4$:     INC     RETRY           ;INCREMENT THE RETRY COUNT
3604  014530  023737  001314  001316          CMP     RETRY,LRETRY    ;RETRY LIMIT ?
3605  014536  101734                          BLOS    2$              ;BR IF NOT
3606  014540  104110                          ERROR   110             ;REPORT UNSUCESSFUL RETRY
3607  014542  000410                          BR      6$              ;CONTINUE
3610  014544  005737  001314          5$:     TST     RETRY           ;COMING FROM A SUCESSFUL RETRY ?
3611  014550  001405                          BEQ     6$              ;BR IF NOT
3612  014552  112737  000111  001114          MOVB    #111,$ITEMB     ;RETRY COUNT MESSAGE NUMBER
3613  014560  004737  017232                  JSR     PC,TYPERR       ;REPORT THE RETRY COUNT
3614  014564  032777  000010  164346  6$:     BIT     #SW03,@SWR      ;COMPARE THE DATA ?
3615  014572  001002                          BNE     7$              ;NO--BRANCH
3616  014574  004737  024546                  JSR     PC,DATCMP       ;YES--DO IT
3617  014600  004737  024046          7$:     JSR     PC,INCADR       ;INCREMENT THE DISK ADDRESS
3620  014604  000703                          BR      1$              ;CONTINUE
3621  014606  000004                  EXIT14: SCOPE                   ;LOOP ?
3622  014610  004737  023404                  JSR     PC,RESLDR       ;RESTORE THE LOADER IF IT WAS MOVED
3623  014614  000137  016606                  JMP     SEOP            ;GO TO END OF TEST
3624
3625                          ;;#################################################################
3626                          ;#TEST 15       POWER FAIL TEST
3627
3630                          ;#TEST THE ABILITY OF THE RP11E TO SENSE POWER FAILURE AND FOR THE DRIVES
3631                          ;#TO RETRACT THEIR HEADS.  WHEN POWER IS TURNED ON AGAIN
3632                          ;#THE CYLINDER ADDRESS IS TESTED FOR ZERO.  AFTER TYPING THE MESSAGE
3633                          ;#REQUESTING POWER TO BE TURNED OFF, THE PROGRAM GOES INTO A LOOP
```

```
                                        ;*READING FROM THE SELECTED DISK DRIVE.  AFTER POWER IS RESTORED,
                                        ;*MEMORY IS CHECKED TO SEE THAT THE DISK DID NOT TRANSFER ANY DATA
                                        ;*TO MEMORY WHILE POWER WAS GOING DOWN.

                                        ;;*********************************************************************
014620                                  TST15:
014620 033737 001432 001276                     BIT     BITS+<$TN*2>,TSTNMS  ;IS THIS TEST SELECTED
014626 001002                                   BNE     +6                   ;BR IF YES
014630 000137 015362                            JMP     @#TST16              ;GO TO THE NEXT TEST
014634 012737 014726 001110                     MOV     #TEST15,$LPERR       ;SETUP THE ERROR LOOP ADDRESS
014642 012737 014620 001106                     MOV     #TST15,$LPADR        ;SETUP THE SCOPE LOOP ADDRESS
014650 012737 000015 001102                     MOV     #15,$TSTNM           ;SETUP TEST NUMBER AND
                                                                             ;CLEAR THE ERROR FLAG ($ERFLG)
014656 013777 001102 164256                     MOV     $TSTNM,@#DISPLAY     ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
014664 012737 015272 001200                     MOV     #EXIT1$,BYPASS       ;SETUP BYPASS ADDRESS
014672 013704 001324                            MOV     RPADR,R4             ;RP11E BUS ADDRESS
014676 105037 002501                            CLRB    DPB+1                ;CLEAR ANY EXTENDED MEMORY BITS
014702 105037 002503                            CLRB    DPB+3                ;CLEAR 'MODE' & 'HDR' BITS
014706 104401 027357                            TYPE    ,MSG15A              ;'POWER FAIL TEST'
014712 004737 023300                            JSR     PC,MOVLDR            ;MOVE THE LOADER
014716 012637 001240                            MOV     (SP)+,LOMEM          ;BUFFER STARTING ADDRESS
014722 005037 001242                            CLR     LOMEM+2              ;CLEAR THE UPPER BITS JUST TO BE NICE
014726 012706 001100                    TEST15: MOV     #STACK,SP            ;SETUP THE STACK POINTER
014732 004737 025606                            JSR     PC,CLRP              ;CLEAR THE RP11
014736 012737 015304 000024                     MOV     #TST15B,@#24         ;SET UP POWER FAIL VECTOR
014744 012737 000340 000026                     MOV     #PR7,@#26            ;LOCKOUT INTERRUPTS
014752 013700 001240                            MOV     LOMEM,R0             ;START OF FREE MEMORY SPACE
014756 012720 125252                    1$:     MOV     #125252,(R0)+        ;FILL MEMORY WITH CHECKERBOARD
014762 020037 001302                            CMP     R0,MEMSIZ            ;PATTERN
014766 101773                                   BLOS    1$                   ;BR IF NOT FINISHED
014770 013737 001210 002510                     MOV     MAXCYL,$CYL          ;LOAD MAXIMUM CYLINDER
014776 005037 002512                            CLR     $SEC                 ;CLEAR SECTOR/TRACK
015002 012737 000256 002504                     MOV     #256,$WC             ;WORD COUNT = 1 SECTOR
015010 013737 001240 002506                     MOV     LOMEM,$BUF           ;BUFFER ADDRESS
015016 112737 000003 002502                     MOVB    #WRTSEK,DPB+2        ;WRITE OPERATION
015024 004737 022740                            JSR     PC,RP11              ;START THE COMMAND
015030 004737 023616                            JSR     PC,CONRDY            ;WAIT FOR THE CONTROLLER TO BE READY
015034 004737 023172                            JSR     PC,SAVRP             ;STORE THE REGISTERS
015040 032737 100000 001340                     BIT     #ERR,$RPCS           ;ERROR DURING OPERATION ?
015046 001401                                   BEQ     2$                   ;BR IF NOT
015050 104004                                   ERROR   4                    ;REPORT THE ERROR
015052 104401 027402                    2$:     TYPE    ,MSG15B              ;'TURN CPU POWER OFF AFTER MESSAGE'
015056 004737 025606                    3$:     JSR     PC,CLRP              ;CLEAR THE RP11
015062 112737 000005 002502                     MOVB    #RDSEK,DPB+2         ;READ COMMAND
015070 004737 022740                            JSR     PC,RP11              ;INITIATE THE READ
015074 004737 023616                            JSR     PC,CONRDY            ;WAIT FOR CONTROLLER READY
015100 000766                                   BR      3$                   ;LOOP, READING THE DISK

                                        ;AFTER MACHINE IS POWERED DOWN AND UP, CONTROL IS TRANSFERRED HERE.

015102 004737 022536                    TST15A: JSR     PC,RPINIT            ;INITIALZIE THE RP11E
015106 104401 027576                            TYPE    ,MSG15D              ;'PROGRAM WILL LOOP, WAITING FOR DRIVE TO COME ONLINE'
015112 012737 015112 001110             1$:     MOV     #1$,$LPERR           ;SETUP LOOP ON ERROR ADDRESS
015120 012737 002500 001220                     MOV     #2500,STALLT         ;LOAD STALL VALUE
015126 004737 023134                            JSR     PC,STALL             ;STALL FOR 2500 MILLISECONDS
015132 013701 001176                            MOV     CHKDRV,R1            ;DRIVE ADDRESS
```

```
3658  015136  004737  022636                  JSR     PC,DRVINT       ;CHECK THE DRIVE'S STATUS
3659  015142  105761  001370                  TSTB    DRVSTA(R1)      ;IS THE DRIVE ONLINE ?
3660  015146  001761                          BEQ     1$              ;CONTINUE WAITING FOR DRIVE TO COME ONLINE
3661  015150  003002                          BGT     2$              ;BR IF IT IS ONLINE
3662  015152  104112                          ERROR   112             ;DRIVE UNSAFE AFTER POWER FAIL
3663  015154  000446                          BR      EXIT15          ;DRIVE UNSAFE, EXIT
3664  015156  113764  001176  000005  2$:     MOVB    CHKDRV,RPCS+1(R4)  ;SELECT THE DRIVE
3665  015164  004737  023172                  JSR     PC,SAVRP        ;STORE THE REGISTERS
3666  015170  005737  001354                  TST     $SUCA           ;SELECTED CYLINDER SHOULD BE ZERO
3667  015174  001404                          BEQ     3$              ;BR IF IT DID
3668  015176  012737  015206  001164          MOV     #3$,$ESCAPE     ;ESCAPE TO 3$ ON ERROR
3669  015204  104063                          ERROR   63              ;DRIVE DID NOT RETURN TO CYLINDER 0
3670                                                                  ;ON POWER FAILURE
3671  015206                          3$:
3672  015206  012737  015266  001164          MOV     #6$,$ESCAPE     ;ESCAPE TO 6$ ON ERROR
3673  015214  013700  001240                  MOV     LOMEM,R0        ;BUFFER ADDRESS
3674  015220  022720  125252          4$:     CMP     #125252,(R0)+   ;DID MEMORY RETAIN PATTERN ON POWER FAIL ?
3675  015224  001004                          BNE     5$              ;BRANCH IF NOT
3676  015226  020037  001302                  CMP     R0,MEMSIZ       ;END OF MEMORY ?
3677  015232  001372                          BNE     4$              ;BR IF NOT
3678  015234  000414                          BR      6$              ;EXIT
3679  015236  012737  125252  001124  5$:     MOV     #125252,$GDDAT  ;EXPECTED MEMORY CONTENTS
3680  015244  016037  177776  001126          MOV     -2(R0),$BDDAT   ;ACTUAL MEMORY CONTENTS
3681  015252  010037  001122                  MOV     R0,$BDADR       ;ADDRESS
3682  015256  162737  000002  001122          SUB     #2,$BDADR       ;CORRECT IT
3683  015264  104064                          ERROR   64              ;CONTENTS OF MEMORY CHANGED AFTER POWER FAIL
3684  015266  104401  027543          6$:     TYPE    ,MSG15C         ;'END OF POWER FAIL TEST'
3685  015272  000004                  EXIT15: SCOPE                   ;LOOP ?
3686  015274  004737  023404                  JSR     PC,RESLDR       ;RESTORE THE LOADER
3687  015300  000137  003676                  JMP     START2          ;RETURN TO 'CONVERSATION MODE' ENTRANCE
3688
3689                                  ;POWER FAIL TRAP HANDLER
3690
3691  015304  012737  015314  000024  TST15B: MOV     #TST15C,@#24    ;SET UP POWER UP VECTOR
3692  015312  000774                          BR      TST15B          ;LOOP
3693
3694                                  ;POWER UP TRAP HANDLER
3695
3696  015314  012737  000026  000024  TST15C: MOV     #26,@#24        ;RESTORE TRAP CATCHER
3697  015322  005037  000026                  CLR     @#26            ;CATCHER HALT
3698  015326  012706  001100                  MOV     #STACK,SP       ;SETUP STACK POINTER
3699  015332  005037  177776                  CLR     PS              ;RETURN TO PRIORITY ZERO
3700  015336  012737  011610  001220          MOV     #5000.,STALLT   ;LOAD STALL VALUE
3701  015344  004737  023134                  JSR     PC,STALL        ;STALL FOR 5000. MILLISECONDS
3702  015350  004737  020374                  JSR     PC,$TKINT       ;INITIALIZE THE TTY KEYBOARD
3703  015354  013704  001324                  MOV     RPADR,R4        ;RP11E ADDRESS
3704  015360  000650                          BR      TST15A          ;VERIFY POWER DOWN SEQUENCE
3705
3706
3707                                  ;;*****************************************************************
3708                                  ;*TEST 16        HEAD ALIGNMENT ROUTINE
3709
3710                                  ;*THIS ROUTINE PERFORMS HEAD SELECTION AS DIRECTED FROM THE KEYBOARD.
3711                                  ;*THE ALIGNMENT OF THE SELECTED HEAD MAY BE CHECKED OR ADJUSTED.
3712
3713                                  ;;*****************************************************************
```

```
015362                                  TST16:
015362  033737  001434  001276                  BIT     BITS+<$TN*2>,TSTNMS     ;IS THIS TEST SELECTED
015370  001002                                  BNE     .+6                     ;BR IF YES
015372  000137  015614                          JMP     TST17                   ;GO TO THE NEXT TEST
015376  012737  015450  001110                  MOV     #TEST16,$LPERR          ;SETUP THE ERROR LOOP ADDRESS
015404  012737  015362  001106                  MOV     #TST16,$LPADR           ;SETUP THE SCOPE LOOP ADDRESS
015412  012737  000016  001102                  MOV     #16,$TSTNM              ;SETUP TEST NUMBER AND
                                                                                ;CLEAR THE ERROR FLAG ($ERFLG)
015420  013777  001102  163514                  MOV     $TSTNM,@DISPLAY         ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
015426  012737  015610  001200                  MOV     #EXIT16,BYPASS          ;SETUP BYPASS ADDRESS
015434  013704  001324                          MOV     RPADR,R4                ;RP11E BUS ADDRESS
015440  105037  002501                          CLRB    DPB+1                   ;CLEAR ANY EXTENDED MEMORY BITS
015444  105037  002503                          CLRB    DPB+3                   ;CLEAR 'MODE' & 'HDR' BITS
015450  012706  001100                  TEST16: MOV     #STACK,SP               ;SETUP THE STACK POINTER
015454  104401  027667                          TYPE    ,MSG16A                 ;'HEAD ALIGNMENT ROUTINE'
015460  000410                                  BR      3$                      ;GET THE HEAD ADDRESS WITHOUT THE SW07 NONESENSE
015462  032777  000200  163450          1$:     BIT     #SW07,@SWR              ;SWITCH 7 SET ?
015470  001774                                  BEQ     1$                      ;BR IF NOT
015472  032777  000200  163440          2$:     BIT     #SW07,@SWR              ;SWITCH 7 RESET ?
015500  001374                                  BNE     2$                      ;BR IF NOT
015502  104401  030201                  3$:     TYPE    ,MSG16B                 ;'ENTER HEAD ADDRESS (IN OCTAL)'
015506  104411                                  RDLIN                           ;GET THE ENTRY
015510  012601                                  MOV     (SP)+,R1                ;STARTING ADDRESS OF INPUT ASCII STRING
015512  004037  027062                          JSR     R0,CK.NUM               ;CHECK THE NUMBER
015516  015502                                          3$                      ;ILLEGAL INPUT
015520  015524                                          4$                      ;TERMINATED WITH A 'CR'
015522  015502                                          3$                      ;NO ENTRY - 'CR' ONLY
015524  122702  000024                  4$:     CMPB    #20.,R2                 ;VALID ?
015530  101764                                  BLOS    3$                      ;BR IF NOT
015532  110237  002513                          MOVB    R2,$TRK                 ;LOAD NEW HEAD ADDRESS
015536  136237  001360  001304                  BITB    ATABIT(R2),DRVTYP       ;RP02 OR RP03 ?
015544  001004                                  BNE     5$                      ;BR IF RP03
015546  012737  000111  002510                  MOV     #73.,$CYL               ;RP02 ALIGNMENT CYLINDER
015554  000403                                  BR      6$                      ;CONTINUE
015556  012737  000222  002510          5$:     MOV     #146.,$CYL              ;RP03 ALIGNMENT CYLINDER
015564  112737  000011  002502          6$:     MOVB    #SEEK,DPB+2             ;SEEK COMMAND
015572  004737  025606                          JSR     PC,CLRP                 ;CLEAR THE RP11
015576  004737  022740                          JSR     PC,RP11                 ;SEEK AND LOAD THE HEAD
015602  004737  023500                          JSR     PC,DRVRDY               ;WAIT FOR THE DRIVE TO BECOME READY
015606  000725                                  BR      1$                      ;WAIT FOR SWITCH 7 TO BE TOGGLED
015610  000000                          EXIT16: HALT                            ;'EXIT' FOR COMPATABILITY WITH OTHER TESTS
015612  000776                                  BR      .-2                     ;LOCK THE HALT


                                        ;;*****************************************************************
                                        ;*TEST 17        RP11E/DRIVE CONTROL PANEL SWITCH TEST

                                        ;*THIS TEST TEST THE 'ENABLE/DISABLE' AND 'READ ONLY' SWITCHES ON
                                        ;*THE DRIVE CONTROL PANEL AND TESTS THE 'WRITE LOCKOUT' AND 'LOA'
                                        ;*SWITCHES ON THE RP11E/DRIVE CONTROL PANEL.

                                        ;;*****************************************************************
015614                                  TST17:
015614  033737  001436  001276                  BIT     BITS+<$TN*2>,TSTNMS     ;IS THIS TEST SELECTED
015622  001002                                  BNE     .+6                     ;BR IF YES
015624  000137  016606                          JMP     @#$EOP                  ;GO TO END OF TEST
```

```
3770  015630  012737  015706  001110          MOV     #TEST17,$LPERR  ;SETUP THE ERROR LOOP ADDRESS
3771  015636  012737  015614  001106          MOV     #TST17,$LPADR   ;SETUP THE SCOPE LOOP ADDRESS
3772  015644  012737  000017  001102          MOV     #17,$TSTNM      ;SETUP TEST NUMBER AND
3773                                                                  ;CLEAR THE ERROR FLAG ($ERFLG)
3774  015652  013777  001102  163262          MOV     $TSTNM,@DISPLAY ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
3775  015660  012737  016574  001200          MOV     #EXIT17,BYPASS  ;SETUP BYPASS ADDRESS
3776  015666  013704  001324                  MOV     RPADR,R4        ;RP11E BUS ADDRESS
3777  015672  105037  002501                  CLRB    DPB+1           ;CLEAR ANY EXTENDED MEMORY BITS
3778  015676  105037  002503                  CLRB    DPB+3           ;CLEAR 'MODE' & 'HDR' BITS
3779  015702  104401  030243                  TYPE    ,MSG17A         ;'RP11E CONTROL PANEL SWITCH TEST'
3780  015706  012706  001100          TEST17: MOV     #STACK,SP       ;SETUP THE STACK POINTER
3781
3782                                  ;CHECK THE ENABLE/DISABLE SWITCH ON THE DRIVE
3783
3784  015712  104401  030306                  TYPE    ,MSG17B ;'TYPE A CR TO START TEST AFTER EACH SETUP'
3785  015716  104401  030365          2$:     TYPE    ,MSG17C         ;'DISABLE DRIVE # '
3786  015722  013746  001176                  MOV     CHKDRV,-(SP)    ;;SAVE CHKDRV FOR TYPEOUT
3787                                                                  ;;TYPE DRIVE NUMBER
3788  015726  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
3789  015730     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
3790  015731     001                          .BYTE   1               ;;TYPE LEADING ZEROS
3791  015732  104411                          RDLIN                   ;WAIT FOR THE 'CR'
3792  015734  012737  015734  001110  3$:     MOV     #3$,$LPERR      ;CHANGE LOOP ON ERROR ADDRESS
3793  015742  004737  025606                  JSR     PC,CLRP         ;CLEAR THE RP11
3794  015746  113764  001176  000005          MOVB    CHKDRV,RPCS+1(R4)  ;SELECT DRIVE
3795  015754  004737  023172                  JSR     PC,SAVRP        ;SAVE THE RP11 REGISTERS
3796  015760  032737  040000  001334          BIT     #SUOL,$RPDS     ;IS THE UNIT ON LINE?
3797  015766  001401                          BEQ     4$              ;BRANCH IF NOT
3798  015770  104065                          ERROR   65              ;SELECTED UNIT ON LINE WITH THE
3799                                                                  ;DRIVE DISABLED
3800  015772  104401  030410          4$:     TYPE    ,MSG17D         ;'ENABLE DRIVE # '
3801  015776  013746  001176                  MOV     CHKDRV,-(SP)    ;;SAVE CHKDRV FOR TYPEOUT
3802                                                                  ;;TYPE DRIVE NUMBER
3803  016002  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
3804  016004     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
3805  016005     001                          .BYTE   1               ;;TYPE LEADING ZEROS
3806  016006  104411                          RDLIN                   ;WAIT FOR THE 'CR'
3807  016010  012737  016010  001110  5$:     MOV     #5$,$LPERR      ;CHANGE THE LOOP ON ERROR ADDRESS
3808  016016  004737  025606                  JSR     PC,CLRP         ;CLEAR THE RP11
3809  016022  113764  001176  000005          MOVB    CHKDRV,RPCS+1(R4)  ;RESELECT THE UNIT
3810  016030  004737  023172                  JSR     PC,SAVRP        ;SAVE THE RP11 REGISTERS
3811  016034  032737  040000  001334          BIT     #SUOL,$RPDS     ;IS THE UNIT ON LINE?
3812  016042  001001                          BNE     RDSW            ;BR IF ONLINE
3813  016044  104066                          ERROR   66              ;SELECTED UNIT NOT ON LINE WITH
3814                                                                  ;THE DRIVE ENABLED
3815
3816                                  ;CHECK THE DRIVE'S WRITE LOCKOUT SWITCH
3817
3818  016046  104401  030432          RDSW:   TYPE    ,MSG17E         ;'SET READ ONLY ON DRIVE '
3819  016052  013746  001176                  MOV     CHKDRV,-(SP)    ;;SAVE CHKDRV FOR TYPEOUT
3820                                                                  ;;TYPE DRIVE NUMBER
3821  016056  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
3822  016060     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
3823  016061     001                          .BYTE   1               ;;TYPE LEADING ZEROS
3824  016062  104411                          RDLIN                   ;WAIT FOR 'CR'
3825  016064  012737  016064  001110  2$:     MOV     #2$,$LPERR      ;LOAD LOOP ON ERROR ADDRESS
```

```
3826  016072  004737  025606                      JSR     PC,CLRP         ;CLEAR THE RP11
3827  016076  113764  001176  000005              MOVB    CHKDRV,RPCS+1(R4)  ;SELECT THE DRIVE
3828  016104  004737  023172                      JSR     PC,SAVRP        ;SAVE THE RP11 REGISTERS
3829  016110  032737  000400  001334              BIT     #SUWP,$RPDS     ;IS SELECTED UNIT WRITE PROTECTED ?
3830  016116  001001                              BNE     3$              ;BR IF IT IS
3831  016120  104067                              ERROR   67              ;SELECTED UNIT WRITE PROTECT DID
3832                                                                      ;NOT SET WITH READ ONLY SET
3833  016122  104401  030466          3$:         TYPE    ,MSG17F         ;'SET READ/WRITE ON DRIVE '
3834  016126  013746  001176                      MOV     CHKDRV,-(SP)    ;;SAVE CHKDRV FOR TYPEOUT
3835                                                                      ;;TYPE DRIVE NUMBER
3836  016132  104403                              TYPOS                   ;;GO TYPE--OCTAL ASCII
3837  016134     001                              .BYTE   1               ;;TYPE 1 DIGIT(S)
3838  016135     001                              .BYTE   1               ;;TYPE LEADING ZEROS
3839  016136  104411                              RDLIN                   ;WAIT FOR 'CR'
3840  016140  012737  016140  001110  4$:         MOV     #4$,$LPERR      ;CHANGE THE LOOP ON ERROR ADDRESS
3841  016146  004737  025606                      JSR     PC,CLRP         ;CLEAR THE RP11
3842  016152  113764  001176  000005              MOVB    CHKDRV,RPCS+1(R4)  ;RESELECT THE DRIVE
3843  016160  004737  023172                      JSR     PC,SAVRP        ;SAVE THE RP11 REGISTERS
3844  016164  032737  000400  001334              BIT     #SUWP,$RPDS     ;IS SELECTED UNIT WRITE PROTECTED ?
3845  016172  001401                              BEQ     WRSW            ;BR IF NOT
3846  016174  104070                              ERROR   70              ;SELECTED UNIT WRITE PROTECT SET
3847                                                                      ;WITH DRIVE WRITE ENABLED
3848
3849                          ;TEST THE SETTING OF WRITE LOCKOUT AND THE 'LOA' SWITCHES ON THE
3850                          ;RP11E CONTROL PANEL.
3851
3852  016176  104401  030523  WRSW:   TYPE    ,MSG17G         ;'SET WRITE LOCKOUT ' ,ETC
3853  016202  104411                  RDLIN                   ;WAIT FOR 'CR' FROM THE OPERATOR
3854  016204  012737  016204  001110  1$:     MOV     #1$,$LPERR      ;CHANGE LOOP ON ERROR ADDRESS
3855  016212  005064  000012                  CLR     RPCA(R4)        ;CLEAR THE CYLINDER REGISTER
3856  016216  004737  023172                  JSR     PC,SAVRP        ;SAVE THE RP11 REGISTERS
3857  016222  032737  000400  001334          BIT     #SUWP,$RPDS     ;IS SELECTED UNIT WRITE PROTECT SET ?
3858  016230  001001                          BNE     2$              ;BR IF SET
3859  016232  104071                          ERROR   71              ;SELECTED UNIT WRITE PROTECT SHOULD
3860                                                                  ;BE SET - RPCA EQUAL TO ZERO
3861  016234  012764  000002  000012  2$:     MOV     #2,RPCA(R4)     ;LOAD A 2 INTO RPCA
3862  016242  004737  023172                  JSR     PC,SAVRP        ;SAVE THE RP11 REGISTERS
3863  016246  032737  000400  001334          BIT     #SUWP,$RPDS     ;IS SELECTED UNIT WRITE PROTECTED ?
3864  016254  001401                          BEQ     WRSW1           ;BR IF NOT PROTECTED
3865  016256  104072                          ERROR   72              ;SELECTED UNIT WRITE PROTECT IS SET
3866                                                                  ;WITH EQUAL TO 2.
3867
3868                          ;CHECK THE CYLINDER 'LOA' SWITCHES
3869
3870  016260  012700  000002  WRSW1:  MOV     #2,R0           ;INITIALIZE SWITCH PATTERN
3871  016264  104401  030640          TYPE    ,MSG17H         ;'SET FOLLOWING VALUES IN  THE
3872                                                          ;CYLINDER LOA SWITCHES'
3873  016270  104401  030740  1$:     TYPE    ,MSG17I         ;'SET '
3874  016274  006200                  ASR     R0              ;SETUP PATTERN FOR TYPOUT
3875  016276  010046                  MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
3876                                                          ;;TYPE THE SWITCH VALUE
3877  016300  104403                  TYPOS                   ;;GO TYPE--OCTAL ASCII
3878  016302     003                  .BYTE   3               ;;TYPE 3 DIGIT(S)
3879  016303     001                  .BYTE   1               ;;TYPE LEADING ZEROS
3880  016304  006300                  ASL     R0              ;RESTORE THE PATTERN
3881  016306  104411                  RDLIN                   ;WAIT FOR 'CR' FROM THE OPERATOR
```

```
 3882  016310  012737  016310  001110  2$:     MOV     #2$,$LPERR      ;CHANGE LOOP ON ERROR ADDRESS
 3883  016316  010064  000012                  MOV     R0,RPCA(R4)     ;LOAD PATTERN INTO RPCA
 3884  016322  004737  023172                  JSR     PC,SAVRP        ;SAVE THE RP11 REGISTERS
 3885  016326  032737  000400  001334          BIT     #SUWP,$RPDS     ;IS UNIT WRITE PROTECT SET ?
 3886  016334  001001                          BNE     3$              ;BR IF SET
 3887  016336  104073                          ERROR   73              ;UNIT WRITE PROTECT IS NOT SET
 3888                                                                  ;RPCA EQUALS CONTENTS OF BOUNDARY
 3889                                                                  ;SWITCHES
 3890  016340  062764  000002  000012  3$:     ADD     #2,RPCA(R4)     ;INCREMENT CYLINDER ADDRESS
 3891  016346  012737  016346  001110  4$:     MOV     #4$,$LPERR      ;CHANGE THE LOOP ON ERROR ADDRESS
 3892  016354  004737  023172                  JSR     PC,SAVRP        ;SAVE THE RP11 REGISTERS
 3893  016360  032737  000400  001334          BIT     #SUWP,$RPDS     ;IS UNIT WRITE PROTECT SET ?
 3894  016366  001401                          BEQ     5$              ;BR IF NOT
 3895  016370  104074                          ERROR   74              ;UNIT WRITE PROTECT IS SET WITH
 3896                                                                  ;RPCA ONE GREATER THAN BOUNDARY
 3897                                                                  ;SWITCH SETTINGS
 3898  016372  006300                  5$:     ASL     R0              ;UPDATE TEST PATTERN
 3899  016374  032700  001000                  BIT     #BIT09,R0       ;IS PATTERN EXCEEDED?
 3900  016400  001733                          BEQ     1$              ;BRANCH IF NOT
 3901
 3902                          ;CHECK DRIVE 'LOA' SWITCHES
 3903
 3904  016402  005037  001176          WRSW2:  CLR     CHKDRV          ;START WITH DRIVE 0
 3905  016406  104401  030763                  TYPE    ,MSG17J         ;'RESET THE CYLINDER LOA SWITCHES
 3906                                                                  ;'SET THE FOLLOWING VALUES IN THE DRIVE
 3907                                                                  ;'LOA SWITCHES '
 3908  016412  012764  000002  000012          MOV     #2,RPCA(R4)     ;LOAD CYLINDER ADDRESS
 3909  016420  113764  001176  000005  1$:     MOVB    CHKDRV,RPCS+1(R4)  ;SELECT THE FIRST DRIVE NUMBER
 3910  016426  104401  030740                  TYPE    ,MSG17I         ;'SET SWITCHES TO '
 3911  016432  013746  001176                  MOV     CHKDRV,-(SP)    ;;SAVE CHKDRV FOR TYPEOUT
 3912                                                                  ;;TYPE SWITCH SETTING
 3913  016436  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
 3914  016440     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
 3915  016441     001                          .BYTE   1               ;;TYPE LEADING ZEROS
 3916  016442  104411                          RDLIN                   ;WAIT FOR THE OPERATOR TO ENTER A 'CR'
 3917  016444  012737  016444  001110  2$:     MOV     #2$,$LPERR      ;LOOP ON ERROR ADDRESS
 3918  016452  004737  023172                  JSR     PC,SAVRP        ;STORE THE RP11 REGISTERS
 3919  016456  032737  000400  001334          BIT     #SUWP,$RPDS     ;IS THE SELECTED DRIVE WRITE PROTECTED ?
 3920  016464  001401                          BEQ     3$              ;BR IF NOT
 3921  016466  104075                          ERROR   75              ;'SUWP' SET WHEN DRIVE LOA SWITCHES
 3922                                                                  ;'EQUAL THE SELECTED DRIVE'
 3923  016470  005737  001176          3$:     TST     CHKDRV          ;THROUGH HERE THE FIRST TIME ?
 3924  016474  001004                          BNE     4$              ;BR IF NOT
 3925  016476  012737  000001  001176          MOV     #1,CHKDRV       ;SET FOR DRIVE 1
 3926  016504  000406                          BR      5$              ;CONTINUE
 3927  016506  006337  001176          4$:     ASL     CHKDRV          ;SHIFT TEST ADDRESS
 3928  016512  032737  000010  001176          BIT     #BIT03,CHKDRV   ;FINISHED WITH ADDRESSES ?
 3929  016520  001025                          BNE     EXIT17          ;BR IF FINISHED
 3930  016522  104401  030740          5$:     TYPE    ,MSG17I         ;'SET SWITCHES TO '
 3931  016526  013746  001176                  MOV     CHKDRV,-(SP)    ;;SAVE CHKDRV FOR TYPEOUT
 3932                                                                  ;;TYPE SWITCH SETTING
 3933  016532  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
 3934  016534     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
 3935  016535     001                          .BYTE   1               ;;TYPE LEADING ZEROS
 3936  016536  104411                          RDLIN                   ;WAIT FOR THE OPERATOR TO ENTER A 'CR'
 3937  016540  012737  016540  001110  6$:     MOV     #6$,$LPERR      ;CHANGE ERROR LOOP ADDRESS
```

```
3938  016546  004737  023172                  JSR     PC,SAVRP        ;STORE THE REGISTERS
3939  016552  032737  000400  001334          BIT     #SUWP,$RPDS     ;IS THE DRIVE WRITE PROTECTED ?
3940  016560  001001                          BNE     7$              ;BR IF IT IS
3941  016562  104113                          ERROR   113             ;'SUWP' NOT SET WHEN DRIVE LOA SWITCHES
3942                                                                  ;'GREATER THAN SELECTED DRIVE NUMBER
3943  016564  113764  001176  000005  7$:     MOVB    CHKDRV,RPCS+1(R4)  ;SELECT THE NEXT DRIVE
3944  016572  000724                          BR      2$              ;CONTINUE WITH THE TEST
3945  016574  000004                  EXIT17: SCOPE                   ;LOOP ?
3946  016576  104401  031122                  TYPE    ,MSG17K         ;'END OF CONTROL PANEL TEST - RETURN SWITCHES TO NORMAL'
3947  016602  000137  003676                  JMP     START2          ;RETURN TO THE 'CONVERSATION MODE' ENTRANCE
3948
3949                                  .SBTTL  END OF PASS ROUTINE
3950
3951                                  ;;*****************************************************************
3952                                  ;*INCREMENT THE PASS NUMBER ($PASS)
3953                                  ;*INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
3954                                  ;*IF THERES A MONITOR GO TO IT
3955                                  ;*IF THERE ISN'T JUMP TO RESTART
3956
3957  016606                          $EOP:
3958  016606  104401  016614                  TYPE    ,65$            ;;TYPE ASCIZ STRING
3959  016612  000410                          BR      64$             ;;GET OVER THE ASCIZ
3960                                  ;;65$:  .ASCIZ  <15><12><12>/END OF PASS/
3961  016634                          64$:
3962  016634  005737  001312                  TST     @#DRVSEL        ;ANY DRIVES SELECTED
3963  016640  001436                          BEQ     1$              ;NO, BRANCH
3964  016642  104401  016650                  TYPE    ,67$            ;;TYPE ASCIZ STRING
3965  016646  000407                          BR      66$             ;;GET OVER THE ASCIZ
3966                                  ;;67$:  .ASCIZ  / USING DRIVE/
3967  016666                          66$:
3968  016666  013746  001176                  MOV     @#CHKDRV,-(SP)  ;;SAVE @#CHKDRV FOR TYPEOUT
3969  016672  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
3970  016674     002                          .BYTE   2               ;;TYPE 2 DIGIT(S)
3971  016675     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
3972  016676  104401  016704                  TYPE    ,69$            ;;TYPE ASCIZ STRING
3973  016702  000412                          BR      68$             ;;GET OVER THE ASCIZ
3974                                  ;;69$:  .ASCIZ  /   ERRORS DETECTED=/
3975  016730                          68$:
3976  016730  013746  001112                  MOV     @#$ERTTL,-(SP)  ;;SAVE @#$ERTTL FOR TYPEOUT
3977  016734  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3978  016736  005037  001112          1$:     CLR     @#$ERTTL        ;ZERO ERROR TOTAL
3979  016742  005037  001102                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
3980  016746  005037  001162                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
3981  016752  005237  001100                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
3982  016756  042737  100000  001100          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
3983  016764  005327                          DEC     (PC)+           ;;LOOP?
3984  016766  000001                  $EOPCT: .WORD   1
3985  016770  003027                          BGT     $DOAGN          ;;YES
3986  016772  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
3987  016774  000001                  $ENDCT: .WORD   1
3988  016776  016766                          $EOPCT
3989  017000  104401  017006                  TYPE    ,65$            ;;TYPE ASCIZ STRING
3990  017004  000407                          BR      64$             ;;GET OVER THE ASCIZ
3991                                  ;;65$:  .ASCIZ  <15><12>/END OF TEST/
3992  017024                          64$:
3993  017024  104401  017054                  TYPE    ,$ENULL         ;TYPE NULL CHARACTER
```

```
 3994  017030  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
 3995  017034  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
 3996  017036  000005                          RESET                   ;;CLEAR THE WORLD
 3997  017040  004710                  $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
 3998  017042  000240                          NOP                     ;;SAVE ROOM
 3999  017044  000240                          NOP                     ;;FOR
 4000  017046  00024C                          NOP                     ;;ACT11
 4001  017050                          $DOAGN:
 4002  017050  000137                          JMP     @(PC)+          ;;RETURN
 4003  017052  004752                  $RTNAD: .WORD   RESTART
 4004  017054     377     377     000  $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
 4005          017060                          .EVEN
 4006
 4007
 4008                                  .SBTTL  *** SUBROUTINES ***
 4009
 4010                                  .SBTTL  ERROR HANDLER ROUTINE
 4011
 4012                                  ;;*****************************************************************
 4013                                  ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
 4014                                  ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
 4015                                  ;*AND GO TO TYPERR ON ERROR
 4016                                  ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
 4017                                  ;*SW15=1         HALT ON ERROR
 4018                                  ;*SW13=1         INHIBIT ERROR TYPEOUTS
 4019                                  ;*SW10=1         BELL ON ERROR
 4020                                  ;*SW09=1         LOOP ON ERROR
 4021                                  ;*CALL
 4022                                  ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
 4023
 4024
 4025  017060                          $ERROR:
 4026  017060  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
 4027  017062  105237  001103          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
 4028  017066  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
 4029  017070  013777  001102  162044          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
 4030  017076  032777  002000  162034          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
 4031  017104  001402                          BEQ     1$              ;;NO - SKIP
 4032  017106  104401  001166                  TYPE    ,$BELL          ;;RING BELL
 4033  017112  005237  001112          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
 4034  017116  011637  001116                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
 4035  017122  162737  000002  001116          SUB     #2,$ERRPC
 4036  017130  117737  161762  001114          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
 4037  017136  032777  020000  161774          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
 4038  017144  001004                          BNE     20$             ;;SKIP TYPEOUTS
 4039  017146  004737  017232                  JSR     PC,TYPERR       ;;GO TO USER ERROR ROUTINE
 4040  017152  104401  001173                  TYPE    ,$CRLF
 4041  017156                          20$:
 4042  017156  005777  161756          2$:     TST     @SWR            ;;HALT ON ERROR
 4043  017162  100002                          BPL     3$              ;;SKIP IF CONTINUE
 4044  017164  000000                          HALT                    ;;HALT ON ERROR!
 4045  017166  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
 4046  017170  032777  001000  161742  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
 4047  017176  001402                          BEQ     4$              ;;BR IF NO
 4048  017200  013716  001110                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
 4049  017204  005737  001164          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
```

```
 4050  017210  001402                        BEQ     5$                ;;BR IF NONE
 4051  017212  013716  001164                MOV     $ESCAPE,(SP)      ;;FUDGE RETURN ADDRESS FOR ESCAPE
 4052  017216                          5$:
 4053  017216  022737  017040  000042        CMP     #$ENDAD,@#42      ;;ACT-11 AUTO-ACCEPT?
 4054  017224  001001                        BNE     6$                ;;BRANCH IF NO
 4055  017226  000000                        HALT                      ;;YES
 4056  017230                          6$:
 4057  017230  000002                        RTI                       ;;RETURN
 4058
 4059                                  ;;*****************************************************************
 4060
 4061                                  ;THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE
 4062                                  ;WHICH ERROR IS TO BE REPORTED, IT THEN OBTAINS, FROM THE "ERROR
 4063                                  ;TABLE" ($ERRTB), AND REPORTS THE APPROPIATE INFORMATION
 4064                                  ;CONCERNING THE ERROR.
 4065                                  ;CALL
 4066                                  ;       JSR     PC,TYPERR
 4067                                  ;       RETURN
 4068
 4069  017232  104412                  TYPERR: SAVREG                    ;SAVE R0-R5
 4070  017234  032777  020000  161676        BIT     #SW13,@SWR        ;INHIBIT TYPEOUTS ?
 4071  017242  001105                        BNE     20$               ;BR IF YES
 4072  017244  005037  001160                CLR     $TMPO             ;CLEAR LOCATION FOR TEST NUMBER
 4073  017250  113737  001102  001160        MOVB    $TSTNM,$TMPO      ;LOAD TEST NUMBER FOR TYPEOUT
 4074  017256  005000                        CLR     R0                ;CLEAR R0 FOR ERROR NUMBER
 4075  017260  113700  001114                MOVB    $ITEMB,R0         ;ERROR NUMBER
 4076  017264  005300                        DEC     R0                ;FORM INDEX FOR ERROR TABLE
 4077  017266  006300                        ASL     R0
 4078  017270  006300                        ASL     R0
 4079  017272  006300                        ASL     R0
 4080  017274  062700  002514          1$:   ADD     #$ERRTB,R0        ;FORM ADDRESS
 4081  017300  012037  017314                MOV     (R0)+,2$          ;GET ERROR MESSAGE (EM) POINTER
 4082  017304  001404                        BEQ     3$                ;BRANCH IF THERE ISN'T ONE
 4083  017306  104401  001173                TYPE    ,$CRLF            ;"CARRIAGE RETURN - LINE FEED
 4084  017312  104401                        TYPE
 4085  017314  000000                  2$:   .WORD   0                 ;"EM" POINTER GOES HERE
 4086  017316  012037  017332          3$:   MOV     (R0)+,4$          ;PICK UP DATA HEADER (DH) POINTER
 4087  017322  001404                        BEQ     5$                ;BRANCH IF NONE
 4088  017324  104401  001173                TYPE    ,$CRLF            ;CARRIAGE RETURN-LINE FEED
 4089  017330  104401                        TYPE
 4090  017332  000000                  4$:   .WORD   0                 ;"DH" POINTER GOES HERE
 4091  017334  012001                  5$:   MOV     (R0)+,R1          ;PICKUP DATA TABLE (DT) POINTER
 4092  017336  001447                        BEQ     20$               ;BRANCH IF NONE
 4093  017340  005005                        CLR     R5                ;SET INDENT SWITCH
 4094  017342  012000                        MOV     (R0)+,R0          ;DATA FORMAT (DF) POINTER
 4095  017344  012002                        MOV     (R0)+,R2          ;NUMBER OF DH'S TO TYPE
 4096  017346  001443                        BEQ     20$               ;BRANCH IF DH NUMBER IS 0
 4097  017350  005105                        COM     R5                ;NO INDENT
 4098  017352  112003                  10$:  MOVB    (R0)+,R3          ;NUMBER OF DATA WORDS TO TYPE
 4099  017354  001440                        BEQ     20$               ;BR IF NO WORDS TO TYPE
 4100  017356  112004                        MOVB    (R0)+,R4          ;AND HOW TO TYPE THEM
 4101  017360  104401  001173                TYPE    ,$CRLF            ;CR-LF
 4102  017364  005705                        TST     R5                ;INDENT ?
 4103  017366  001002                        BNE     11$               ;BR IF NOT
 4104  017370  104401  027354                TYPE    ,BLNKS2           ;BLANKS
 4105  017374  006004                  11$:  ROR     R4                ;OCTAL OR DECIMAL?
```

```
 4106  017376  103403                        BCS     12$             ;DECIMAL--BRANCH
 4107  017400  013146                        MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
 4108  017402  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 4109  017404  000402                        BR      13$
 4110  017406                          12$:
 4111  017406  013146                        MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
 4112  017410  104405                        TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
 4113  017412  005303                  13$:  DEC     R3              ;MORE NUMBERS TO TYPE?
 4114  017414  001403                        BEQ     14$             ;NO--BRANCH
 4115  017416  104401  027354                TYPE    ,BLNKS2         ;YES--TYPE SEPARATORS
 4116  017422  000764                        BR      11$             ;LOOP
 4117  017424  005302                  14$:  DEC     R2              ;MORE DH'S?
 4118  017426  003413                        BLE     20$             ;NO--BRANCH
 4119  017430  104401  001173                TYPE    ,$CRLF          ;YES--START A NEW LINE
 4120  017434  005105                        COM     R5              ;INDENT?
 4121  017436  001002                        BNE     15$             ;NO--BRANCH
 4122  017440  104401  027354                TYPE    ,BLNKS2         ;YES--TYPE SPACES
 4123  017444  012037  017452          15$:  MOV     (R0)+,16$       ;GET NEXT DH
 4124  017450  104401                        TYPE                    ;AND TYPE IT
 4125  017452  000000                  16$:  .WORD   0               ;DH POINTER GOES HERE
 4126  017454  000736                        BR      10$             ;LOOP
 4127  017456  104413                  20$:  RESREG                  ;RESTORE R0-R5
 4128  017460  000207                        RTS     PC              ;RETURN
 4129
 4130                                  .SBTTL  TYPE ROUTINE
 4131
 4132                                  ;;**************************************************************
 4133                                  ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
 4134                                  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
 4135                                  ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
 4136                                  ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
 4137                                  ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
 4138                                  ;*
 4139                                  ;*CALL:
 4140                                  ;*1) USING A TRAP INSTRUCTION
 4141                                  ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
 4142                                  ;*OR
 4143                                  ;*      TYPE
 4144                                  ;*      MESADR
 4145                                  ;*
 4146
 4147  017462  105737  001157          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
 4148  017466  100002                        BPL     1$              ;;BR IF YES
 4149  017470  000000                        HALT                    ;;HALT HERE IF NO TERMINAL
 4150  017472  000407                        BR      3$              ;;LEAVE
 4151  017474  010046                  1$:   MOV     R0,-(SP)        ;;SAVE R0
 4152  017476  017600  000002                MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
 4153  017502  112046                  2$:   MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
 4154  017504  001005                        BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
 4155  017506  005726                        TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
 4156  017510  012600                  6$:   MOV     (SP)+,R0        ;;RESTORE R0
 4157  017512  062716  000002          3$:   ADD     #2,(SP)         ;;ADJUST RETURN PC
 4158  017516  000002                        RTI                     ;;RETURN
 4159  017520  122716  000011          4$:   CMPB    #HT,(SP)        ;;BRANCH IF <HT>
 4160  017524  001430                        BEQ     8$
 4161  017526  122716  000200                CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
```

```
4162  017532  001006                        BNE     5$
4163  017534  005726                        TST     (SP)+           ;;POP  <CR><LF> EQUIV
4164  017536  104401                        TYPE                    ;;TYPE A CR AND LF
4165  017540  001173                        $CRLF
4166  017542  105037  017676                CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
4167  017546  000755                        BR      2$              ;;GET NEXT CHARACTER
4168  017550  004737  017632        5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
4169  017554  123726  001156        6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
4170  017560  001350                        BNE     2$              ;;IF NO GO GET NEXT CHAR.
4171  017562  013746  001154                MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
4172                                                                ;;AND THE NULL CHAR.
4173  017566  105366  000001        7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
4174  017572  002770                        BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
4175  017574  004737  017632                JSR     PC,$TYPEC       ;;GO TYPE A NULL
4176  017600  105337  017676                DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
4177  017604  000770                        BR      7$              ;;LOOP
4178
4179                                ;HORIZONTAL TAB PROCESSOR
4180
4181  017606  112716  000040        8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
4182  017612  004737  017632        9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
4183  017616  132737  000007  017676        BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
4184  017624  001372                        BNE     9$              ;;TAB STOP
4185  017626  005726                        TST     (SP)+           ;;POP SPACE OFF STACK
4186  017630  000724                        BR      2$              ;;GET NEXT CHARACTER
4187  017632  105777  161312        $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
4188  017636  100375                        BPL     $TYPEC
4189  017640  116677  000002  161304        MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
4190  017646  122766  000015  000002        CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
4191  017654  001003                        BNE     1$              ;;BRANCH IF NO
4192  017656  105037  017676                CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
4193  017662  000406                        BR      $TYPEX          ;;EXIT
4194  017664  122766  000012  000002  1$:   CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
4195  017672  001402                        BEQ     $TYPEX          ;;BRANCH IF YES
4196  017674  105227                        INCB    (PC)+           ;;COUNT THE CHARACTER
4197  017676  000000        $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
4198  017700  000207        $TYPEX: RTS     PC
4199
4200
4201                                .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
4202
4203                                ;;##################################################################
4204                                ;#THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
4205                                ;#OCTAL (ASCII) NUMBER AND TYPE IT.
4206                                ;#$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
4207                                ;#CALL:
4208                                ;#      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
4209                                ;#      TYPOS                   ;;CALL FOR TYPEOUT
4210                                ;#      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
4211                                ;#      .BYTE   M               ;;M=1 OR 0
4212                                ;#                                      ;;1=TYPE LEADING ZEROS
4213                                ;#                                      ;;0=SUPPRESS LEADING ZEROS
4214                                ;#
4215                                ;#$TYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
4216                                ;#$TYPOS OR $TYPOC
4217                                ;#CALL:
```

```
                                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                        ;*      TYPON                   ;;CALL FOR TYPEOUT
                                        ;*
                                        ;*STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                        ;*      TYPOC                   ;;CALL FOR TYPEOUT
                                        ;*
017702 017646 000000            STYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
017706 116637 000001 020125             MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
017714 112637 020127                    MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
017720 062716 000002                    ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
017724 000406                           BR      STYPON
017726 112737 000001 020125     STYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
017734 112737 000006 020127             MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
017742 112737 000005 020124     STYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
017750 010346                           MOV     R3,-(SP)        ;;SAVE R3
017752 010446                           MOV     R4,-(SP)        ;;SAVE R4
017754 010546                           MOV     R5,-(SP)        ;;SAVE R5
017756 113704 020127                    MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
017762 005404                           NEG     R4
017764 062704 000006                    ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
017770 110437 020126                    MOVB    R4,$OMODE       ;;SAVE IT FOR USE
017774 113704 020125                    MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
020000 016605 000012                    MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
020004 005003                           CLR     R3              ;;CLEAR THE OUTPUT WORD
020006 006105                   1$:     ROL     R5              ;;ROTATE MSB INTO "C"
020010 000404                           BR      3$              ;;GO DO MSB
020012 006105                   2$:     ROL     R5              ;;FORM THIS DIGIT
020014 006105                           ROL     R5
020016 006105                           ROL     R5
020020 010503                           MOV     R5,R3
020022 006103                   3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
020024 105337 020126                    DECB    $OMODE          ;;TYPE THIS DIGIT?
020030 100016                           BPL     7$              ;;BR IF NO
020032 042703 177770                    BIC     #177770,R3      ;;GET RID OF JUNK
020036 001002                           BNE     4$              ;;TEST FOR 0
020040 005704                           TST     R4              ;;SUPPRESS THIS 0?
020042 001403                           BEQ     5$              ;;BR IF YES
020044 005204                   4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
020046 052703 000060                    BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
020052 052703 000040            5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
020056 110337 020122                    MOVB    R3,8$           ;;SAVE FOR TYPING
020062 104401 020122                    TYPE    ,8$             ;;GO TYPE THIS DIGIT
020066 105337 020124            7$:     DECB    $OCNT           ;;COUNT BY 1
020072 003347                           BGT     2$              ;;BR IF MORE TO DO
020074 002402                           BLT     6$              ;;BR IF DONE
020076 005204                           INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
020100 000744                           BR      2$              ;;GO DO THE LAST DIGIT
020102 012605                   6$:     MOV     (SP)+,R5        ;;RESTORE R5
020104 012604                           MOV     (SP)+,R4        ;;RESTORE R4
020106 012603                           MOV     (SP)+,R3        ;;RESTORE R3
020110 016666 000002 000004             MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
020116 012616                           MOV     (SP)+,(SP)
020120 000002                           RTI                     ;;RETURN
020122     000                  8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
```

```
                                        .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
                                $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
                                $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
                                $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE

                                .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

                                ;;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                                ;#THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
                                ;#SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
                                ;#NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
                                ;#BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
                                ;#REPLACED WITH SPACES.
                                ;#CALL:
                                ;#      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
                                ;#      TYPDS                   ;;GO TO THE ROUTINE

020130                          $TYPDS:
020130  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
020132  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
020134  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
020136  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
020140  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
020142  012746  020200                  MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
020146  016605  000020                  MOV     20(SP),R5       ;;GET THE INPUT NUMBER
020152  100004                          BPL     1$              ;;BR IF INPUT IS POS.
020154  005405                          NEG     R5              ;;MAKE THE BINARY NUMBER POS.
020156  112766  000055  000001          MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
020164  005000                  1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
020166  012703  020344                  MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
020172  112723  000040                  MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
020176  005002                  2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
020200  016001  020334                  MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
020204  160105                  3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
020206  002402                          BLT     4$              ;;BR IF DONE
020210  005202                          INC     R2              ;;INCREASE THE BCD DIGIT BY 1
020212  000774                          BR      3$
020214  060105                  4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
020216  005702                          TST     R2              ;;CHECK IF BCD DIGIT=0
020220  001002                          BNE     5$              ;;FALL THROUGH IF 0
020222  105716                          TSTB    (SP)            ;;STILL DOING LEADING 0'S?
020224  100407                          BMI     7$              ;;BR IF YES
020226  106316                  5$:     ASLB    (SP)            ;;MSD?
020230  103003                          BCC     6$              ;;BR IF NO
020232  116663  000001  177777          MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
020240  052702  000060          6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
020244  052702  000040          7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
020250  110223                          MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
020252  005720                          TST     (R0)+           ;;JUST INCREMENTING
020254  020027  000010                  CMP     R0,#10          ;;CHECK THE TABLE INDEX
020260  002746                          BLT     2$              ;;GO DO THE NEXT DIGIT
020262  003002                          BGT     8$              ;;GO TO EXIT
020264  010502                          MOV     R5,R2           ;;GET THE LSD
020266  000764                          BR      6$              ;;GO CHANGE TO ASCII
020270  105726                  8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
020272  100003                          BPL     9$              ;;BR IF NO
```

```
4330  020274  116663  177777  177776          MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
4331  020302  105013                  9$:     CLRB    (R3)            ;;SET THE TERMINATOR
4332  020304  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
4333  020306  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
4334  020310  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
4335  020312  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
4336  020314  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
4337  020316  104401  020344                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
4338  020322  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
4339  020330  012616                          MOV     (SP)+,(SP)
4340  020332  000002                          RTI                     ;;RETURN TO USER
4341  020334  023420                  $DTBL:  10000.
4342  020336  001750                          1000.
4343  020340  000144                          100.
4344  020342  000012                          10.
4345  020344  000004                  $DBLK:  .BLKW   4
4346
4347                                  .SBTTL  TTY INPUT ROUTINE
4348
4349                                  ;;******************************************************************
4350                                  .ENABL  LSB
4351  020354  000000                  $TKCNT: .WORD   0               ;;NUMBER OF ITEMS IN QUEUE
4352  020356  000000                  $TKQIN: .WORD   0               ;;INPUT POINTER
4353  020360  000000                  $TKQOUT: .WORD  0               ;;OUTPUT POINTER
4354  020362  000012                  $TKQSRT: .BLKB  10.             ;;TTY KEYBOARD QUEUE
4355          020374                  $TKQEND=.
4356
4357                                  ;*TK INITIALIZE ROUTINE
4358                                  ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
4359                                  ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
4360                                  ;*
4361                                  ;*CALL:
4362                                  ;*      JSR     PC,$TKINT
4363                                  ;*      RETURN
4364                                  ;*
4365  020374  005037  020354          $TKINT: CLR     $TKCNT          ;;CLEAR COUNT OF ITEMS IN QUEUE
4366  020400  012737  020362  020356          MOV     #$TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
4367  020406  013737  020356  020360          MOV     $TKQIN,$TKQOUT  ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
4368  020414  012737  020444  000060          MOV     #$TKSRV,@#TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
4369  020422  012737  000200  000062          MOV     #200,@#TKVEC+2  ;;"BR" LEVEL 4
4370  020430  005777  160512                  TST     @$TKB           ;;CLEAR DONE FLAG
4371  020434  012777  000100  160502          MOV     #100,@$TKS      ;;ENABLE TTY KEYBOARD INTERRUPT
4372  020442  000207                          RTS     PC              ;;RETURN TO CALLER
4373
4374                                  ;*TK SERVICE ROUTINE
4375                                  ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
4376                                  ;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
4377                                  ;*IT IN THE QUEUE.
4378                                  ;*IF THE CHARACTER IS A "CONTROL-C" (^C) $TKINT IS CALLED AND
4379                                  ;*UPON RETURN EXIT IS MADE TO THE "CONTROL-C" RESTART ADDRESS (START2)
4380                                  ;*
4381  020444  117746  160476          $TKSRV: MOVB    @$TKB,-(SP)     ;;PICKUP THE CHARACTER
4382  020450  042716  177600                  BIC     #^C177,(SP)     ;;STRIP THE JUNK
4383  020454  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL C?
4384  020460  001007                          BNE     1$              ;;BRANCH IF NO
4385  020462  104401  021562                  TYPE    ,$CNTLC         ;;TYPE A CONTROL-C (^C)
```

```
4386  020466  004737  020374                JSR     PC,$TKINT       ;;INIT THE KEYBOARD
4387  020472  005726                        TST     (SP)+           ;;CLEAN UP STACK
4388  020474  000137  003676                JMP     START2          ;;CONTROL C RESTART
4389  020500  021627  000007        1$:     CMP     (SP),#7         ;;IS IT A CONTROL G?
4390  020504  001004                        BNE     2$              ;;BRANCH IF NO
4391  020506  022737  000176  001140        CMP     #SWREG,SWR      ;;IS SOFT-SWR SELECTED?
4392  020514  001500                        BEQ     6$              ;;GO TO SWR CHANGE
4393
4394  020516                        2$:
4395  020516  022737  000012  020354        CMP     #10.,$TKCNT     ;;IS THE QUEUE FULL?
4396  020524  001004                        BNE     3$              ;;BRANCH IF NO
4397  020526  104401  001166                TYPE    $BELL           ;;RING THE TTY BELL
4398  020532  005726                        TST     (SP)+           ;;CLEAN CHARACTER OFF OF STACK
4399  020534  000451                        BR      5$              ;;EXIT
4400  020536  021627  000023        3$:     CMP     (SP),#23        ;;IS IT A CONTROL-S?
4401  020542  001021                        BNE     32$             ;;BRANCH IF NO
4402  020544  005077  160374                CLR     @STKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
4403  020550  005726                        TST     (SP)+           ;;CLEAN CHAR OFF STACK
4404  020552  105777  160366        31$:    TSTB    @STKS           ;;WAIT FOR A CHAR
4405  020556  100375                        BPL     31$             ;;LOOP UNTIL ITS THERE
4406  020560  117746  160362                MOVB    @STKB,-(SP)     ;;GET THE CHARACTER
4407  020564  042716  177600                BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
4408  020570  022627  000021                CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
4409  020574  001366                        BNE     31$             ;;BRANCH IF NO
4410  020576  012777  000100  160340        MOV     #100,@STKS      ;;REENABLE TTY KEYBOARD INTERRUPTS
4411  020604  000002                        RTI                     ;;RETURN
4412  020606  005237  020354        32$:    INC     $TKCNT          ;;COUNT THIS CHARACTER
4413  020612  021627  000140                CMP     (SP),#140       ;;IS IT UPPER CASE?
4414  020616  002405                        BLT     4$              ;;BRANCH IF YES
4415  020620  021627  000175                CMP     (SP),#175       ;;IS IT A SPECIAL CHAR?
4416  020624  003002                        BGT     4$              ;;BRANCH IF YES
4417  020626  042716  000040                BIC     #40,(SP)        ;;MAKE IT UPPER CASE
4418  020632  112677  177520        4$:     MOVB    (SP)+,@STKQIN   ;;AND PUT IT IN QUEUE
4419  020636  005237  020356                INC     $TKQIN          ;;UPDATE THE POINTER
4420  020642  023727  020356  020374        CMP     $TKQIN,#$TKQEND ;;GO OFF THE END?
4421  020650  001003                        BNE     5$              ;;BRANCH IF NO
4422  020652  012737  020362  020356        MOV     #$TKQSRT,$TKQIN ;;RESET THE POINTER
4423  020660  000002                5$:     RTI                     ;;RETURN
4424
4425                                  ;;**********************************************************************
4426                                  ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
4427                                  ;*ROUTINE IS ENTERED  FROM THE TRAP HANDLER, AND WILL
4428                                  ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
4429                                  ;*CALL WHEN OPERATING IN TTY INTERRUPT MODE.
4430  020662  022737  000176  001140  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED
4431  020670  001124                        BNE     15$             ;;EXIT IF NOT
4432  020672  105777  160246                TSTB    @STKS           ;;IS A CHAR WAITING?
4433  020676  100121                        BPL     15$             ;;IF NOT, EXIT
4434  020700  117746  160242                MOVB    @STKB,-(SP)     ;;YES
4435  020704  042716  177600                BIC     #^C177,(SP)     ;;MAKE IT  7-BIT ASCII
4436  020710  021627  000007                CMP     (SP),#7         ;;IS IT A CONTROL-G?
4437  020714  001300                        BNE     2$              ;;IF NOT, PUT IT IN THE TTY QUEUE
4438                                                                ;;AND EXIT
4439
4440                                  ;;**********************************************************************
4441                                  ;*CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
```

```
4442                                    ;*ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
4443                                    ;*CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
4444  020716  123727  001134  000001  6$:     CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
4445  020724  001674                          BEQ     2$              ;;BRANCH IF YES
4446  020726  005726                          TST     (SP)+           ;;CLEAR CONTROL-G OFF STACK
4447  020730  004737  020374                  JSR     PC,$TKINT       ;;FLUSH THE TTY INPUT QUEUE
4448  020734  005077  160204                  CLR     @#TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
4449  020740  112737  000001  001135          MOVB    #1,$INTAG       ;;SET INTERRUPT MODE INDICATOR
4450
4451  020746  104401  021574          SGTSWR: TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (↑G)
4452  020752  104401  021601                  TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
4453  020756  013746  000176                  MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
4454  020762  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
4455  020764  104401  021612                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
4456  020770  005046                  19$:     CLR     -(SP)           ;;CLEAR COUNTER
4457  020772  005046                          CLR     -(SP)           ;;THE NEW SWR
4458  020774  105777  160144          7$:     TSTB    @#TKS           ;;CHAR THERE?
4459  021000  100375                          BPL     7$              ;;IF NOT TRY AGAIN
4460
4461  021002  117746  160140                  MOVB    @#TKB,-(SP)     ;;PICK UP CHAR
4462  021006  042716  177600                  BIC     #↑C177,(SP)     ;;MAKE IT 7-BIT ASCII
4463
4464  021012  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL-C?
4465  021016  001015                          BNE     9$              ;;BRANCH IF NOT
4466  021020  104401  021562                  TYPE    ,$CNTLC         ;;YES, ECHO CONTROL-C (↑C)
4467  021024  062706  000006                  ADD     #6,SP           ;;CLEAN UP STACK
4468  021030  123727  001135  000001          CMPB    $INTAG,#1       ;;REENABLE TTY KEYBOARD INTERRUPTS?
4469  021036  001003                          BNE     8$              ;;BRANCH IF NO
4470  021040  012777  000100  160076          MOV     #100,@#TKS      ;;ALLOW TTY KEYBOARD INTERRUPTS
4471  021046  000137  003676          8$:     JMP     START2          ;;CONTROL-C RESTART
4472
4473
4474  021052  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
4475  021056  001005                          BNE     10$             ;;BRANCH IF NOT
4476  021060  104401  021567                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (↑U)
4477  021064  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
4478  021070  000737                          BR      19$             ;;LET'S TRY IT AGAIN
4479
4480
4481  021072  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
4482  021076  001022                          BNE     16$             ;;BRANCH IF NO
4483  021100  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
4484  021104  001403                          BEQ     11$             ;;BRANCH IF YES
4485  021106  016677  000002  160024          MOV     2(SP),@SWR      ;;SAVE NEW SWR
4486  021114  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
4487  021120  104401  001173          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
4488  021124  123727  001135  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
4489  021132  001003                          BNE     15$             ;;BRANCH IF NOT
4490  021134  012777  000100  160002          MOV     #100,@#TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
4491  021142  000002                  15$:    RTI                     ;;RETURN
4492  021144  004737  017632          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
4493  021150  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
4494  021154  002420                          BLT     18$             ;;BRANCH IF YES
4495  021156  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
4496  021162  003015                          BGT     18$             ;;BRANCH IF YES
4497  021164  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
```

```
 4498  021170  005766  000002                    TST     2(SP)           ;;IS THIS THE FIRST CHAR
 4499  021174  001403                            BEQ     17$             ;;BRANCH IF YES
 4500  021176  006316                            ASL     (SP)            ;;NO, SHIFT PRESENT
 4501  021200  006316                            ASL     (SP)            ;;   CHAR OVER TO MAKE
 4502  021202  006316                            ASL     (SP)            ;;   ROOM FOR NEW ONE.
 4503  021204  005266  000002          17$:      INC     2(SP)           ;;KEEP COUNT OF CHAR
 4504  021210  056616  177776                    BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
 4505  021214  000667                            BR      7$              ;;GET THE NEXT ONE
 4506  021216  104401  001172          18$:      TYPE    ,$QUES          ;;TYPE ?<CR><LF>
 4507  021222  000720                            BR      20$             ;;SIMULATE CONTROL-U
 4508                                  .DSABL    LSB
 4509
 4510
 4511                                  ;;*****************************************************************
 4512                                  ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
 4513                                  ;*CALL:
 4514                                  ;*      RDCHR                   ;;GET A CHARACTER FROM THE QUEUE
 4515                                  ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
 4516                                  ;*                              ;;WITH PARITY BIT STRIPPED OFF
 4517                                  ;
 4518
 4519  021224  011646                  $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC AND
 4520  021226  016666  000004  000002          MOV     4(SP),2(SP)     ;;THE PS
 4521  021234  005066  000004                  CLR     4(SP)           ;;GET READY FOR A CHARACTER
 4522  021240  005046                          CLR     -(SP)           ;;PUT NEW PS ON STACK
 4523  021242  012746  021250                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
 4524  021246  000002                          RTI                     ;;POP NEW PC AND PS
 4525  021250                          64$:
 4526  021250  005737  020354          1$:     TST     $TKCNT          ;;WAIT ON A CHARACTER
 4527  021254  001775                          BEQ     1$
 4528  021256  005337  020354                  DEC     $TKCNT          ;;DECREMENT THE COUNTER
 4529  021262  117766  177072  000004          MOVB    @$TKQOUT,4(SP)  ;;GET ONE CHARACTER
 4530  021270  005237  020360                  INC     $TKQOUT         ;;UPDATE THE POINTER
 4531  021274  023727  020360  020374          CMP     $TKQOUT,#$TKQEND ;;DID IT GO OFF OF THE END?
 4532  021302  001003                          BNE     2$              ;;BRANCH IF NO
 4533  021304  012737  020362  020360          MOV     #$TKQSRT,$TKQOUT ;;RESET THE POINTER
 4534  021312  000002                  2$:     RTI                     ;;RETURN
 4535                                  ;;*****************************************************************
 4536                                  ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
 4537                                  ;*CALL:
 4538                                  ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
 4539                                  ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
 4540                                  ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
 4541
 4542  021314  010346                  $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
 4543  021316  005046                          CLR     -(SP)           ;;CLEAR THE RUBOUT KEY
 4544  021320  012703  021550          1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
 4545  021324  022703  021562          2$:     CMP     #$TTYIN+10.,R3  ;;BUFFER FULL?
 4546  021330  101456                          BLOS    4$              ;;BR IF YES
 4547  021332  104410                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
 4548  021334  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
 4549  021336  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
 4550  021342  001022                          BNE     5$              ;;BR IF NO
 4551  021344  005716                          TST     (SP)            ;;IS THIS THE FIRST RUBOUT?
 4552  021346  001007                          BNE     6$              ;;BR IF NO
 4553  021350  112737  000134  021546          MOVB    #'\,9$          ;;TYPE A BACK SLASH
```

```
                                                TYPE    @S
                                                MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
                                        6$:     DEC     R3              ;;BACKUP BY ONE
                                                CMP     R3,#$TTYIN      ;;STACK EMPTY?
                                                BLO     4$              ;;BR IF YES
                                                MOVB    (R3),@S         ;;SETUP TO TYPEOUT THE DELETED CHAR.
                                                TYPE    ,@S             ;;GO TYPE
                                                BR      2$              ;;GO READ ANOTHER CHAR.
                                        5$:     TST     (SP)            ;;RUBOUT KEY SET?
                                                BEQ     7$              ;;BR IF NO
                                                MOVB    #'\,@S          ;;TYPE A BACK SLASH
                                                TYPE    @S
                                                CLR     (SP)            ;;CLEAR THE RUBOUT KEY
                                        7$:     CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
                                                BNE     8$              ;;BR IF NO
                                                TYPE    $CNTLU          ;;TYPE A CONTROL "U"
                                                BR      1$              ;;GO START OVER
                                        8$:     CMPB    #22,(R3)        ;;IS CHARACTER A "↑R"?
                                                BNE     3$              ;;BRANCH IF NO
                                                CLRB    (R3)            ;;CLEAR THE CHARACTER
                                                TYPE    ,$CRLF          ;;TYPE A "CR" & "LF"
                                                TYPE    ,$TTYIN         ;;TYPE THE INPUT STRING
                                                BR      2$              ;;GO PICKUP ANOTHER CHACTER
                                        4$:     TYPE    $QUES           ;;TYPE A '?'
                                                BR      1$              ;;CLEAR THE BUFFER AND LOOP
                                        3$:     MOVB    (R3),@S         ;;ECHO THE CHARACTER
                                                TYPE    @S
                                                CMPB    #15,(R3)+       ;;CHECK FOR RETURN
                                                BNE     2$              ;;LOOP IF NOT RETURN
                                                CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
                                                TYPE    $LF             ;;TYPE A LINE FEED
                                                TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
                                                MOV     (SP)+,R3        ;;RESTORE R3
                                                MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
                                                MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
                                                MOV     #$TTYIN,4(SP)
                                                RTI                     ;;RETURN
                                        @S:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
                                                .BYTE   0               ;;TERMINATOR
                                        $TTYIN: .BLKB   10.             ;;RESERVE 10. BYTES FOR TTY INPUT
                                        $CNTLC: .ASCIZ  /↑C/<15><12>    ;;CONTROL "C"
                                        $CNTLU: .ASCIZ  /↑U/<15><12>    ;;CONTROL "U"
                                        $CNTLG: .ASCIZ  /↑G/<15><12>    ;;CONTROL "G"
                                        $MSWR:  .ASCIZ  <15><12>/SWR = /
                                        $MNEW:  .ASCIZ  /  NEW = /

                                                .EVEN

                                                .SBTTL  SCOPE HANDLER ROUTINE

                                        ;;**************************************************************
                                        ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
                                        ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
                                        ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
                                        ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
```

```
                                        ;*SW14=1        LOOP ON TEST
                                        ;*SW11=1        INHIBIT ITERATIONS
                                        ;*SW09=1        LOOP ON ERROR
                                        ;*CALL
                                        ;*      SCOPE           ;;SCOPE=IOT
                                        $SCOPE:
                                                CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
                                        1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
                                                BNE     $OVER           ;;YES IF SW14=1
                                        ;#####START OF CODE FOR THE XOR TESTER#####
                                        $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                        ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
                                                MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
                                                MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
                                                TST     @#177060        ;;TIME OUT ON XOR?
                                                MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
                                                BR      $SVLAD          ;;GO TO THE NEXT TEST
                                        5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
                                                MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
                                                BR      7$              ;;LOOP ON THE PRESENT TEST
                                        6$:;#####END OF CODE FOR THE XOR TESTER#####
                                        2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
                                                BEQ     3$              ;;BR IF NO
                                                CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
                                                BHI     3$              ;;BR IF NO
                                                BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
                                                BEQ     4$              ;;BR IF NO
                                        7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
                                                BR      $OVER
                                        4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
                                                CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
                                                BR      1$              ;;ESCAPE TO THE NEXT TEST
                                        3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
                                                BNE     1$              ;;BR IF YES
                                                INC     $ICNT           ;;INCREMENT ITERATION COUNT
                                                CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
                                                BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
                                        1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
                                                MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
                                        $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
                                                MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
                                                MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
                                                CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
                                                MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
                                        $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
                                                MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
                                                RTI                     ;;FIXES PS
                                        $MXCNT: 1                       ;;MAX. NUMBER OF ITERATIONS

                                        .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES

                                        ;;**************************************************************
                                        ;*SAVE R0-R5
                                        ;*CALL:
                                        ;*      SAVREG
```

```
                                        ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
                                        ;*
                                        ;*TOP---(+16)
                                        ;*  +2---(+18)
                                        ;*  +4---R5
                                        ;*  +6---R4
                                        ;*  +8---R3
                                        ;* +10---R2
                                        ;* +12---R1
                                        ;* +14---R0

022050                                  $SAVREG:
022050  010046                                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
022052  010146                                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
022054  010246                                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
022056  010346                                  MOV     R3,-(SP)        ;;PUSH R3 ON STACK
022060  010446                                  MOV     R4,-(SP)        ;;PUSH R4 ON STACK
022062  010546                                  MOV     R5,-(SP)        ;;PUSH R5 ON STACK
022064  016646  000022                         MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
022070  016646  000022                         MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
022074  016646  000022                         MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
022100  016646  000022                         MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
022104  000002                                  RTI

                                        ;*RESTORE R0-R5
                                        ;*CALL:
                                        ;*      RESREG
022106                                  $RESREG:
022106  012666  000022                          MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
022112  012666  000022                          MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
022116  012666  000022                          MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
022122  012666  000022                          MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
022126  012605                                  MOV     (SP)+,R5        ;;POP STACK INTO R5
022130  012604                                  MOV     (SP)+,R4        ;;POP STACK INTO R4
022132  012603                                  MOV     (SP)+,R3        ;;POP STACK INTO R3
022134  012602                                  MOV     (SP)+,R2        ;;POP STACK INTO R2
022136  012601                                  MOV     (SP)+,R1        ;;POP STACK INTO R1
022140  012600                                  MOV     (SP)+,R0        ;;POP STACK INTO R0
022142  000002                                  RTI

                                        .SBTTL  ROUTINE TO SIZE MEMORY

                                        ;;*****************************************************************
                                        ;*CALL:
                                        ;*      JSR     PC,$SIZE
                                        ;*      RETURN
                                        ;*$LSTAD WILL CONTAIN:
                                        ;*      WITH KT11 OPTION        -- LAST VIRTUAL ADDRESS OF THE LAST BANK
                                        ;*      WITHOUT KT11 OPTION     -- LAST ABSOLUTE ADDRESS OF AVAILABLE MEMORY
                                        ;*$LSTBK WILL CONTAIN THE LAST BANK AS A SAF
                                        ;*$KT11 IS THE MEMORY MANAGEMENT KEY
                                        ;*BIT07 = 0 DON'T USE MEMORY MANAGEMENT
                                        ;*      MUST BE SETUP BEFORE THE CALL
                                        ;*BIT15 = 0 DON'T HAVE MEMORY MANAGEMENT OPTION
                                        ;*      DETERMINED BY ROUTINE
```

```
 4722  022144  010046                    $SIZE:  MOV     R0,-(SP)        ;;SAVE R0 ON THE STACK
 4723  022146  010146                            MOV     R1,-(SP)        ;;SAVE R1 ON THE STACK
 4724  022150  010246                            MOV     R2,-(SP)        ;;SAVE R2 ON THE STACK
 4725  022152  010346                            MOV     R3,-(SP)        ;;SAVE R3 ON THE STACK
 4726  022154  013746  000004                    MOV     @#ERRVEC,-(SP)  ;;SAVE PRESENT ERROR VECTOR PS & PC
 4727  022160  013746  000006                    MOV     @#ERRVEC+2,-(SP)
 4728  022164  010600                            MOV     SP,R0           ;;SAVE THE STACK POINTER
 4729                                    ;;SET THE ERRVEC PS TO THE PRESENT PS
 4730  022166  104400                            TRAP                    ;;PUSH OLD PSW AND PC ON STACK
 4731  022170  012637  000006                    MOV     (SP)+,@#ERRVEC+2        ;;SAVE THE PSW IN @#ERRVEC+2
 4732  022174  012701  003776                    MOV     #3776,R1        ;;SETUP ADDRESS
 4733  022200  105727                            TSTB    (PC)+           ;;USE MEMORY MANAGEMENT?
 4734  022202  000200                    $KT11:  .WORD   200             ;;SET TO USE MEMORY MANAGEMENT
 4735  022204  100062                            BPL     $CORE           ;;BR IF NO
 4736  022206  012737  022344  000004            MOV     #$KTNEX,@#ERRVEC        ;;SET FOR TIMEOUT
 4737  022214  005737  177572                    TST     @#SR0           ;;KT11 ARE YOU THERE?
 4738  022220  052737  100000  022202            BIS     #100000,$KT11   ;;YES--SET KT11 KEY
 4739  022226  005046                            CLR     -(SP)           ;;INITIALIZE FOR "PAR" LOADING
 4740  022230  012702  172340                    MOV     #KIPAR0,R2      ;;ADDRESS OF FIRST "PAR"
 4741  022234  012703  000010                    MOV     #10,R3          ;;LOAD EIGHT "PAR.'S" AND EIGHT "PDR.'S"
 4742  022240  012762  077406  177740    1$:     MOV     #77406,-40(R2)  ;;PDR = 4K, UP, READ/WRITE
 4743  022246  011622                            MOV     (SP),(R2)+      ;;LOAD "PAR"
 4744  022250  062716  000200                    ADD     #200,(SP)       ;;UPDATE FOR NEXT "PAR"
 4745  022254  077307                            SOB     R3,1$           ;;LOOP UNTIL ALL EIGHT ARE LOADED
 4746  022256  012742  177600                    MOV     #177600,-(R2)   ;;SETUP KIPAR7 FOR I/O
 4747  022262  005042                            CLR     -(R2)           ;;SETUP KIPAR6 FOR TESTING
 4748  022264  012737  022302  000004            MOV     #2$,@#ERRVEC    ;;CATCH TIMEOUT IF NO SR3
 4749  022272  012737  000020  172516            MOV     #20,@#SR3       ;;ENABLE 22 BIT MODE
 4750  022300  000401                            BR      3$              ;;THIS PDP-11 HAS A SR3 REGISTER
 4751  022302  022626                    2$:     CMP     (SP)+,(SP)+     ;;CLEAN OFF THE STACK--NO SR3
 4752  022304  005237  177572            3$:     INC     @#SR0           ;;TURN ON MEMORY MANAGEMENT
 4753  022310  012737  022334  000004            MOV     #$KTOUT,@#ERRVEC        ;;SET FOR TIME OUT
 4754  022316  005737  143776            4$:     TST     @#143776        ;;TRAP ON NON-EX-MEM
 4755  022322  062712  000040                    ADD     #40,(R2)        ;;MAKE A 1K STEP
 4756  022326  023712  172356                    CMP     @#KIPAR7,(R2)   ;;LAST ONE?
 4757  022332  101371                            BHI     4$              ;;NO--TRY IT
 4758  022334  011202                    $KTOUT: MOV     (R2),R2         ;;GET LAST BANK+1
 4759  022336  005037  177572                    CLR     @#SR0           ;;TURN OFF MEMORY MANAGEMENT
 4760  022342  000421                            BR      $SIZEX
 4761  022344  042737  100000  022202    $KTNEX: BIC     #100000,$KT11   ;;KT11 NON-EXISTENT
 4762  022352  012737  022402  000004    $CORE:  MOV     #$CROUT,@#ERRVEC        ;;SET FOR TIMEOUT
 4763  022360  005002                            CLR     R2              ;;SET UP BANK
 4764  022362  062701  004000            1$:     ADD     #4000,R1        ;;INCREMENT BY 1K
 4765  022366  062702  000040                    ADD     #40,R2          ;;1K STEP
 4766  022372  005711                            TST     (R1)            ;;TRAP ON TIME OUT
 4767  022374  022701  177776                    CMP     #177776,R1      ;;LAST ONE
 4768  022400  001370                            BNE     1$              ;;NO--TRY AGAIN
 4769  022402  162701  004000    $CROUT: SUB     #4000,R1
 4770  022406  162702  000040    $SIZEX: SUB     #40,R2          ;;DROP BACK
 4771  022412  010006                            MOV     R0,SP           ;;RESTORE THE STACK
 4772  022414  012637  000006                    MOV     (SP)+,@#ERRVEC+2        ;;RESTORE ERROR VECTOR
 4773  022420  012637  000004                    MOV     (SP)+,@#ERRVEC
 4774  022424  010137  022446                    MOV     R1,$LSTAD       ;;LAST ADDRESS
 4775  022430  010237  022450                    MOV     R2,$LSTBK       ;;LAST BANK
 4776  022434  012603                            MOV     (SP)+,R3        ;;RESTORE R3
 4777  022436  012602                            MOV     (SP)+,R2        ;;RESTORE R2
```

```
 4778  022440  012601                        MOV     (SP)+,R1        ;;RESTORE R1
 4779  022442  012600                        MOV     (SP)+,R0        ;;RESTORE R0
 4780  022444  000207                        RTS     PC
 4781  022446  000000              $LSTAD: .WORD   0               ;;CONTAINS THE LAST ADDRESS
 4782  022450  000000              $LSTBK: .WORD   0               ;;CONTAINS THE LAST BANK
 4783
 4784                              .SBTTL  TRAP DECODER
 4785
 4786                              ;;**************************************************************
 4787                              ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
 4788                              ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
 4789                              ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
 4790                              ;*GO TO THAT ROUTINE.
 4791
 4792  022452  010046              $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
 4793  022454  016600  000002              MOV     2(SP),R0        ;;GET TRAP ADDRESS
 4794  022460  005740                      TST     -(R0)           ;;BACKUP BY 2
 4795  022462  111000                      MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
 4796  022464  006300                      ASL     R0              ;;POSITION FOR INDEXING
 4797  022466  016000  022506              MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
 4798  022472  000200                      RTS     R0              ;;GO TO ROUTINE
 4799
 4800
 4801                              ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
 4802
 4803  022474  011646              $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
 4804  022476  016666  000004  000002      MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
 4805  022504  000002                      RTI                     ;;RESTORE THE PSW
 4806
 4807                              .SBTTL  TRAP TABLE
 4808
 4809                              ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
 4810                              ;*BY THE "TRAP" INSTRUCTION.
 4811
 4812                              ;       ROUTINE
 4813                              ;       -------
 4814  022506  022474              $TRPAD: .WORD   $TRAP2
 4815  022510  017462                      $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
 4816  022512  017726                      $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
 4817  022514  017702                      $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
 4818  022516  017742                      $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
 4819  022520  020130                      $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
 4820
 4821  022522  020752                      $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING
 4822
 4823  022524  020662                      $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
 4824  022526  021224                      $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
 4825  022530  021314                      $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
 4826  022532  022050                      $SAVREG ;;CALL=SAVREG   TRAP+12(104412) SAVE R0-R5 ROUTINE
 4827  022534  022106                      $RESREG ;;CALL=RESREG   TRAP+13(104413) RESTORE R0-R5 ROUTINE
 4828
 4829
 4830                              ;THIS ROUTINE CLEARS THE RP11 AND DETERMINES WHICH DRIVES ARE AVAILABLE.
 4831                              ;THE TABLE 'DRVSTA' IS LOADED TO REFLECT THE SYSTEM STATUS.
 4832
 4833  022536  104412              RPINIT: SAVREG                  ;SAVE R0-R5
```

```
4834  022540  013701  001326          MOV     RPVEC,R1        ;VECTOR ADDRESS
4835  022544  005021                  CLR     (R1)+           ;SET INTERRUPT ADDRESS TO ZERO
4836  022546  013711  001332          MOV     RPPRIO,(R1)     ;RP11 PRIORITY
4837  022552  005037  001304          CLR     DRVTYP          ;CLEAR DRIVE TYPE STORAGE
4838  022556  005037  001370          CLR     DRVSTA          ;SET DRIVE STATUS TO OFFLINE
4839  022562  005037  001372          CLR     DRVSTA+2        ;SET DRIVE STATUS TO OFFLINE
4840  022566  005037  001374          CLR     DRVSTA+4        ;SET DRIVE STATUS TO OFFLINE
4841  022572  005037  001376          CLR     DRVSTA+6        ;SET DRIVE STATUS TO OFFLINE
4842  022576  013704  001324          MOV     RPADR,R4        ;PUT RP11 ADDRESS INTO R4
4843  022602  012764  000001  000004  MOV     #CLEAR,RPCS(R4)  ;CLEAR THE RP11
4844  022610  012764  000001  000004  MOV     #CLEAR,RPCS(R4)  ;CLEAR THE RP11 AGAIN
4845  022616  012701  000007          MOV     #7,R1           ;'DRVSTA' TABLE INDEX
4846  022622  004737  022636  1$:     JSR     PC,DRVINT       ;CHECK THE DRIVE'S STATUS
4847  022626  005301                  DEC     R1              ;DECREMENT THE TABLE INDEX
4848  022630  100374                  BPL     1$              ;BR IF NOT FINISHED
4849  022632  104413                  RESREG                  ;RESTORE R0-R5
4850  022634  000207                  RTS     PC              ;RETURN
4851
4852                          ;ROUTINE TO CHECK THE DRIVE'S STATUS.  DRIVE NUMBER MUST BE IN R1
4853
4854  022636  110164  000005  DRVINT: MOVB    R1,RPCS+1(R4)   ;SELECT DRIVE
4855  022642  032764  001000  000000  BIT     #SUFU,RPDS(R4)  ;SEE IF DRIVE UNSAFE
4856  022650  001404                  BEQ     1$              ;BR IF NOT UNSAFE
4857  022652  112761  177777  001370  MOVB    #-1,DRVSTA(R1)  ;SET INDICATOR TO 'UNSAFE'
4858  022660  000417                  BR      3$              ;EXIT
4859  022662  032764  040000  000000  1$:     BIT     #SUOL,RPDS(R4)  ;IS DRIVE ONLINE ?
4860  022670  001422                  BEQ     4$              ;BR IF NOT
4861  022672  032764  004000  000000  BIT     #SUSI,RPDS(R4)  ;SEEK INCOMPLETE SET ?
4862  022700  001004                  BNE     2$              ;BR IF SET
4863  022702  032764  100000  000000  BIT     #SURDY,RPDS(R4)  ;IS DRIVE READY ?
4864  022710  001412                  BEQ     4$              ;BR IF NOT
4865  022712  112761  000001  001370  2$:     MOVB    #1,DRVSTA(R1)  ;SET DRIVE INDICATOR TO 'ONLINE'
4866  022720  032764  020000  000000  3$:     BIT     #SURP03,RPDS(R4)  ;IS THE DRIVE AN RP03 ?
4867  022726  001403                  BEQ     4$              ;BR IF NOT
4868  022730  156137  001360  001304  BISB    ATABIT(R1),DRVTYP  ;SET RP03 INDICATOR
4869  022736  000207                  4$:     RTS     PC              ;RETURN
4870
4871                          ;DO THE REQUESTED COMMAND.  THE RP11 ADDRESS MUST BE IN R4
4872
4873  022740  104412          RP11:   SAVREG                  ;SAVE R0 - R5
4874  022742  113764  002500  000005  MOVB    DPB,RPCS+1(R4)  ;SELECT THE DRIVE
4875  022750  005001                  CLR     R1              ;CLEAR R1
4876  022752  113701  002500          MOVB    DPB,R1          ;DRIVE ADDRESS
4877  022756  004737  023052          JSR     PC,CIR          ;SEE IF DRIVE IS READY
4878  022762  013746  002504          MOV     SWC,-(SP)       ;COMPLEMENT THE WORD COUNT
4879  022766  005416                  NEG     (SP)            ;COMPLEMENT
4880  022770  012664  000006          MOV     (SP)+,RPWC(R4)  ;LOAD THE WORD COUNT REGISTER
4881  022774  013764  002506  000010  MOV     $BUF,RPBA(R4)   ;BUFFER ADDRESS
4882  023002  013764  002510  000012  MOV     $CYL,RPCA(R4)   ;CYLINDER ADDRESS
4883  023010  013764  002512  000014  MOV     $SEC,RPDA(R4)   ;SECTOR/TRACK ADDRESS
4884  023016  113764  002500  000005  MOVB    DPB,RPCS+1(R4)  ;SELECT THE DRIVE
4885  023024  153764  002503  000005  BISB    DPB+3,RPCS+1(R4)  ;SET 'MODE' & 'HDR' BITS
4886  023032  013746  002502          MOV     DPB+2,-(SP)     ;COMMAND CODE
4887  023036  153716  002501          BISB    DPB+1,(SP)      ;'MEX' BITS
4888  023042  112664  000004          MOVB    (SP)+,RPCS(R4)  ;START THE COMMAND
4889  023046  104413                  RESREG                  ;RESTORE R0 - R5
```

```
                                        RTS     PC              ;RETURN

                                ;SEE IF DRIVE IS STILL ONLINE

                                CIR:
                                        MOV     #2$,$ESCAPE     ;;ESCAPE TO 2$ ON ERROR
                                        JSR     PC,DRVINT       ;CHECK ON THE DRIVE'S STATUS
                                        JSR     PC,SAVRP        ;STORE RP11 REGISTERS IN CASE OF ERROR
                                        TSTB    DRVSTA(R1)      ;IS THE DRIVE STILL ONLINE ?
                                        BGT     3$              ;BR IF IT IS
                                        BLT     1$              ;BR IF DRIVE IS UNSAFE
                                        ERROR   76              ;DRIVE HAS GONE OFFLINE
                                1$:     ERROR   77              ;DRIVE HAS BECOME UNSAFE
                                2$:     BIT     #SWO9,@SWR      ;LOOP ON THE ERROR ?
                                        BNE     CIR             ;BR IF LOOP
                                        BISB    ATABIT(R1),DRVBAD  ;SETUP TO DEASSIGN THE DRIVE
                                        JMP     @#SEOP          ;GO TO THE END OF PASS ROUTINE
                                3$:     CLR     $ESCAPE         ;CLEAR THE 'ESCAPE' FLAG
                                        RTS     PC              ;RETURN

                                ;THIS ROUTINE WILL PROVIDE A STALL IN MILLISECONDS FOR A SPECIFIC
                                ;AMOUNT OF TIME.  STALL VALUE IS STORED IN 'STALLT'.
                                ;       JSR     PC,STALL

                                STALL:  MOV     STALLT,-(SP)    ;PICKUP THE STALL TIME
                                        CLR     -(SP)           ;CLEAR TEMP. LOCATION
                                1$:     SUB     #1,2(SP)        ;MORE STALL REQUIRED?
                                        BLO     3$              ;NO--BRANCH
                                        MOV     #100.,(SP)      ;STALL FOR ABOUT 1 MILLISECOND
                                2$:     TST     R0              ;NOP TO KILL TIME
                                        DEC     (SP)            ;COUNT
                                        BNE     2$              ;LOOP IF MORE COUNTS NEEDED
                                        BR      1$
                                3$:     CMP     (SP)+,(SP)+     ;CLEAN OFF THE STACK
                                        RTS     PC              ;EXIT

                                ;THIS ROUTINE STORES THE RP11E REGISTERS FOR USE BY THE PROGRAM.
                                ;THE PROGRAM DOES NOT USE THE 'LIVE' REGISTERS FOR ERROR DETERMINATION.
                                ;CALL
                                ;       JSR     PC,SAVRP
                                ;       RETURN

                                SAVRP:  MOV     R4,-(SP)        ;SAVE R4
                                        MOV     RPADR,R4        ;RP11 ADDRESS
                                        MOV     RPDS(R4),$RPDS  ;STORE RPDS
                                        MOV     RPER(R4),$RPER  ;STORE RPER
                                        MOV     RPCS(R4),$RPCS  ;STORE RPCS
                                        MOV     RPWC(R4),$RPWC  ;STORE RPWC
                                        MOV     RPBA(R4),$RPBA  ;STORE RPBA
                                        MOV     RPCA(R4),$RPCA  ;STORE RPCA
                                        MOV     RPDA(R4),$RPDA  ;STORE RPDA
                                        MOV     RPMI(R4),$RPMI  ;STORE RPMI
                                        MOV     SUCA(R4),$SUCA  ;STORE SUCA
                                        MOV     SILO(R4),$SILO  ;STORE SILO
                                        MOV     (SP)+,R4        ;RESTORE R4
                                        RTS     PC              ;RETURN
```

```
                                        ;ROUTINE TO MOVE THE LOADER FROM HIGH MEMORY TO JUST ABOVE THE PROGRAM
                                        ;
023300  105737  000041        MOVLDR: TSTB    41              ;WHO LOADED THE PROGRAM
023304  001005                        BNE     1$              ;BR IF NOT 'ABS'
023306  012702  000144                MOV     #100.,R2        ;'ABS' LOADER SIZE
023312  012701  044112                MOV     #BUFFER+200.,R1 ;RELOCATE TO HERE
023316  000404                        BR      2$              ;FINISH
023320  012702  002734        1$:     MOV     #1500.,R2       ;'XXDP' SIZE
023324  012701  051472                MOV     #BUFFER+3000.,R1  ;RELOCATE TO HERE (UPPER BOUNDARY)
023330  012703  157776        2$:     MOV     #157776,R3      ;TRY THIS UPPER MEMORY ADDRESS
023334  005737  022202                TST     $KT11           ;MEMORY MANAGEMENT ?
023340  100402                        BMI     3$              ;BR IF YES
023342  013703  022446                MOV     $LSTAD,R3       ;USE ACTUAL HIGH ADDRESS
023346  005046        3$:     CLR     -(SP)           ;CLEAR THE STACK
023350  016616  000002                MOV     2(SP),(SP)      ;MOVE THE RETURN ADDRESS DOWN THE STACK
023354  010166  000002                MOV     R1,2(SP)        ;BEGINNING ADDRESS OF BUFFER
023360  005737  001322                TST     LDRFLG          ;LOADER ALREADY RELOCATED ?
023364  001006                        BNE     5$              ;BR IF YES
023366  014341        4$:     MOV     -(R3),-(R1)     ;MOVE A WORD
023370  005302                        DEC     R2              ;COUNT IT
023372  001375                        BNE     4$              ;BR IF MORE TO GO
023374  012737  177777  001322        MOV     #-1,LDRFLG      ;SET THE LOADER MOVED FLAG
023402  000207        5$:     RTS     PC              ;RETURN
                                        ;ROUTINE TO RELOCATE THE LOADER TO UPPER MEMORY
                                        ;
023404  104412        RESLDR: SAVREG                  ;SAVE R0 - R5
023406  005737  001322                TST     LDRFLG          ;LOADER MOVED ?
023412  001430                        BEQ     5$              ;BR IF NOT
023414  105737  000041                TSTB    41              ;'ABS' LOADER
023420  001005                        BNE     1$              ;BR IF NOT
023422  012702  000144                MOV     #100.,R2        ;'ABS' LOADER SIZE
023426  012701  044112                MOV     #BUFFER+200.,R1 ;RELOCATE FROM HERE
023432  000404                        BR      3$              ;FINISH
023434  012702  002734        1$:     MOV     #1500.,R2       ;'XXDP' SIZE
023440  012701  051472                MOV     #BUFFER+3000.,R1  ;RELOCATE FROM HERE (UPPER BOUNDARY)
023444  012703  157776        3$:     MOV     #157776,R3      ;TRY THIS UPPER MEMORY ADDRESS
023450  005737  022202                TST     $KT11           ;MEMORY MANAGEMENT ?
023454  100402                        BMI     4$              ;BR IF YES
023456  013703  022446                MOV     $LSTAD,R3       ;USE ACTUAL HIGH ADDRESS
023462  014143        4$:     MOV     -(R1),-(R3)     ;MOVE A WORD
023464  005302                        DEC     R2              ;COUNT IT
023466  001375                        BNE     4$              ;BR IF MORE TO GO
023470  005037  001322                CLR     LDRFLG          ;CLEAR THE LOADER MOVED FLAG
023474  104413        5$:     RESPEG                  ;RESTORE R0 - R5
023476  000207                        RTS     PC              ;RETURN
                                        ;
                                        ;ROUTINE TO WAIT FOR DRIVE READY
                                        ;
023500  032764  100000  000000  DRVRDY: BIT     #SURDY,RPDS(R4)  ;DRIVE READY ?
023506  001042                        BNE     6$              ;BR IF READY
023510  013746  001212                MOV     TIMOUT,-(SP)    ;WAIT TIME: APPROX 1 SEC
023514  005046                        CLR     -(SP)           ;CLEAR WORKING LOCATION
023516  162766  000001  000002  1$:     SUB     #1,2(SP)        ;ANY TIME LEFT IN WAIT LOOP ?
023524  001411                        BEQ     3$              ;BR IF NOT
```

```
023526  012716  000144                  MOV     #100.,(SP)              ;VALUE FOR APPROX 1 MS
023532  032764  100000  000000  2$:     BIT     #SURDY,RPDS(R4)         ;DRIVE READY ?
023540  001023                          BNE     5$                      ;BR IF READY
023542  005316                          DEC     (SP)                    ;COUNT SOME TIME
023544  001372                          BNE     2$                      ;BR IF NOT FINISHED
023546  000763                          BR      1$                      ;COUNT MAJOR TIME AGAIN
023550  004737  023172          3$:     JSR     PC,SAVRP                ;STORE RP11 REGISTERS FOR ERROR MESSAGE
023554  012737  023564  001164          MOV     #4$,$ESCAPE             ;;ESCAPE TO 4$ ON ERROR
023562  104100                          ERROR   100                     ;DRIVE TIMED OUT
023564  005037  001164          4$:     CLR     $ESCAPE                 ;CLEAR THE 'ESCAPE' FLAG
023570  062706  000004                  ADD     #4,SP                   ;CORRECT  THE STACK POINTER
023574  032777  001000  155336          BIT     #SW09,@SWR              ;LOOP ON THE ERROR ?
023602  001336                          BNE     DRVRDY                  ;LOOP IF SET
023604  000177  155370                  JMP     @BYPASS                 ;GO THE 'BYPASS' ADDRESS
023610  062706  000004          5$:     ADD     #4,SP                   ;CORRECT THE STACK POINTER
023614  000207                  6$:     RTS     PC                      ;RETURN


                                ;ROUTINE TO WAIT FOR CONTROLLER READY

023616  032764  000200  000004  CONRDY: BIT     #RDY,RPCS(R4)           ;CONTROLLER READY ?
023624  001042                          BNE     6$                      ;BR IF READY
023626  013746  001212                  MOV     TIMOUT,-(SP)            ;WAIT TIME: APPROX 1 SEC
023632  005046                          CLR     -(SP)                   ;CLEAR WORKING LOCATION
023634  162766  000001  000002  1$:     SUB     #1,2(SP)                ;ANY TIME LEFT IN WAIT LOOP ?
023642  001411                          BEQ     3$                      ;BR IF NOT
023644  012716  000144                  MOV     #100.,(SP)              ;VALUE FOR APPROX 1 MS
023650  032764  000200  000004  2$:     BIT     #RDY,RPCS(R4)           ;CONTROLLER READY ?
023656  001023                          BNE     5$                      ;BR IF READY
023660  005316                          DEC     (SP)                    ;COUNT SOME TIME
023662  001372                          BNE     2$                      ;BR IF NOT FINISHED
023664  000763                          BR      1$                      ;COUNT MAJOR TIME AGAIN
023666  004737  023172          3$:     JSR     PC,SAVRP                ;STORE RP11 REGISTERS FOR ERROR MESSAGE
023672  012737  023702  001164          MOV     #4$,$ESCAPE             ;;ESCAPE TO 4$ ON ERROR
023700  104101                          ERROR   101                     ;CONTROLLER TIMED OUT
023702  005037  001164          4$:     CLR     $ESCAPE                 ;CLEAR THE 'ESCAPE' FLAG
023706  062706  000004                  ADD     #4,SP                   ;CORRECT THE STACK POINTER
023712  032777  001000  155220          BIT     #SW09,@SWR              ;LOOP ON THE ERROR ?
023720  001336                          BNE     CONRDY                  ;BR IF LOOPING
023722  000177  155252                  JMP     @BYPASS                 ;'BYPASS' THE TEST
023726  062706  000004          5$:     ADD     #4,SP                   ;CORRECT THE STACK POINTER
023732  000207                  6$:     RTS     PC                      ;RETURN

                                ;ROUTINE TO DETERMINE THE NUMBER OF SECTORS ON THE DRIVE BEING TESTED

023734  010046                  SECNBR: MOV     R0,-(SP)                ;SAVE R0
023736  012737  117230  001244          MOV     #40600.,PAKSIZ          ;START WITH THE NUMBER OF SECTORS ON
023744  005037  001246                  CLR     PAKSIZ+2                ;AN RP02
023750  005000                          CLR     R0                      ;USE R0 AS AN INDEX
023752  113700  002500                  MOVB    DPB,R0                  ;GET THE DRIVE NUMBER
023756  136037  001360  001304          BITB    ATABIT(R0),DRVTYP   ;IS THE DRIVE AN RP03 ?
023764  001404                          BEQ     1$                      ;BR IF NOT
023766  006337  001244                  ASL     PAKSIZ                  ;DOUBLE THE SECTOR COUNT
023772  006137  001246                  ROL     PAKSIZ+2                ;GET THE CARRY
023776  012600                  1$:     MOV     (SP)+,R0                ;RESTORE R0
024000  000207                          RTS     PC                      ;RETURN
```

```
5058
5059                                    ;ROUTINE TO SELECT A PATTERN
5060
5061  024002  010046                    PATSEL: MOV     R0,-(SP)        ;SAVE R0
5062  024004  005237  001204            1$:     INC     PATNUM          ;INCREMENT THE PATTERN NUMBER
5063  024010  022737  000020  001204            CMP     #16.,PATNUM     ;NUMBER AT MAXIMUM
5064  024016  101002                            BHI     2$              ;BR IF NOT
5065  024020  005037  001204                    CLR     PATNUM          ;START AT ZERO AGAIN
5066  024024  013700  001204            2$:     MOV     PATNUM,R0       ;SEE IF THE PATTERN IS IN USE
5067  024030  006300                            ASL     R0              ;CONVERT INTO A TABLE INDEX
5068  024032  036037  001400  001300            BIT     BITS(R0),PATRN  ;INUSE INDICATOR SET FOR THIS PATTERN ?
5069  024040  001761                            BEQ     1$              ;BR IF NOT
5070  024042  012600                            MOV     (SP)+,R0        ;RESTORE R0
5071  024044  000207                            RTS     PC              ;RETURN
5072
5073                                    ;ROUTINE TO INCREMENT THE DISK ADDRESS
5074                                    ;       RETURN + 2 WHEN THE END OF THE PACK IS REACHED
5075
5076  024046  005737  001260            INCADR: TST     OPRSEL          ;USING AN OPERATOR SPECIFIED ADDRESS ?
5077  024052  001066                            BNE     5$              ;BR IF YES
5078  024054  005737  001246                    TST     PAKSIZ+2        ;SECTORS LEFT ?
5079  024060  003017                            BGT     1$              ;BR IF AN UPPER COUNT
5080  024062  005737  001244                    TST     PAKSIZ          ;LOWER COUNT AT ZERO ?
5081  024066  001460                            BEQ     5$              ;BR IF IT IS
5082  024070  023737  001244  001254            CMP     PAKSIZ,BLKS14   ;ENOUGH SECTORS LEFT FOR A FULL TRANSFER ?
5083  024076  103010                            BHIS    1$              ;BR IF THERE ARE
5084  024100  013737  001244  002504            MOV     PAKSIZ,$WC      ;USE THE RESIDUE COUNT FOR THIS TRANSFER
5085  024106  000337  002504                    SWAB    $WC             ;CONVERT TO A WORD COUNT
5086  024112  005037  001244                    CLR     PAKSIZ          ;FORCE COUNT TO ZERO
5087  024116  000410                            BR      2$              ;GO AND INCREMENT THE ADDRESS
5088  024120  013737  001256  002504  1$:       MOV     WC14,$WC        ;USE THE STANDARD WORD COUNT
5089  024126  163737  001254  001244            SUB     BLKS14,PAKSIZ   ;DECREMENT THE TOTAL SECTORS AVAILABLE
5090  024134  005637  001246                    SBC     PAKSIZ+2        ;SUBTRACT ANY CARRY
5091  024140  005046                    2$:     CLR     -(SP)           ;CLEAR THE STACK
5092  024142  113716  002512                    MOVB    $SEC,(SP)       ;GET CURRENT SECTOR
5093  024146  063716  001250                    ADD     INCSEC,(SP)     ;ADD THE SECTOR INCREMENT
5094  024152  121627  000012                    CMPB    (SP),#10.       ;EXCEED THE MAXIMUM ?
5095  024156  002404                            BLT     3$              ;BR IF NOT
5096  024160  162716  000012                    SUB     #10.,(SP)       ;CORRECT THE RESIDUE
5097  024164  105237  002513                    INCB    $TRK            ;INCREMENT THE TRACK
5098  024170  111637  002512            3$:     MOVB    (SP),$SEC       ;NEW SECTOR ADDRESS
5099  024174  113716  002513                    MOVB    $TRK,(SP)       ;GET CURRENT TRACK VALUE
5100  024200  063716  001252                    ADD     INCTRK,(SP)     ;ADD THE TRACK INCREMENT
5101  024204  121627  000024                    CMPB    (SP),#20.       ;EXCEED THE MAXIMUM ?
5102  024210  002404                            BLT     4$              ;BR IF NOT
5103  024212  162716  000024                    SUB     #20.,(SP)       ;CORRECT THE SIZE
5104  024216  005237  002510                    INC     $CYL            ;INCREMENT THE CYLINDER ADDRESS
5105  024222  112637  002513            4$:     MOVB    (SP)+,$TRK      ;NEW TRACK ADDRESS
5106  024226  000402                            BR      6$              ;EXIT
5107  024230  062716  000002            5$:     ADD     #2,(SP)         ;INCREMENT RETURN - AT END
5108  024234  000207                    6$:     RTS     PC              ;RETURN
5109
5110                                    ;ROUTINE TO SETUP A BUFFER WITH THE SEQUENTIAL HEADERS FOR CYLINDER 0
5111                                    ;       TRACK 0
5112
5113  024236  104412                    LODSEC: SAVREG                  ;SAVE R0 - R5
```

```
5114  024240  013701  002506          MOV     $BUF,R1         ;BUFFER ADDRESS
5115  024244  005003                  CLR     R3              ;SECTOR COUNTER
5116  024246  005021          1$:     CLR     (R1)+           ;UPPER WORD OF HEADER
5117  024250  005021                  CLR     (R1)+           ;CYLINDER/TRACK WORD OF HEADER
5118  024252  010321                  MOV     R3,(R1)+        ;SECTOR ADDRESS
5119  024254  005203                  INC     R3              ;INCREMENT THE SECTOR ADDRESS
5120  024256  022703  000012          CMP     #10.,R3         ;FINISHED ?
5121  024262  001371                  BNE     1$              ;BR IF NOT
5122  024264  104413                  RESREG                  ;RESTORE R0 - R5
5123  024266  000207                  RTS     PC              ;RETURN
5124
5125                          ;FILL THE BUFFER WITH THE SECTOR ADDRESS (CONTAINED IN '$SEC')
5126
5127  024270  104412          FILSEC: SAVREG                  ;STORE R0-R5
5128  024272  005003                  CLR     R3              ;CLEAR THE INDEX
5129  024274  113703  002512          MOVB    $SEC,R3         ;SECTOR ADDRESS
5130  024300  001002                  BNE     1$              ;BR IF NOT SECTOR ZERO
5131  024302  012703  000012          MOV     #10.,R3         ;USE 10. AS VALUE FOR SECTOR ZERO
5132  024306  013702  002504  1$:     MOV     $WC,R2          ;WORD COUNT
5133  024312  013701  002506          MOV     $BUF,R1         ;BUFFER START ADDRESS
5134  024316  010321          2$:     MOV     R3,(R1)+        ;MOVE DATA INTO THE BUFFER
5135  024320  005302                  DEC     R2              ;DECREMENT THE COUNTER
5136  024322  001375                  BNE     2$              ;BR IF MORE TO DO
5137  024324  104413                  RESREG                  ;RESTORE R0-R5
5138  024326  000207                  RTS     PC              ;RETURN
5139
5140                          ;ROUTINE TO FILL THE BUFFER WITH THE TRACK ADDRESS (CONTAINED IN '$TRK')
5141
5142  024330  104412          FILTRK: SAVREG                  ;STORE R0-R5
5143  024332  012701  043603          MOV     #BUFFER+1,R1    ;BUFFER ADDRESS
5144  024336  013702  002504          MOV     $WC,R2          ;WORD COUNT
5145  024342  005003                  CLR     R3              ;CLEAR THE DATA STORAGE
5146  024344  113711  002513  1$:     MOVB    $TRK,(R1)       ;MOVE TRACK ADDRESS INTO THE BUFFER
5147  024350  062701  000002          ADD     #2,R1           ;INCREMENT THE BUFFER POINTER
5148  024354  005302                  DEC     R2              ;DECREMENT THE COUNTER
5149  024356  001372                  BNE     1$              ;BR IF MORE TO DO
5150  024360  104413                  RESREG                  ;RESTORE R0-R5
5151  024362  000207                  RTS     PC              ;RETURN
5152
5153                          ;THIS ROUTINE WILL MOVE THE 16 WORDS OF THE
5154                          ;DESIRED PATTERN INTO THE DATA BUFFER.
5155
5156  024364  104412          SETBUF: SAVREG                  ;SAVE R0 - R5
5157  024366  013700  001204          MOV     PATNUM,R0       ;GET THE PATTERN NUMBER
5158  024372  006300                  ASL     R0              ;CONVERT NUMBER TO INDEX
5159  024374  013701  002506          MOV     $BUF,R1         ;FIRST ADDRESS
5160  024400  013702  002504          MOV     $WC,R2          ;WORD COUNT
5161  024404  016003  001440  1$:     MOV     PAT.PT(R0),R3   ;PICKUP PATTERN POINTER
5162  024410  012704  000020          MOV     #16.,R4         ;PATTERN COUNTER
5163  024414  012321          2$:     MOV     (R3)+,(R1)+     ;MOVE WORD 'N' INTO DATA BUFFER
5164  024416  005302                  DEC     R2              ;DECREMENT THE COUNTER
5165  024420  001403                  BEQ     3$              ;BR IF DONE
5166  024422  005304                  DEC     R4              ;DECREMENT THE PATTERN COUNTER
5167  024424  001767                  BEQ     1$              ;START PATTERN AGAIN
5168  024426  000772                  BR      2$              ;CONTINUE
5169  024430  104413          3$:     RESREG                  ;RESTORE R0 - R5
```

```
5170  024432  000207                          RTS     PC              ;RETURN
5171
5172                                  ;THIS ROUTINE LOADS THE PHYSICAL TRANSFER ADDRESS INTO 'SBUF'
5173
5174  024434  013737  001224  002506  GETBUF: MOV     ACTMEM,SBUF     ;LOWER 16 BITS OF PHYSICAL ADDRESS
5175  024442  013746  001226          MOV     ACTMEM+2,-(SP)  ;UPPER ADDRESS BITS ON THE STACK
5176  024446  042716  177774          BIC     #^C3,(SP)       ;LEAVE ONLY THE LOWER 2 BITS
5177  024452  006316                  ASL     (SP)            ;ALIGN UPPER ADDRESS BITS
5178  024454  006316                  ASL     (SP)            ;ALIGN UPPER ADDRESS BITS
5179  024456  006316                  ASL     (SP)            ;ALIGN UPPER ADDRESS BITS
5180  024460  006316                  ASL     (SP)            ;ALIGN UPPER ADDRESS BITS
5181  024462  112637  002501          MOVB    (SP)+,DPB+1     ;LOAD 'MEX' BITS
5182  024466  013746  002504          MOV     $WC,-(SP)       ;PUT WORD COUNT ON THE STACK
5183  024472  006316                  ASL     (SP)            ;DOUBLE THE WORD COUNT
5184  024474  062637  001224          ADD     (SP)+,ACTMEM    ;FORM NEXT STARTING ADDRESS
5185  024500  005537  001226          ADC     ACTMEM+2        ;ADD ANY CARRY
5186  024504  023737  001226  001236  CMP     ACTMEM+2,HIMEM+2  ;EXCEEDING HIGHEST BANK ADDRESS
5187  024512  101005                  BHI     1$              ;BR IF HIGHER
5188  024514  103413                  BLO     2$              ;BR IF LOWER
5189  024516  023737  001224  001234  CMP     ACTMEM,HIMEM    ;CHECK LOWER 16 BITS
5190  024524  103407                  BLO     2$              ;BR IF LOWER
5191  024526  013737  001240  001224  1$:     MOV     LOMEM,ACTMEM    ;RESET ADDRESS
5192  024534  013737  001242  001226          MOV     LOMEM+2,ACTMEM+2  ;UPPER ADDRESS BITS
5193  024542  000734                  BR      GETBUF          ;TRY AGAIN
5194  024544  000207                  2$:     RTS     PC              ;RETURN
5195
5196                                  ;THIS ROUTINE COMPARES A 16 WORD DATA PATTERN AGAINST THE
5197                                  ;DATA BUFFER.  THE PATTERN NUMBER IS IN 'PATNUM'
5198
5199  024546  104412                  DATCMP: SAVREG                  ;SAVE R0-R5
5200  024550  005737  001222                  TST     MMACTV          ;MEMORY MANAGEMENT ACTIVE ?
5201  024554  001510                  BEQ     2$              ;BR IF NOT
```

```
5202  024556  005037  172340                  CLR     KIPAR0          ;LOAD ADDRESS REGISTER 0
5203  024562  012737  000200  172342          MOV     #200,KIPAR1     ;LOAD ADDRESS REGISTER 1
5204  024570  012737  000400  172344          MOV     #400,KIPAR2     ;LOAD ADDRESS REGISTER 2
5205  024576  012737  177600  172356          MOV     #177600,KIPAR7  ;ACCESS TO I/O PAGE
5206  024604  005046                          CLR     -(SP)           ;MAKE ROOM ON THE STACK
5207  024606  005046                          CLR     -(SP)           ;MAKE ROOM ON THE STACK
5208  024610  013766  002506  000002          MOV     $BUF,2(SP)      ;LOWER 16 BITS OF STARTING ADDRESS
5209  024616  113716  002501                  MOVB    DPB+1,(SP)      ;UPPER ADDRESS BITS
5210  024622  006216                          ASR     (SP)            ;RIGHT JUSTIFY THE ADDRESS BITS
5211  024624  006216                          ASR     (SP)            ;RIGHT JUSTIFY THE ADDRESS BITS
5212  024626  006216                          ASR     (SP)            ;RIGHT JUSTIFY THE ADDRESS BITS
5213  024630  006216                          ASR     (SP)            ;RIGHT JUSTIFY THE ADDRESS BITS
5214  024632  012703  000006                  MOV     #6,R3           ;LOAD COUNTER
5215  024636  006216                  1$:     ASR     (SP)            ;CONVERT ADDRESS TO BLOCK ADDRESS
5216  024640  006066  000002                  ROR     2(SP)           ;LOWER 16 BITS
5217  024644  005303                          DEC     R3              ;DECREMENT THE COUNTER
5218  024646  001373                          BNE     1$              ;CONTINUE UNTIL FINISHED
5219  024650  042766  000177  000002          BIC     #177,2(SP)      ;CLEAR LOWER BLOCK ADDRESS BITS
5220  024656  016637  000002  172346          MOV     2(SP),KIPAR3    ;LOAD ADDRESS REGISTER 3
5221  024664  062766  000200  000002          ADD     #200,2(SP)      ;INCREMENT BLOCK ADDRESS BY 4K
5222  024672  016637  000002  172350          MOV     2(SP),KIPAR4    ;LOAD ADDRESS REGISTER # 4
5223  024700  012737  077406  172300          MOV     #77406,KIPDR0   ;LOAD DESCRIPTOR REGISTER # 0
5224  024706  012737  077406  172302          MOV     #77406,KIPDR1   ;LOAD DESCRIPTOR REGISTER # 1
5225  024714  012737  077406  172304          MOV     #77406,KIPDR2   ;LOAD DESCRIPTOR REGISTER # 2
5226  024722  012737  077406  172306          MOV     #77406,KIPDR3   ;LOAD DESCRIPTOR REGISTER # 3
5227  024730  012737  077406  172310          MOV     #77406,KIPDR4   ;LOAD DESCRIPTOR REGISTER # 4
5228  024736  012737  077406  172316          MOV     #77406,KIPDR7   ;LOAD DESCRIPTOR REGISTER # 7
5229  024744  023727  002504  010000          CMP     $WC,#4096.      ;WAS TRANSFER GREATER THAN 4K ?
5230  024752  101411                          BLOS    2$              ;BR IF NOT
5231  024754  062766  000200  000002          ADD     #200,2(SP)      ;INCREMENT THE ADDRESS BY 4K
5232  024762  016637  000002  172352          MOV     2(SP),KIPAR5    ;LOAD ADDRESS REGISTER # 5
5233  024770  012737  077406  172312          MOV     #77406,KIPDR5   ;LOAD DESCRIPTOR REGISTER # 5
5234  024776  013701  002506          2$:     MOV     $BUF,R1         ;LOAD STARTING ADDRESS
5235  025002  005737  001222                  TST     MMACTV          ;MEMORY MANAGEMENT ACTIVE ?
5236  025006  001425                          BEQ     CMPAR           ;BR IF NOT
5237  025010  042701  140000                  BIC     #140000,R1      ;CHANGE PHYSICAL BASE TO VIRTUAL BASE
5238  025014  052701  060000                  BIS     #60000,R1       ;START AT VIRTUAL 12K
5239  025020  062706  000004                  ADD     #4,SP           ;CORRECT THE STACK POINTER
5240  025024  012737  025042  000004          MOV     #3$,ERRVEC      ;CHANGE THE ERROR VECTOR
5241  025032  012737  000020  172516          MOV     #20,@#SR3       ;ENABLE 22 BIT ADDRESSING MODE
5242  025040  000402                          BR      4$              ;BR IF AN 11/70
5243  025042  062706  000004          3$:     ADD     #4,SP           ;CORRECT THE STACK POINTER
5244  025046  012737  000006  000004  4$:     MOV     #ERRVEC+2,ERRVEC  ;RESTORE THE ERROR VECTOR
5245  025054  012737  000001  177572          MOV     #1,@#SR0        ;ENABLE MEMORY MANAGEMENT
5246  025062  013737  001342  025352  CMPAR:  MOV     $RPWC,CMCNT     ;WORD COUNT TO WORKING LOCATION
5247  025070  063737  002504  025352          ADD     $WC,CMCNT       ;CALCULATE ACTUAL WORDS TRANSFERED
5248  025076  013737  002510  025354          MOV     $CYL,CMCYL      ;CYLINDER ADDRESS
5249  025104  113737  002513  025360          MOVB    $TRK,CMTRK      ;TRACK ADDRESS
5250  025112  113737  002512  025356          MOVB    $SEC,CMSEC      ;SECTOR ADDRESS
5251  025120  013737  001320  025350  CMSTR:  MOV     CMP.CT,LIMIT    ;DISPLAY LIMIT
5252  025126  005237  025350                  INC     LIMIT           ;CONVERT PARAMETER INTO LIMIT VALUE
5253  025132  010137  025362                  MOV     R1,CMBUF        ;STARTING ADDRESS OF SECTOR BUFFER
5254  025136  005037  025344                  CLR     FRSTER          ;CLEAR 'FIRST ERROR' INDICATOR
5255  025142  005037  025346                  CLR     ERCTR           ;CLEAR ERROR COUNTER
5256  025146  023727  025352  000400          CMP     CMCNT,#256.     ;IS BUFFER SIZE GREATER THAN ONE SECTOR ?
5257  025154  101003                          BHI     1$              ;BR IF IT IS
```

```
5258  025156  013702  025352                  MOV     CMCNT,R2        ;LESS THAN, USE REMAINING BUFFER
5259  025162  000402                          BR      2$              ;
5260  025164  012702  000400          1$:     MOV     #256.,R2        ;COMPARE SECTOR
5261  025170  160237  025352          2$:     SUB     R2,CMCNT        ;DECREMENT WORD COUNT
5262  025174  013704  001204                  MOV     PATN[illegible]R4        ;PATTERN NUMBER
5263  025200  006304                          ASL     R4              ;GENERATE INTO AN INDEX
5264  025202  012703  000020          CMDAT:  MOV     #16.,R3         ;R3 IS PATTERN POSITION COUNTER
5265  025206  016405  001440                  MOV     PAT.PT(R4),R5   ;PATTERN ADDRESS
5266  025212  022125                  1$:     CMP     (R1)+,(R5)+     ;COMPARE BUFFER WITH PATTERN
5267  025214  001406                          BEQ     2$              ;BR IF EQUAL
5268  025216  032777  000010  153714          BIT     #SW03,@SWR      ;SWITCH 3 SET ?
5269  025224  001002                          BNE     2$              ;BR IF NOT
5270  025226  004737  025364                  JSR     PC,CMPRT        ;TYPE THE ERROR
5271  025232  005302                  2$:     DEC     R2              ;DECREMENT SIZE COUNT
5272  025234  001403                          BEQ     3$              ;BR WHEN AT END
5273  025236  005303                          DEC     R3              ;DECREMENT PATT POS COUNT
5274  025240  001364                          BNE     1$              ;BR IF NOT AT END OF PATT
5275  025242  000757                          BR      CMDAT           ;RESTART THE PATTERN
5276  025244  004737  025562          3$:     JSR     PC,ENDCMP       ;PRINT LAST LINE (IF ERRORS)
5277  025250  005737  025352                  TST     CMCNT           ;AT END OF BUFFER
5278  025254  003424                          BLE     CMPRX           ;BR IF AT END
5279  025256  005237  025356                  INC     CMSEC           ;INCREMENT SECTOR
5280  025262  023727  025356  000012          CMP     CMSEC,#10.      ;SECTOR GREATER THAN MAX ?
5281  025270  103713                          BLO     CMSTR           ;BR IF NOT GREATER THAN MAX
5282  025272  005037  025356                  CLR     CMSEC           ;CLEAR SECTOR ADDRESS
5283  025276  005237  025360                  INC     CMTRK           ;INCREMENT TRACK
5284  025302  023727  025360  000024          CMP     CMTRK,#20.      ;TRACK GREATER THAN MAX ?
5285  025310  103703                          BLO     CMSTR           ;BR IF NOT GREATER
5286  025312  005037  025360                  CLR     CMTRK           ;RESET TRACK ADDRESS
5287  025316  005237  025354                  INC     CMCYL           ;INCREMENT CYLINDER ADDRESS
5288  025322  000137  025120                  JMP     CMSTR           ;CONTINUE WITH COMPARE
5289  025326  104413          CMPRX:  RESREG                          ;RESTORE R0 - R5
5290  025330  005737  001222                  TST     MMACTV          ;MEMORY MANAGEMENT ACTIVE ?
5291  025334  001402                          BEQ     1$              ;BR IF NOT
5292  025336  005037  177572                  CLR     @#SR0           ;TURN OFF MEMORY MANAGEMENT
5293  025342  000207                  1$:     RTS     PC
5294
5295  025344  000000          FRSTER: .WORD   0               ;FIRST ERROR INDICATOR
5296  025346  000000          ERCTR:  .WORD   0               ;NUMBER OF ERRORS
5297  025350  000000          LIMIT:  .WORD   0               ;DISPLAY LIMIT
5298  025352  000000          CMCNT:  .WORD   0               ;WORD COUNT
5299  025354  000000          CMCYL:  .WORD   0               ;CYLINDER ADDRESS
5300  025356  000000          CMSEC:  .WORD   0               ;SECTOR ADDRESS
5301  025360  000000          CMTRK:  .WORD   0               ;TRACK ADDRESS
5302  025362  000000          CMBUF:  .WORD   0               ;BEGINNING ADDRESS OF SECTOR
5303
5304                          ;TYPE DATA COMPARE ERRORS
5305
5306  025364  005737  025344          CMPRT:  TST     FRSTER          ;FIRST ERROR?
5307  025370  001037                          BNE     CMPRT1          ;BR IF NOT
5308  025372  005737  001222                  TST     MMACTV          ;MEMORY MANAGEMENT ACTIVE ?
5309  025376  001011                          BNE     1$              ;BR IF ACTIVE
5310  025400  012737  025400  001116          MOV     #.,$ERRPC       ;PC FOR ERROR MESSAGE
5311  025406  112737  000102  001114          MOVB    #102,$ITEMB     ;COMPARSION ERROR MESSAGE NUMBER
5312  025414  004737  017232                  JSR     PC,TYPERR       ;REPORT THE ERROR
5313  025420  000412                          BR      2$              ;CONTINUE
```

```
5314  025422  005037  177572          1$:     CLR     @#SR0           ;TURN OFF MEMORY MANAGEMENT
5315  025426  012737  025426  001116          MOV     #.,$ERRPC       ;PC FOR ERROR MESSAGE
5316  025434  112737  000103  001114          MOVB    #103,$ITEMB     ;COMPARSION ERROR MESSAGE WITH MM REGISTERS
5317  025442  004737  017232                  JSR     PC,TYPERR       ;REPORT THE COMPARSION ERROR
5318  025446  012737  177777  025344  2$:     MOV     #-1,FRSTER      ;SET FIRST ERROR INDICATOR
5319  025454  005737  001222                  TST     MMACTV          ;MEMORY MANAGEMENT ACTIVE ?
5320  025460  001403                          BEQ     CMPRT1          ;BR IF OT
5321  025462  012737  000001  177572          MOV     #1,@#SR0        ;ENABLE MEMORY MANAGEMENT AGAIN
5322  025470  005737  025350          CMPRT1: TST     LIMIT           ;TYPEOUT LIMIT REACHED ?
5323  025474  001403                          BEQ     1$              ;BR IF IT HAS
5324  025476  005337  025350                  DEC      LIMIT          ;DECREMENT LIMIT COUNTER
5325  025502  001004                          BNE     2$              ;BR IF NOT AT LIMIT
5326  025504  032777  000020  153426  1$:     BIT     #SW04,@SWR      ;PRINT ALL DATA COMPARE ERRORS ?
5327  025512  001420                          BEQ     3$              ;BR IF NOT
5328  025514  010137  001122          2$:     MOV     R1,$BDADR       ;ADDRESS OF BAD WORD
5329  025520  162737  000002  001122          SUB     #2,$BDADR       ;ADJUST ADDRESS
5330  025526  016537  177776  001124          MOV     -2(R5),$GDDAT   ;GOOD DATA
5331  025534  016137  177776  001126          MOV     -2(R1),$BDDAT   ;BAD DATA
5332  025542  112737  000104  001114          MOVB    #104,$ITEMB     ;ERROR TABLE INDEX
5333  025550  004737  017232                  JSR     PC,TYPERR       ;DISPLAY WORD THAT DIDN'T COMPARE
5334  025554  005237  025346          3$:     INC     ERCTR           ;COUNT THE ERROR
5335  025560  000207                          RTS     PC              ;RETURN
5336
5337                                  ;LAST LINE OF COMPARE ERROR REPORTING
5338
5339  025562  005737  025346          ENDCMP: TST     ERCTR           ;SEE IF ANY ERRORS
5340  025566  001406                          BEQ     2$              ;BR IF NONE
5341  025570  012737  025600  001164          MOV     #1$,$ESCAPE     ;;ESCAPE TO 1$ ON ERROR
5342  025576  104105                          ERROR   105             ;NUMBER OF COMPARE ERRORS
5343  025600  005037  001164          1$:     CLR     $ESCAPE         ;CLEAR THE ERROR ESCAPE ADDRESS
5344  025604  000207                  2$:     RTS     PC              ;RETURN
5345
5346                                  ;ROUTINE TO CLEAR THE RP11E
5347                                  ;
5348                                  ;THE FIRST 'CLEAR' TERMINATES ANY ORDER WHICH IS IN PROGRESS.  THE SECOND
5349                                  ;'CLEAR' CLEARS THE 'PROGRAM ERROR' WHICH MAY HAVE BEEN SET BY THE FIRST
5350                                  ;'CLEAR'.
5351                                  ;
5352                                  ;NOTE:  R4 MUST CONTAIN THE BASE ADDRESS OF THE RP11E
5353
5354  025606  012764  000001  000004  CLRP:   MOV     #CLEAR,RPCS(R4) ;CLEAR THE RP11
5355  025614  012764  000001  000004          MOV     #CLEAR,RPCS(R4) ;CLEAR THE RP11
5356  025622  000207                          RTS     PC              ;RETURN
5357
5358                                  ;ROUTINE TO SIZE MEMORY AND TO SETUP FOR TEST 14
5359
5360  025624  012737  000200  022202  SIZMEM: MOV     #BIT07,$KT11    ;TELL THE 'SIZE' ROUTINE TO USE MM
5361  025632  004737  022144                  JSR     PC,$SIZE        ;FIND OUT HOW MUCH MEMORY AND IF THE
5362                                                                  ;SYSTEM HAS A KT11
5363  025636  005737  022202                  TST     $KT11           ;DOES THE SYSTEM HAVE A KT11 ?
5364  025642  100060                          BPL     2$              ;BR IF NOT
5365  025644  023727  022450  001540          CMP     $LSTBK,#1540    ;MORE THAN 28K ON THE SYSTEM ?
5366  025652  101003                          BHI     1$              ;BR IF MORE THAN 28K
5367  025654  005037  022202                  CLR     $KT11           ;CLEAR MEMORY MANAGEMENT INDICATOR
5368  025660  000761                          BR      SIZMEM          ;DO IT AGAIN
5369  025662  012737  157776  001302  1$:     MOV     #157776,MEMSIZ  ;ADDRESS OF LAST NON MEMORY MANAGEMENT LOCATION
```

```
5370  025670  013737  022450  001230          MOV     $LSTBK,MAXMEM   ;HIGH MEMORY BANK ADDRESS
5371  025676  005037  001232                  CLR     MAXMEM+2        ;CLEAR UPPER MEMORY ADDRESS BITS
5372  025702  006337  001230                  ASL     MAXMEM          ;CONVERT THE BANK COUNT INTO AN
5373  025706  006137  001232                  ROL     MAXMEM+2        ;ABSOLUTE MEMORY ADDRESS
5374  025712  006337  001230                  ASL     MAXMEM          ;CONVERT THE BANK COUNT INTO AN
5375  025716  006137  001232                  ROL     MAXMEM+2        ;ABSOLUTE MEMORY ADDRESS
5376  025722  006337  001230                  ASL     MAXMEM          ;CONVERT THE BANK COUNT INTO AN
5377  025726  006137  001232                  ROL     MAXMEM+2        ;ABSOLUTE MEMORY ADDRESS
5378  025732  006337  001230                  ASL     MAXMEM          ;CONVERT THE BANK COUNT INTO AN
5379  025736  006137  001232                  ROL     MAXMEM+2        ;ABSOLUTE MEMORY ADDRESS
5380  025742  006337  001230                  ASL     MAXMEM          ;CONVERT THE BANK COUNT INTO AN
5381  025746  006137  001232                  ROL     MAXMEM+2        ;ABSOLUTE MEMORY ADDRESS
5382  025752  006337  001230                  ASL     MAXMEM          ;CONVERT THE BANK COUNT INTO AN
5383  025756  006137  001232                  ROL     MAXMEM+2        ;ABSOLUTE MEMORY ADDRESS
5384  025762  063737  022446  001230          ADD     $LSTAD,MAXMEM   ;ADD THE SIZE OF THE LAST BANK
5385  025770  005537  001232                  ADC     MAXMEM+2        ;HANDLE ANY CARRY
5386  025774  012737  020000  001264          MOV     #8192.,MAXWC    ;SETUP FOR MAXIMUM TRANSFER SIZE
5387  026002  000433                          BR      5$              ;EXIT
5388  026004  013737  022446  001302  2$:     MOV     $LSTAD,MEMSIZ   ;HIGH NON MEMORY MANAGEMENT ADDRESS
5389  026012  013737  022446  001230          MOV     $LSTAD,MAXMEM   ;HIGH NON MEMORY MANAGEMENT ADDRESS
5390  026020  005037  001232                  CLR     MAXMEM+2        ;DON'T NEED THE UPPER WORD
5391  026024  013737  001230  001264          MOV     MAXMEM,MAXWC    ;CONVERT MEMORY SIZE TO WORD COUNT
5392  026032  162737  043602  001264          SUB     #BUFFER,MAXWC   ;SUBTRACT STARTING ADDRESS OF BUFFER
5393  026040  000241                          CLC                     ;CLEAR THE 'C' BIT
5394  026042  006037  001264                  ROR     MAXWC           ;CHANGE TO WORD COUNT
5395  026046  105737  000041                  TSTB    41              ;SEE WHICH LOADER
5396  026052  001004                          BNE     4$              ;BR IF 'XXDP'
5397  026054  162737  000144  001264  3$:     SUB     #100.,MAXWC     ;'ABS' LOADER SIZE (PLUS A LITTLE BIT)
5398  026062  000403                          BR      5$              ;EXIT
5399  026064  162737  002734  001264  4$:     SUB     #1500.,MAXWC    ;'XXDP' SIZE
5400  026072  000207                  5$:     RTS     PC              ;RETURN
5401
5402                                          ;THIS ROUTINE IS USED TO ENSURE THAT THE BUS ADDRESS
5403                                          ;OF THE RP11 IS SETUP TO READ THE PROPER VALUE.
5404                                          ;IT WILL ALSO READ THE ADDRESS FROM THE TTY IF
5405                                          ;REQUIRED.
5406                                          ;NOTE: THIS ROUTINE DESTROYS R0 - R4
5407
5408  026074  005737  001216          GETADR: TST     BUSADR          ;INPUT FROM TTY REQUESTED?
5409  026100  001425                          BEQ     3$              ;NO--BRANCH
5410  026102  005037  001216                  CLR     BUSADR          ;YES--CLEAR THE REQUEST FLAG
5411  026106  104401  001173  1$:             TYPE    ,$CRLF          ;CR-LF
5412  026112  104401  032106                  TYPE    ,MRPADR         ;'RPADR='
5413  026116  013746  001324                  MOV     RPADR,-(SP)     ;;SAVE RPADR FOR TYPEOUT
5414                                          ;;RP11 ADDRESS
5415  026122  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
5416  026124     006                          .BYTE   6               ;;TYPE 6 DIGIT(S)
5417  026125     001                          .BYTE   1               ;;TYPE LEADING ZEROS
5418  026126  104401  031246                  TYPE    ,SLASH          ;'/'
5419  026132  104411                          RDLIN                   ;GET THE ENTRY
5420  026134  012601                          MOV     (SP)+,R1        ;STARTING ADDRESS OF INPUT ASCII STRING
5421  026136  004037  027062                  JSR     R0,CK.NUM       ;CHECK THE NUMBER
5422  026142  026106                          1$                      ;ILLEGAL INPUT
5423  026144  026150                          2$                      ;TERMINATED WITH A 'CR'
5424  026146  026154                          3$                      ;NO ENTRY - 'CR' ONLY
5425  026150  010237  001324          2$:     MOV     R2,RPADR        ;STORE THE ADDRESS
```

```
5426  026154  013701  000004          3$:     MOV     ERRVEC,R1       ;SAVE THE ERROR VECTOR
5427  026160  012737  026200  000004          MOV     #4$,ERRVEC      ;SETUP FOR TRAP
5428  026166  005777  153132                  TST     @RPADR          ;CHECK FOR RP11
5429  026172  010137  000004                  MOV     R1,ERRVEC       ;RESTORE ERROR VECTOR
5430  026176  000207                          RTS     PC              ;RETURN
5431  026200  010137  000004          4$:     MOV     R1,ERRVEC       ;RESTORE ERROR VECTOR
5432  026204  022626                          CMP     (SP)+,(SP)+     ;CLEAN OFF THE STACK
5433  026206  012737  026216  001164          MOV     #5$,$ESCAPE     ;ESCAPE TO 5$ ON ERROR
5434  026214  104001                          ERROR   1               ;REPORT THE ERROR
5435  026216  005037  001164          5$:     CLR     $ESCAPE         ;CLEAR ERROR ESCAPE ADDRESS
5436  026222  005737  000042                  TST     @#42            ;IS THERE A MONITOR?
5437  026226  001727                          BEQ     1$              ;NO--GO ASK FOR ADDRESS
5438  026230  005037  016766                  CLR     $EOPCT          ;NO PASSES
5439  026234  000137  016606                  JMP     $EOP            ;GO TO END OF PROGRAM
5440
5441                                  ;THIS ROUTINE ALLOWS THE OPERATOR TO CONTROL THE PROGRAM
5442                                  ;       (CONVERSATION MODE)
5443
5444  026240  104401  031250          ENTPRM: TYPE    ,ENTNM          ;'ENTER TEST NUMBER'
5445  026244  104411                          RDLIN                   ;GET THE ENTRY
5446  026246  012601                          MOV     (SP)+,R1        ;STARTING ADDRESS OF INPUT ASCII STRING
5447  026250  004037  027062                  JSR     R0,CK.NUM       ;CHECK THE NUMBER
5448  026254  026240                          ENTPRM                  ;ILLEGAL INPUT
5449  026256  026262                          1$                      ;TERMINATED WITH A 'CR'
5450  026260  026306                          2$                      ;NO ENTRY - 'CR' ONLY
5451  026262  022702  000020          1$:     CMP     #$TN,R2         ;VALID NUMBER ?
5452  026266  101764                          BLOS    ENTPRM          ;BR IF NOT
5453  026270  006302                          ASL     R2              ;CONVERT INTO A BIT INDEX
5454  026272  016237  001400  001276          MOV     BITS(R2),TSTNMS ;TEST SELECTION BIT
5455  026300  006202                          ASR     R2              ;CHANGE BACK INTO A WORD COUNT
5456  026302  010246                          MOV     R2,-(SP)        ;SAVE THE TEST NUMBER ENTRY
5457  026304  000405                          BR      ENTPR1          ;GET THE DRIVE NUMBER
5458  026306  012737  017777  001276  2$:     MOV     #17777,TSTNMS   ;SELECT ALL TESTS
5459  026314  012746  177777                  MOV     #-1,-(SP)       ;ALL TESTS SELECTED
5460  026320  005037  001312          ENTPR1: CLR     DRVSEL          ;CLEAR DRIVE SELECTION WORD
5461  026324  004737  022536                  JSR     PC,RPINIT       ;CHECK THE DRIVE STATUS
5462  026330  012737  000312  001210          MOV     #202.,MAXCYL    ;ASSUME ONLY RP02'S
5463  026336  005737  001304                  TST     DRVTYP          ;SEE WHICH TYPE
5464  026342  001403                          BEQ     1$              ;BR IF ONLY RP02'S
5465  026344  012737  000625  001210          MOV     #405.,MAXCYL    ;RP03'S
5466  026352  104401  031276          1$:     TYPE    ,DRVNM          ;ENTER DRIVE NUMBER
5467  026356  104411                          RDLIN                   ;GET THE ENTRY
5468  026360  012601                          MOV     (SP)+,R1        ;STARTING ADDRESS OF INPUT ASCII STRING
5469  026362  004037  027062                  JSR     R0,CK.NUM       ;CHECK THE NUMBER
5470  026366  026352                          1$                      ;ILLEGAL INPUT
5471  026370  026374                          2$                      ;TERMINATED WITH A 'CR'
5472  026372  026450                          5$                      ;NO ENTRY - 'CR' ONLY
5473  026374  022702  000010          2$:     CMP     #8.,R2          ;VALID ENTRY ?
5474  026400  101764                          BLOS    1$              ;BR IF NOT
5475  026402  116237  001360  001312          MOVB    ATABIT(R2),DRVSEL ;DRIVE SELECTION BIT
5476  026410  105762  001370                  TSTB    DRVSTA(R2)      ;IS DRIVE ONLINE
5477  026414  003035                          BGT     8$              ;BR IF ONLINE
5478  026416  104401  031323                  TYPE    ,DRIVE          ;'DRIVE'
5479  026422  105762  001370                  TSTB    DRVSTA(R2)      ;CHECK DRIVE STATUS AGAIN
5480  026426  100403                          BMI     3$              ;BR IF UNSAFE
5481  026430  104401  027236                  TYPE    ,OFFLIN         ;'OFFLINE'
```

```
5482  026434  000402                        BR      4$              ;CONTINUE
5483  026436  104401  027222        3$:     TYPE    ,UNSAFE         ;'UNSAFE'
5484  026442  104401  001173        4$:     TYPE    ,$CRLF          ;CR-LF
5485  026446  000724                        BR      ENTPR1          ;TRY AGAIN
5486  026450  005001                5$:     CLR     R1              ;CLEAR R1 - USE IT AS AN INDEX
5487  026452  005716                        TST     (SP)            ;ALL TESTS SELECTED ?
5488  026454  100403                        BMI     6$              ;BR IF THEY WERE
5489  026456  022716  000015              CMP     #15,(SP)        ;TEST 15, 16, OR 17 SELECTED ?
5490  026462  101733                        BLOS    1$              ;BR IF YES, SPECIFIC DRIVE NUMBER REQUIRED
5491  026464  105761  001370        6$:     TSTB    DRVSTA(R1)      ;IS THE DRIVE ONLINE ?
5492  026470  003403                        BLE     7$              ;BR IF NOT
5493  026472  156137  001360  001312        BISB    ATABIT(R1),DRVSEL  ;SET THE SELECT BIT FOR THE DRIVE
5494  026500  005201                7$:     INC     R1              ;INCREMENT THE DRIVE INDEX
5495  026502  020127  000010              CMP     R1,#8.          ;REACHED MAXIMUM ?
5496  026506  103766                        BLO     6$              ;BR IF NOT
5497  026510  022726  000014        8$:     CMP     #14,(SP)+       ;IS TEST 14 SELECTED ?
5498  026514  001137                        BNE     ENTPRX          ;BR IF NOT
5499  026516  104401  031332        ENTPR2: TYPE    ,CHNGPM         ;CHANGE PARAMETERS FOR TEST 14 ?
5500  026522  104411                        RDLIN                   ;GET THE RESPONSE
5501  026524  123627  000131              CMPB    @(SP)+,#'Y      ;'Y' ENTERED ?
5502  026530  001131                        BNE     ENTPRX          ;BR IF NOT
5503  026532  104401  031373              TYPE    ,MXWRDS         ;'MAXIMUM WORD COUNT FOR TEST 14: '
5504  026536  013746  001264              MOV     MAXWC,-(SP)     ;;SAVE MAXWC FOR TYPEOUT
5505                                                                ;;MAXIMUM WORD COUNT FOR TEST 14
5506  026542  104403                        TYPOS                   ;;GO TYPE--OCTAL ASCII
5507  026544     005                        .BYTE   5               ;;TYPE 5 DIGIT(S)
5508  026545     000                        .BYTE   0               ;;SUPPRESS LEADING ZEROS
5509  026546  104401  001173              TYPE    ,$CRLF          ;CR-LF
5510  026552  104401  031447        1$:     TYPE    ,ENTWC          ;'ENTER WORD COUNT FOR TEST 14'
5511  026556  104411                        RDLIN                   ;GET THE ENTRY
5512  026560  012601                        MOV     (SP)+,R1        ;STARTING ADDRESS OF INPUT ASCII STRING
5513  026562  004037  027062              JSR     R0,CK.NUM       ;CHECK THE NUMBER
5514  026566  026552                        1$                      ;ILLEGAL INPUT
5515  026570  026574                        2$                      ;TERMINATED WITH A 'CR'
5516  026572  026614                        ENTPR3                  ;NO ENTRY - 'CR' ONLY
5517  026574  020237  001264        2$:     CMP     R2,MAXWC        ;VALUE TOO LARGE ?
5518  026600  101364                        BHI     1$              ;BR IF TOO LARGE
5519  026602  020237  001262              CMP     R2,LMTBK        ;NEW WORD COUNT LARGER THAN BK ?
5520  026606  101361                        BHI     1$              ;BR IF LARGER
5521  026610  010237  001256              MOV     R2,WC14         ;LOAD NEW WORD COUNT
5522  026614  104401  031511        ENTPR3: TYPE    ,CHNGPT         ;ENTER PATTERN CODE
5523  026620  104411                        RDLIN                   ;READ THE ENTRY
5524  026622  012601                        MOV     (SP)+,R1        ;STARTING ADDRESS OF INPUT ASCII STRING
5525  026624  004037  027062              JSR     R0,CK.NUM       ;CHECK THE NUMBER
5526  026630  026614                        ENTPR3                  ;ILLEGAL INPUT
5527  026632  026636                        1$                      ;TERMINATED WITH A 'CR'
5528  026634  026642                        ENTPR4                  ;NO ENTRY - 'CR' ONLY
5529  026636  010237  001300        1$:     MOV     R2,PATTRN       ;PATTERN CODE
5530  026642  104401  031564        ENTPR4: TYPE    ,CHNGAD         ;USE A SPECIFIC DISK ADDRESS ?
5531  026646  104411                        RDLIN                   ;READ THE ENTRY
5532  026650  123627  000131              CMPB    @(SP)+,#'Y      ;'Y' ENTERED ?
5533  026654  001057                        BNE     ENTPRX          ;BR IF NOT
5534  026656  104401  031637              TYPE    ,ONEPAT         ;'A SINGLE DATA PATTERN MUST HAVE BEEN SPECIFIED;
5535                                                                ;'IF NOT, PATTERN 0 (ALL ZEROS) WILL BE USED'
5536  026662  104401  032000        1$:     TYPE    ,ENTCYL         ;'ENTER CYLINDER ADDRESS'
5537  026666  104411                        RDLIN                   ;GET THE ENTRY
```

```
5538  026670  012601                          MOV     (SP)+,R1        ;STARTING ADDRESS OF INPUT ASCII STRING
5539  026672  004037  027062                  JSR     R0,CK.NUM       ;CHECK THE NUMBER
5540  026676  026662                          1$                      ;ILLEGAL INPUT
5541  026700  026704                          2$                      ;TERMINATED WITH A 'CR'
5542  026702  026662                          1$                      ;NO ENTRY - 'CR' ONLY
5543  026704  020237  001210          2$:     CMP     R2,MAXCYL       ;TOO LARGE ?
5544  026710  101364                          BHI     1$              ;BR IF TOO LARGE
5545  026712  010237  001266                  MOV     R2,SCYL         ;LOAD CYLINDER ADDRESS
5546  026716  104401  032031          3$:     TYPE    ,ENTRK          ;'ENTER TRACK ADDRESS'
5547  026722  104411                          RDLIN                   ;GET THE ENTRY
5548  026724  012601                          MOV     (SP)+,R1        ;STARTING ADDRESS OF INPUT ASCII STRING
5549  026726  004037  027062                  JSR     R0,CK.NUM       ;CHECK THE NUMBER
5550  026732  026716                          3$                      ;ILLEGAL INPUT
5551  026734  026740                          4$                      ;TERMINATED WITH A 'CR'
5552  026736  026716                          3$                      ;NO ENTRY - 'CR' ONLY
5553  026740  020227  000023          4$:     CMP     R2,#19.         ;VALID ENTRY ?
5554  026744  101364                          BHI     3$              ;BR IF NOT
5555  026746  010237  001270                  MOV     R2,STRK         ;TRACK ADDRESS
5556  026752  104401  032057          5$:     TYPE    ,ENTSEC         ;'ENTER SECTOR ADDRESS'
5557  026756  104411                          RDLIN
5558  026760  012601                          MOV     (SP)+,R1        ;STARTING ADDRESS OF INPUT ASCII STRING
5559  026762  004037  027062                  JSR     R0,CK.NUM       ;CHECK THE NUMBER
5560  026766  026752                          5$                      ;ILLEGAL INPUT
5561  026770  026774                          6$                      ;TERMINATED WITH A 'CR'
5562  026772  026752                          5$                      ;NO ENTRY - 'CR' ONLY
5563  026774  020227  000011          6$:     CMP     R2,#9.          ;VALID ENTRY ?
5564  027000  101364                          BHI     5$              ;BR IF NOT
5565  027002  010237  001272                  MOV     R2,SSEC         ;LOAD SECTOR ADDRESS
5566  027006  012737  177777  001260          MOV     #-1,OPRSEL      ;SET ADDRESS FLAG
5567  027014  000207          ENTPRX: RTS     PC                      ;RETURN
5568
5569                                          ;THIS ROUTINE IS USED TO CHECK IF AN ASCII CHARACTER IS A DIGIT
5570                                          ;BETWEEN 0 AND 7.
5571                                          ;
5572                                          ;       MOV     #ADR,R1         ;ADDRESS OF ASCII CHARACTER
5573                                          ;       JSR     R0,CK.OCT       ;CHECK THE CHARACTER
5574                                          ;       RETURN1                 ;CHARACTER IS IN R2 AS A
5575                                          ;                               ;OCTAL DIGIT
5576                                          ;       RETURN2                 ;CHARACTER IS NOT BETWEEN 0-7
5577
5578  027016  121127  000060          CK.OCT: CMPB    (R1),#'0        ;LESS THAN ZERO?
5579  027022  103407                          BLO     1$              ;YES -- BRANCH
5580  027024  121127  000067                  CMPB    (R1),#'7        ;GREATER THAN SEVEN?
5581  027030  101004                          BHI     1$              ;YES -- BRANCH
5582  027032  111102                          MOVB    (R1),R2         ;GET THE CHARACTER
5583  027034  042702  177770                  BIC     #^C7,R2         ;STRIP AWAY THE ASCII
5584  027040  000401                          BR      2$              ;BYPASS RETURN ADJUST
5585  027042  005720          1$:     TST     (R0)+           ;ADJUST FOR RETURN
5586  027044  000200          2$:     RTS     R0              ;RETURN
5587
5588                                          ;THIS ROUTINE WILL CHECK AN ASCII CHARACTER TO
5589                                          ;DETERMINE WHAT IT IS.
5590                                          ;
5591                                          ;       MOV     #ADR,R1         ;ADDRESS OF ASCII CHARACTER
5592                                          ;       JSR     R0,CK.CHR       ;CHECK CHARACTER
5593                                          ;       RETURN  ADR1            ;UNKNOWN CHARACTER
```

```
                                ;       RETURN  ADR2            ;CARRIAGE RETURN # (R1)=ADR+1
                                ;
027046  105711                 CK.CHR: TSTB    (R1)            ;"CARRIAGE RETURN"?
027050  001002                         BNE     1$              ;NO -- BRANCH
027052  005720                         TST     (R0)+           ;CARRIAGE RETURN
027054  005201                         INC     R1              ;MOVE POINTER TO NEXT CHARACTER
027056  011000                 1$:     MOV     (R0),R0         ;UNKNOWN CHARACTER
027060  000200                         RTS     R0              ;RETURN
                                ;
                                ;THIS ROUTINE CHECKS AN ASCIZ STRING FOR LEGAL CHARACTERS
                                ;AND FORMS AN OCTAL NUMBER IN R2
                                ;
                                ;       MOV     @ADR,R1         ;ADDRESS OF ASCIZ STRING
                                ;       JSR     R0,CK.NUM       ;GO FORM THE NUMBER
                                ;       RETURN  ADR1            ;ILLEGAL CHARACTER IN THE INPUT STRING
                                ;       RETURN  ADR2            ;"CR"--R2=NUMBER
                                ;       RETURN  ADR3            ;'CR' WAS FIRST ENTRY
                                ;
027062  010346                 CK.NUM: MOV     R3,-(SP)        ;SAVE R3
027064  005003                         CLR     R3              ;START NUMBER AT ZERO
027066  004037  027016                  JSR     R0,CK.OCT       ;OCTAL DIGIT?
027072  000404                         BR      1$              ;YES--BRANCH
027074  004037  027046                  JSR     R0,CK.CHR       ;CHECK ONE CHARACTER
027100  027154                         6$                      ;ILLEGAL CHARACTER
027102  027146                         8$                      ;CARRIAGE RETURN
027104  005201                 1$:     INC     R1              ;MOVE TO NEXT CHARACTER
027106  006303                         ASL     R3              ;FOR THE OCTAL NUMBER IN R3
027110  103421                         BCS     6$              ;DON'T LET IT GET TO BIG
027112  006303                         ASL     R3
027114  103417                         BCS     6$
027116  006303                         ASL     R3
027120  103415                         BCS     6$
027122  060203                         ADD     R2,R3
027124  004037  027016                  JSR     R0,CK.OCT       ;IS THIS AN OCTAL DIGIT?
027130  000765                         BR      1$              ;YES--MAKE IT PART OF THE NUMBER
027132  010302                 2$:     MOV     R3,R2           ;SAVE THE OCTAL NUMBER
027134  005003                         CLR     R3              ;START WITH ZERO INDEX
027136                         3$:
027136  004037  027046                  JSR     R0,CK.CHR       ;CHECK ONE CHARACTER
027142  027154                         6$                      ;ILLEGAL CHARACTER
027144  027150                         5$                      ;CARRIAGE RETURN
027146  005723                 8$:     TST     (R3)+           ;'CR' ONLY
027150  005723                 5$:     TST     (R3)+           ;"CR"
027152  060300                         ADD     R3,R0           ;YES--SAVE THE OCTAL NUMBER
027154  012603                 6$:     MOV     (SP)+,R3        ;RESTORE R3
027156  011000                         MOV     (R0),R0         ;PICKUP EXIT ADDRESS
027160  000200                         RTS     R0              ;RETURN
                                ;
                                .SBTTL  TELETYPE MESSAGES
                                ;
027162  005015  042012  044522  DRSTAT: .ASCIZ  <15><12><12>@DRIVE STATUS:@<15><12><12>
027170  042526  051440  040524
027176  052524  035123  005015
027204  000012
027206  020040  020040  047440  ONLINE: .ASCIZ  @     ONLINE@
```

```
5650  027214  046116  047111  000105
5651  027222  020040  020040  052440  UNSAFE: .ASCIZ  @    UNSAFE@
5652  027230  051516  043101  000105
5653  027236  020040  020040  047440  OFFLIN: .ASCIZ  @    OFFLINE@
5654  027244  043106  044514  042516
5655  027252     000
5656  027253     040  051040  030120  RP02:   .ASCIZ  @ RP02@
5657  027260  000062
5658  027262  020040  050122  031460  RP03:   .ASCIZ  @ RP03@
5659  027270     000
5660  027271     104  044522  042526  DRVTST: .ASCIZ  @DRIVE(S) TO BE USED @
5661  027276  051450  020051  047524
5662  027304  041040  020105  051525
5663  027312  042105  000040
5664  027316  047516  042516     000  NONE:   .ASCIZ  @NONE@
5665  027323     054     000  COMMA:  .ASCIZ  @,@
5666  027325     015  005012  042524  TSTING: .ASCIZ  <15><12><12>@TESTING WITH DRIVE @
5667  027332  052123  047111  020107
5668  027340  044527  044124  042040
5669  027346  044522  042526  000040
5670  027354  020040     000  BLNKS2: .ASCIZ  @ @
5671
5672  027357     015  005012  047520  MSG15A: .ASCIZ  <15><12><12>@POWER FAIL TEST@
5673  027364  042527  020122  040506
5674  027372  046111  052040  051505
5675  027400  000124
5676  027402  005015  052012  051125  MSG15B: .ASCII  <15><12><12>@TURN CPU POWER  O F F  AFTER THIS MESSAGE, WAIT@<15><12>
5677  027410  020116  050103  020125
5678  027416  047520  042527  020122
5679  027424  047440  043040  043040
5680  027432  020040  043101  042524
5681  027440  020122  044124  051511
5682  027446  046440  051505  040523
5683  027454  042507  020054  040527
5684  027462  052111  005015
5685  027466  020065  042523  047503          .ASCIZ  @5 SECONDS, THEN TURN CPU POWER  O N  AGAIN@<15><12>
5686  027474  042116  026123  052040
5687  027502  042510  020116  052524
5688  027510  047122  041440  052520
5689  027516  050040  053517  051105
5690  027524  020040  020117  020116
5691  027532  040440  040507  047111
5692  027540  005015     000
5693  027543     015  042412  042116  MSG15C: .ASCIZ  <15><12>@END OF POWER FAIL TEST@<15><12>
5694  027550  047440  020106  047520
5695  027556  042527  020122  040506
5696  027564  046111  052040  051505
5697  027572  006524  000012
5698  027576  005015  050012  047522  MSG15D: .ASCIZ  <15><12><12>@PROGRAM WILL LOOP, WAITING FOR DRIVE TO COME ONLINE@<15><12
5699  027604  051107  046501  053440
5700  027612  046111  020114  047514
5701  027620  050117  020054  040527
5702  027626  052111  047111  020107
5703  027634  047506  020122  051104
5704  027642  053111  020105  047524
5705  027650  041440  046517  020105
```

```
5706  027656  047117  044514  042516
5707  027664  005015     000
5708
5709  027667     015  005012  042510  MSG16A: .ASCII  <15><12><12>@HEAD ALIGNMENT ROUTINE@
5710  027674  042101  040440  044514
5711  027702  047107  042515  052116
5712  027710  051040  052517  044524
5713  027716  042516
5714  027720  005015  041412  047117          .ASCII  <15><12><12>@CONNECT A JUMPER BETWEEN PINS J13M2 & J13C2 ON THE RP11E BA
5715  027726  042516  052103  040440
5716  027734  045040  046525  042520
5717  027742  020122  042502  053524
5718  027750  042505  020116  044520
5719  027756  051516  045040  031461
5720  027764  031115  023040  045040
5721  027772  031461  031103  047440
5722  030000  020116  044124  020105
5723  030006  050122  030461  020105
5724  030014  040502  045503  046120
5725  030022  047101     105
5726  030025     015  020012  051050          .ASCII  <15><12>@ (REFER TO THE APPLICABLE MAINTENANCE MANUAL FOR THE DRIVE SETU
5727  030032  043105  051105  052040
5728  030040  020117  044124  020105
5729  030046  050101  046120  041511
5730  030054  041101  042514  046440
5731  030062  044501  052116  047105
5732  030070  047101  042503  046440
5733  030076  047101  040525  020114
5734  030104  047506  020122  044124
5735  030112  020105  051104  053111
5736  030120  020105  042523  052524
5737  030126  027120     051
5738  030131     015  005012  047524          .ASCIZ  <15><12><12>@TOGGLE SWITCH 7 FOR HEAD SELECTION@<15><12>
5739  030136  043507  042514  051440
5740  030144  044527  041524  020110
5741  030152  020067  047506  020122
5742  030160  042510  042101  051440
5743  030166  046105  041505  044524
5744  030174  047117  005015     000
5745  030201     015  042412  052116  MSG16B: .ASCIZ  <15><12>@ENTER HEAD ADDRESS (IN OCTAL): @
5746  030206  051105  044040  040505
5747  030214  020104  042101  051104
5748  030222  051505  020123  044450
5749  030230  020116  041517  040524
5750  030236  024514  020072     000
5751
5752  030243     015  005012  050122  MSG17A: .ASCIZ  <15><12><12>@RP11E CONTROL PANEL SWITCH TEST@
5753  030250  030461  020105  047503
5754  030256  052116  047522  020114
5755  030264  040520  042516  020114
5756  030272  053523  052111  044103
5757  030300  052040  051505  000124
5758  030306  005015  054524  042520  MSG17B: .ASCIZ  <15><12>@TYPE A 'CR' TO START TEST AFTER EACH SETUP@<15><12>
5759  030314  040440  023440  051103
5760  030322  020047  047524  051440
5761  030330  040524  052122  052040
```

```
 5762  030336  051505  020124  043101
 5763  030344  042524  020122  040505
 5764  030352  044103  051440  052105
 5765  030360  050125  005015  000
 5766  030365  015  042012  051511  MSG17C: .ASCIZ  <15><12>@DISABLE DRIVE # @
 5767  030372  041101  042514  042040
 5768  030400  044522  042526  021440
 5769  030406  000040
 5770  030410  005015  042105  041101  MSG17D: .ASCIZ  <15><12>@ENABLE DRIVE # @
 5771  030416  042514  042040  044522
 5772  030424  042526  021440  000040
 5773  030432  005015  042523  020124  MSG17E: .ASCIZ  <15><12>@SET 'READ ONLY' ON DRIVE @
 5774  030440  051047  040505  020104
 5775  030446  047117  054514  020047
 5776  030454  047117  042040  044522
 5777  030462  042526  000040
 5778  030466  005015  042523  020124  MSG17F: .ASCIZ  <15><12>@SET 'READ-WRITE' ON DRIVE @
 5779  030474  051047  040505  026504
 5780  030502  051127  052111  023505
 5781  030510  047440  020116  051104
 5782  030516  053111  020105  000
 5783  030523  015  051413  052105  MSG17G: .ASCII  <15><12>@SET 'WRITE LOCKOUT' AND CLEAR THE LOA SWITCHES@
 5784  030530  023440  051127  052111
 5785  030536  020105  047514  045503
 5786  030544  052517  023524  040440
 5787  030552  042116  041440  042514
 5788  030560  051101  052040  042510
 5789  030566  046040  040517  051440
 5790  030574  044527  041524  042510
 5791  030602  123
 5792  030603  015  047412  020116          .ASCIZ  <15><12>@ON THE RP11E CONTROL PANEL@
 5793  030610  044124  020105  050122
 5794  030616  030461  020105  047503
 5795  030624  052116  047522  020114
 5796  030632  040520  042516  000114
 5797  030640  005015  042523  020124  MSG17H: .ASCIZ  <15><12>@SET THE FOLLOWING OCTAL VALUES IN THE CYLINDER LOA SWITCHES: @
 5798  030646  044124  020105  047506
 5799  030654  046114  053517  047111
 5800  030662  020107  041517  040524
 5801  030670  020114  040526  052514
 5802  030676  051505  044440  020116
 5803  030704  044124  020105  054503
 5804  030712  044514  042116  051105
 5805  030720  046040  040517  051440
 5806  030726  044527  041524  042510
 5807  030734  036123  000040
 5808  030740  005015  042523  020124  MSG17I: .ASCIZ  <15><12>@SET SWITCHES TO @
 5809  030746  053523  052111  044103
 5810  030754  051505  052040  020117
 5811  030762  000
 5812  030763  015  051012  051505  MSG17J: .ASCII  <15><12>@RESET THE CYLINDER LOA SWITCHES@
 5813  030770  052105  052040  042510
 5814  030776  041440  046131  047111
 5815  031004  042504  020122  047514
 5816  031012  020101  053523  052111
 5817  031020  044103  051505
```

```
 5818  031024  005015  051412  052105          .ASCIZ  <15><12><12>@SET THE FOLLOWING OCTAL VALUES IN THE DRIVE LOA SWITCHES: @
 5819  031032  052040  042510  043040
 5820  031040  046117  047514  044527
 5821  031046  043516  047440  052103
 5822  031054  046101  053040  046101
 5823  031062  042525  020123  047111
 5824  031070  052040  042510  042040
 5825  031076  044522  042526  046040
 5826  031104  040517  051440  044527
 5827  031112  041524  042510  035123
 5828  031120  000040
 5829  031122  005015  042412  042116  MSG17K: .ASCII  <15><12><12>@END OF CONROL PANEL SWITCH TEST@
 5830  031130  047440  020106  047503
 5831  031136  051116  046117  050040
 5832  031144  047101  046105  051440
 5833  031152  044527  041524  020110
 5834  031160  042524  052123
 5835  031164  005015  042522  052524          .ASCIZ  <15><12>@RETURN RP11E CONTROL PANEL SWITCHES TO NORMAL@<15><12>
 5836  031172  047122  051040  030520
 5837  031200  042461  041440  047117
 5838  031206  051124  046117  050040
 5839  031214  047101  046105  051440
 5840  031222  044527  041524  042510
 5841  031230  020123  047524  047040
 5842  031236  051117  040515  006514
 5843  031244  000012
 5844
 5845  031246  000057                  SLASH:  .ASCIZ  @/@
 5846  031250  005015  047105  042524  ENTNM:  .ASCIZ  <15><12>@ENTER TEST NUMBER: @
 5847  031256  020122  042524  052123
 5848  031264  047040  046525  042502
 5849  031272  035122  000040
 5850  031276  047105  042524  020122  DRVNM:  .ASCIZ  @ENTER DRIVE NUMBER: @
 5851  031304  051104  053111  020105
 5852  031312  052516  041115  051105
 5853  031320  020072     000
 5854  031323     104  044522  042526  DRIVE:  .ASCIZ  @DRIVE @
 5855  031330  000040
 5856  031332  044103  047101  042507  CHNGPM: .ASCIZ  @CHANGE PARAMETERS FOR TEST 14 ? @
 5857  031340  050040  051101  046501
 5858  031346  052105  051105  020123
 5859  031354  047506  020122  042524
 5860  031362  052123  030440  020064
 5861  031370  020077     000
 5862  031373     115  054101  046511  MXWRDS: .ASCIZ  @MAXIMUM WORD COUNT FOR TEST 14 (IN OCTAL): @
 5863  031400  046525  053440  051117
 5864  031406  020104  047503  047125
 5865  031414  020124  047506  020122
 5866  031422  042524  052123  030440
 5867  031430  020064  044450  020116
 5868  031436  041517  040524  024514
 5869  031444  020072     000
 5870  031447     105  052116  051105  ENTWC:  .ASCIZ  @ENTER NEW WORD COUNT (IN OCTAL): @
 5871  031454  047040  053505  053440
 5872  031462  051117  020104  047503
 5873  031470  047125  020124  044450
```

```
 5874  031476  020116  041517  040524
 5875  031504  024514  020072     000
 5876  031511     105  052116  051105  CHNGPT: .ASCIZ  @ENTER PATTERN SELECTION CODE FOR TEST 14: @
 5877  031516  050040  052101  042524
 5878  031524  047122  051440  046105
 5879  031532  041505  044524  047117
 5880  031540  041440  042117  020105
 5881  031546  047506  020122  042524
 5882  031554  052123  030440  035064
 5883  031562  000040
 5884  031564  051525  020105  020101  CHNGAD: .ASCIZ  @USE A SPECIFIC DISK ADDRESS FOR TEST 14 ? @
 5885  031572  050123  041505  043111
 5886  031600  041511  042040  051511
 5887  031606  020113  042101  051104
 5888  031614  051505  020123  047506
 5889  031622  020122  042524  052123
 5890  031630  030440  020064  020077
 5891  031636     000
 5892  031637     040  040450  051440  ONEPAT: .ASCII  @ (A SINGLE DATA PATTERN MUST HAVE BEEN SPECIFIED@<15><12>
 5893  031644  047111  046107  020105
 5894  031652  040504  040524  050040
 5895  031660  052101  042524  047122
 5896  031666  046440  051525  020124
 5897  031674  040510  042526  041040
 5898  031702  042505  020116  050123
 5899  031710  041505  043111  042511
 5900  031716  006504     012
 5901  031721     040  043111  047040          .ASCIZ  @ IF NOT, PATTERN 0 (ALL ZEROS) WILL BE USED)@<15><12>
 5902  031726  052117  020054  040520
 5903  031734  052124  051105  020116
 5904  031742  020060  040450  046114
 5905  031750  055040  051105  051517
 5906  031756  020051  044527  046114
 5907  031764  041040  020105  051525
 5908  031772  042105  006451  000012
 5909  032000  047105  042524  020122  ENTCYL: .ASCIZ  @ENTER CYLINDER ADDRESS: @
 5910  032006  054503  044514  042116
 5911  032014  051105  040440  042104
 5912  032022  042522  051523  020072
 5913  032030     000
 5914  032031     105  052116  051105  ENTRK:  .ASCIZ  @ENTER TRACK ADDRESS: @
 5915  032036  052040  040522  045503
 5916  032044  040440  042104  042522
 5917  032052  051523  020072     000
 5918  032057     105  052116  051105  ENTSEC: .ASCIZ  @ENTER SECTOR ADDRESS: @
 5919  032064  051440  041505  047524
 5920  032072  020122  042101  051104
 5921  032100  051505  035123  000040
 5922
 5923  032106  050122  042101  020122  MRPADR: .ASCIZ  @RPADR = @
 5924  032114  020075     000
 5925
 5926                                  .SBTTL  ERROR MESSAGES
 5927
 5928  032117     122  030520  020061  EM1:    .ASCIZ  @RP11 DIDN'T RESPOND TO ADDRESSING@
 5929  032124  044504  047104  052047
```

```
5930  032132  051040  051505  047520
5931  032140  042116  052040  020117
5932  032146  042101  051104  051505
5933  032154  044523  043516     000
5934  032161     103  047101  052047  EM2:    .ASCIZ  @CAN'T START READ COMMAND@
5935  032166  051440  040524  052122
5936  032174  051040  040505  020104
5937  032202  047503  046515  047101
5938  032210  000104
5939  032212  041447  042514  051101  EM3:    .ASCIZ  @'CLEAR' COMMAND DIDN'T TERMINATE DATA TRANSFER OPERATION@
5940  032220  020047  047503  046515
5941  032226  047101  020104  044504
5942  032234  047104  052047  052040
5943  032242  051105  044515  040516
5944  032250  042524  042040  052101
5945  032256  020101  051124  047101
5946  032264  043123  051105  047440
5947  032272  042520  040522  044524
5948  032300  047117     000
5949  032303     105  051122  051117  EM4:    .ASCIZ  @ERROR WRITING TEST SECTOR(S)@
5950  032310  053440  044522  044524
5951  032316  043516  052040  051505
5952  032324  020124  042523  052103
5953  032332  051117  051450  000051
5954  032340  051127  052111  020105  EM5:    .ASCIZ  @WRITE CHECK ERROR CHECKING TEST SECTOR(S)@
5955  032346  044103  041505  020113
5956  032354  051105  047522  020122
5957  032362  044103  041505  044513
5958  032370  043516  052040  051505
5959  032376  020124  042523  052103
5960  032404  051117  051450  000051
5961  032412  054105  042520  052103  EM6:    .ASCIZ  @EXPECTED 'WCE' DIDN'T OCCUR@
5962  032420  042105  023440  041527
5963  032426  023505  042040  042111
5964  032434  023516  020124  041517
5965  032442  052503  000122
5966  032446  051105  047522  020122  EM7:    .ASCIZ  @ERROR WRITING LESS THAN A FULL SECTOR@
5967  032454  051127  052111  047111
5968  032462  020107  042514  051523
5969  032470  052040  040510  020116
5970  032476  020101  052506  046114
5971  032504  051440  041505  047524
5972  032512  000122
5973  032514  047117  020105  043117  EM11:   .ASCIZ  @ONE OF 2 WORDS WRITTEN IN THE PARTIAL WRITE DIDN'T COMPARE@
5974  032522  031040  053440  051117
5975  032530  051504  053440  044522
5976  032536  052124  047105  044440
5977  032544  020116  044124  020105
5978  032552  040520  052122  040511
5979  032560  020114  051127  052111
5980  032566  020105  044504  047104
5981  032574  052047  041440  046517
5982  032602  040520  042522     000
5983  032607     116  047117  055055  EM12:   .ASCII  @NON-ZERO DATA IN PART OF SECTOR WHICH SHOULD HAVE@<15><12>
5984  032614  051105  020117  040504
5985  032622  040524  044440  020116
```

```
 5986  032630  040520  052122  047440
 5987  032636  020106  042523  052103
 5988  032644  051117  053440  044510
 5989  032652  044103  051440  047510
 5990  032660  046125  020104  040510
 5991  032666  042526  005015
 5992  032672  042502  047105  055040          .ASCIZ  @BEEN ZERO FILLED DURING PARTIAL WRITE@
 5993  032700  051105  020117  044506
 5994  032706  046114  042105  042040
 5995  032714  051125  047111  020107
 5996  032722  040520  052122  040511
 5997  032730  020114  051127  052111
 5998  032736  000105
 5999  032740  042447  050117  020047  EM13:   .ASCII  @'EOP' DIDN'T SET DURING A 2 SECTOR WRITE BEGINNING@
 6000  032746  044504  047104  052047
 6001  032754  051440  052105  042040
 6002  032762  051125  047111  020107
 6003  032770  020101  020062  042523
 6004  032776  052103  051117  053440
 6005  033004  044522  042524  041040
 6006  033012  043505  047111  044516
 6007  033020  043516
 6008  033022  005015  052101  052040          .ASCIZ  <15><12>@AT THE LAST SECTOR OF THE PACK@
 6009  033030  042510  046040  051501
 6010  033036  020124  042523  052103
 6011  033044  051117  047440  020106
 6012  033052  044124  020105  040520
 6013  033060  045503     000
 6014  033063     047  051105  023522  EM14:   .ASCIZ  @'ERR' DIDN'T SET WITH 'EOP'@
 6015  033070  042040  042111  023516
 6016  033076  020124  042523  020124
 6017  033104  044527  044124  023440
 6018  033112  047505  023520     000
 6019  033117     047  047505  023520  EM15:   .ASCIZ  @'EOP' DIDN'T CLEAR RPCA@
 6020  033124  042040  042111  023516
 6021  033132  020124  046103  040505
 6022  033140  020122  050122  040503
 6023  033146     000
 6024  033147     123  041525  020101  EM16:   .ASCIZ  @SUCA NOT CORRECT AFTER 'EOP'@
 6025  033154  047516  020124  047503
 6026  033162  051122  041505  020124
 6027  033170  043101  042524  020122
 6028  033176  042447  050117  000047
 6029  033204  051105  047522  020122  EM17:   .ASCIZ  @ERROR READING TEST SECTOR@
 6030  033212  042522  042101  047111
 6031  033220  020107  042524  052123
 6032  033226  051440  041505  047524
 6033  033234  000122
 6034  033236  047503  052116  047105  EM20:   .ASCII  @CONTENTS OF FIRST SECTOR ON PACK CHANGED AFTER@<15><12>
 6035  033244  051524  047440  020106
 6036  033252  044506  051522  020124
 6037  033260  042523  052103  051117
 6038  033266  047440  020116  040520
 6039  033274  045503  041440  040510
 6040  033302  043516  042105  040440
 6041  033310  052106  051105  005015
```

```
6042  033316  047506  041522  047111          .ASCIZ  @FORCING 'EOP' DURING WRITE@
6043  033324  020107  042447  050117
6044  033332  020047  052504  044522
6045  033340  043516  053440  044522
6046  033346  042524     000
6047  033351     047  051120  043517  EM21:   .ASCII  @'PROG' ERROR DIDN'T SET WHEN READ COMMAND ISSUED@
6048  033356  020047  051105  047522
6049  033364  020122  044504  047104
6050  033372  052047  051440  052105
6051  033400  053440  042510  020116
6052  033406  042522  042101  041440
6053  033414  046517  040515  042116
6054  033422  044440  051523  042525
6055  033430     104
6056  033431     015  053412  044510          .ASCIZ  <15><12>@WHILE CONTROLLER BUSY@
6057  033436  042514  041440  047117
6058  033444  051124  046117  042514
6059  033452  020122  052502  054523
6060  033460     000
6061  033461     047  051105  023522  EM22:   .ASCIZ  @'ERR' DIDN'T SET WITH 'PROG'@
6062  033466  042040  042111  023516
6063  033474  020124  042523  020124
6064  033502  044527  044124  023440
6065  033510  051120  043517  000047
6066  033516  051105  047522  020122  EM23:   .ASCIZ  @ERROR ATTEMPTING TO MISFORMAT SECTOR 0@
6067  033524  052101  042524  050115
6068  033532  044524  043516  052040
6069  033540  020117  044515  043123
6070  033546  051117  040515  020124
6071  033554  042523  052103  051117
6072  033562  030040     000
6073  033565     105  051122  051117  EM24:   .ASCIZ  @ERROR VERIFYING THE MISFORMATTED TEST HEADER@
6074  033572  053040  051105  043111
6075  033600  044531  043516  052040
6076  033606  042510  046440  051511
6077  033614  047506  046522  052101
6078  033622  042524  020104  042524
6079  033630  052123  044040  040505
6080  033636  042504  000122
6081  033642  044515  043123  051117  EM25:   .ASCIZ  @MISFORMATTED TEST HEADER IS NOT CORRECT@
6082  033650  040515  052124  042105
6083  033656  052040  051505  020124
6084  033664  042510  042101  051105
6085  033672  044440  020123  047516
6086  033700  020124  047503  051122
6087  033706  041505  000124
6088  033712  044047  043116  020047  EM26:   .ASCIZ  @'HNF' DIDN'T SET WHEN SEARCHING FOR MISFORMATTED SECTOR@
6089  033720  044504  047104  052047
6090  033726  051440  052105  053440
6091  033734  042510  020116  042523
6092  033742  051101  044103  047111
6093  033750  020107  047506  020122
6094  033756  044515  043123  051117
6095  033764  040515  052124  042105
6096  033772  051440  041505  047524
6097  034000  000122
```

```
 6098  034002  042047  045523  020047  EM27:   .ASCIZ  @'DSK' NOT SET WITH 'HNF'@
 6099  034010  047516  020124  042523
 6100  034016  020124  044527  044124
 6101  034024  023440  047110  023506
 6102  034032     000
 6103  034033     047  042510  020047  EM30:   .ASCIZ  @'HE' DIDN'T SET WITH 'HNF'@
 6104  034040  044504  047104  052047
 6105  034046  051440  052105  053440
 6106  034054  052111  020110  044047
 6107  034062  043116  000047
 6108  034066  042447  051122  020047  EM31:   .ASCIZ  @'ERR' NOT SET WITH 'HNF'@
 6109  034074  047516  020124  042523
 6110  034102  020124  044527  044124
 6111  034110  023440  047110  023506
 6112  034116     000
 6113  034117     105  051122  051117  EM32:   .ASCIZ  @ERROR WHILE RESTORING MISFORMATTED TEST HEADER@
 6114  034124  053440  044510  042514
 6115  034132  051040  051505  047524
 6116  034140  044522  043516  046440
 6117  034146  051511  047506  046522
 6118  034154  052101  042524  020104
 6119  034162  042524  052123  044040
 6120  034170  040505  042504  000122
 6121  034176  047503  052116  047522  EM33:   .ASCII  @CONTROLLER DIDN'T BECOME BUSY WHEN READ COMMAND@
 6122  034204  046114  051105  042040
 6123  034212  042111  023516  020124
 6124  034220  042502  047503  042515
 6125  034226  041040  051525  020131
 6126  034234  044127  047105  051040
 6127  034242  040505  020104  047503
 6128  034250  046515  047101     104
 6129  034255     015  044412  051523          .ASCIZ  <15><12>@ISSUED TO SEEKING DRIVE@
 6130  034262  042525  020104  047524
 6131  034270  051440  042505  044513
 6132  034276  043516  042040  044522
 6133  034304  042526     000
 6134  034307     105  051122  051117  EM34:   .ASCIZ  @ERROR DURING READ COMMAND WHICH WAS ISSUED TO SEEKING DRIVE@
 6135  034314  042040  051125  047111
 6136  034322  020107  042522  042101
 6137  034330  041440  046517  040515
 6138  034336  042116  053440  044510
 6139  034344  044103  053440  051501
 6140  034352  044440  051523  042525
 6141  034360  020104  047524  051440
 6142  034366  042505  044513  043516
 6143  034374  042040  044522  042526
 6144  034402     000
 6145  034403     104  052101  020101  EM35:   .ASCIZ  @DATA INCORRECT FROM HEADER READ COMMAND ISSUED TO A SEEKING DRIVE@
 6146  034410  047111  047503  051122
 6147  034416  041505  020124  051106
 6148  034424  046517  044040  040505
 6149  034432  042504  020122  042522
 6150  034440  042101  041440  046517
 6151  034446  040515  042116  044440
 6152  034454  051523  042525  020104
 6153  034462  047524  040440  051440
```

```
6154  034470  042505  044513  043516
6155  034476  042040  044522  042526
6156  034504     000
6157  034505     047  054116  042515  EM36:   .ASCIZ  @'NXME' DIDN'T SET WHEN LOCATION 760000 ADDRESSED@
6158  034512  020047  044504  047104
6159  034520  052047  051440  052105
6160  034526  053440  042510  020116
6161  034534  047514  040503  044524
6162  034542  047117  033440  030066
6163  034550  030060  020060  042101
6164  034556  051104  051505  042523
6165  034564  000104
6166  034566  044047  023505  042040  EM37:   .ASCIZ  @'ME' DIDN'T SET WITH 'NXME'@
6167  034574  042111  023516  020124
6168  034602  042523  020124  044527
6169  034610  044124  023440  054116
6170  034616  042515  000047
6171  034622  042447  051122  020047  EM40:   .ASCIZ  @'ERR' DIDN?T SET WITH 'NXME'@
6172  034630  044504  047104  052067
6173  034636  051440  052105  053440
6174  034644  052111  020110  047047
6175  034652  046530  023505     000
6176  034657     105  051122  051117  EM41:   .ASCIZ  @ERROR FORMATTING TRACK 0@
6177  034664  043040  051117  040515
6178  034672  052124  047111  020107
6179  034700  051124  041501  020113
6180  034706  000060
6181  034710  051105  047522  020122  EM42:   .ASCIZ  @ERROR READING THE HEADER FROM THE TEST SECTOR@
6182  034716  042522  042101  047111
6183  034724  020107  044124  020105
6184  034732  042510  042101  051105
6185  034740  043040  047522  020115
6186  034746  044124  020105  042524
6187  034754  052123  051440  041505
6188  034762  047524  000122
6189  034766  042523  052103  051117  EM43:   .ASCIZ  @SECTOR FIELD FROM HEADER IS NOT CORRECT@
6190  034774  043040  042511  042114
6191  035002  043040  047522  020115
6192  035010  042510  042101  051105
6193  035016  044440  020123  047516
6194  035024  020124  047503  051122
6195  035032  041505  000124
6196  035036  051105  047522  020122  EM44:   .ASCIZ  @ERROR WRITING SECTOR ADDRESS IN DATA FIELD@
6197  035044  051127  052111  047111
6198  035052  020107  042523  052103
6199  035060  051117  040440  042104
6200  035066  042522  051523  044440
6201  035074  020116  040504  040524
6202  035102  043040  042511  042114
6203  035110     000
6204  035111     047  047523  023524  EM45:   .ASCIZ  @'SOT' OR SECTOR ADDRESS REIGSTER IS NOT CORRECT@
6205  035116  047440  020122  042523
6206  035124  052103  051117  040440
6207  035132  042104  042522  051523
6208  035140  051040  044505  051507
6209  035146  042524  020122  051511
```

```
6210  035154  047040  052117  041440
6211  035162  051117  042522  052103
6212  035170     000
6213  035171     104  052101  020101  EM46:   .ASCIZ  @DATA READ FROM THE SECTOR IS NOT CORRECT@
6214  035176  042522  042101  043040
6215  035204  047522  020115  044124
6216  035212  020105  042523  052103
6217  035220  051117  044440  020123
6218  035226  047516  020124  047503
6219  035234  051122  041505  000124
6220  035242  042523  052103  051117  EM47:   .ASCII  @SECTOR CONTENTS WRONG, DATA SHOULD BE THE@<15><12>
6221  035250  041440  047117  042524
6222  035256  052116  020123  051127
6223  035264  047117  026107  042040
6224  035272  052101  020101  044123
6225  035300  052517  042114  041040
6226  035306  020105  044124  006505
6227  035314     012
6228  035315     124  040522  045503          .ASCIZ  @TRACK NUMBER OF THE CURRENTLY SELECTED HEAD@
6229  035322  047040  046525  042502
6230  035330  020122  043117  052040
6231  035336  042510  041440  051125
6232  035344  042522  052116  054514
6233  035352  051440  046105  041505
6234  035360  042524  020104  042510
6235  035366  042101     000
6236  035371     105  051122  051117  EM50:   .ASCIZ  @ERROR AFTER 2 WORD READ STARTING AT LOC 1777776@
6237  035376  040440  052106  051105
6238  035404  031040  053440  051117
6239  035412  020104  042522  042101
6240  035420  051440  040524  052122
6241  035426  047111  020107  052101
6242  035434  046040  041517  030440
6243  035442  033467  033467  000066
6244  035450  046447  054105  023460  EM51:   .ASCII  @'MEX0' DIDN'T SET AFTER 2 WORD READ STARTING@<15><12>
6245  035456  042040  042111  023516
6246  035464  020124  042523  020124
6247  035472  043101  042524  020122
6248  035500  020062  047527  042122
6249  035506  051040  040505  020104
6250  035514  052123  051101  044524
6251  035522  043516  005015
6252  035526  052101  046040  041517          .ASCIZ  @AT LOC 1777776@
6253  035534  030440  033467  033467
6254  035542  000066
6255  035544  046447  054105  023461  EM52:   .ASCIZ  @'MEX1' SET AFTER 2 WORD READ STARTING AT LOC 1777776@
6256  035552  051440  052105  040440
6257  035560  052106  051105  031040
6258  035566  053440  051117  020124
6259  035574  042522  042101  051440
6260  035602  040524  052122  047111
6261  035610  020107  052101  046040
6262  035616  041517  030440  033467
6263  035624  033467  000066
6264  035630  047503  052116  047105  EM53:   .ASCII  @CONTENTS OF LOC 200000 WRONG AFTER 2 WORD READ@<15><12>
6265  035636  051524  047440  020106
```

```
6266  035644  047514  020103  030062
6267  035652  030060  030060  053440
6268  035660  047522  043516  040440
6269  035666  052106  051105  031040
6270  035674  053440  051117  020104
6271  035702  042522  042101  005015
6272  035710  052123  051101  044524          .ASCIZ  @STARTING AT LOC 1777776@
6273  035716  043516  040440  020124
6274  035724  047514  020103  033461
6275  035732  033467  033067      000
6276  035737     105  051122  051117  EM54:   .ASCIZ  @ERROR AFTER 2 WORD READ STARTING AT LOC 3777776@
6277  035744  040440  052106  051105
6278  035752  031040  053440  051117
6279  035760  020104  042522  042101
6280  035766  051440  040524  052122
6281  035774  047111  020107  052101
6282  036002  046040  041517  031440
6283  036010  033467  033467  000066
6284  036016  046447  054105  023460  EM55:   .ASCIZ  @'MEXO' DIDN'T CLEAR AFTER 2 WORD STARTING AT LOC 3777776@
6285  036024  042040  042111  023516
6286  036032  020124  046103  040505
6287  036040  020122  043101  042524
6288  036046  020122  020062  047527
6289  036054  042122  051440  040524
6290  036062  052122  047111  020107
6291  036070  052101  046040  041517
6292  036076  031440  033467  033467
6293  036104  000066
6294  036106  046447  054105  023461  EM56:   .ASCIZ  @'MEX1' DIDN'T SET AFTER 2 WORD READ STARTING AT LOC 3777776@
6295  036114  042040  042111  023516
6296  036122  020124  042523  020124
6297  036130  043101  042524  020122
6298  036136  020062  047527  042122
6299  036144  051040  040505  020104
6300  036152  052123  051101  044524
6301  036160  043516  040440  020124
6302  036166  047514  020103  033463
6303  036174  033467  033067      000
6304  036201     103  047117  042524  EM57:   .ASCII  @CONTENTS OF LOC 400000 WRONG AFTER 2 WORD READ@<15><12>
6305  036206  052116  020123  043117
6306  036214  046040  041517  032040
6307  036222  030060  030060  020060
6308  036230  051127  047117  020107
6309  036236  043101  042524  020122
6310  036244  020062  047527  042122
6311  036252  051040  040505  006504
6312  036260     012
6313  036261     123  040524  052122          .ASCIZ  @STARTING AT LOCATION 3777776@
6314  036266  047111  020107  052101
6315  036274  046040  041517  052101
6316  036302  047511  020116  033463
6317  036310  033467  033067      000
6318  036315     105  051122  051117  EM60:   .ASCII  @ERROR RETURNING SECTOR TO PDP-11 MODE USING A@<15><12>
6319  036322  051040  052105  051125
6320  036330  044516  043516  051440
6321  036336  041505  047524  020122
```

```
6322  036344  047524  050040  050104
6323  036352  030455  020061  047515
6324  036360  042504  052440  044523
6325  036366  043516  040440  005015
6326  036374  020062  047527  042122          .ASCIZ  @2 WORD WRITE@
6327  036402  053440  044522  042524
6328  036410     000
6329  036411     105  051122  051117  EM61:   .ASCIZ  @ERROR WRITING TEST SECTOR(S)@
6330  036416  053440  044522  044524
6331  036424  043516  052040  051505
6332  036432  020124  042523  052103
6333  036440  051117  051450  000051
6334  036446  051105  047522  020122  EM62:   .ASCIZ  @ERROR READING TEST SECTOR(S)@
6335  036454  042522  042101  047111
6336  036462  020107  042524  052123
6337  036470  051440  041505  047524
6338  036476  024122  024523     000
6339  036503     104  044522  042526  EM63:   .ASCII  @DRIVE DIDN'T RETURN TO CYL 0 FROM THE MAXIMUM@<15><12>
6340  036510  042040  042111  023516
6341  036516  020124  042522  052524
6342  036524  047122  052040  020117
6343  036532  054503  020114  020060
6344  036540  051106  046517  052040
6345  036546  042510  046440  054101
6346  036554  046511  046525  005015
6347  036562  054503  020114  047117          .ASCIZ  @CYL ON CONTROLLER POWER FAIL@
6348  036570  041440  047117  051124
6349  036576  046117  042514  020122
6350  036604  047520  042527  020122
6351  036612  040506  046111     000
6352  036617     103  047117  042524  EM64:   .ASCII  @CONTENTS OF MEMORY CHANGED DURING POWER@<15><12>
6353  036624  052116  020123  043117
6354  036632  046440  046505  051117
6355  036640  020131  044103  047101
6356  036646  042507  020104  052504
6357  036654  044522  043516  050040
6358  036662  053517  051105  005015
6359  036670  040506  046111  053440          .ASCIZ  @FAIL WHILE READING THE DISK@
6360  036676  044510  042514  051040
6361  036704  040505  044504  043516
6362  036712  052040  042510  042040
6363  036720  051511  000113
6364  036724  051447  047525  023514  EM65:   .ASCIZ  @'SUOL' SET WITH DRIVE DISABLED@
6365  036732  051440  052105  053440
6366  036740  052111  020110  051104
6367  036746  053111  020105  044504
6368  036754  040523  046102  042105
6369  036762     000
6370  036763     047  052523  046117  EM66:   .ASCIZ  @'SUOL' NOT SET WITH DRIVE ENABLED@
6371  036770  020047  047516  020124
6372  036776  042523  020124  044527
6373  037004  044124  042040  044522
6374  037012  042526  042440  040516
6375  037020  046102  042105     000
6376  037025     047  052523  050127  EM67:   .ASCIZ  @'SUMP' NOT SET WITH DRIVE SET TO 'READ ONLY'@
6377  037032  020047  047516  020124
```

```
 6378  037040  042523  020124  044527
 6379  037046  044124  042040  044522
 6380  037054  042526  051440  052105
 6381  037062  052040  020117  051047
 6382  037070  040505  020104  047117
 6383  037076  054514  000047
 6384  037102  051447  053525  023520  EM70:  .ASCIZ  @'SUWP' SET WITH DRIVE SET TO 'READ/WRITE'@
 6385  037110  051440  052105  053440
 6386  037116  052111  020110  051104
 6387  037124  053111  020105  042523
 6388  037132  020124  047524  023440
 6389  037140  042522  042101  053457
 6390  037146  044522  042524  000047
 6391  037154  051447  053525  023520  EM71:  .ASCII  @'SUWP' SET WITH RP11 'WRITE LOCKOUT' SET, CYLINDER@<15><12>
 6392  037162  051440  052105  053440
 6393  037170  052111  020110  050122
 6394  037176  030461  023440  051127
 6395  037204  052111  020105  047514
 6396  037212  045503  052517  023524
 6397  037220  051440  052105  020054
 6398  037226  054503  044514  042116
 6399  037234  051105  005015
 6400  037240  047514  023501  020123         .ASCIZ  @LOA'S = 0, AND RPCA = 0@
 6401  037246  020075  026060  040440
 6402  037254  042116  051040  041520
 6403  037262  020101  020075  000060
 6404  037270  051447  053525  023520  EM72:  .ASCIZ  @'SUWP' SET WITH RPCA = 2 & CYL LOA'S = 0@
 6405  037276  051440  052105  053440
 6406  037304  052111  020110  050122
 6407  037312  040503  036440  031040
 6408  037320  023040  041440  046131
 6409  037326  046040  040517  051447
 6410  037334  036440  030040     000
 6411  037341     047  052523  050127  EM73:  .ASCIZ  @'SUWP' NOT SET; RPCA = VALUE IN CYL LOA'S@
 6412  037346  020047  047516  020124
 6413  037354  042523  035524  051040
 6414  037362  041520  020101  020075
 6415  037370  040526  052514  020105
 6416  037376  047111  041440  046131
 6417  037404  046040  040517  051447
 6418  037412     000
 6419  037413     047  052523  050127  EM74:  .ASCIZ  @'SUWP' SET WITH RPCA ONE GREATER THAN CYL LOA VALUE@
 6420  037420  020047  042523  020124
 6421  037426  044527  044124  051040
 6422  037434  041520  020101  047117
 6423  037442  020105  051107  040505
 6424  037450  042524  020122  044124
 6425  037456  047101  041440  046131
 6426  037464  046040  040517  053040
 6427  037472  046101  042525     000
 6428  037477     047  052523  050127  EM75:  .ASCIZ  @'SUWP' SET WITH DRIVE LOA SWITCHES EQUAL TO SELECTED DRIVE@
 6429  037504  020047  042523  020124
 6430  037512  044527  044124  042040
 6431  037520  044522  042526  046040
 6432  037526  040517  051440  044527
 6433  037534  041524  042510  020123
```

```
6434  037542  050505  040525  020114
6435  037550  047524  051440  046105
6436  037556  041505  042524  020104
6437  037564  051104  053111  000105
6438  037572  051104  053111  020105  EM76:   .ASCIZ  @DRIVE HAS GONE OFFLINE@
6439  037600  040510  020123  047507
6440  037606  042516  047440  043106
6441  037614  044514  042516     000
6442  037621     104  044522  042526  EM77:   .ASCIZ  @DRIVE IS UNSAFE@
6443  037626  044440  020123  047125
6444  037634  040523  042506     000
6445  037641     104  044522  042526  EM100:  .ASCIZ  @DRIVE TIMED OUT@
6446  037646  052040  046511  042105
6447  037654  047440  052125     000
6448  037661     103  047117  051124  EM101:  .ASCIZ  @CONTROLLER TIMED OUT@
6449  037666  046117  042514  020122
6450  037674  044524  042515  020104
6451  037702  052517  000124
6452  037706  040504  040524  041440  EM102:  .ASCIZ  @DATA COMPARE ERROR@
6453  037714  046517  040520  042522
6454  037722  042440  051122  051117
6455  037730     000
6456  037731     104  052101  020101  EM103:  .ASCIZ  @DATA COMPARE ERROR (MEMORY MANAGEMENT ENABLED)@
6457  037736  047503  050115  051101
6458  037744  020105  051105  047522
6459  037752  020122  046450  046505
6460  037760  051117  020131  040515
6461  037766  040516  042507  042515
6462  037774  052116  042440  040516
6463  040002  046102  042105  000051
6464  040010  042447  051122  020047  EM106:  .ASCIZ  @'ERR' DIDN'T SET WITH 'WCE'@
6465  040016  044504  047104  052047
6466  040024  051440  052105  053440
6467  040032  052111  020110  053447
6468  040040  042503  000047
6469  040044  044047  023505  042040  EM107:  .ASCIZ  @'HE' DIDN'T SET WITH 'PROG'@
6470  040052  042111  023516  020124
6471  040060  042523  020124  044527
6472  040066  044124  023440  051120
6473  040074  043517  000047
6474  040100  051104  053111  020105  EM112:  .ASCIZ  @DRIVE UNSAFE AFTER POWER FAIL@
6475  040106  047125  040523  042506
6476  040114  040440  052106  061105
6477  040122  050040  053517  061105
6478  040130  043040  044501  000114
6479  040136  051447  053525  023520  EM113:  .ASCIZ  @'SUMP' NOT SET WITH DRIVE LOA SWITCHES GREATER THAN SELECTED DRIVE@
6480  040144  047040  052117  051440
6481  040152  052105  053440  052111
6482  040160  020110  051104  053111
6483  040166  020105  047514  020101
6484  040174  053523  052111  044103
6485  040202  051505  043440  042522
6486  040210  052101  051105  052040
6487  040216  040510  020116  042523
6488  040224  042514  052103  042105
6489  040232  042040  044522  042526
```

```
 6490  040240      000
 6491
 6492                                ;'DH' MESSAGE LINES
 6493
 6494  040241      122  040520  051104  DH1:    .ASCIZ  @RPADR@
 6495  040246      000
 6496  040247      124  051505  020124  DH2:    .ASCIZ  @TEST #  ERR PC  DRV #   RPDS    RPER    RPCS    RPWC@
 6497  040254   020043  042440  051122
 6498  040262   050040  020103  042040
 6499  040270   053122  021440  020040
 6500  040276   051040  042120  020123
 6501  040304   020040  051040  042520
 6502  040312   020122  020040  051040
 6503  040320   041520  020123  020040
 6504  040326   051040  053520  000103
 6505  040334   050122  040502  020040  DH2A:   .ASCIZ  @RPBA    RPCA    RPDA    RPM1    SUCA    SILO@
 6506  040342   020040  050122  040503
 6507  040350   020040  020040  050122
 6508  040356   040504  020040  020040
 6509  040364   050122  030515  020040
 6510  040372   020040  052523  040503
 6511  040400   020040  020040  044523
 6512  040406   047514      000
 6513  040411      124  051505  020124  DH5:    .ASCIZ  @TEST #  ERR PC  DRV #   PATRN # RPDS    RPER    RPCS@
 6514  040416   020043  042440  051122
 6515  040424   050040  020103  042040
 6516  040432   053122  021440  020040
 6517  040440   050040  052101  047122
 6518  040446   021440  051040  042120
 6519  040454   020123  020040  051040
 6520  040462   042520  020122  020040
 6521  040470   051040  041520  000123
 6522  040476   050122  041527  020040  DH5A:   .ASCIZ  @RPWC    RPBA    RPCA    RPDA    RPM1    SUCA    SILO@
 6523  040504   020040  050122  040502
 6524  040512   020040  020040  050122
 6525  040520   040503  020040  020040
 6526  040526   050122  040504  020040
 6527  040534   020040  050122  030515
 6528  040542   020040  020040  052523
 6529  040550   040503  020040  020040
 6530  040556   044523  047514      000
 6531  040563      124  051505  020124  DH11:   .ASCIZ  @TEST #  ERR PC  DRV #   EXPT'D 1ST WORD 2ND WORD@
 6532  040570   020043  042440  051122
 6533  040576   050040  020103  042040
 6534  040604   053122  021440  020040
 6535  040612   042440  050130  023524
 6536  040620   020104  051461  020124
 6537  040626   047527  042122  031040
 6538  040634   042116  053440  051117
 6539  040642   000104
 6540  040644   042524  052123  021440  DH12:   .ASCIZ  @TEST #  ERR PC  DRV #   EXPT'D  RECV'D  LOCN@
 6541  040652   020040  051105  020122
 6542  040660   041520  020040  051104
 6543  040666   020126  020043  020040
 6544  040674   054105  052120  042047
 6545  040702   020040  042522  053103
```

```
6546  040710  042047  020040  047514
6547  040716  047103     000
6548  040721     124  051505  020124  DH14:   .ASCIZ  @TEST #  ERR PC  DRV #   RPDS    RPER    RPCS@
6549  040726  020043  042440  051122
6550  040734  050040  020103  042040
6551  040742  053122  021440  020040
6552  040750  051040  042120  020123
6553  040756  020040  051040  042520
6554  040764  020122  020040  051040
6555  040772  041520  000123
6556  040776  020040  020040  020040  DH16:   .ASCII  @                        EXPT'D  ACTUAL@<15><12>
6557  041004  020040  020040  020040
6558  041012  020040  020040  020040
6559  041020  020040  020040  020040
6560  041026  054105  052120  042047
6561  041034  020040  041501  052524
6562  041042  046101  005015
6563  041046  042524  052123  021440          .ASCIZ  @TEST #  ERR PC  DRV #   SUCA    SUCA@
6564  041054  020040  051105  020122
6565  041062  041520  020040  051104
6566  041070  020126  020043  020040
6567  041076  052523  040503  020040
6568  041104  020040  052523  040503
6569  041112     000
6570  041113     124  051505  020124  DH25:   .ASCIZ  @TEST #  ERR PC  DRV #   EXPT'D  RECV'D@
6571  041120  020043  042440  051122
6572  041126  050040  020103  042040
6573  041134  053122  021440  020040
6574  041142  042440  050130  023524
6575  041150  020104  051040  041505
6576  041156  023526  000104
6577  041162  042510  042101  051105  DH25A:  .ASCIZ  @HEADER CONTENTS@
6578  041170  041440  047117  042524
6579  041176  052116  000123
6580  041202  020040  020040  020040  DH43:   .ASCII  @                        EXPT'D  ACTUAL@<15><12>
6581  041210  020040  020040  020040
6582  041216  020040  020040  020040
6583  041224  020040  020040  020040
6584  041232  054105  052120  042047
6585  041240  020040  041501  052524
6586  041246  046101  005015
6587  041252  042524  052123  021440          .ASCIZ  @TEST #  ERR PC  DRV #   SECTOR  SECTOR@
6588  041260  020040  051105  020122
6589  041266  041520  020040  051104
6590  041274  020126  020043  020040
6591  041302  042523  052103  051117
6592  041310  020040  042523  052103
6593  041316  051117     000
6594  041321     122  042120  020123  DH43A:  .ASCIZ  @RPDS    RPER    RPCS    RPWC    RPBA    RPCA    RPDA@
6595  041326  020040  051040  042520
6596  041334  020122  020040  051040
6597  041342  041520  020123  020040
6598  041350  051040  053520  020103
6599  041356  020040  051040  041120
6600  041364  020101  020040  051040
6601  041372  041520  020101  020040
```

```
  6602  041400  051040  042120  000101
  6603  041406  042510  042101  051105  DH43B:  .ASCIZ  @HEADER CONTENTS@
  6604  041414  041440  047117  042524
  6605  041422  052116  000123
  6606  041426  020040  020040  020040  DH45:   .ASCII  @                        EXPT'D  'SOT'   SECTOR@<15><12>
  6607  041434  020040  020040  020040
  6608  041442  020040  020040  020040
  6609  041450  020040  020040  020040
  6610  041456  054105  052120  042047
  6611  041464  020040  051447  052117
  6612  041472  020047  020040  042523
  6613  041500  052103  051117  005015
  6614  041506  042524  052123  021440          .ASCIZ  @TEST #  ERR PC  DRV #   VALUE  COUNTER  ADDR@
  6615  041514  020040  051105  020122
  6616  041522  041520  020040  051104
  6617  041530  020126  020043  020040
  6618  041536  040526  052514  020105
  6619  041544  041440  052517  052116
  6620  041552  051105  020040  042101
  6621  041560  051104     000
  6622  041563     122  042120  020123  DH45A:  .ASCIZ  @RPDS    RPER    RPCS    RPDA@
  6623  041570  020040  051040  042520
  6624  041576  020122  020040  051040
  6625  041604  041520  020123  020040
  6626  041612  051040  042120  000101
  6627  041620  020040  020040  020040  DH46:   .ASCII  @                                EXPT'D  RECV'D@<15><12>
  6628  041626  020040  020040  020040
  6629  041634  020040  020040  020040
  6630  041642  020040  020040  020040
  6631  041650  020040  020040  020040
  6632  041656  020040  054105  052120
  6633  041664  042047  020040  042522
  6634  041672  053103  042047  005015
  6635  041700  042524  052123  021440          .ASCIZ  @TEST #  ERR PC  DRV #   SECTOR  DATA    DATA@
  6636  041706  020040  051105  020122
  6637  041714  041520  020040  051104
  6638  041722  020126  020043  020040
  6639  041730  042523  052103  051117
  6640  041736  020040  040504  040524
  6641  041744  020040  020040  040504
  6642  041752  040524     000
  6643  041755     124  051505  020124  DH47:   .ASCIZ  @TEST #  ERR PC  DRV #   EXPT'D  RECV'D@
  6644  041762  020043  042440  051122
  6645  041770  050040  020103  042040
  6646  041776  053122  021440  020040
  6647  042004  042440  050130  023524
  6648  042012  020104  051040  041505
  6649  042020  023526  000104
  6650  042024  050122  051504  020040  DH47A:  .ASCIZ  @RPDS    RPER    RPCS    RPCA    RPDA@
  6651  042032  020040  050122  051105
  6652  042040  020040  020040  050122
  6653  042046  051503  020040  020040
  6654  042054  050122  040503  020040
  6655  042062  020040  050122  040504
  6656  042070     000
  6657  042071     124  051505  020124  DH51:   .ASCIZ  @TEST #  ERR PC  DRV #   RPDS    RPER    RPCS    RPWC    RPBA@
```

```
 6658  042076  020043  042440  051122
 6659  042104  050040  020103  042040
 6660  042112  053122  021440  020040
 6661  042120  051040  042120  020123
 6662  042126  020040  051040  042520
 6663  042134  020122  020040  051040
 6664  042142  041520  020123  020040
 6665  042150  051040  053520  020103
 6666  042156  020040  051040  041120
 6667  042164  000101
 6668  042166  042524  052123  021440  DH63:   .ASCIZ  @TEST #  ERR PC  DRV #   RPDS    RPER    RPCS    SUCA@
 6669  042174  020040  051105  020122
 6670  042202  041520  020040  051104
 6671  042210  020126  020043  020040
 6672  042216  050122  051504  020040
 6673  042224  020040  050122  051105
 6674  042232  020040  020040  050122
 6675  042240  051503  020040  020040
 6676  042246  052523  040503     000
 6677  042253     124  051505  020124  DH71:   .ASCIZ  @TEST #  ERR PC  DRV #   RPDS    RPER    RPCS    RPCA@
 6678  042260  020043  042440  051122
 6679  042266  050040  020103  042040
 6680  042274  053122  021440  020040
 6681  042302  051040  042120  020123
 6682  042310  020040  051040  042520
 6683  042316  020122  020040  051040
 6684  042324  041520  020123  020040
 6685  042332  051040  041520  000101
 6686  042340  042524  052123  021440  DH102:  .ASCIZ  @TEST #  ERR PC  DRV #  SEC STRT CYL     TRK     SEC     PATTERN@
 6687  042346  020040  051105  020122
 6688  042354  041520  020040  051104
 6689  042362  020126  020043  051440
 6690  042370  041505  051440  051124
 6691  042376  020124  054503  020114
 6692  042404  020040  020040  051124
 6693  042412  020113  020040  020040
 6694  042420  042523  020103  020040
 6695  042426  020040  040520  052124
 6696  042434  051105  000116
 6697  042440  020040  020040  020040  DH102A: .ASCII  @        GOOD    BAD@<15><12>
 6698  042446  020040  047507  042117
 6699  042454  020040  020040  040502
 6700  042462  006504     012
 6701  042465     101  042104  020122          .ASCIZ  @ADDR    DATA    DATA@
 6702  042472  020040  042040  052101
 6703  042500  020101  020040  042040
 6704  042506  052101  000101
 6705  042512  044513  040520  031522  DH103:  .ASCIZ  @KIPAR3  KIPAR4  KIPAR5  KIPDR3  KIPDR4  KIPDR5@
 6706  042520  020040  044513  040520
 6707  042526  032122  020040  044513
 6708  042534  040520  032522  020040
 6709  042542  044513  042120  031522
 6710  042550  020040  044513  042120
 6711  042556  032122  020040  044513
 6712  042564  042120  032522     000
 6713  042571     126  051111  052524  DH103A: .ASCII  @VIRTUAL GOOD    BAD@<15><12>
```

```
 6714  042576  046101  043440  047517
 6715  042604  020104  020040  041040
 6716  042612  042101  005015
 6717  042616  042101  051104  020040          .ASCIZ  @ADDR    DATA    DATA@
 6718  042624  020040  040504  040524
 6719  042632  020040  020040  040504
 6720  042640  040524     000
 6721  042643     124  052117  046101  DH105:  .ASCIZ  @TOTAL COMPARE ERRORS:@
 6722  042650  041440  046517  040520
 6723  042656  042522  042440  051122
 6724  042664  051117  035123     000
 6725  042671     125  051516  042503  DH110:  .ASCIZ  @UNSCESSFUL AFTER 3 RETRIES@
 6726  042676  051523  052506  020114
 6727  042704  043101  042524  020122
 6728  042712  020063  042522  051124
 6729  042720  042511  000123
 6730  042724  042522  051124  020131  DH111:  .ASCII  @RETRY SUCESSFUL@<15><12>
 6731  042732  052523  042503  051523
 6732  042740  052506  006514     012
 6733  042745     122  052105  054522          .ASCIZ  @RETRY COUNT:@
 6734  042752  041440  052517  052116
 6735  042760  000072
 6736
 6737                                  .EVEN
 6738
 6739                                  ;'DT' WORDS
 6740
 6741  042762  001324                  DT1:    .WORD   RPADR
 6742  042764  001160  001116  001176  DT2:    .WORD   $TMPO,$ERRPC,CHKDRV,$RPDS,$RPER,$RPCS,$RPWC
 6743  042772  001334  001336  001340
 6744  043000  001342
 6745  043002  001344  001346  001350          .WORD   $RPBA,$RPCA,$RPDA,$RPM1,$SUCA,$SILO
 6746  043010  001352  001354  001356
 6747  043016  001160  001116  001176  DT5:    .WORD   $TMPO,$ERRPC,CHKDRV,PATNUM,$RPDS,$RPER,$RPCS
 6748  043024  001204  001334  001336
 6749  043032  001340
 6750  043034  001342  001344  001346          .WORD   $RPWC,$RPBA,$RPCA,$RPDA,$RPM1,$SUCA,$SIL O
 6751  043042  001350  001352  001354
 6752  043050  001356
 6753  043052  001160  001116  001176  DT11:   .WORD   $TMPO,$ERRPC,CHKDRV,$GDDAT,BUFFER,BUFFER+2
 6754  043060  001124  043602  043604
 6755  043066  001160  001116  001176  DT12:   .WORD   $TMPO,$ERRPC,CHKDRV,$GDDAT,$BDDAT,$BDADR
 6756  043074  001124  001126  001122
 6757  043102  001160  001116  001176  DT14:   .WORD   $TMPO,$ERRPC,CHKDRV,$RPDS,$RPER,$RPCS
 6758  043110  001334  001336  001340
 6759  043116  001160  001116  001176  DT16:   .WORD   $TMPO,$ERRPC,CHKDRV,MAXCYL,$SUCA
 6760  043124  001210  001354
 6761  043130  001160  001116  001176  DT25:   .WORD   $TMPO,$ERRPC,CHKDRV,$GDDAT,$BDDAT
 6762  043136  001124  001126
 6763  043142  043602  043604  043606          .WORD   BUFFER,BUFFER+2,BUFFER+4
 6764  043150  001160  001116  001176  DT43:   .WORD   $TMPO,$ERRPC,CHKDRV,$GDDAT,BUFFER+4
 6765  043156  001124  043606
 6766  043162  001334  001336  001340          .WORD   $RPDS,$RPER,$RPCS,$RPWC,$RPBA,$RPCA,$RPDA
 6767  043170  001342  001344  001346
 6768  043176  001350
 6769  043200  043602  043604  043606          .WORD   BUFFER,BUFFER+2,BUFFER+4
```

```
 6770  043206  001160  001116  001176  DT45:   .WORD   STMPO,SERRPC,CHKDRV,$GDDAT,$BDDAT,$BDADR
 6771  043214  001124  001126  001122
 6772  043222  001334  001336  001340          .WORD   $RPDS,$RPER,$RPCS,$RPDA
 6773  043230  001350
 6774  043232  001160  001116  001176  DT46:   .WORD   STMPO,SERRPC,CHKDRV,$GDADR,$GDDAT,$BDDAT
 6775  043240  001120  001124  001126
 6776  043246  001160  001116  001176  DT47:   .WORD   STMPO,SERRPC,CHKDRV,$GDDAT,BUFFER
 6777  043254  001124  043602
 6778  043260  001334  001336  001340          .WORD   $RPDS,$RPER,$RPCS,$RPCA,$RPDA
 6779  043266  001346  001350
 6780  043272  001160  001116  001176  DT51:   .WORD   STMPO,SERRPC,CHKDRV,$RPDS,$RPER,$RPCS,$RPWC,$RPBA
 6781  043300  001334  001336  001340
 6782  043306  001342  001344
 6783  043312  001160  001116  001176  DT63:   .WORD   STMPO,SERRPC,CHKDRV,$RPDS,$RPER,$RPCS,$SUCA
 6784  043320  001334  001336  001340
 6785  043326  001354
 6786  043330  001160  001116  001176  DT71:   .WORD   STMPO,SERRPC,CHKDRV,$RPDS,$RPER,$RPCS,$RPCA
 6787  043336  001334  001336  001340
 6788  043344  001346
 6789  043346  001160  001116  001176  DT102:  .WORD   STMPO,SERRPC,CHKDRV,CMBUF,CMCYL,CMTRK,CMSEC,PATNUM
 6790  043354  025362  025354  025360
 6791  043362  025356  001204
 6792  043366  001122  001124  001126          .WORD   $BDADR,$GDDAT,$BDDAT
 6793  043374  001160  001116  001176  DT103:  .WORD   STMPO,SERRPC,CHKDRV,CMBUF,CMCYL,CMTRK,CMSEC,PATNUM
 6794  043402  025362  025354  025360
 6795  043410  025356  001204
 6796  043414  172346  172350  172352          .WORD   KIPAR3,KIPAR4,KIPAR5,KIPDR3,KIPDR4,KIPDR5
 6797  043422  172306  172310  172312
 6798  043430  001122  001124  001126  DT104:  .WORD   $BDADR,$GDDAT,$BDDAT
 6799  043436  025346            DT105:  .WORD   ERCTR
 6800  043440  001314            DT111:  .WORD   RETRY
 6801
 6802                            ;'DF' BLOCKS
 6803
 6804  043442  000001            DF1:    .WORD   1
 6805  043444     001                    .BYTE   1
 6806  043445     000                    .BYTE   0
 6807
 6808  043446  000002            DF2:    .WORD   2
 6809  043450     007                    .BYTE   7
 6810  043451     000                    .BYTE   0
 6811  043452  040334                    .WORD   DH2A
 6812  043454     006                    .BYTE   6
 6813  043455     000                    .BYTE   0
 6814
 6815  043456  000002            DF5:    .WORD   2
 6816  043460     007                    .BYTE   7
 6817  043461     010                    .BYTE   10
 6818  043462  040476                    .WORD   DH5A
 6819  043464     007                    .BYTE   7
 6820  043465     000                    .BYTE   0
 6821
 6822  043466  000001            DF11:   .WORD   1
 6823  043470     006                    .BYTE   6
 6824  043471     000                    .BYTE   0
 6825
```

```
6826  043472  000002          DF25:   .WORD   2
6827  043474      005                 .BYTE   5
6828  043475      000                 .BYTE   0
6829  043476  041162                  .WORD   DH25A
6830  043500      003                 .BYTE   3
6831  043501      000                 .BYTE   0
6832
6833  043502  000003          DF43:   .WORD   3
6834  043504      006                 .BYTE   6
6835  043505      000                 .BYTE   0
6836  043506  041321                  .WORD   DH43A
6837  043510      005                 .BYTE   5
6838  043511      000                 .BYTE   0
6839  043512  041406                  .WORD   DH43B
6840  043514      003                 .BYTE   3
6841  043515      000                 .BYTE   0
6842
6843  043516  000002          DF45:   .WORD   2
6844  043520      006                 .BYTE   6
6845  043521      000                 .BYTE   0
6846  043522  041563                  .WORD   DH45A
6847  043524      004                 .BYTE   4
6848  043525      000                 .BYTE   0
6849
6850  043526  000002          DF47:   .WORD   2
6851  043530      005                 .BYTE   5
6852  043531      000                 .BYTE   0
6853  043532  042024                  .WORD   DH47A
6854  043534      005                 .BYTE   5
6855  043535      000                 .BYTE   0
6856
6857  043536  000001          DF51:   .WORD   1
6858  043540      010                 .BYTE   8.
6859  043541      000                 .BYTE   0
6860
6861  043542  000001          DF53:   .WORD   1
6862  043544      005                 .BYTE   5
6863  043545      000                 .BYTE   0
6864
6865  043546  000001          DF63:   .WORD   1
6866  043550      007                 .BYTE   7
6867  043551      000                 .BYTE   0
6868
6869  043552  000002          DF102:  .WORD   2
6870  043554      010                 .BYTE   8.
6871  043555      200                 .BYTE   200
6872  043556  042440                  .WORD   DH102A
6873  043560      000                 .BYTE   0
6874  043561      000                 .BYTE   0
6875
6876  043562  000003          DF103:  .WORD   3
6877  043564      010                 .BYTE   8.
6878  043565      200                 .BYTE   200
6879  043566  042512                  .WORD   DH103
6880  043570      006                 .BYTE   6
6881  043571      000                 .BYTE   0
```

```
                                                .WORD   DH103A
                                                .BYTE   0
                                                .BYTE   0

                                DF104:          .WORD   1
                                                .BYTE   3
                                                .BYTE   0


                                .EVEN

                        043602  BUFFER=.                ;BUFFER STARTS HERE

043602  005015  046412  044501  TITLE:  .ASCII  <15><12><12>@MAINDEC-11-DZRPY-B@
043610  042116  041505  030455
043616  026461  055104  050122
043624  026531      102
043627      015  051012  030520          .ASCIZ  <15><12>@RP11-E FUNCTIONAL LOGIC & READ/WRITE TEST@<15><12><12>
043634  026461  020105  052506
043642  041516  044524  047117
043650  046101  046040  043517
043656  041511  023040  051040
043664  040505  027504  051127
043672  052111  020105  042524
043700  052123  005015  000012
        000001                  .END
```

```
ACTMEM  001224          1236#  3491*  3492*  5174   5175   5184*  5185*  5186   5189   5191*  5192*
AIE   = 020000          1175#
ATABIT  001360          1289#  2307   2328   3744   4868   4905   5052   5475   5493
BITS    001400          1315#  2398   2445   2524   2613   2697   2747   2842   2903   2970   3088   3164   3294
                        3463   3608   3715   3767   5068   5454
BIT0  = 000001          1080#
BIT00 = 000001          1070#  1080   1315   2808
BIT01 = 000002          1069#  1079   1316   2654
BIT02 = 000004          1068#  1078   1317   2927
BIT03 = 000010          1067#  1077   1318   2499   3928
BIT04 = 000020          1066#  1076   1319
BIT05 = 000040          1065#  1075   1320
BIT06 = 000100          1064#  1074   1321
BIT07 = 000200          1063#  1073   1322   5360
BIT08 = 000400          1062#  1072   1323
BIT09 = 001000          1061#  1071   1324   3899   4046   4636
BIT1  = 000002          1079#
BIT10 = 002000          1060#  1325   2722   4030
BIT11 = 004000          1059#  1326   4643
BIT12 = 010000          1058#  1327   2804
BIT13 = 020000          1057#  1328   4037
BIT14 = 040000          1056#  1329   4618
BIT15 = 100000          1055#  1330
BIT2  = 000004          1078#
BIT3  = 000010          1077#
BIT4  = 000020          1076#
BIT5  = 000040          1075#
BIT6  = 000100          1074#
BIT7  = 000200          1073#
BIT8  = 000400          1072#
BIT9  = 001000          1071#
BLKS14  001254          1243#  3493*  3494*  3495*  3496*  3497   3517   3557   5082   5089
BLNKS2  027354          4104   4115   4122   5670#
BPTVEC= 000014          1087#
BUFFER= 043602          2413   2464   2489*  2540   2577   2579   2582   2584   2630   2680   2714   2763*  2764*
                        2765*  2789   2795   2819*  2820*  2821*  2868   2884   2888   2986   3018   3068   3104
                        3131   3186*  3187*  3188   3308   3325   3337   3345   3354   3358   3389*  3390*  3391
                        3489   4952   4955   4979   4982   5143   5392   6753   6763   6764   6769   6776   6893#
BUSADR  031216          1233#  2199*  2201*  2204*  2206*  5408   5410*
BYPASS  001200          1226#  2275*  2406*  2453*  2532*  2621*  2705*  2755*  2850*  2911*  2978*  3096*  3172*
                        3302*  3471*  3616*  3723*  3775*  5015   5041
CHKDRV  001176          1225#  2362*  2366*  2372   2374   3657   3664   3786   3794   3801   3809   3819   3827
                        3834   3842   3904*  3909   3911   3923   3925*  3927*  3928   3931   3943   3968   6742
                        6747   6753   6755   6757   6759   6761   6764   6770   6774   6776   6780   6783   6786
                        6789   6793
CHNGAD  031564          5530   5884#
CHNGPM  031332          5499   5856#
CHNGPT  031511          5522   5876#
CIR     023052          4877   4894#  4904
CKSWR = 104407          4026   4045   4617   4823#
CK.CHR  027046          5596#  5617   5633
CK.NUM  027062          3737   5421   5447   5469   5513   5525   5539   5549   5559   5613#
CK.OCT  027016          5578#  5615   5628
CLEAR = 000001          1148#  2424   2425   4843   4844   5354   5355
CLRP    025606          2381   2412   2466   2475   2483   2495   2539   2555   2567   2628   2644   2668   2711
                        2762   2779   2799   2818   2856   2917   2985   3002   3039   3058   3107   3122   3185
```

```
                    3201    3236    3321    3336    3385    3521    3538    3563    3625    3644    3750    3793    3808
                    3826    3841    5354#
CMBUF   025362      5253#   5302#   6789    6793
CMCNT   025352      5246#   5247#   5256    5258    5261#   5277    5298#
CMCYL   025354      5248#   5287#   5299#   6789    6793
CMDAT   025202      5264#   5275
CMPAR   025062      5236    5246#
CMPRT   025364      5270    5306#
CMPRTI  025470      5307    5320    5322#
CMPRX   025326      5278    5289#
CMP.CT  001320      1265#   3376    5251
CMSEC   025356      5250#   5279#   5280    5282#   5300#   6789    6793
CMSTR   025120      5251#   5281    5285    5288
CMTRK   025360      5249#   5283#   5284    5286#   5301#   6789    6793
CNTRLC  001206      1269#   2202#   2207#   2319    2339    2361#
COMMA   027323      2357    5665#
CONRDY  023616      2469    2478    2497    2548    2560    2571    2637    2652    2674    2772    2782    2802    2825
                    2877    2925    2994    3007    3044    3060    3111    3125    3194    3215    3242    3329    3348
                    3393    3531    3541    3568    3638    3647    5022#   5040
CR      = 000015     995#   4190    4200
CRLF    = 000200     996#   4161    4200
DATCMP  024546      3588    5199#
DDISP   = 177570    1002#   1208    2234
DF1     043442      1655    2146    2174    6804#
DF102   043552      2125    6869#
DF103   043562      2132    6876#
DF104   043576      2139    6886#
DF11    043466      1711    1719    1734    1763    1771    1778    1792    1813    1820    1827    1871    1878    1920
                    2026    2033    2040    2047    2054    2090    2153    2160    2188    6822#
DF2     043446      1662    1669    1676    1697    1727    1741    1755    1785    1806    1834    1842    1849    1864
                    1885    1892    1906    1935    1965    1996    2097    2104    2111    2118    2181    6808#
DF25    043472      1799    1857    6826#
DF43    043502      1899    6833#
DF45    043516      1913    6843#
DF47    043526      1928    6850#
DF5     043456      1683    1690    2003    2010    6815#
DF51    043536      1943    1950    1973    1980    6857#
DF53    043542      1748    1958    1988    6861#
DF63    043546      2018    2062    2069    2076    2083    6865#
DH1     040241      1653    6494#
DH102   042340      2123    2130    6686#
DH102A  042440      6697#   6872
DH103   042512      6705#   6879
DH103A  042571      6713#   6882
DH105   042643      2144    6721#
DH11    040563      1709    1761    6531#
DH110   042671      2165    6725#
DH111   042724      2172    6730#
DH12    040644      1717    2024    6540#
DH14    040721      1732    1769    1776    1790    1811    1818    1825    1869    1876    2031    2038    2045    2052
                    2088    2151    2158    2186    6548#
DH16    040776      1746    6556#
DH2     040247      1660    1667    1674    1695    1725    1739    1753    1783    1804    1832    1840    1847    1862
                    1883    1890    1904    1933    1963    1994    2095    2102    2109    2116    2179    6496#
DH2A    040334      6505#   6811
DH25    041113      1797    1855    1956    1986    6570#
```

```
DH2SA   041162          6577#   6829
DH43    041202          1897    6580#
DH43A   041321          6594#   6836
DH43B   041406          6603#   6839
DH45    041426          1911    6606#
DH45A   041563          6622#   6846
DH46    041620          1918    6627#
DH47    041755          1926    6643#
DH47A   042024          6650#   6853
DH5     040411          1681    1688    2001    2008    6513#
DH5A    040476          6522#   6818
DH51    042071          1941    1948    1971    1978    6657#
DH63    042166          2016    6668#
DH71    042253          2060    2067    2074    2081    6677#
DISPLA  001142          1208#   2234*   2242*   2405*   2452*   2531*   2620*   2704*   2754*   2849*   2910*   2977*   3095*
                        3171*   3301*   3470*   3615*   3722*   3774*   4029*   4655*
DISPRE  000174           959#   2242
DPB     002500          1625#   2372*   2382*   2408*   2409*   2417*   2455*   2456*   2467*   2476*   2534*   2535*   2546*
                        2558*   2569*   2623*   2624*   2635*   2672*   2707*   2708*   2716*   2757*   2758*   2769*   2770*
                        2780*   2800*   2822*   2823*   2852*   2853*   2859*   2863*   2869*   2913*   2914*   2921*   2923*
                        2980*   2981*   2991*   2992*   3003*   3004*   3040*   3042*   3049*   3057*   3098*   3099*   3108*
                        3123*   3174*   3175*   3192*   3213*   3239*   3240*   3304*   3305*   3326*   3327*   3346*   3386*
                        3388*   3473*   3474*   3529*   3539*   3566*   3618*   3619*   3636*   3645*   3725*   3726*   3749*
                        3777*   3778*   4874    4876    4884    4885    4886    4887    5051    5181*   5209
DRIVE   031323          5478    5854#
DRSTAT  027162          2292    5645#
DRVBAD  001310          1259#   2272*   2369    4905*
DRVINT  022636          3658    4846    4854#   4896
DRVMSK  001306          1258#   2363*   2364    2367*
DRVNM   031276          5466    5850#
DRVOK   004774          2365    2372#
DRVRDY  023500          2384    2861    2871    3752    4996#   5014
DRVSEL  001312          1262#   2323*   2328*   2335    2364    2369*   3962    5460*   5475*   5493*
DRVSTA  001370          1304*   2301    2326    3659    4838*   4839*   4840*   4841*   4857*   4865*   4898    5476    5479
                        5491
DRVTST  027271          2332    5660#
DRVTYP  001304          1256#   2285    2307    3744    4837*   4868*   5052    5463
DSWR  =  177570         1001#   1207    2233
DT1     042762          1654    6741#
DT102   043346          2124    6789#
DT103   043374          2131    6793#
DT104   043430          2138    6798#
DT105   043436          2145    6799#
DT11    043052          1710    1762    6753#
DT111   043440          2173    6800#
DT12    043066          1718    2025    6755#
DT14    043102          1733    1770    1777    1791    1812    1819    1826    1870    1877    2032    2039    2046    2053
                        2089    2152    2159    2187    6757#
DT16    043116          1747    6759#
DT2     042764          1661    1668    1675    1696    1726    1740    1754    1784    1805    1833    1841    1848    1863
                        1884    1891    1905    1934    1964    1995    2096    2103    2110    2117    2180    6742#
DT25    043130          1798    1856    1957    1987    6761#
DT43    043150          1898    6764#
DT45    043206          1912    6770#
DT46    043232          1919    6774#
DT47    043246          1927    6776#
```

DT5	043016	1682	1689	2002	2009	6747#
DT51	043272	1942	1949	1972	1979	6780#
DT63	043312	2017	6783#			
DT71	043330	2061	2068	2075	2082	6786#
EMTVEC=	000030	1090#	2219*	2220*		
EM1	032117	1652	5928#			
EM100	037641	2108	6445#			
EM101	037661	2115	6448#			
EM102	037706	2122	6452#			
EM103	037731	2129	6456#			
EM106	040010	2150	6464#			
EM107	040044	2157	6469#			
EM11	032514	1708	5973#			
EM112	040100	2178	6474#			
EM113	040136	2185	6479#			
EM12	032607	1715	5983#			
EM13	032740	1723	5999#			
EM14	033063	1731	6014#			
EM15	033117	1738	6019#			
EM16	033147	1745	6024#			
EM17	033204	1752	6029#			
EM2	032161	1659	5934#			
EM20	033236	1759	6034#			
EM21	033351	1767	6047#			
EM22	033461	1775	6061#			
EM23	033516	1782	6066#			
EM24	033565	1789	6073#			
EM25	033642	1796	6081#			
EM26	033712	1803	6088#			
EM27	034002	1810	6098#			
EM3	032212	1666	5939#			
EM30	034033	1817	6103#			
EM31	034066	1824	6108#			
EM32	034117	1831	6113#			
EM33	034176	1838	6121#			
EM34	034307	1846	6134#			
EM35	034403	1853	6145#			
EM36	034505	1861	6157#			
EM37	034562	1868	6166#			
EM4	032303	1673	5949#			
EM40	034622	1875	6171#			
EM41	034657	1882	6176#			
EM42	034710	1889	6181#			
EM43	034766	1896	6189#			
EM44	035036	1903	6196#			
EM45	035111	1910	6204#			
EM46	035171	1917	6213#			
EM47	035242	1924	6220#			
EM5	032340	1680	5954#			
EM50	035371	1932	6236#			
EM51	035450	1939	6244#			
EM52	035544	1947	6255#			
EM53	035630	1954	6264#			
EM54	035737	1962	6276#			
EM55	036016	1970	6284#			
EM56	036106	1977	6294#			

```
EM57    036201          1984    6304#
EM6     032412          1687    5961#
EM60    036315          1992    6318#
EM61    036411          2000    6329#
EM62    036446          2007    6334#
EM63    036503          2014    6339#
EM64    036617          2022    6352#
EM65    036724          2030    6364#
EM66    036763          2037    6370#
EM67    037025          2044    6376#
EM7     032446          1694    5966#
EM70    037102          2051    6384#
EM71    037154          2058    6391#
EM72    037270          2066    6404#
EM73    037341          2073    6411#
EM74    037413          2080    6419#
EM75    037477          2087    6428#
EM76    037572          2094    6438#
EM77    037621          2101    6442#
ENDCMP  025562          5276    5339#
ENTCYL  032000          5536    5909#
ENTNM   031250          5444    5846#
ENTPRM  026240          2321    5444#   5448    5452
ENTPRX  027014          5498    5502    5533    5567#
ENTPR1  026320          5457    5460#   5485
ENTPR2  026516          5499#
ENTPR3  026614          5516    5522#   5526
ENTPR4  026642          5528    5530#
ENTRK   032031          5546    5914#
ENTSEC  032057          5556    5918#
ENTWC   031447          5510    5870#
ERCTR   025346          5255*   5296#   5334*   5339    6799
ERR     = 100000        1177#   2471    2480    2503    2550    2562    2573    2639    2658    2676    2731    2774    2783
                        2814    2827    2879    2934    2996    3012    3046    3062    3113    3127    3196    3217    3244
                        3331    3350    3395    3533    3543    3570    3640
ERRVEC= 000004          1083#   2231    2232*   2243*   4623    4624*   4626*   4629*   4726    4727    4731*   4736*   4748*
                        4753*   4762*   4772*   4773*   5240*   5244*   5426    5427*   5429*   5431*
EXIT0   005276          2406    2423    2428    2433#
EXIT1   005712          2453    2507    2510#
EXIT10  011520          2978    2999    3073    3076#
EXIT11  012050          3096    3135    3139#
EXIT12  012702          3172    3199    3233    3256    3260#
EXIT13  013562          3302    3396    3398#
EXIT14  014606          3471    3565    3591#
EXIT15  015272          3616    3663    3685#
EXIT16  015610          3723    3754#
EXIT17  016574          3775    3929    3945#
EXIT2   006370          2532    2553    2565    2576    2590    2598#
EXIT3   007046          2621    2642    2679    2682    2686#
EXIT4   007276          2705    2727    2732    2734#
EXIT5   010022          2755    2830#
EXIT6   010356          2828    2850    2876    2890    2892#
EXIT7   010600          2911    2930    2935    2937#
FILSEC  024270          3041    5127#
FILTRK  024330          3109    5142#
FLAG    001202          1227#
```

```
FRSTER  025344          5254* 5295# 5306  5318*
GETADR  026074          2282  5408#
GETBUF  024434          3561  5174# 5193
GNS   = ******  U        958  3960  3966  3974  3991  4815  4816  4817  4818  4819  4821  4823  4824
                        4825  4826  4827
GTSWR = 104406          2254  4821#
HDR   = 004000          1173# 2769  2822  2869  2991  3003
HE    = 040000          1176# 2728  2811  2931
HIMEM   001234          1238# 3483* 3484* 3487* 3488* 5186  5189
HOMSEK= 000015          1154# 2382  2859
HT    = 000011           993# 4159  4200
ICNT    001214          1232#
IDE   = 000100          1171#
INCADR  024046          3546  3589  5076#
INCSEC  001250          1241# 3499* 3505* 5093
INCTRK  001252          1242# 3498* 3502* 5100
IOTVEC= 000020          1088# 2217* 2218*
KIPAR0= 172340          1121# 3203* 4712  4740  5202*
KIPAR1= 172342          1122# 3204* 5203*
KIPAR2= 172344          1123# 3205* 3234* 5204*
KIPAR3= 172346          1124# 5220* 6796
KIPAR4= 172350          1125# 5222* 6796
KIPAR5= 172352          1126# 5232* 6796
KIPAR6= 172354          1127#
KIPAR7= 172356          1128# 3202* 4756  5205*
KIPDR0= 172300          1110# 3206* 5223*
KIPDR1= 172302          1111# 3207* 5224*
KIPDR2= 172304          1112# 3208* 5225*
KIPDR3= 172306          1113# 5226* 6796
KIPDR4= 172310          1114# 5227* 6796
KIPDR5= 172312          1115# 5233* 6796
KIPDR6= 172314          1116#
KIPDR7= 172316          1117# 3209* 5228*
LDRFLG  001322          1266# 4963  4968* 4974  4990*
LF    = 000012           994# 4194  4200
LIMIT   025350          5251* 5252* 5297# 5322  5324*
LMTBK   001262          1246# 2266  2268  5519
LODSEC  024236          2990  5113#
LOMEM   001240          1239# 3482* 3489* 3490* 3491  3492  3512  3622* 3623* 3628  3635  3673  5191
                        5192
LRETRY  001316          1264# 3578
MAXCYL  001210          1230# 2284* 2287* 2648  2664  3632  5462* 5465* 5543  6759
MAXMEM  001230          1237# 3180  3232  3483  3484  5370* 5371* 5372* 5373* 5374* 5375* 5376* 5377*
                        5378* 5379* 5380* 5381* 5382* 5383* 5384* 5385* 5389* 5390* 5391
MAXWC   001264          1247# 2265  2266  5386* 5391* 5392* 5394* 5397* 5399* 5504  5517
MEMSIZ  001302          1254# 3488  3630  3676  5369* 5388*
MEX0  = 000020          1169# 2921  3220  3239  3247
MEX1  = 000040          1170# 2921  3224  3251
MMACTV  001222          1235# 2280* 3480* 3486* 5200  5235  5290  5308  5319
MMVEC = 000250          1099#
MODE  = 010000          1174# 2769  2822  2869  2991  3003  3326
MOVLDR  023300          3481  3621  4949#
MRPADR  032106          5412  5923#
MSG15A  027357          3620  5672#
MSG15B  027402          3643  5676#
MSG15C  027543          3684  5693#
```

```
MSG15D  027576          3653    5698#
MSG16A  027667          3728    5709#
MSG16B  030201          3734    5745#
MSG17A  030243          3779    5752#
MSG17B  030306          3784    5758#
MSG17C  030365          3785    5766#
MSG17D  030410          3800    5770#
MSG17E  030432          3818    5773#
MSG17F  030466          3833    5778#
MSG17G  030523          3852    5783#
MSG17H  030640          3871    5797#
MSG17I  030740          3873    3910    3930    5808#
MSG17J  030763          3905    5812#
MSG17K  031122          3946    5829#
MXWRDS  031373          5503    5862#
NONE    027316          2337    5664#
OFFLIN  027236          2313    5481    5653#
ONEPAT  031637          5534    5892#
ONLINE  027206          2304    5649#
OPRSEL  001260          1245#   2261#   3515    3555    5076    5566#
PAKSIZ  001244          1240#   3517#   3518#   3557#   3558#   5048#   5049#   5054#   5055#   5078    5080    5082    5084
                        5086#   5089#   5090#
PATNUM  001204          1228#   2462#   2486    2506    2508#   2544#   2631#   2645#   3514#   3554#   5062#   5063    5065#
                        5066    5157    5262    6747    6789    6793
PATSEL  024002          3519    3560    5061#
PATTRN  001300          1253#   2269#   5068    5529#
PAT.PT  001440          1332#   5161    5265
PAT0    001500          1332    1351#
PAT1    001540          1333    1368#
PAT10   002200          1342    1521#
PAT11   002240          1343    1538#
PAT12   002300          1344    1555#
PAT13   002340          1345    1572#
PAT14   002400          1346    1589#
PAT15   002440          1347    1606#
PAT2    001600          1334    1385#
PAT3    001640          1335    1402#
PAT4    001700          1336    1419#
PAT5    001740          1337    1436#
PAT6    002000          1338    1453#
PAT7    002040          1339    1470#
PAT8    002100          1340    1487#
PAT9    002140          1341    1504#
PC      =%000007        1014#   2248#   2258#   2259#   2282#   2283#   2321#   2381#   2383#   2384#   2412#   2418#   2421#
                        2426#   2432#   2463#   2466#   2468#   2469#   2470#   2475#   2477#   2478#   2479#   2483#   2495#
                        2496#   2497#   2498#   2539#   2545#   2547#   2548#   2549#   2555#   2556#   2559#   2560#   2561#
                        2567#   2570#   2571#   2572#   2628#   2632#   2636#   2637#   2638#   2644#   2646#   2651#   2652#
                        2653#   2668#   2673#   2674#   2675#   2711#   2717#   2719#   2721#   2762#   2771#   2772#   2773#
                        2779#   2781#   2782#   2799#   2801#   2802#   2803#   2818#   2824#   2825#   2826#   2856#   2860#
                        2861#   2864#   2866#   2871#   2872#   2877#   2878#   2917#   2924#   2925#   2926#   2985#   2990#
                        2993#   2994#   2995#   3002#   3006#   3007#   3008#   3010#   3039#   3041#   3043#   3044#   3045#
                        3058#   3059#   3060#   3061#   3107#   3109#   3110#   3111#   3112#   3122#   3124#   3125#   3126#
                        3185#   3193#   3194#   3195#   3201#   3214#   3215#   3216#   3236#   3241#   3242#   3243#   3321#
                        3328#   3329#   3330#   3336#   3347#   3348#   3349#   3372#   3374#   3385#   3392#   3393#   3394#
                        3481#   3507#   3519#   3521#   3526#   3530#   3531#   3532#   3538#   3540#   3541#   3542#   3546#
                        3548#   3560#   3561#   3563#   3567#   3568#   3569#   3585#   3588#   3589#   3592#   3621#   3625#
```

```
                    3637#  3638#  3639#  3644#  3646#  3647#  3652#  3656#  3658#  3665#  3686#  3701#  3702#
                    3750#  3751#  3752#  3793#  3795#  3808#  3810#  3826#  3828#  3841#  3843#  3856#  3862#
                    3884#  3892#  3918#  3938#  3983#  3986#  3997#  4002   4039#  4128#  4168#  4175#  4182#
                    4196#  4198#  4372#  4386#  4447#  4492#  4733   4780#  4846#  4850#  4869#  4877#  4890#
                    4896#  4697#  4908#  4924#  4945#  4969#  4992#  5008#  5017#  5034#  5043#  5057#  5071#
                    5108#  5123#  5138#  5151#  5170#  5194#  5270#  5276#  5293#  5312#  5317#  5333#  5335#
                    5344#  5356#  5361#  5400#  5430#  5461#  5567#
PIRQ   = 177772     1000#
PIRQVE= 000240      1094#
PR0    = 000000     1017#
PR1    = 000040     1018#
PR2    = 000100     1019#
PR3    = 000140     1020#
PR4    = 000200     1021#
PR5    = 000240     1022#
PR6    = 000300     1023#
PR7    = 000340     1024#  3627
PS     = 177776      997#   998   2281#  3699#
PSW    = 177776      998#  2271#
PWRVEC= 000024      1089#
ROCHR  = 104410     4547   4824#
RDERR    001274     1251#
RDLIN  = 104411     3735   3791   3806   3824   3839   3853   3881   3916   3936   4825#  5419   5445   5467
                    5500   5511   5523   5531   5537   5547   5557
RDSEK  = 000005     1150#  2417   2569   2672   2716   2720   2780   2923   3004   3049   3057   3123   3213
                    3240   3346   3566   3645
RDSW     016046     3812   3818#
RDY    = 000200     1172#  2419   2427   2873   5022   5029
READ   = 000017     1155#  2870
RESLDR   023404     2258   3592   3686   4973#
RESREG= 104413      4127   4827#  4849   4889   4991   5122   5137   5150   5169   5289
RESTAR   004752     2366#  4003
RESVEC= 000010      1084#
RETRY    001314     1263#  3559#  3572   3577#  3578   3582   6800
RPADR    001324     1270#  2380   2407   2454   2533   2622   2706   2756   2851   2912   2979   3097   3173
                    3303   3472   3617   3703   3724   3776   4842   4933   5413   5425#  5428   6741
RPBA   = 000010     1137#  2868#  4881#  4938
RPCA   = 000012     1138#  3855#  3861#  3883#  3890#  3908#  4882#  4939
RPCS   = 000004     1135#  2419   2424#  2425#  2720#  2870#  3664#  3794#  3809#  3827#  3842#  3909#  3943#
                    4843#  4844#  4854#  4874#  4884#  4885#  4888#  4936   5022   5029   5354#  5355#
RPDA   = 000014     1139#  3011   4883#  4940
RPDS   = 000000     1133#  4855   4859   4861   4863   4866   4934   4996   5003
RPER   = 000002     1134#  4935
RPINIT   022536     2283   3652   4833#  5461
RPM1   = 000016     1140#  4941
RPM2   = 000020     1141#
RPM3   = 000022     1142#
RPPRIO   001332     1272#  4836
RPVEC    001326     1271#  4834
RPWC   = 000006     1136#  2867#  4880#  4937
RP02     027253     2309   5656#
RP03     027262     2311   5658#
RP11     022740     2383   2418   2468   2477   2496   2547   2559   2570   2636   2651   2673   2717   2771
                    2781   2801   2824   2860   2864   2924   2993   3006   3043   3059   3110   3124   3193
                    3214   3241   3328   3347   3392   3530   3540   3567   3637   3646   3751   4873#
RSTRT1   004730     2362#  2371
```

```
RSTRT2  004742          2364#  2368
RSTRT3  004750          2365#
R0     =%000000         1005#  2324*  2326   2328   2329*  2334*  2350   2358*  2584*  2586   2591   2593   2788*
                        2789   2791*  2792   2795   3628*  3629*  3630   3673*  3674   3676   3680   3681   3737*
                        3870*  3874*  3875   3880*  3883   3898*  3899   3994*  3997   4074*  4075*  4076*  4077*
                        4078*  4079*  4080*  4081   4086   4091   4094*  4095   4098   4100   4123   4151   4152*
                        4153   4156*  4292   4302*  4306   4322   4323   4336*  4678   4703*  4722   4728*  4771
                        4779*  4792   4793*  4794   4795*  4796*  4797*  4798*  4919   5047   5050*  5051*  5052
                        5056*  5061   5066*  5067*  5068   5070*  5157*  5158*  5161   5421*  5447*  5469*  5513*
                        5525*  5539*  5549*  5559*  5585   5586*  5598   5600*  5601*  5615*  5617*  5628*  5633*
                        5638*  5640*  5641*
R1     =%000001         1006#  2293*  2296   2301   2307   2315*  2325*  2330*  2335*  2345*  2347*  2355   2484*
                        2488*  2489   2585*  2588*  3308*  3310*  3311*  3312*  3315*  3316*  3317*  3337*  3338*
                        3340*  3354*  3362   3364   3657*  3659   3736*  4091*  4107   4111   4293   4306*  4307
                        4311   4335*  4679   4702*  4723   4732*  4764*  4766   4767   4769*  4774   4778*  4834*
                        4835*  4836*  4845*  4847*  4854   4857*  4865*  4868   4875*  4876*  4898   4905   4952*
                        4955*  4962   4965*  4979*  4982*  4987   5114*  5116*  5117*  5118*  5133*  5134*  5143*
                        5146*  5147*  5159*  5163*  5234*  5237*  5238*  5253   5266   5328   5331   5420*  5426*
                        5429   5431   5446*  5468*  5486*  5491   5493   5494*  5495   5512*  5524*  5538*  5548*
                        5558*  5578   5580   5582   5596   5599*  5620*
R2     =%000002         1007#  2294*  2316*  2485*  2489*  2490   2492*  3309*  3313*  3318*  3339*  3341*  3355*
                        3380*  3741   3743   3744   4095*  4117*  4294   4305*  4309*  4312   4319*  4320*  4321
                        4326*  4334*  4680   4701*  4724   4740*  4742*  4743*  4746*  4747*  4755*  4756   4758*
                        4763*  4765*  4770*  4775   4777*  4951*  4954*  4966*  4978*  4981*  4988*  5132*  5135*
                        5144*  5148*  5160*  5164*  5258*  5260*  5261   5271*  5425   5451   5453*  5454   5455*
                        5456   5473   5475   5476   5479   5517   5519   5521   5529   5543   5545   5553   5555
                        5563   5565   5582*  5583*  5627   5630*
R3     =%000003         1008#  3356*  3357*  3358*  3362   3365   3366   4098*  4113*  4234   4243*  4249*  4250*
                        4253*  4258*  4259*  4260   4269*  4295   4303*  4304*  4318*  4321*  4330*  4331*  4333*
                        4542   4544*  4545   4548*  4549   4556*  4557   4559   4567   4571   4573*  4579   4581
                        4583*  4586*  4681   4700*  4725   4741*  4745*  4776*  4956*  4959*  4965   4983*  4986*
                        4987*  5115*  5118   5119*  5120   5128*  5129*  5131*  5134   5145*  5161*  5163   5214*
                        5217*  5264*  5273*  5613   5614*  5621*  5623*  5625*  5627*  5630   5631*  5636   5637
                        5638   5639*
R4     =%000004         1009#  2380*  2407*  2419   2424*  2425*  2454*  2533*  2622*  2706*  2720*  2756*  2851*
                        2867*  2868*  2870*  2912*  2979*  3011   3097*  3173*  3303*  3472*  3617*  3664*  3703*
                        3724*  3776*  3794*  3809*  3827*  3842*  3855*  3861*  3883*  3890*  3908*  3909*  3943*
                        4100*  4105*  4235   4237*  4238*  4239*  4240   4241*  4255   4257*  4265*  4268*  4682
                        4699*  4842*  4843*  4844*  4854*  4855   4859   4861   4863   4866   4874*  4880*  4881*
                        4882*  4883*  4884*  4885*  4888*  4932   4933*  4934   4935   4936   4937   4938   4939
                        4940   4941   4942   4943   4944*  4996   5003   5022   5029   5162*  5166*  5262*  5263*
                        5265   5354*  5355*
R5     =%000005         1010#  4093*  4097*  4102   4120*  4236   4242*  4244*  4246*  4247*  4248*  4249   4267*
                        4296   4298*  4300*  4307*  4311*  4326   4332*  4683   4698*  5265*  5266   5330
R6     =%000006         1011#  1013   2211*  2212*  2213
R7     =%000007         1012#  1014
SAVREG= 104412          4069   4826#  4833   4873   4973   5113   5127   5142   5156   5199
SAVRP   023172          2421   2426   2470   2479   2498   2549   2561   2572   2638   2653   2675   2721   2773
                        2803   2826   2872   2878   2926   2935   3008   3045   3061   3112   3126   3195   3216
                        3243   3330   3349   3394   3532   3542   3569   3639   3665   3795   3810   3828   3843
                        3856   3862   3884   3892   3918   3938   4897   4932#  5008   5034
SCYL    001266          1248#  2262*  3511   3552   5545*
SECNBR  023734          3507   3548   5047#
SEEK   = 000011         1152#  2863   3749
SETBUF  024364          2463   2545   2556   2632   2646   3526   5156#
SILO   = 000026         1144#  4943
```

```
SIZMEM  025624          2259    5360#   5368
SLASH   031246          5418    5845#
SP     =X000006         1013#   2215*   2231*   2239*   2243    2296*   2350*   2374*   2411*   2458*   2537*   2626*   2710*
                        2760*   2855*   2916*   2983*   3105*   3183*   3307*   3475*   3482    3497*   3500*   3504*   3505
                        3506*   3522*   3523*   3524    3622    3624*   3698*   3727*   3736    3780*   3786*   3801*   3819*
                        3834*   3875*   3911*   3931*   3968*   3976*   4034    4048*   4051*   4107*   4111*   4151*   4152
                        4153*   4155    4156    4157*   4159    4161    4163    4169    4171*   4173*   4181*   4185    4189
                        4190    4194    4226*   4227    4228    4229*   4234*   4235*   4236*   4242    4267    4268    4269
                        4270*   4271*   4292*   4293*   4294*   4295*   4296*   4297*   4298    4301*   4314    4316*   4318
                        4328    4330    4332    4333    4334    4335    4336    4338*   4339*   4381*   4382*   4383    4387
                        4389    4398    4400    4403    4406*   4407*   4408    4413    4415    4417*   4418    4434*   4435*
                        4436    4446    4453*   4456*   4457*   4461*   4462*   4464    4467*   4474    4477*   4481    4483
                        4485    4486*   4493    4495    4497*   4498    4500*   4501*   4502*   4503*   4504*   4519*   4520*
                        4521*   4522*   4523*   4529*   4542*   4543*   4548    4551    4555*   4562    4566*   4585    4586
                        4587*   4588*   4589*   4623*   4626    4628    4629    4651    4652    4656*   4678*   4679*   4680*
                        4681*   4682*   4683*   4684*   4685*   4686*   4687*   4694*   4695*   4696*   4697*   4698    4699
                        4700    4701    4702    4703    4722*   4723*   4724*   4725*   4726*   4727*   4728    4731    4739*
                        4743    4744*   4751    4771*   4772    4773    4776    4777    4778    4779    4792*   4793    4803*
                        4804*   4878*   4879*   4880    4886*   4887*   4888    4914*   4915*   4916*   4918*   4920*   4923
                        4932*   4944    4960*   4961*   4962*   4998*   4999*   5000*   5002*   5005*   5012*   5016*   5024*
                        5025*   5026*   5028*   5031*   5038*   5042*   5047*   5056    5061*   5070    5091*   5092*   5093*
                        5094    5096*   5098    5099*   5100*   5101    5103*   5105    5107*   5175*   5176*   5177*   5178*
                        5179*   5180*   5181    5182*   5183*   5184    5206*   5207*   5208*   5209*   5210*   5211*   5212*
                        5213*   5215*   5216*   5219*   5220    5221*   5222    5231*   5232    5239*   5243*   5413*   5420
                        5432    5446    5456*   5459*   5468    5487    5489    5497    5501    5504*   5512    5524    5532
                        5538    5548    5558    5613*   5639
SR0    = 177572         1103#   3210*   3260*   4737    4752*   4759*   5245*   5292*   5314*   5321*
SR1    = 177574         1104#
SR2    = 177576         1105#
SR3    = 172516         1106#   4749*   5241*
SSEC    001272          1250#   2264*   3509    3550    5565*
STACK  = 001100          988#   2215    2411    2458    2537    2626    2710    2760    2855    2916    2983    3105    3183
                        3307    3475    3506    3624    3698    3727    3780
STALL   023134          2719    2866    3010    3656    3701    4914#
STALLT  001220          1234#   2718*   2865*   3009*   3655*   3700*   4914
START   003710          2203    2208#
START1  003654           977    2201#
START2  003676           979    2206#   3687    3947    4388    4471
START3  003644           981    2199#
START4  003666           983    2204#
START5  004136          2246    2248#
START6  004350          2281#
START7  004534          2291    2319#
STKLMT= 177774           999#
STRK    001270          1249#   2263*   3510    3551    5555*
STRT1A  003660          2200    2202#
STRT2A  003702          2205    2207#
SUCA   = 000024         1143#   4942
SUFU   = 001000         1161#   4855
SUOL   = 040000         1164#   3796    3811    4859
SURDY  = 100000         1165#   4863    4996    5003
SURP03= 020000          1163#   4866
SUSI   = 004000         1162#   4861
SUWP   = 000400         1160#   3829    3844    3857    3863    3885    3893    3919    3939
SWR     001140          1207#   2213    2233*   2235    2241*   2252    2290    3360    3378    3476    3522    3527    3536
                        3564    3574    3586    3730    3732    4030    4037    4042    4046    4070    4391    4430    4485*
```

```
                    4618    4636    4643    4903    5013    5039    5268    5326
SWREG   000176       960#   2241    2252    4391    4430    4453
SWD     = 000001    1052#
SW00    = 000001    1042#   1052    3524    3527
SW01    = 000002    1041#   1051    3524    3536
SW02    = 000004    1040#   1050    3564
SW03    = 000010    1039#   1049    3360    3586    5268
SW04    = 000020    1038#   1048    3378    5326
SW05    = 000040    1037#   1047    3574
SW06    = 000100    1036#   1046    3476
SW07    = 000200    1035#   1045    2290    3730    3732
SW08    = 000400    1034#   1044
SW09    = 001000    1033#   1043    4903    5013    5039
SW1     = 000002    1051#
SW10    = 002000    1032#
SW11    = 004000    1031#
SW12    = 010000    1030#
SW13    = 020000    1029#   4070
SW14    = 040000    1028#
SW15    = 100000    1027#
SW2     = 000004    1050#
SW3     = 000010    1049#
SW4     = 000020    1048#
SW5     = 000040    1047#
SW6     = 000100    1046#
SW7     = 000200    1045#
SW8     = 000400    1044#
SW9     = 001000    1043#
TBITVE= 000014    1085#
TEST0   005150    2401    2411#
TEST1   005374    2448    2458#
TEST10  010676    2973    2983#   2984
TEST11  011642    3091    3105#
TEST12  012172    3167    3181    3183#   3184
TEST13  013004    3297    3307#
TEST14  014056    3466    3506#
TEST15  014726    3611    3624#
TEST16  015450    3718    3727#
TEST17  015706    3770    3780#
TEST2   006010    2527    2537#   2538
TEST3   006466    2616    2626#   2627
TEST4   007144    2700    2710#
TEST5   007374    2750    2760#   2761
TEST6   010120    2845    2855#
TEST7   010454    2906    2916#
TIMOUT  001212    1231#   4998    5024
TITLE   043602    2247    6895#
TKVEC = 000060    1092#   4368*   4369*
TPVEC = 000064    1093#
TRAPVE= 000034    1091#   2221*   2222*
TRTVEC= 000014    1086#
TSTING  027325    2373    5666#
TSTNMS  001276    1252#   2270*   2398    2445    2524    2613    2697    2747    2842    2903    2970    3088    3164
                  3294    3463    3608    3715    3767    5454*   5458*
TST0    005054    2385    2397#   2402
TST1    005300    2400    2444#   2449
```

```
TST10   010602          2905    2969#   2974
TST11   011522          2972    3087#   3092
TST12   012052          3090    3163#   3168
TST12B  012524          3228    3232#
TST13   012710          3166    3179    3182    3293#   3298
TST13A  013464          3334    3353    3361    3379    3382    3384#
TST14   013564          3296    3462#   3467
TST14A  014332          3525    3548#
TST15   014620          3465    3607#   3612
TST15A  015102          3652#   3704
TST15B  015304          3626    3691#   3692
TST15C  015314          3691    3696#
TST16   015362          3610    3714#   3719
TST17   015614          3717    3766#   3771
TST2    005714          2447    2523#   2528
TST3    006372          2526    2612#   2617
TST4    007050          2615    2696#   2701
TST5    007300          2699    2746#   2751
TST5A   007624          2794    2798#
TST5B   007732          2777    2807    2815    2817#
TST6    010024          2749    2841#   2846
TST7    010360          2844    2902#   2907
TYPDS = 104405          4112    4819#
TYPE  = 104401          2247    2292    2304    2306    2309    2311    2313    2314    2318    2332    2337    2357    2373
                        2379    3620    3643    3653    3684    3728    3734    3779    3784    3785    3800    3818    3833
                        3852    3871    3873    3905    3910    3930    3946    3958    3964    3972    3989    3993    4032
                        4040    4083    4084    4088    4089    4101    4104    4115    4119    4122    4124    4164    4261
                        4337    4385    4397    4451    4452    4455    4466    4476    4487    4506    4554    4560    4565
                        4569    4574    4575    4577    4580    4584    4815#   5411    5412    5418    5444    5466    5478
                        5481    5483    5484    5499    5503    5509    5510    5522    5530    5534    5536    5546    5556
TYPERR  017232          3372    3374    3585    4039    4069#   5312    5317    5333
TYPOC = 104402          3977    4108    4454    4816#
TYPON = 104404          4818#
TYPOS = 104403          2298    2352    2376    3788    3803    3821    3836    3877    3913    3933    3969    4817#   5415
                        5506
UNSAFE  027222          2306    5483    5651#
WC14    001256          1244#   2265*   2268*   3493    3513    3553    5088    5521#
WRITE = 000013          1153#
WRSW    016176          3845    3852#
WRSW1   016260          3864    3870#
WRSW2   016402          3904#
WRTCHK= 000007          1151#   2476    3539
WRTSEK= 000003          1149#   2467    2546    2558    2635    2770    2800    2823    2992    3042    3108    3192    3327
                        3388    3529    3636
$AUTOB  001134          1204#   2256*   4444    4601
$BOADR  001122          1199#   2593*   2594*   3031*   3032*   3035    3366*   3367*   3681*   3682*   5328*   5329*   6755
                        6770    6792    6798
$BODAT  001126          1201#   2582*   2591*   2680*   2681    2795*   2894*   2885    2888*   2889    3025*   3026*   3027*
                        3028*   3029*   3030*   3033    3035    3230*   3258*   3365*   3680*   5331*   6755    6761    6770
                        6774    6792    6798
$BELL   001166          1220#   4032    4058    4397    4594
$BUF    002506          1630#   2413*   2464*   2540*   2630*   2714*   2922*   2986*   3104*   3188*   3212*   3238*   3325*
                        3343*   3344*   3345*   3391*   3512*   3635*   4881    5114    5133    5159    5174*   5208    5234
$CHARC  017676          4166*   4176*   4183    4192*   4197#
$CKSWR  020662          4430#   4823
$CMTAG  001100          1187#   2210    2211    2219    2223    2224    2225
```

```
$CM3  = 000000  1217#
$CM4  = 000001  1217#  1218#
$CNTLC  021562  4385  4466  4594#
$CNTLG  021574  4451  4596#
$CNTLU  021567  4476  4569  4595#
$CORE  022352  4735  4762#
$CRLF  001173  1222#  2314  2318  2379  4040  4058  4083  4088  4101  4119  4165  4200  4487
               4574  4594  5411  5484  5509
$CROUT  022402  4762  4769#
$CYL  002510  1631#  2415#  2460#  2543#  2634#  2648#  2671#  2713#  2768#  2858#  2862#  2920#  2988#
              3102#  3191#  3323#  3511#  3552#  3632#  3746#  3748#  4882  5104#  5248
$DBLK  020344  4303  4337  4345#
$DOAGN  017050  3985  3995  4001#
$DTBL  020334  4306  4341#
$ENDAD  017040  968  3997#  4053
$ENDCT  016774  2223  2333#  2349#  2360  3987#
$ENULL  017054  3993  4004#
$EOP  016606  2275  2338  3593  3769  3957#  4906  5439
$EOPCT  016766  2223#  2360#  3984#  3988  5438#
$ERFLG  001103  1190#  4027#  4058  4608  4632  4634  4640#  4659
$ERMAX  001115  1196#  2226#  2277#  4634  4654#  4659
$ERROR  017060  2219  4025#
$ERRPC  001116  1197#  3370#  4034#  4035#  4036  4058  5310#  5315#  6742  6747  6753  6755  6757
                6759  6761  6764  6770  6774  6776  6780  6783  6786  6789  6793
$ERRTB  002514  1649#  4080
$ERTTL  001112  1194#  2260#  3976  3978#  4033#  4058
$ESCAP  001164  1219#  2225#  2274#  3668#  3672#  4049  4051  4058  4653#  4895#  4907#  5009#  5011#
                5035#  5037#  5341#  5343#  5433#  5435#
$FILLC  001156  1215#  4169  4200
$FILLS  001155  1214#  4200
$GDADR  001120  1198#  3066#  3067#  6774
$GDDAT  001124  1200#  2581#  2592#  2683#  2787#  2789  2883#  2885  2887#  2889  3016#  3017#  3018
                3021#  3022  3024#  3033  3056#  3068  3074#  3119#  3131  3137#  3229#  3257#  3364#
                3679#  5330#  6753  6755  6761  6764  6770  6774  6776  6792  6798
$GET42  017030  3994#
$GTSWR  020752  4452#  4821
$HD  = 000000  931
$ICNT  001104  1191#  2276#  4645#  4646  4648#  4658
$INTAG  001135  1205#  4449#  4468  4488  4601
$ITEMB  001114  1195#  3371#  3373#  3584#  4036#  4058  4075  5311#  5316#  5332#
$KTNEX  022344  4736  4761#
$KTOUT  022334  4753  4758#
$KT11  022202  3177  3478  4734#  4738#  4761#  4957  4984  5360#  5363  5367#
$LF  001174  1223#  4058  4200  4584  4594
$LPADR  001106  1192#  2227#  2278#  2402#  2449#  2528#  2617#  2701#  2751#  2846#  2907#  2974#  3092#
                3168#  3298#  3467#  3612#  3719#  3771#  4638#  4651#  4656  4658
$LPERR  001110  1193#  2228#  2279#  2401#  2448#  2465#  2474#  2494#  2527#  2538#  2554#  2566#  2616#
                2627#  2643#  2667#  2700#  2750#  2761#  2778#  2798#  2817#  2845#  2906#  2973#  2984#
                3001#  3038#  3053#  3091#  3106#  3121#  3167#  3184#  3200#  3235#  3297#  3320#  3335#
                3384#  3466#  3520#  3562#  3611#  3654#  3718#  3770#  3792#  3807#  3825#  3840#  3854#
                3882#  3891#  3917#  3937#  4048  4638  4652#  4658
$LSTAD  022446  4774#  4781#  4959  4986  5384  5388  5389
$LSTBK  022450  4775#  4782#  5365  5370
$MAIL  = ****** U  2245  2252  4042  4153  4651
$MNEW  021612  4455  4599#
$MSWR  021601  4452  4597#
```

```
$MXCNT  022046          4649    4658#
$NULL   001154          1213#   4171    4200
$NWTST= 000001          2390#   2392    2436#   2438    2513#   2515    2601#   2603    2689#   2691    2737#   2739    2833#
                        2835    2895#   2897    2940#   2942    3079#   3081    3142#   3144    3264#   3266    3400#   3402
                        3595#   3597    3707#   3709    3758#   3760
$OCNT   020124          4233*   4262*   4275#
$OMODE  020126          4228*   4232*   4237    4240*   4251*   4277#
$OVER   022032          4619    4639    4647    4655#
$PASS   001100          1188#   3981*   3982*   4004
$QUES   001172          1221#   4058    4200    4506    4577    4594
$RDCHR  021224          4519#   4824
$RDDEC= ****** U        4826
$RDLIN  021314          4542#   4825
$RDOCT= ****** U        4826
$RDSZ = 000012          4535#
$RESRE  022106          4693#   4827
$RPBA   001344          1280#   4938*   6745    6750    6766    6780
$RPCA   001346          1281#   2661    4939*   6745    6750    6766    6778    6786
$RPCS   001340          1278#   2427    2471    2480    2503    2550    2562    2573    2639    2658    2676    2728    2731
                        2774    2783    2811    2814    2827    2873    2879    2931    2934    2996    3012    3046    3062
                        3113    3127    3196    3217    3220    3224    3244    3247    3251    3331    3350    3395    3533
                        3543    3570    3640    4936*   6742    6747    6757    6766    6772    6778    6780    6783    6786
$RPDA   001350          1282#   3011*   3025    3031    4940*   6745    6750    6766    6772    6778
$RPDS   001334          1276#   2804    3796    3811    3829    3844    3857    3863    3885    3893    3919    3939    4934*
                        6742    6747    6757    6765    6772    6778    6780    6783    6786
$RPER   001336          1277#   2499    2654    2722    2808    2927    4935*   6742    6747    6757    6766    6772    6778
                        6780    6783    6786
$RPMI   001352          1283#   4941*   6745    6750
$RPWC   001342          1279#   4937*   5246    6742    6750    6766    6780
$RTNAD  017052          4003#
$R2A  = ****** U        4828
$SAVRE  022050          4677#   4826
$SCOPE  021624          2217    4616#
$SEC    002512          1632#   2416*   2459*   2542*   2633*   2649*   2670*   2712*   2767*   2857*   2919*   2989*   3000*
                        3017    3050*   3051    3055*   3067    3071*   3072    3074    3101*   3120*   3190*   3322*   3509*
                        3550*   3633*   4883    5092    5098*   5129    5250
$SETUP= 000127          2191#   2216    2217    2219    2221    2223    2224    2225    2227    2249    3979    4026    4045
                        4053    4389    4394    4395    4425    4601    4617
$SILO   001356          1285#   4943*   6745    6750
$SIZE   022144          4722#   5361
$SIZEX  022406          4760    4770#
$STUP = 177777          2191#
$SUCA   001354          1284#   2664    3666    4942*   6745    6750    6759    6783
$SVLAD  022004          4627    4650#
$SVPC = 000200           966#    971
$SWR  = 167000           921#    931     935     936     937     938     939     940     941    1218    1219    1220    2224
                        2225    2227    2228    2410    2457    2536    2625    2709    2759    2854    2915    2982    3100
                        3176    3306    3475    3620    3727    3779    3954    3980    3996    4002    4004    4017    4018
                        4019    4020    4021    4030    4037    4042    4046    4058    4609    4610    4611    4612    4613
                        4618    4630    4632    4633    4634    4641    4642    4643    4652    4655    4658
$SWRMK= 000000          4613
$TIMES  001162          1210#   2224*   2273#   2410#   2457#   2536#   2625#   2709#   2759#   2854#   2915#   2982#   3100#
                        3176*   3306#   3980#   4641#   4646    4649#   4658
$TKB    001146          1210#   4350    4370    4381    4406    4434    4461
$TKCNT  020354          4351#   4365*   4395    4412*   4526    4528*
$TKINT  020374          2248    2432    3702    4365#   4386    4447
```

```
STKQEN=  020374          4355#  4420   4531
STKQIN   020356          4352#  4366#  4367   4418#  4419#  4420   4422#
STKQOU   020360          4353#  4367#  4529   4530#  4531   4533#
STKQSR   020362          4354#  4366   4422   4533
STKS     001144          1209#  4350   4371#  4402#  4404   4410#  4432   4448#  4458   4470#  4490#
STXSRV   020444          4360   4381#
STNPO    001160          1217#  4072#  4073#  6742   6747   6753   6755   6757   6759   6761   6764   6770   6774
                         6776   6780   6783   6786   6789   6793
STN    = 000020           921#   931   2390   2397   2398   2410#  2436   2444   2445   2457#  2513   2523   2524
                         2536#  2601   2612   2613   2625#  2689   2696   2697   2709#  2737   2746   2747   2759#
                         2833   2841   2842   2854#  2895   2902   2903   2915#  2940   2969   2970   2982#  3079
                         3087   3088   3100#  3142   3163   3164   3176#  3264   3293   3294   3306#  3400   3462
                         3463   3475#  3595   3607   3608   3620#  3707   3714   3715   3727#  3758   3766   3767
                         3779#  5451
STPB     001152          1212#  4189#  4200
STPFLG   001157          1216#  4147   4200
STPS     001150          1211#  4187   4200
STRAP    022452          2221   4792#
STRAP2   022474          4803#  4814
STRK     002513          1633#  2650#  3116#  3117   3134   3136#  3510#  3551#  3743#  5097#  5099   5105#  5146
                         5249
STRP   = 000014          4807#  4816#  4817#  4818#  4819#  4820#  4821   4822#  4823   4824#  4825#  4826#  4827#
                         4828#
STRPAD   022506          4797   4814#
STSTNM   001102          1189#  2403#  2405   2450#  2452   2529#  2531   2618#  2620   2702#  2704   2752#  2754
                         2847#  2849   2908#  2910   2975#  2977   3093#  3095   3169#  3171   3299#  3301   3468#
                         3470   3613#  3615   3720#  3722   3772#  3774   3979#  4029   4058   4073   4608   4650#
                         4655   4659
STTYIN   021550          4544   4545   4557   4575   4589   4593#
STYPBN=  ******  U       4820
STYPDS   020130          4291#  4819
STYPE    017462          4147#  4807   4815
STYPEC   017632          4168   4175   4182   4187#  4188   4492
STYPEX   017700          4193   4195   4198#
STYPOC   017726          4231#  4816
STYPON   017742          4230   4233#  4818
STYPOS   017702          4226#  4817
SWC      002504          1629#  2414#  2461#  2541#  2557#  2568#  2629#  2647#  2669#  2715#  2766#  2918#  2987#
                         3005#  3054#  3103#  3189#  3324#  3338   3339   3343   3344   3355   3356   3357   3387#
                         3513#  3553#  3634#  4878   5084#  5085#  5088#  5132   5144   5160   5182   5229   5247
                         4621#
SXTSTR   021636          3996#
$$GET4=  000000          4227#  4231#  4241   4276#
$$FILL   020125          4039   4618
$$OCAT=  ******  U        954#   958#   966    967#   969#   971#   975#   1186#  1224   2214   2227   2228   2278
.      = 043706          2279   2399   2446   2525   2614   2698   2748   2843   2904   2971   3023   3089   3165
                         3295   3370   3464   3609   3716   3755   3768   3961#  3967#  4004   4005#  4058   4200
                         4345#  4350   4354#  4355   4356   4593#  4594   4601#  4658   4659   5310   5315   6893
```

```
CKCHR   2191#  5617   5632
CKNUM   2191#  3736   5420   5446   5468   5512   5524   5538   5548   5558
COMMEN     1#  1095#
ENDCOM     1#  1095#
ENDPAS  2191#  3989
ERROR    989#  2422   2429   2473   2482   2501   2505   2552   2564   2575   2583   2595   2641   2656   2660
        2663   2666   2678   2684   2724   2730   2733   2776   2785   2796   2806   2810   2813   2816   2829
        2875   2881   2891   2929   2933   2936   2998   3014   3020   3037   3048   3064   3070   3115   3129
        3133   3198   3219   3222   3226   3231   3246   3249   3253   3259   3333   3352   3397   3535   3545
        3576   3580   3642   3662   3669   3683   3798   3813   3831   3846   3859   3865   3887   3895   3921
        3941   4901   4902   5010   5036   5342   5434
ESCAPE     1#  1095#  3668   3671   4894   5009   5035   5341   5433
GETPRI     1#  1095#  4730
GETSWR     1#   921#  1095#  2249
MORETN  1179#  1225
MORE.S  2191#  2397   2444   2523   2612   2696   2746   2841   2902   2969   3087   3163   3293   3462   3607
        3714   3766
MSG     2389#  2392   2435#  2438   2512#  2515   2600#  2603   2688#  2691   2736#  2739   2832#  2835   2894#
        2897   2939#  2942   3078#  3081   3141#  3144   3263#  3266   3400#  3402   3595#  3597   3706#  3709
        3757#  3760
MULT       1#  1095#
NEWTST     1#  1095#  2390   2436   2513   2601   2689   2737   2833   2895   2940   3079   3142   3264   3400
        3595   3707   3758
POP        1#  1095#  4332   4698
PUSH       1#  1095#  4291   4678
REPORT     1#  1095#
SCOPE    990#  2433   2510   2598   2686   2734   2830   2892   2937   3076   3139   3261   3398   3591   3685
        3945
SETPRI     1#  1095#  4522
SETTRA  4807#  4816   4817   4818   4819   4821   4823   4824   4825   4826   4827
SETUP      1#  1095#  2209
SKIP       1#  1095#
SLASH      1#  1095#
SPACE   1095#
STARS      1#   964   1095#  1182   1224   2192   2196   2390   2396   2436   2443   2513   2522   2601   2611
        2689   2695   2737   2745   2833   2840   2895   2901   2940   2968   3079   3086   3142   3162   3264
        3292   3400   3461   3595   3606   3707   3713   3758   3765   3951   4013   4059   4132   4203   4281
        4349   4425   4440   4511   4535   4605   4662   4708   4786
STIME   2191#  2718   2865   3009   3655   3700
SWRSU      1#  1095#  2229#
TRMTRP  4807#
TYPBIN     1#  1095#
TYPDEC     1#  1095#  4110
TYPEND  2191#  3958
TYPNAM     1#  1095#
TYPNUM     1#  1095#
TYPOCS     1#  1095#  2295   2350   2374   3786   3801   3819   3834   3875   3911   3931   3968   5413   5504
TYPOCT     1#  1095#  3976   4107   4453
TYPTXT     1#  1095#  3958   3964   3972   3989
$$CMRE  1180#
$$CMTN  1180#  1217
$$ESCA     1#  1095#
$$NEWT     1#  1095#  2390   2436   2513   2601   2689   2737   2833   2895   2940   3079   3142   3264   3400
        3595   3707   3758
$$SET   4807#  4816   4817   4818   4819   4821   4823   4824   4825   4826   4827
$$SKIP     1#  1095#
```

```
.EQUAT          1#      921#     985
.HEADE          1#      921#
.KT11           1#      921#    1095
.SETUP          1#      921#    2191
.SWRHI          1#      921#     931
.SWRLO        941#      944      945     946     947     948     949     950
.SYCTI          1#      921#     962
.SYPTB          1#
.SYPTH          1#
.SYPTY          1#
.SYSTA          1#
.SYCTC          1#      921#     952
.SYMTA          1#      921#    1180
.SYB2D          1#
.SYB2O          1#
.SYDIV          1#
.SYEOP          1#      921#    3949
.SYERRO         1#      921#    4011
.SYERRT         1#
.SYMLT          1#
.SYPWE          1#
.SYRAND         1#      921#
.SYRDE          1#
.SYRDC          1#
.SYREAD         1#      921#    4347
.SY2AZ          1#
.SYSAVE         1#      921#    4660
.SYB2D          1#
.SYB2O          1#
.SYSCOP         1#      921#    4603
.SYSIZE         1#      921#    4706
.SYSUPR         1#
.SYTRAP         1#      921#    4784
.SYTYPB         1#
.SYTYPD         1#      921#    4279
.SYTYPE         1#      921#    4130
.SYTYPO         1#      921#    4201
.S40CA          1#
.1170           1#
```

```
ADC     5185  5385
ADD     2492  2791  3338  3344  3345  3357  3358  3504  3890  4080  4157  4229  4239  4311  4467
        4477  4486  4744  4755  4764  4765  5012  5016  5038  5042  5093  5100  5107  5147  5184
        5221  5231  5239  5243  5247  5384  5627  5638
ASL     3880  3898  3927  4077  4078  4079  4500  4501  4502  4796  5054  5067  5158  5177  5178
        5179  5180  5183  5263  5372  5374  5376  5378  5380  5382  5453  5621  5623  5625
ASLB    2330  2367  4316
ASR     2345  2347  3027  3028  3029  3030  3874  5210  5211  5212  5213  5215  5455
BCC     2348  2368  4317
BCS     4106  5622  5624  5626
BEQ     2286  2289  2302  2320  2342  2356  2420  2472  2481  2491  2507  2551  2563  2574  2580
        2640  2662  2665  2677  2682  2775  2784  2828  2874  2880  2890  2997  3013  3019  3036
        3047  3063  3069  3073  3114  3128  3132  3135  3197  3218  3225  3228  3245  3248  3256
        3314  3332  3351  3363  3379  3396  3525  3534  3544  3571  3573  3575  3583  3641  3660
        3667  3731  3797  3845  3864  3894  3900  3920  3963  3995  4028  4031  4047  4050  4082
        4087  4092  4096  4099  4114  4160  4195  4256  4392  4445  4484  4499  4527  4563  4633
        4637  4856  4860  4864  4867  4975  5001  5027  5053  5069  5081  5165  5167  5201  5236
        5267  5272  5291  5320  5323  5327  5340  5409  5437  5464
BGE     4647
BGT     3661  3985  4263  4325  4416  4496  4899  5079  5477
BHI     4635  4757  5064  5187  5257  5366  5518  5520  5544  5554  5564  5581
BHIS    3377  5083
BIC     2369  3026  3032  3982  4253  4382  4407  4417  4435  4462  4497  4761  5176  5219  5237
        5583
BIS     2869  4258  4259  4319  4320  4504  4738  5238
BISB    2328  4868  4885  4887  4905  5493
BIT     2290  2364  2398  2419  2427  2445  2471  2480  2499  2503  2524  2550  2562  2573  2613
        2639  2654  2658  2676  2697  2722  2728  2731  2747  2774  2783  2804  2808  2811  2814
        2827  2842  2873  2879  2903  2927  2931  2934  2970  2996  3012  3046  3062  3088  3113
        3127  3164  3196  3217  3220  3224  3244  3247  3251  3294  3331  3350  3360  3378  3395
        3463  3476  3524  3527  3533  3536  3543  3564  3570  3574  3586  3608  3640  3715  3730
        3732  3767  3796  3811  3829  3844  3857  3863  3885  3893  3899  3919  3928  3939  4030
        4037  4046  4070  4618  4636  4643  4855  4859  4861  4863  4866  4903  4996  5003  5013
        5022  5029  5039  5068  5268  5326
BITB    2307  3744  4183  5052
BLE     2327  4118  5278  5492
BLO     3023  3118  3233  4558  4917  5188  5190  5281  5285  5496  5579
BLOS    2267  3579  3631  3742  4546  5230  5452  5474  5490
BLT     4174  4264  4308  4324  4414  4494  4900  5095  5102
BMI     2303  3178  3501  4315  4958  4985  5480  5488
BNE     2214  2236  2246  2251  2253  2291  2308  2317  2331  2336  2340  2344  2365  2399  2428
        2446  2487  2500  2504  2526  2578  2587  2589  2614  2655  2659  2698  2723  2729  2732
        2748  2790  2793  2805  2809  2812  2815  2843  2886  2904  2928  2932  2935  2971  3034
        3052  3089  3165  3181  3221  3252  3295  3319  3342  3361  3369  3381  3464  3477  3516
        3528  3537  3556  3565  3587  3609  3675  3677  3716  3733  3745  3768  3812  3830  3858
        3886  3924  3929  3940  4038  4054  4071  4103  4121  4154  4162  4170  4184  4191  4254
        4313  4384  4390  4396  4401  4409  4421  4431  4437  4465  4469  4475  4482  4489  4532
        4550  4552  4568  4572  4582  4619  4644  4768  4862  4904  4921  4950  4964  4967  4977
        4989  4997  5004  5006  5014  5023  5030  5032  5040  5077  5121  5130  5136  5149  5218
        5269  5274  5307  5309  5325  5396  5498  5502  5533  5597
BPL     3479  4043  4148  4188  4252  4299  4329  4405  4433  4459  4735  4848  5364
BR      2200  2203  2205  2238  2255  2305  2310  2312  2322  2346  2359  2371  2423  2493  2502
        2508  2553  2565  2576  2590  2642  2657  2679  2727  2777  2794  2807  2876  2930  3015
        3065  3075  3130  3138  3334  3353  3382  3485  3503  3547  3581  3590  3648  3663  3678
        3692  3704  3729  3747  3753  3755  3926  3944  3959  3965  3973  3990  4109  4116  4126
        4150  4167  4177  4186  4193  4230  4245  4266  4310  4327  4399  4478  4505  4507  4561
```

```
        4570  4576  4578  4621  4627  4630  4639  4642  4750  4760  4858  4922  4953  4980  5007
        5033  5087  5106  5168  5193  5242  5259  5275  5313  5368  5387  5398  5457  5482  5485
        5584  5616  5629
CLC     5393
CLR     2201  2202  2206  2212  2224  2225  2260  2261  2262  2263  2264  2271  2272  2273  2274
        2280  2281  2293  2323  2324  2333  2334  2361  2362  2415  2416  2459  2460  2484  2485
        2542  2543  2592  2631  2633  2634  2670  2671  2683  2712  2713  2767  2768  2788  2819
        2820  2821  2857  2858  2887  2919  2920  2988  2989  3016  3024  3066  3101  3102  3119
        3120  3186  3190  3191  3203  3211  3237  3260  3322  3323  3340  3359  3389  3390  3486
        3487  3490  3498  3499  3559  3623  3633  3697  3699  3855  3904  3978  3979  3980  4072
        4074  4093  4243  4302  4305  4365  4402  4448  4456  4457  4521  4522  4543  4566  4641
        4653  4739  4747  4759  4763  4835  4837  4838  4839  4840  4841  4875  4907  4915  4960
        4990  4999  5011  5025  5037  5049  5050  5065  5086  5091  5115  5116  5117  5128  5145
        5202  5206  5207  5254  5255  5282  5286  5292  5314  5343  5367  5371  5390  5410  5435
        5438  5460  5486  5614  5631
CLRB    2408  2409  2455  2456  2534  2535  2623  2624  2707  2708  2757  2758  2852  2853  2913
        2914  2980  2981  3000  3040  3055  3098  3099  3174  3175  3304  3305  3386  3473  3474
        3495  3618  3619  3725  3726  3777  3778  4166  4192  4331  4573  4583  4640
CMP     2213  2235  2252  2266  2486  2490  2506  2577  2579  2664  2789  2792  2885  2889  3018
        3022  3033  3035  3068  3227  3232  3255  3362  3376  3578  3630  3674  3676  4053  4323
        4383  4389  4391  4395  4400  4408  4413  4415  4420  4430  4436  4464  4474  4481  4493
        4495  4531  4545  4557  4628  4646  4751  4756  4767  4923  5063  5082  5120  5186  5189
        5229  5256  5266  5280  5284  5365  5432  5451  5473  5489  5495  5497  5517  5519  5543
        5553  5563
CMPB    2343  3051  3072  3117  3131  3134  3741  4159  4161  4169  4190  4194  4444  4468  4488
        4549  4567  4571  4581  4634  5094  5101  5501  5532  5578  5580
COM     3523  4097  4120
DEC     2316  2588  3313  3318  3341  3380  3983  4076  4113  4117  4528  4556  4847  4920  4966
        4988  5005  5031  5135  5148  5164  5166  5217  5271  5273  5324
DECB    4173  4176  4251  4262
EMT      989
HALT     958  3754  4044  4055  4149
INC     2245  2288  2315  2329  2349  2358  2366  3021  3375  3496  3502  3577  3981  4033  4257
        4265  4309  4412  4419  4503  4530  4645  4752  5062  5104  5119  5252  5279  5283  5287
        5334  5494  5599  5620
INCB    3050  3071  3116  3136  3137  4027  4196  4650  5097
IOT      990
JMP      977   979   981   983  2338  2385  2400  2447  2526  2615  2699  2749  2844  2905  2972
        2999  3090  3166  3179  3182  3199  3296  3465  3593  3610  3687  3717  3769  3947  4002
        4388  4471  4906  5015  5041  5208  5439
JSR     2248  2258  2259  2282  2283  2321  2381  2383  2384  2412  2418  2421  2426  2432  2463
        2466  2468  2469  2470  2475  2477  2478  2479  2483  2495  2496  2497  2498  2539  2545
        2547  2548  2549  2555  2556  2559  2560  2561  2567  2570  2571  2572  2628  2632  2636
        2637  2638  2644  2646  2651  2652  2653  2668  2673  2674  2675  2711  2717  2719  2721
        2762  2771  2772  2773  2779  2781  2782  2799  2801  2802  2803  2818  2824  2825  2826
        2856  2860  2861  2864  2866  2871  2872  2877  2878  2917  2924  2925  2926  2985  2990
        2993  2994  2995  3002  3006  3007  3008  3010  3039  3041  3043  3044  3045  3058  3059
        3060  3061  3107  3109  3110  3111  3112  3122  3124  3125  3126  3185  3193  3194  3195
        3201  3214  3215  3216  3236  3241  3242  3243  3321  3328  3329  3330  3336  3347  3348
        3349  3372  3374  3385  3392  3393  3394  3481  3507  3519  3521  3526  3530  3531  3532
        3538  3540  3541  3542  3546  3548  3560  3561  3563  3567  3568  3569  3585  3588  3589
        3592  3621  3625  3637  3638  3639  3644  3646  3647  3652  3656  3658  3665  3686  3701
        3702  3737  3750  3751  3752  3793  3795  3808  3810  3826  3828  3841  3843  3856  3862
        3884  3892  3918  3938  3997  4039  4168  4175  4182  4386  4447  4492  4846  4877  4896
        4897  5008  5034  5270  5276  5312  5317  5333  5361  5421  5447  5461  5469  5513  5525
        5539  5549  5559  5615  5617  5628  5633
```

DZRPYB.CMB CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

```
MOV     2198  2204  2207  2211  2215  2217  2218  2219  2220  2221  2222  2223  2227  2228  2231
        2232  2233  2234  2239  2241  2242  2243  2265  2268  2269  2270  2275  2276  2278  2279
        2284  2287  2294  2296  2325  2335  2350  2360  2363  2374  2380  2401  2402  2403  2405
        2406  2407  2410  2411  2413  2414  2424  2425  2448  2449  2450  2452  2453  2454  2457
        2458  2461  2462  2464  2465  2467  2474  2476  2488  2489  2494  2508  2527  2528  2529
        2531  2532  2533  2536  2537  2538  2540  2541  2544  2546  2554  2557  2566  2568  2569
        2581  2582  2584  2585  2591  2593  2616  2617  2618  2620  2621  2622  2625  2626  2627
        2629  2630  2643  2645  2647  2648  2649  2667  2669  2672  2680  2700  2701  2702  2704
        2705  2706  2709  2710  2714  2715  2718  2750  2751  2752  2754  2755  2756  2759  2760
        2761  2763  2764  2765  2766  2769  2778  2787  2795  2798  2800  2817  2822  2845  2846
        2847  2849  2850  2851  2854  2855  2862  2865  2867  2868  2883  2884  2888  2906  2907
        2908  2910  2911  2912  2915  2916  2918  2922  2973  2974  2975  2977  2978  2979  2982
        2983  2984  2986  2987  2991  3001  3003  3005  3009  3011  3025  3031  3038  3053  3054
        3056  3091  3092  3093  3095  3096  3097  3100  3103  3104  3105  3106  3121  3167  3168
        3169  3171  3172  3173  3176  3183  3184  3187  3188  3189  3192  3200  3202  3204  3205
        3206  3207  3208  3209  3210  3212  3229  3230  3234  3235  3238  3257  3258  3297  3298
        3299  3301  3302  3303  3306  3307  3308  3309  3310  3311  3312  3315  3316  3317  3320
        3324  3325  3326  3335  3337  3339  3343  3354  3355  3356  3364  3365  3366  3370  3384
        3387  3391  3466  3467  3468  3470  3471  3472  3475  3480  3482  3483  3484  3488  3489
        3491  3492  3493  3497  3505  3506  3511  3512  3513  3514  3520  3522  3552  3553  3554
        3562  3611  3612  3613  3615  3616  3617  3622  3624  3626  3627  3628  3629  3632  3634
        3635  3654  3655  3657  3668  3672  3673  3679  3680  3681  3691  3696  3698  3700  3703
        3718  3719  3720  3722  3723  3724  3727  3736  3746  3748  3770  3771  3772  3774  3775
        3776  3780  3786  3792  3801  3807  3819  3825  3834  3840  3854  3861  3870  3875  3882
        3883  3891  3908  3911  3917  3925  3931  3937  3968  3976  3986  3994  4029  4034  4048
        4051  4081  4086  4091  4094  4095  4107  4111  4123  4151  4152  4156  4171  4226  4234
        4235  4236  4242  4249  4267  4268  4269  4270  4271  4292  4293  4294  4295  4296  4297
        4298  4303  4306  4326  4332  4333  4334  4335  4336  4338  4339  4366  4367  4368  4369
        4371  4410  4422  4453  4470  4485  4490  4519  4520  4523  4533  4542  4544  4555  4586
        4587  4588  4589  4623  4624  4626  4629  4638  4648  4649  4651  4652  4655  4656  4670
        4679  4680  4681  4682  4683  4684  4685  4686  4687  4694  4695  4696  4697  4698  4699
        4700  4701  4702  4703  4722  4723  4724  4725  4726  4727  4728  4731  4732  4736  4740
        4741  4742  4743  4746  4748  4749  4753  4758  4762  4771  4772  4773  4774  4775  4776
        4777  4778  4779  4792  4793  4797  4803  4804  4834  4836  4842  4843  4844  4845  4878
        4880  4881  4882  4883  4886  4895  4914  4918  4932  4933  4934  4935  4936  4937  4938
        4939  4940  4941  4942  4943  4944  4951  4952  4954  4955  4956  4959  4961  4962  4965
        4968  4978  4979  4981  4982  4983  4986  4987  4998  5002  5009  5024  5028  5035  5047
        5048  5056  5061  5066  5070  5084  5088  5114  5118  5131  5132  5133  5134  5143  5144
        5157  5159  5160  5161  5162  5163  5174  5175  5182  5191  5192  5203  5204  5205  5208
        5214  5220  5222  5223  5224  5225  5226  5227  5228  5232  5233  5234  5240  5241  5244
        5245  5246  5248  5251  5253  5258  5260  5262  5264  5265  5310  5315  5318  5321  5328
        5330  5331  5341  5354  5355  5360  5369  5370  5386  5388  5389  5391  5413  5420  5425
        5426  5427  5429  5431  5433  5446  5454  5456  5458  5459  5462  5465  5468  5504  5512
        5521  5524  5529  5538  5545  5548  5555  5558  5565  5566  5600  5613  5630  5639  5640
MOVB    2226  2256  2277  2372  2382  2417  2558  2635  2650  2716  2720  2770  2780  2823  2859
        2863  2870  2921  2923  2992  3004  3017  3042  3049  3057  3067  3074  3108  3123  3213
        3239  3240  3327  3346  3371  3373  3388  3509  3510  3529  3539  3550  3551  3566  3584
        3636  3645  3664  3743  3749  3794  3809  3827  3842  3909  3943  4036  4073  4075  4098
        4100  4153  4181  4189  4227  4228  4231  4232  4233  4237  4240  4241  4260  4301  4304
        4318  4321  4330  4381  4406  4418  4434  4449  4461  4529  4548  4553  4559  4564  4579
        4654  4795  4854  4857  4865  4874  4876  4884  4888  5051  5092  5098  5099  5105  5129
        5146  5181  5209  5249  5250  5311  5316  5332  5475  5582
NEG     4238  4300  4879
NOP     3998  3999  4000
RESET   2208  2431  3996
ROL     4244  4246  4247  4248  4250  5055  5373  5375  5377  5379  5381  5383
```

```
ROR     4105  5216  5394
RTI     2240  4057  4158  4272  4340  4411  4423  4491  4524  4534  4590  4657  4688  4704  4805
RTS     4128  4198  4372  4780  4798  4850  4869  4890  4908  4924  4945  4969  4992  5017  5043
        5057  5071  5108  5123  5138  5151  5170  5194  5293  5335  5344  5356  5400  5430  5567
        5586  5601  5641
SBC     3518  3558  5090
SOB     4745
SUB     2594  3367  3500  3517  3557  3682  4035  4307  4769  4770  4916  5000  5026  5089  5096
        5103  5261  5329  5392  5397  5399
SWAB    3494  5085
TRAP    4730  4807  4816  4817  4818  4819  4821  4823  4824  4825  4826  4827
TST     2250  2285  2319  2339  2341  2355  2586  2661  2681  3177  3180  3368  3478  3515  3555
        3572  3582  3666  3923  3962  4042  4049  4102  4155  4163  4185  4255  4312  4322  4370
        4387  4398  4403  4446  4483  4498  4526  4551  4562  4585  4625  4737  4754  4766  4794
        4919  4957  4963  4974  4984  5076  5078  5080  5200  5235  5277  5290  5306  5308  5319
        5322  5339  5363  5408  5428  5436  5463  5487  5585  5598  5636  5637
TSTB    2301  2326  3659  4147  4187  4314  4328  4404  4432  4458  4632  4733  4898  4949  4976
        5395  5476  5479  5491  5596
.ASCII  1221  1222  5676  5709  5714  5726  5783  5812  5829  5892  5983  5999  6034  6047  6121
        6220  6244  6264  6304  6318  6339  6352  6391  6556  6580  6606  6627  6697  6713  6730
        6895
.ASCIZ  1220  1223  3961  3967  3975  3992  4594  4595  4596  4597  4599  5645  5649  5651  5653
        5656  5658  5660  5664  5665  5666  5670  5672  5685  5693  5698  5738  5745  5752  5758
        5766  5770  5773  5778  5792  5797  5808  5818  5835  5845  5846  5850  5854  5856  5862
        5870  5876  5884  5901  5909  5914  5918  5923  5928  5934  5939  5949  5954  5961  5966
        5973  5992  6008  6014  6019  6024  6029  6042  6056  6061  6066  6073  6081  6088  6098
        6103  6108  6113  6129  6134  6145  6157  6166  6171  6176  6181  6189  6196  6204  6213
        6228  6236  6252  6255  6272  6276  6284  6294  6313  6326  6329  6334  6347  6359  6364
        6370  6376  6384  6400  6404  6411  6419  6428  6438  6442  6445  6448  6452  6456  6464
        6469  6474  6479  6494  6496  6505  6513  6522  6531  6540  6548  6563  6570  6577  6587
        6594  6603  6614  6622  6635  6643  6650  6657  6668  6677  6686  6701  6705  6717  6721
        6725  6733  6899
.BLKB    4354  4593
.BLKW    4345
.BYTE   1189  1190  1195  1196  1204  1205  1213  1214  1215  1216  1289  1290  1291  1292  1293
        1294  1295  1296  1304  1305  1306  1307  1308  1309  1310  1311  1625  1626  1627  1628
        1632  1633  2299  2300  2353  2354  2377  2378  3789  3790  3804  3805  3822  3823  3837
        3838  3878  3879  3914  3915  3934  3935  3970  3971  4004  4273  4274  4275  4276  4591
        4592  5416  5417  5507  5508  6805  6806  6809  6810  6812  6813  6816  6817  6819  6820
        6823  6824  6827  6828  6830  6831  6834  6835  6837  6838  6840  6841  6844  6845  6847
        6848  6851  6852  6854  6855  6858  6859  6862  6863  6866  6867  6870  6871  6873  6874
        6877  6878  6880  6881  6883  6884  6887  6888
.DSABL   4508
.ENABL      1   921  4350
.END    6907
.ENDC    926   938   940   941   961   965   969   971   989  1081  1095  1107  1118  1129  1183
        1187  1189  1217  1218  1219  1220  1221  1225  1635  2191  2193  2197  2215  2216  2219
        2221  2223  2224  2225  2227  2229  2245  2252  2258  2300  2301  2354  2355  2378  2379
        2391  2392  2396  2397  2401  2402  2410  2411  2437  2438  2443  2444  2448  2449  2457
        2458  2514  2515  2522  2523  2527  2528  2536  2537  2602  2603  2611  2612  2616  2617
        2625  2626  2690  2691  2695  2696  2700  2701  2709  2710  2738  2739  2745  2746  2750
        2751  2759  2760  2834  2835  2840  2841  2845  2846  2854  2855  2896  2897  2901  2902
        2906  2907  2915  2916  2941  2942  2968  2969  2973  2974  2982  2983  3080  3081  3086
        3087  3091  3092  3100  3101  3143  3144  3162  3163  3167  3168  3176  3177  3265  3266
        3292  3293  3297  3298  3306  3307  3401  3402  3461  3462  3466  3467  3475  3596  3597
        3606  3607  3611  3612  3620  3669  3673  3708  3709  3713  3714  3718  3719  3727  3759
```

```
            3760  3765  3766  3770  3771  3779  3790  3791  3805  3806  3823  3824  3838  3839  3879
            3880  3915  3916  3935  3936  3952  3954  3956  3961  3967  3971  3972  3975  3979  3985
            3988  3989  3992  3994  3996  4002  4004  4005  4014  4017  4027  4034  4039  4040  4041
            4042  4053  4057  4058  4060  4133  4153  4204  4282  4350  4380  4389  4393  4424  4426
            4441  4472  4508  4512  4523  4535  4536  4544  4546  4549  4577  4594  4595  4601  4606
            4609  4613  4618  4620  4631  4632  4634  4636  4643  4645  4650  4651  4655  4658  4659
            4663  4709  4732  4739  4783  4787  4793  4796  4815  4816  4817  4818  4819  4820  4821
            4822  4823  4824  4825  4826  4827  4828  4896  5010  5036  5342  5417  5418  5434  5508
            5509
.EQUIV       989   990   998  1013  1014  1043  1044  1045  1046  1047  1048  1049  1050  1051  1052
            1071  1072  1073  1074  1075  1076  1077  1078  1079  1080
.EVEN       3961  3967  3975  3992  4005  4601  6737  6891
.IF          922   938   939   940   941   961   964   967   969   987  1053  1081  1107  1118  1129
            1182  1186  1188  1217  1218  1219  1220  1224  1225  2191  2192  2196  2210  2215  2217
            2219  2221  2223  2224  2225  2227  2245  2249  2252  2299  2300  2353  2354  2377  2378
            2390  2392  2396  2400  2401  2410  2411  2436  2438  2443  2447  2448  2457  2458  2513
            2515  2522  2526  2527  2536  2537  2601  2603  2611  2615  2616  2625  2626  2689  2691
            2695  2699  2700  2709  2710  2737  2739  2745  2749  2750  2759  2760  2833  2835  2840
            2844  2845  2854  2855  2895  2897  2901  2905  2906  2915  2916  2940  2942  2968  2972
            2973  2982  2983  3079  3081  3086  3090  3091  3100  3101  3142  3144  3162  3166  3167
            3176  3177  3264  3266  3292  3296  3297  3306  3307  3400  3402  3461  3465  3466  3475
            3595  3597  3606  3610  3611  3620  3668  3672  3707  3709  3713  3717  3718  3727  3758
            3760  3765  3769  3770  3779  3789  3790  3804  3805  3822  3823  3837  3838  3878  3879
            3914  3915  3934  3935  3951  3952  3953  3954  3955  3956  3958  3960  3966  3970  3971
            3974  3984  3987  3989  3991  3994  3996  4002  4004  4005  4013  4016  4027  4030  4037
            4039  4040  4042  4046  4053  4057  4058  4059  4132  4153  4203  4281  4349  4351  4378
            4383  4389  4425  4426  4440  4464  4511  4512  4522  4535  4543  4545  4549  4550  4593
            4594  4601  4605  4608  4613  4618  4630  4632  4633  4634  4643  4645  4651  4652  4657
            4658  4659  4662  4708  4712  4731  4739  4786  4792  4796  4807  4816  4817  4818  4819
            4820  4821  4823  4824  4825  4826  4827  4828  4895  5009  5035  5341  5416  5417  5433
            5507  5508
.IFF          938   940   941   965   969   971   987  1183  1186  1188  1217  1225  2193  2197  2215
            2300  2354  2378  2390  2391  2392  2397  2401  2402  2410  2436  2437  2438  2444  2448
            2449  2457  2513  2514  2515  2523  2527  2528  2536  2601  2602  2603  2612  2616  2617
            2625  2689  2690  2691  2696  2700  2701  2709  2737  2738  2739  2746  2750  2751  2759
            2833  2834  2835  2841  2845  2846  2854  2895  2896  2897  2902  2906  2907  2915  2940
            2941  2942  2969  2973  2974  2982  3079  3080  3081  3087  3091  3092  3100  3142  3143
            3144  3163  3167  3168  3176  3264  3265  3266  3293  3297  3298  3306  3400  3401  3402
            3462  3466  3467  3475  3595  3596  3597  3607  3611  3612  3620  3668  3672  3707  3708
            3709  3714  3718  3719  3727  3758  3759  3760  3766  3769  3771  3779  3790  3791  3805
            3806  3823  3824  3838  3839  3879  3880  3915  3916  3935  3936  3952  3955  3958  3971
            3984  3987  4004  4014  4016  4030  4053  4058  4060  4133  4204  4282  4350  4426  4441
            4451  4512  4515  4523  4535  4536  4545  4577  4593  4606  4631  4632  4634  4658  4663
            4709  4721  4771  4787  4793  4895  5009  5035  5341  5417  5418  5433  5508
.IFT        3961  3967  3975  3992  4040  4514  4519  4642  4724  4732  4775  4782
.IFTF       3961  3967  3975  3992  4039  4451  4512  4515  4640  4721  4726  4771  4778
.IIF         921   926   931   935   936   937   938   941   942   944   945   946   947   948   949
             950   958  1224  2216  2219  2223  2224  2225  2227  2228  2297  2351  2375  3787  3802
            3820  3835  3876  3912  3932  3954  3969  3977  3979  3980  4004  4005  4017  4018  4019
            4020  4021  4026  4045  4053  4058  4108  4112  4200  4350  4356  4394  4395  4454  4585
            4594  4601  4609  4610  4611  4612  4613  4617  4641  4642  4655  4658  4659  4815  4816
            4817  4818  4819  4821  4823  4824  4825  4826  4827  5414  5505
.IRP        1225  1316  2191  2390  2436  2513  2601  2689  2737  2833  2895  2940  3079  3142  3206
            3264  3400  3595  3707  3758  3958  4292  4332  4678  4698  4934  4940  5223
.LIST          1   921   958  1095  1217  1218  2191  2229  2390  2410  2436  2457  2513  2536  2601
            2625  2689  2709  2737  2759  2833  2854  2895  2915  2940  2982  3079  3100  3142  3176
```

DZRPYB.CMB CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

```
                3264    3306    3400    3475    3595    3620    3707    3727    3758    3779    3961    3967    3975    3979    3992
                3996    4053    4535    4613    4807    4815    4816    4817    4818    4819    4620    4821    4822    4823    4824
                4825    4826    4827    4828
.M3CRO             1     941    1179    1180    2191    2389    2435    2512    2600    2688    2736    2832    2894    2939    3078
                3141    3263    3400    3595    3706    3757    4807
.MCALL           921    1095    2229
.NLIST             1     921     958    1095    1217    1218    2191    2229    2390    2410    2436    2457    2513    2536    2601
                2625    2689    2709    2737    2759    2833    2854    2895    2915    2940    2982    3079    3100    3142    3176
                3264    3306    3400    3475    3595    3620    3707    3727    3758    3779    3961    3967    3975    3979    3992
                3996    4053    4535    4613    4807    4815    4816    4817    4818    4819    4820    4821    4822    4823    4824
                4825    4826    4827    4828
.PAGE           1180    1635    2190
.REM               1
.REPT            958    1217    3027    5177    5210    5372
.SBTTL           931     952     962     973     985    1095    1180    1635    2194    2209    2249    2387    2390    2436    2513
                2601    2689    2737    2833    2995    2940    3079    3142    3264    3400    3595    3707    3758    3949    4009
                4011    4130    4201    4279    4347    4603    4660    4706    4784    4807    5643    5926
.TITLE           921
.WORD            958     959     960     970    1188    1191    1192    1193    1194    1197    1198    1199    1200    1201    1202
                1203    1206    1207    1208    1217    1225    1226    1227    1228    1229    1230    1231    1232    1233    1234
                1235    1236    1237    1238    1239    1240    1241    1242    1243    1244    1245    1246    1247    1248    1249
                1250    1251    1252    1253    1254    1256    1258    1259    1262    1263    1264    1265    1266    1270    1271
                1272    1276    1277    1278    1279    1280    1281    1282    1283    1284    1285    1315    1316    1317    1318
                1319    1320    1321    1322    1323    1324    1325    1326    1327    1328    1329    1330    1332    1333    1334
                1335    1336    1337    1338    1339    1340    1341    1342    1343    1344    1345    1346    1347    1351    1352
                1353    1354    1355    1356    1357    1358    1359    1360    1361    1362    1363    1364    1365    1366    1368
                1369    1370    1371    1372    1373    1374    1375    1376    1377    1378    1379    1380    1381    1382    1383
                1385    1386    1387    1388    1389    1390    1391    1392    1393    1394    1395    1396    1397    1398    1399
                1400    1402    1403    1404    1405    1406    1407    1408    1409    1410    1411    1412    1413    1414    1415
                1416    1417    1419    1420    1421    1422    1423    1424    1425    1426    1427    1428    1429    1430    1431
                1432    1433    1434    1436    1437    1438    1439    1440    1441    1442    1443    1444    1445    1446    1447
                1448    1449    1450    1451    1453    1454    1455    1456    1457    1458    1459    1460    1461    1462    1463
                1464    1465    1466    1467    1468    1470    1471    1472    1473    1474    1475    1476    1477    1478    1479
                1480    1481    1482    1483    1484    1485    1487    1488    1489    1490    1491    1492    1493    1494    1495
                1496    1497    1498    1499    1500    1501    1502    1504    1505    1506    1507    1508    1509    1510    1511
                1512    1513    1514    1515    1516    1517    1518    1519    1521    1522    1523    1524    1525    1526    1527
                1528    1529    1530    1531    1532    1533    1534    1535    1536    1538    1539    1540    1541    1542    1543
                1544    1545    1546    1547    1548    1549    1550    1551    1552    1553    1555    1556    1557    1558    1559
                1560    1561    1562    1563    1564    1565    1566    1567    1568    1569    1570    1572    1573    1574    1575
                1576    1577    1578    1579    1580    1581    1582    1583    1584    1585    1586    1587    1589    1590    1591
                1592    1593    1594    1595    1596    1597    1598    1599    1600    1601    1602    1603    1604    1606    1607
                1608    1609    1610    1611    1612    1613    1614    1615    1616    1617    1618    1619    1620    1621    1629
                1630    1631    3383    3984    3987    4003    4085    4090    4125    4197    4277    4351    4352    4353    4734
                4781    4782    4814    5295    5296    5297    5298    5299    5300    5301    5302    6741    6742    6745    6747
                6750    6753    6755    6757    6759    6761    6763    6764    6766    6769    6770    6772    6774    6776    6778
                6780    6783    6786    6789    6792    6793    6796    6798    6799    6800    6804    6808    6811    6815    6818
                6822    6826    6829    6833    6836    6839    6843    6846    6850    6853    6857    6861    6865    6869    6872
                6876    6879    6882    6886
```

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

*,NOW.SEQ/SOL/CRF/ML:TOC/PAGNUM=DZRPYB.SML,DZRPYB.CMB

RUN-TIME: 50 74 10 SECONDS
RUN-TIME RATIO: 472/135=3.4
CORE USED: 38K (75 PAGES)

L12

Spooler runtime 24 Seconds, 102 KCS, 6% disk reads, 3 disk writes, 153 pages