RK06

DISK DRIVE DIAG PART 1

MD-11-DZR6H-B

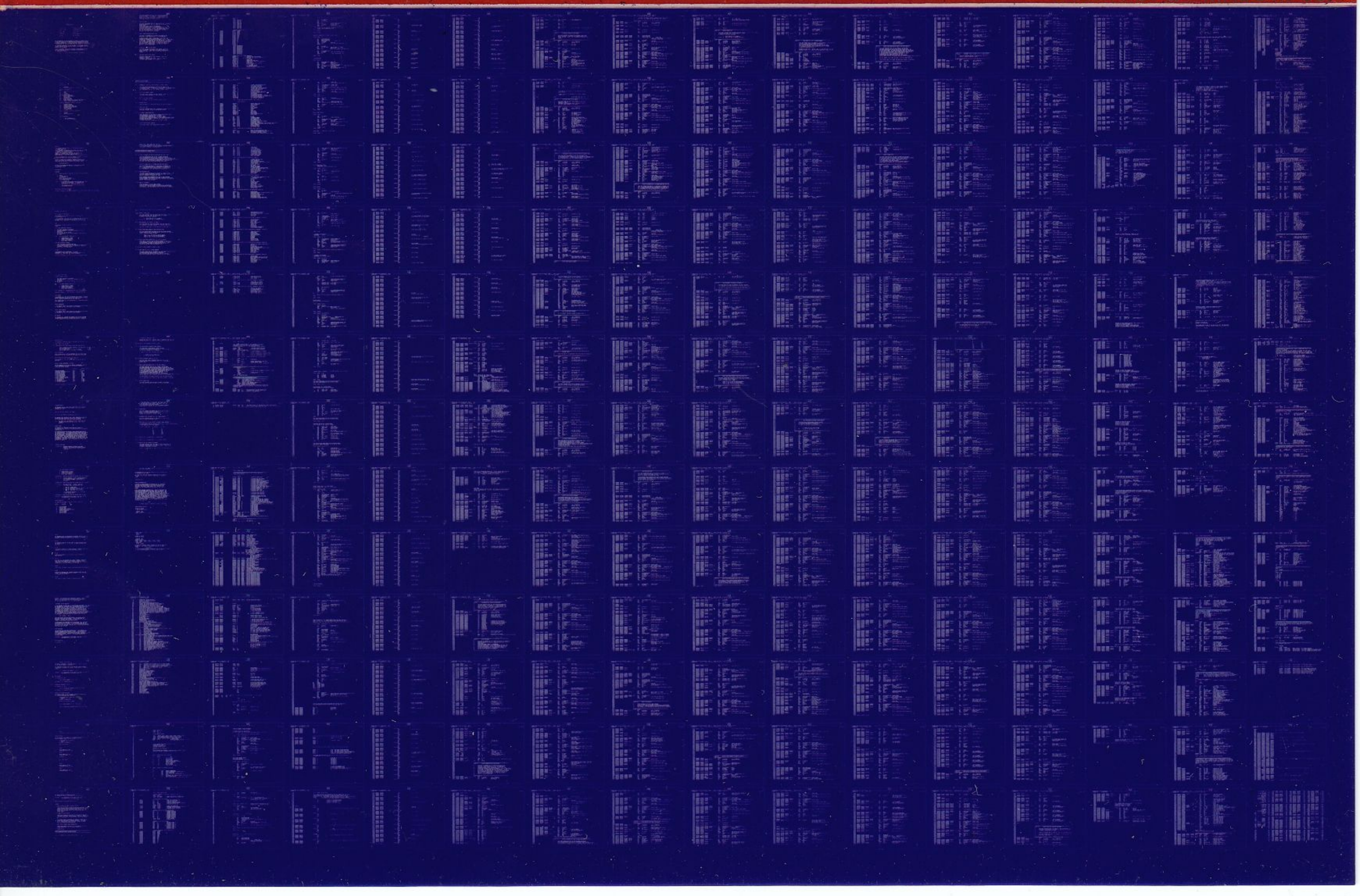
EP-DZR6H-B-DL-A

FICHE 1 OF 2

NOV 1976

digital

MADE IN USA



RK06

UNIBUS DIAG. PART 1

MD-11-DZR6H-B

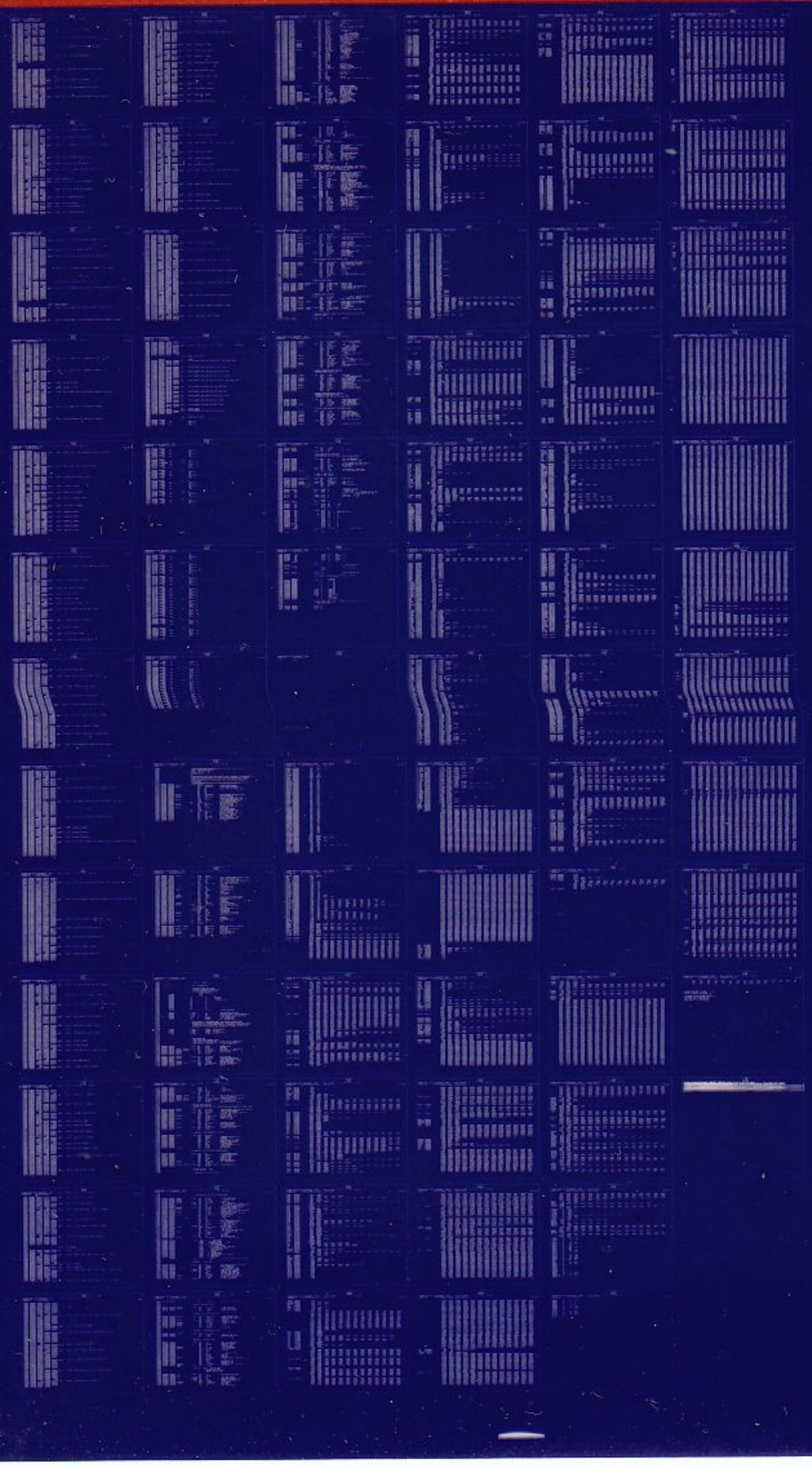
EP-DZR6H-B-DL-A

FICHE 2 OF 2

NOV 1976

digital

MADE IN USA



IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZR6H-B-D
PRODUCT NAME:	UNIBUS RK06 DISK DRIVE DIAGNOSTIC: PART 1
DATE:	AUGUST 1976
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	GARY PAPAZIAN

TABLE OF CONTENTS

1.0 ABSTRACT

THIS PROGRAM PERFORMS PART 1 OF THE DRIVE DIAGNOSTICS
TO INSURE THAT THE
DISK IS CAPABLE OF PERFORMING ALL STATIC & CYCLE UP TESTS.
IT INSURES THAT THE DRIVE CAN WRITE AND READ HEADERS
IN BOTH 20 & 22 SECTOR FORMATS.
FINALLY, IT INSURES THAT THE DISK CAN PERFORM SEEK
OPERATIONS BY DOING SEVERAL SEEK PATTERNS.
ERROR DETECTION LOGIC IS CHECKED BY SOFTWARE ERROR FORCING.

AFTER A SUCCESSFUL RUN (WITH NO ERRORS) OF PART 1, THE DRIVE
IS READY FOR PART 2 OF THE DRIVE DIAGNOSTICS.

TESTING IS BASED ON A HIERARCHY APPROACH STARTING WITH
BASIC LOGIC TESTS AND PROCEEDING THRU DYNAMIC TESTING. THE
TESTS WILL BE KEPT SMALL TO FACILITATE SCOPING LOOPS.

*****CAUTION*****

HALTING THIS PROGRAM ANYWHERE BUT AT THE END OF A PASS,
MAY LEAVE THE HEADERS IN THE DISK CARTRIDGE IN AN
UNDETERMINED STATE.

2.0 REQUIREMENTS

2.1 HARDWARE

THE FOLLOWING HARDWARE IS REQUIRED TO RUN THE DISK DIAGNOSTIC:

PDP-11
CONSOLE TELETYPE
16K MEMORY
KW11-L OR KW11-P OCK
RK06 UNIBUS CONTROLLER (RK611)
1 TO 8 RK06 DRIVES

NOTES: 1. IF NEITHER KW11-L OR P CLOCK IS USED, ALL TIMING
TESTS WILL BE BYPASSED. A MESSAGE AT THE
BEGINNING OF THE TESTS WILL CONFIRM THIS.

2. THE PROGRAM CAN WORK OFF EITHER FORMATTED OR
NON-FORMATTED PACKS.

2.2 PRELIMINARY TESTING & PROGRAMS

THE RK611 DISKLESS CONTROLLER DIAGNOSTICS (ALL PARTS) SHOULD FIRST RUN SUCCESSFULLY.

3.0 PROGRAM CONSIDERATIONS

3.1 PDP-11 FAMILY COMPATIBILITY

THIS PROGRAM CAN BE USED BY THE PDP-11/04,05,10,20, 34,35,40,45,50, & 70.

IT IS COMPATABLE WITH THE LSI-11 INSTRUCTION SET AND CAN TEST THE RK06 ONLY IF THE DRIVE CONTROLLER FOR THE LSI-11 IS DESIGNED TO BE DIAGNOSTICALLY COMPATABLE WITH THE RK611.

3.2 XXDP

THIS PROGRAM CAN BE CHAINED BY XXDP & WILL NOT OVERLAY THE LOADER.

CHAIN MODE OPERATION (MONITOR)

1. THE INPUT DIALOGUE IS BYPASSED.
2. THE BUSS ADDRESS & CONTROLLER INTERRUPT VECTOR IS DEFAULTED.
3. DRIVE 0 WILL NOT BE TESTED.
4. ALL OTHER DRIVES IN THE 'DRIVE PRESENT' CONDITION WILL BE TESTED.

NOTE: THE DRIVE PRESENT CONDITION IS:

A. HEADS MANUALLY LOADED
B. CORRECT PORT SELECTED
C. WRITE LOCK DISABLED
D. DRIVE READY INDICATOR ON

DUMP MODE OPERATION (MANUAL)

1. INPUT DIALOGUE IF STARTED FROM 220.
2. DRIVE 0 CAN BE TESTED, BUT THE OPERATOR IS FIRST GIVEN A MESSAGE TO REPLACE THE PACK IN DR0 WITH A SCRATCH PACK & TYPE <CR> WHEN DONE.

3.3 ACT/APT

THIS PROGRAM IS ACT COMPATIBLE. IT IS APT COMPATABLE TO THE EXTENT THAT APT HOOKS WILL BE IN THE PROGRAM & WILL WORK THRU THE 'UPTON INTERFACE'.

FOR OTHER INTERFACES, APT MAY ONLY LOAD & START THE PROGRAM.

I.E. LOAD & DUMP MODE.

AUTOMATIC MODE (MONITOR)

1. THE INPUT DIALOGUE IS BYPASSED.
2. THE BUSS ADDRESS & CONTROLLER INTERRUPT VECTOR IS DEFAULTED.
3. ALL DRIVES IN THE 'DRIVE PRESENT' CONDITION WILL BE TESTED.

NOTE: THE DRIVE PRESENT CONDITION IS:

A. HEADS MANUALLY LOADED
B. CORRECT PORT SELECTED
C. WRITE LOCK DISABLED
D. DRIVE READY INDICATOR ON

NOTE: SEVERAL APT CONSIDERATIONS ARE STILL TO BE DEFINED.

DUMP MODE (MANUAL): INPUT DIALOGUE IF STARTED FROM 220.

3.4 DUAL ACCESS

THIS PROGRAM WILL NOT TEST OR SUPPORT DUAL-ACCESS. A DRIVE EQUIPED WITH DUAL ACCESS MUST BE SWITCHED TO THE PORT UNDER TEST TO PREVENT CONTENTION WITH THE OTHER PORT.

DUAL ACCESS TESTS WILL BE INCORPORATED IN A SEPARATE PROGRAM AT A LATER DATE.

3.5 MEMORY MANAGEMENT

IF THE MEMORY PARITY CHECK OPTION IS AVAILABLE ON THE SYSTEM, THE PROGRAM WILL RUN WITH MEMORY CHECK ENABLED.

3.6 PARITY CHECK ENABLED

IF THE MEMORY PARITY CHECK OPTION IS AVAILABLE ON THE SYSTEM, THE PROGRAM WILL RUN WITH MEMORY CHECK ENABLED.

3.7 BAD SECTOR

THE PROGRAM WILL COMPARE DATA ERRORS WITH THE BAD SECTOR INFORMATION CONTAINED ON CYLINDER 410, HEAD 2. PRINTOUTS OF DATA ERRORS DUE TO BAD SECTORS/TRACKS WILL BE MASKED OUT.

3.8 EXECUTION TIME

THE EXECUTION TIMES SHOWN BELOW ARE BASED ON THE PDP 11/50.

TOTAL TIME: 5 MIN, 30 SEC

A BREAKDOWN OF THE MORE LENGTHY TESTS ARE SHOWN BELOW:

TEST 16 STATIC CYLINDER ADDRESS & DIFF REGS-PART 2	: 2 MIN, 15 SEC
TEST 36 FORMAT PACK	: 1 MIN
TEST 41 SEEK FROM CYL 0 TO ALL CYLINDERS	: 40 SEC
TEST 42 SEEK FROM CYL 410 TO ALL CYLINDERS	: 40 SEC

3.9 FAULT ISOLATION

FAULT ISOLATION WILL NOT BE PERFORMED FOR THE FIRST STAGE RELEASE BUT WILL BE INCLUDED FOR THE SECOND STAGE RELEASE.

3.10 ERROR CORRECTION AND FAILURE RATE ANALYSIS

THIS PROGRAM WILL NOT DO ERROR CORRECTION OF FAILURE RATE ANALYSIS.

3.11 DEFAULT UNIBUS ADDRESSES & VECTORS

THE FOLLOWING IS A LIST OF ALL DEFAULT ADDRESSES & VECTORS OF ALL HARDWARE TO BE USED & THEIR MEMORY ADDRESSES WHERE THEY CAN BE CHANGED.

	LOCATION	DEFAULT CONTENTS
RK06 BUSS ADDRESS	1264	177440
CONTROLLER INTERRUPT VECTOR	1334	210
CONTROLLER PRIORITY	1336	240
P-CLOCK STATUS REG	1340	172540
P-CLOCK SET BUFFER	1342	172542
P-CLOCK READ BUFFER	1344	172544
L-CLOCK STATUS REG	1346	177546
L-CLOCK INTERRUPT VECTOR	1350	100
P-CLOCK INTERRUPT VECTOR	1352	104
TTY KB STATUS REG	1144	177560
TTY KB BUFFER	1146	177562
TTY PRINTER STATUS REG	1150	177564
TTY PRINTER BUFFER	1152	177566

4.0 OPERATING PROCEDURE & CONTROL FUNCTIONS

4.1 PROGRAM LOADING

THE PROGRAM CAN BE LOADED FROM PAPER TAPE USING STANDARD PROCEDURE FOR ABSOLUTE LOADER TAPES; OR FROM ANY MEDIA SUPPORTED BY XXDP.

4.1.1 LOAD THE STARTING ADDRESS (SEE SEC 4.2).

4.1.2 SET SWITCH REGISTERS AS DESIRED (SEE SEC 4.3).

4.1.3 SET DRIVES TO BE TESTED IN THE 'LOAD' CONDITION & WITH THE APPROPRIATE PORT SELECTED & WRITE LOCK DISABLED. DRIVES NOT TO BE TESTED MUST HAVE BOTH PORTS DESELECTED.

NOTE: THE DRIVE WILL NOT RESPOND TO THE 'START SPINDLE' COMMAND IF THE RUN/STOP SWITCH IS IN THE 'STOP' POSITION.

4.1.4 PRESS 'START'

THE PROGRAM WILL IDENTIFY ITSELF AND WILL BEGIN A DIALOGUE WITH THE OPERATOR TO DETERMINE DRIVES TO BE TESTED (SEE SEC 4.5).

THE PROGRAM BEGINS TESTING ONLY THOSE DRIVES SPECIFIED BY THE INPUT DIALOGUE. IF A SPECIFIED DRIVE CANNOT BE FOUND BY THE PROGRAM IT WILL BE FLAGGED AS AN ERROR THAT THE DRIVE WAS NOT AVAILABLE. THEN BEGINNING WITH THE LOWEST NUMERICAL DRIVE AND PROCEEDING IN SEQUENTIAL ORDER, ALL VALID DRIVES WILL BE TESTED. ONE PASS THROUGH THE TEST SEQUENCE WILL BE PERFORMED ON EACH DRIVE BEFORE MOVING TO THE NEXT DRIVE IN SEQUENCE. THE DRIVE TO BE TESTED WILL BE TYPED AT THE BEGINNING OF EACH PASS. "END OF PASS" WILL BE TYPED AFTER TESTING ALL DRIVES.

4.2 STARTING LOCATIONS

LOCATION 200 - STARTING ADDRESS TO DEFAULT THE BUSS ADDRESS & THE CONTROLLER INTERRUPT VECTOR & TEST ALL DRIVES IN THE 'DRIVE PRESENT' CONDITION.

NOTE: THE DRIVE PRESENT CONDITION IS:

A. HEADS MANUALLY LOADED
B. CORRECT PORT SELECTED
C. WRITE LOCK DISABLED
D. DRIVE READY INDICATOR ON

LOCATION 204 - SAME AS 200 START BUT BYPASS TEST 16 (N SQUARE)

LOCATION 220 - STARTING ADDRESS TO INPUT TESTING PARAMETERS VIA THE INPUT DIALOGUE. BUSS ADDRESS & CONT. INTERRUPT VECTOR INPUTTED ONLY ON 1ST PASS.

LOCATION 230 - SAME AS 220 START BUT BYPASS TEST 16 (N SQUARE)

LOCATION 260 - RUN MODULE TEST ...DEFAULT MODE ONLY.
THIS SKIPS OVER THE FOLLOWING TESTS:

1. TEST 35 FORMAT PACK
2. TEST 36 DECREMENT FROM CYL 410 TO 0 & READ HEADERS
3. TEST 40 SEEK FROM CYL 0 TO ALL
4. TEST 41 SEEK FROM CYL 410 TO ALL

THE PURPOSE OF BYPASSING IS TO PROVIDE A QUICK MODULE TEST

LOCATION 270 - SAME AS 260 START BUT BYPASS TEST 16 ALSO.

IMPORTANT: FOR VARIATIONS OF THE ABOVE, SEE XXDP, ACT/APT CONSIDERATIONS IN SECTIONS 3.2 & 3.3.

4.3 SWITCH REGISTER

THE SWITCHES ARE USED TO PROVIDE CONTROL FUNCTIONS.

SWITCH	FUNCTION
15	HALT ON ERROR
14	LOOP ON TEST
13	INHIBIT ERROR TYPEOUT
12	BYPASS DRIVE AFTER 20 ERRORS
11	INHIBIT ITERATION
10	BELL ON ERROR
9	LOOP ON ERROR
8	LOOP ON TEST IN SW<07:00>

4.3.1 SW<15>

THE PROGRAM HALTS ON ENCOUNTERING AN ERROR, AFTER TYPING OUT THE ERROR MESSAGE AND PERTINENT INFORMATION, IF SW13=0. PRESSING "CONTINUE" RESTORES NORMAL OPERATION OF THE PROGRAM.

4.3.2 SW<14>

THE PROGRAM LOOPS ON THE TEST THAT IS BEING EXECUTED WHEN THE SWITCH IS PUT ON. THIS SWITCH IS NORMALLY USED ALONG WITH SW15.

4.3.3 SW<13>

THIS SWITCH INHIBITS ALL ERROR MESSAGES. NORMALLY USED WHEN LOOPING ON TEST (SW14) OR LOOPING ON ERROR (SW9).

4.3.4 SW<12>

THIS SWITCH BYPASSES A GIVEN DRIVE AFTER 20 ERRORS HAVE BEEN DETECTED.

4.3.5 SW<11>

EACH TEST WILL BE EXECUTED ONLY ONCE. NORMALLY AFTER THE FIRST PASS, EACH SUBTEST IS ITERATED A NUMBER OF TIMES (USUALLY 50, 5 IN SOME CASES). SETTING THIS SWITCH INHIBITS ITERATIONS, SO THAT QUICK PASSES CAN BE MADE.

4.3.6 SW<10>

RINGS A BELL ON ERROR. USEFUL WHEN ERROR TYPEOUT IS INHIBITED.

4.3.7 SW<09>

THIS SWITCH PROVIDES THE TIGHTEST POSSIBLE SCOPE LOOP FOR ERRORS. IF THE PROGRAM DETECTS AN ERROR, IT WILL LOOP BACK TO THE BEGINNING OF TEST.

4.3.8 SW<08>

THIS SWITCH IS USED TO SELECT A PARTICULAR TEST (AS PER

SW(00-7)) FOR EXECUTION AND SUBSEQUENT LOOPING. THUS IF
TEST 15 IS TO BE SELECTED THE SWITCH SETTING WOULD BE -
000415. IT SHOULD BE NOTED THAT BEFORE SELECTING TEST 15;
ALL THE PREVIOUS TESTS (1-14) WILL BE EXECUTED.

4.4 'SOFTWARE' SWITCH REGISTER

IF THE PROGRAM IS BEING RUN ON A SWITCHLESS PROCESSOR (I.E. AN 11/04 OR 11/34)
THE PROGRAM WILL DETERMINE THAT THE HARDWARE SWITCH REGISTER IS
NOT PRESENT AND WILL USE A 'SOFTWARE' SWITCH REGISTER. THE
'SOFTWARE' SWITCH REGISTER IS LOCATED AT LOCATION 176 (8). THE
SETTINGS OF THE "SOFTWARE" SWITCHES ARE CONTROLLED THROUGH
A KEYBOARD ROUTINE WHICH IS CALLED BY TYPING A 'CONTROL G'.
THE PROGRAM WILL RECOGNIZE THE 'CONTROL G' AT ANY TIME EXCEPT
WHEN THE PROGRAM IS AT A HIGHER PRIORITY PROCESSING AN RK06 INTERRUPT.
THE 'SOFTWARE' SWITCH VALUES ARE ENTERED AS AN OCTAL NUMBER
IN RESPONSE TO THE PROMPT FROM THE SWITCH ENTRY ROUTINE:

SWR = NNNNNN NEW =

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

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

4.5 INPUT DIALOGUE

THE DIALOGUE WILL BE DONE INTERACTIVELY. THE PROGRAM WILL
REQUEST A PARAMETER BY CONSOLE TYPEOUT. THE PARAMETER MAY
THEN BE ENTERED AS SPECIFIED BELOW OR ALLOWED TO DEFAULT
BY A CARRIAGE RETURN. UNRECOGNIZED OR ILLEGAL RESPONSES WILL
BE ECHOED BACK FOLLOWED BY "?". THE PROPER RESPONSE MAY
THEN BE ENTERED.

IMPORTANT: FOR VARIATIONS OF THE ABOVE, SEE XXDP, ACT/APT
CONSIDERATIONS IN SECTIONS 3.2 & 3.3.

4.5.1 DRIVE SELECTION

THE REQUEST WILL BE:

DRIVES TO BE TESTED:

THE DEFAULT RESPONSE IS CARRIAGE RETURN TO TEST ALL DRIVES IN THE 'DRIVE PRESENT' CONDITION.

THE OPERATOR CAN ALSO TYPE IN THE SPECIFIC DRIVE NUMBERS TO BE TESTED, SEPARATED BY COMMAS & TERMINATED BY A CARRIAGE RETURN.

E.G. DRIVES TO BE TESTED: 1,2,4,6

IMPORTANT: FOR VARIATIONS OF THE ABOVE, SEE XXDP, ACT/APT CONSIDERATIONS IN SECTIONS 3.2 & 3.3.

4.5.2 BUS ADDRESS

THE REQUEST WILL BE:

TYPE IN BUSS ADDRESS IF NOT 177440

THE DEFAULT IS A CARRIAGE RETURN

4.5.3 CONTROLLER INTERRUPT VECTOR

THE REQUEST WILL BE:

TYPE IN CONTROLLER INTERRUPT VECTOR IF NOT 210

THE DEFAULT IS A CARRIAGE RETURN.

4.5.4 EXAMPLE OF PROGRAM DIALOGUE

THE EXAMPLE SHOWN IS FOR A PROGRAM STARTED AT ADDRESS 220.
ALL OPERATOR RESPONSES ARE UNDERLINED.

```
UNIBUS RK06 DRIVE DIAGNOSTIC
    PART 1
MAINDEC-11-DZR6H-B-PB

DRIVES TO BE TESTED: 1,3<CR>
                     -------

TYPE IN BUSS ADDRESS IF NOT 177440 <CR>
                                   ----

TYPE IN CONTROLLER INTERRUPT VECTOR IF NOT 210 <CR>
                                               ----
```

```
                WILL TEST DRIVES:
                1
                3

                DRIVE 1

                (THE REST IS IDENTICAL TO THE EXAMPLE SHOWN IN 4.6 BELOW)
```

4.6 PROGRAM EXAMPLE

THE FOLLOWING IS AN EXAMPLE OF A PROGRAM STARTED AT THE DEFAULT ADDRESS (200) & WITH 2 DRIVES ON THE LINE.

```
                UNIBUS RK06 DRIVE DIAGNOSTIC
                    PART 1
                MAINDEC-11-DZR6H-B-PB

                WILL TEST DRIVES:
                0
                1

                DRIVE 0

                DRIVE SERIAL NO. AAA
                CARTRIDGE SERIAL NO. BBB

                DRIVE 1

                DRIVE SERIAL NO. CCC
                CARTRIDGE SERIAL NO. DDD

                END PASS #1


                WILL TEST DRIVES:
                0
                1

                DRIVE 0

                DRIVE SERIAL NO. AAA
                CARTRIDGE SERIAL NO. BBB

                DRIVE 1

                DRIVE SERIAL NO. CCC
                CARTRIDGE SERIAL NO. DDD
```

END PASS # 2

(ETC)

THE ABOVE ASSUMES NO ERRORS DETECTED.
THE NUMBER OF PASSES IS DETERMINED BY ACT/APT/XXDP

IMPORTANT: FOR VARIATIONS OF THE ABOVE, SEE XXDP, ACT/APT CONSIDERATIONS IN SECTIONS 3.2 & 3.3.

5.0 DRIVE DIAGNOSTIC FUNCTIONAL DESCRIPTION

5.1 GENERAL

A. BASIC CONTROLLER TESTS, SIZING & SETUP

THESE TESTS DO BASIC CONTROLLER REGISTER REFERENCE TESTS, CHECKS OPERATOR INPUTS AGAINST DRIVES SEEN ON THE LINE OR DEFAULTS TO TEST ALL THE DRIVES SEEN ON THE LINE. IT CHECKS THE EXISTENCE OF AN L OR P CLOCKS FOR USE IN THE TIMING TESTS.

B. STATIC & CYCLE UP TESTS

THESE TESTS CHECK OUT THE ABILITY TO SELECT & DESELECT THE DRIVE; TO DETECT PARITY, UNSAFE, AND FAULT CONDITIONS WITH THE DRIVE READY TO OPERATE BUT WITHOUT THE SPINDLE ON.

THE ENTIRE POWER UP SEQUENCE IS TESTED BY VERIFYING ALL STATUS BITS SET/RESET IN PROPER SEQUENCE: THE BRUSH CYCLE, INNER-OUTER LIMIT DETECTION, FORWARD, REVERSE, PIP...ETC STATUS BITS ARE CHECKED.

C. SEEK, WRITE HEADER, READ HEADER TESTS

THESE TESTS CHECK THE ABILITY OF THE DRIVE TO DO SEEKS, HEADER OPERATIONS & 20, 22 SECTOR FORMATTING.

5.2 TEST DESCRIPTIONS

****
BASIC CONTROLLER TESTS, SIZING & SETUP
****

TEST 1 REFERENCE ALL CONTROLLER REGISTERS

THIS TEST VERIFIES THAT ALL THE CONTROLLER REGISTERS CAN BE ACCESSED. THE INABILITY TO BE ACCESSED WILL RESULT IN A TIMEOUT TRAP WITH AN ERROR MESSAGE. ANY ERROR IN THIS TEST WILL RESULT IN ABORTING ALL OTHER TESTS AND JUMPING TO 'END OF PASS'

TEST 2 SIZE THE BUSS

THIS TEST IS ENTERED ONLY IF 'DRIVE SELECTION' IS DEFAULTED EITHER BY RUNNING IN THE ALTO MODE OR A 200 START IN THE MANUAL MODE.
EVERY DRIVE FROM 0 THRU 7 IS ADDRESSED.
CONTROLLER ERROR (CERR) IS EXAMINED AND IF NOT SET, THE DRIVE WILL BE TESTED. IF SET, THE PROGRAM WILL BYPASS TESTING THAT DRIVE ONLY IF THE ERROR WAS A RESULT OF MDS, UFE OR NED BEING SET; OR BOTH NED & DRA RESET INDICATING THE OTHER PORT IS ACCESSED.

TEST 3 VERIFY OPERATOR DRIVE SELECTIONS

THIS TEST IS ENTERED ONLY IF DRIVE SELECTION IS NOT DEFAULTED. EVERY DRIVE FROM 0 TO 7 IS ADDRESSED &

CONTROLLER ERROR (CERR) IS EXAMINED. IF NOT SET, THE PROGRAM WILL ASSUME THE DRIVE IS PRESENT. IT WILL THEN CHECK TO SEE THAT THE DRIVE WAS INPUTTED FOR TESTING. IF NOT, IT WILL BE AN ERROR. IF CERR WAS SET, THAT DRIVE WILL BE BYPASSED ONLY IF THE ERROR WAS A RESULT OF MDS OR UFE SET OR BOTH NED & DRA RESET (WRONG PORT). IF CERR IS A RESULT OF NED ONLY, IT IS CHECKED AGAINT THE INPUTTED INFOR TO VERIFY IT WAS NOT SPECIFIED.

TEST 4 UNLOAD ALL DRIVES TO BE TESTED

THIS TEST UNLOADS THE DRIVES TO BE TESTED, WAITS FOR ATTN & VERIFIES IT CAME FROM THE CORRECT DRIVE WITHIN APPROX 100 USEC. DRDY & DSC IN RKMR2 ARE CHECKED TO BE CORRECT.

TEST 5 FIND NEXT DRIVE TO BE TESTED

THIS TEST FINDS THE NEXT DRIVE PRESENT & PUTS THAT ADDRESS IN 'DRVAD'.
THROUGHOUT THE FOLLOWING TESTS, THE DRIVE TESTED IS THE DRIVE WHOSE ADDRESS IS IN 'DRVAD'.

****
STATIC & CYCLE UP TESTS
****

TEST 6 REFERENCE & CHECK ALL STATUS BYTES IN RKMR2 & RKMR3

THIS TEST WAITS FOR APPROX 10 SEC FOR THE HEADS TO BE
IN THE HOME POSITION & SPEED CK TO GO LOW FROM THE 'UNLOAD' GIVEN
IN A PREVIOUS TEST. IT THEN CHECKS THE ABILITY TO REFERENCE ALL
DRIVE REGISTERS AND THAT THEY CONTAIN CORRECT STATUS.

TEST 7 PRINT DRIVE SERIAL NUMBER

THIS TEST READS & PRINTS THE DRIVE SERIAL # FROM MSG A, WORD 11
IN DECIMAL & IS PERFORMED ON THE 1ST PASS ONLY

TEST 10 SET VV WITH PACK COMMAND

IF VV IS RESET, THE PACK COMMAND IS USED TO SET IT.

TEST 11 RELEASE DRIVE

TESTS THE ABILITY TO RECOGNIZE THE RLS BIT AND NOT RAISE SACK

TEST 12 DRIVE TYPE TEST

THIS TEST COMPARES DRIVE TYPE IN MESSAGE A AGAINST 'DDT' IN RKDS.
WRONG CDT IN RKCS1 IS SENT & ERRORS ARE VERIFIED.

TEST 13 C-D PARITY ERROR DETECTION

TESTS THE ABILITY OF THE DRIVE TO DETECT EVEN PARITY SENT BY
THE CONTROLLER BY SETTING 'PAT' ON RKMR1.
THE DRIVE SHOULD RESPOND WITH 'C-D PARITY ERROR'
THE DRIVE STILL SENDS ODD PARITY TO THE CONTROLLER WHICH IS NOW
CHECKING FOR EVEN PARITY THEREFORE THE CONTROLLER SHOULD DETECT
AN ERROR AND SET SPAR.
THE ERROR CONDITION IS RESET WITH THE CLEAR COMMAND

TEST 14 VERIFY START SPINDLE COMMAND

THE PROGRAM CHECKS THE ENTIRE STARTUP SEQUENCE, IE:

BRUSH CYCLE, HEADS HOME, FWD, REV ETC.
BY VERIFYING ALL APPROPRIATE STATUS BITS FOR PROPER SEQUENCING.
THE CYLINDER ADDRESS & CYLINDER DIFFERENCE REGS ARE CHECKED
TO BE ZERO AT THE END OF THE SEQUENCE.

****
SEEK/READ HEADER/WRITE HEADER TESTS
****

TEST 15 STATIC CYLINDER DIFF AND CYLINDER ADDR REG TEST; PART 1

THIS TEST CHECKS EACH BIT OF THE CYLINDER DIFFERENCE
AND CYLINDER ADDRESS REGISTERS BY PERFORMING SEEKS TO ALL
MAJOR CYLINDERS (0,1,2,4,8,16,32,64,128,256) WITH EVEN PARITY SET.
THIS FREEZES THE INFORMATION IN THE ABOVE REGISTERS & ALLOWS FOR CHECKING.
THIS TEST VERIFIES C-D PARITY ERROR BIT SET, THAT HEADS DID
NOT MOVE & ALL OTHER APPLICABLE STATUS BITS & REGS.

TEST 16 STATIC CYLINDER DIFF & CYLINDER ADDR REG TEST-PART 2

THIS TEST CHECKS THE ABILITY OF THE DRIVE TO PROPERLY SET THE CYL
DIFF. & CYL ADDR REGS FOR ALL COMBINATIONS BY SEEKING TO
ALL CYLINDERS FROM EVERY OTHER CYLINDER. (N SQUARE SEEKS).
IT IS PERFORMED IN THE SAME MANNER AS THE ABOVE TEST.

TEST 17 HEAD REGISTER TEST

THIS TEST CHECKS THE ABILITY TO SELECT ALL HEADS (0,1,2)
VIA RKDA & READING BACK FROM MSG B3 BY THE SELECT DRIVE CMD.
HEAD 3 IS CHECKED TO PRODUCE INV. ADDR.

SINCE CHANGING HEAD ADDRESSES ARE TIED TO SEEK COMMANDS,
SELECTING HEAD 3 MUST RESULT IN A SEEK INCOMPLETE ALONG WITH
ILLEGAL ADDRESS. IF NOT, THIS MEANS THAT CHANGING HEAD ADDRESSES
ARE NOT TIED TO SEEK COMMANDS

TEST 20 SEEK TO CYLINDER 0

TESTS THE ABILITY TO DO A SEEK COMMAND.
VERIFIES THERE WAS NO MOVEMENT BY CHECKING ALL APPROPRIATE
STATUS BITS. VERIFIES COMMAND COMPLETION BETWEEN 10-15USEC.
READ HEADER IS NOT PERFORMED AS THE PACK MAY NOT BE FORMATTED.

TEST 21 TEST SECTOR COUNT REG. FOR 22 & 20 SECTOR FORMAT

TEST 22 DETECT OUTER LIMIT

THIS TEST VERIFIES THAT THE ABOVE TEST DID ACTUALLY POSITION ON CYLINDER 0
BY DETECTING OUTER LIMIT AS THE ADJACENT CYLINDER.
AN ERROR IN THIS TEST INDICATES:

A. HEADS WERE NOT ON CYLINDER 0
AND/OR B. COULD NOT SEEK IN REVERSE DIRECTION.

TEST 23 READ HEADERS, ALL TRACKS, CYLINDER 0

THIS TEST DOES NOT CHECK THE HEADERS, BUT ONLY THE FACT
THAT HEADERS CAN BE READ CORRECTLY BY CHECKING CERR & DTE.

THIS TEST IS VALID ONLY IF THE PACK HAS NOT BEEN ZERO'D OUT.

TEST 24 BASIC WRITE/READ HEADER & HEAD SWITCHING TEST

THIS TEST CHECKS HEAD SWITCHING BY WRITING UNIQUE HEADERS
ON EACH TRACK OF CYL 0, READING BACK & VERIFYING THEY REMAINED
UNIQUE. 22 SECTOR FORMAT IS USED

I.E. TRACK 0: ALL 0'S FOR ALL SECTOR HEADERS
TRACK 1: 0101 FOR ALL SECTOR HEADERS
TRACK 2: ALL 1'S FOR ALL SECTOR HEADERS

TEST 25 BASIC WRITE/READ HEADER TEST; ALL 1'S, 20 SECTORS

USING HEAD 0, WRITE & READ 20 SECTOR HEADERS BY WRITING ALL
1'S AS HEADERS. ATTEMPT TO FIND SECTORS 20 & 21. VERIFY
THEY ARE NO LONGER THERE BY READING 22 SECTORS AND NOT
FINDING 0'S AS DATA FROM THE PREVIOUS TEST.

TEST 26 WRITE & READ HEADERS CYLINDER 0, HEAD 0

TEST 27 SEEK FROM CYL 0 TO 1 & READ HEADERS

THIS TEST CHECKS MSG A & B WORDS 0,1,2 FOR CORRECT STATUS AFTER RDY
IS RECEIVED FROM A SEEK COMMAND TO DETERMINE
THAT THE HEADS ARE ACTUALLY MOVING & THE CYL DIFF IS 1.
AFTER ATTN IS RECEIVED, CERR IS EXAMINED FOR ANY ERRORS.

F02

CYL DIFFERENCE IN MSG A2 IS VERIFIED TO BE 0 & CYL ADDR

IN MSG B2 IS VERIFIED TO BE 1.

HEADERS ARE READ FROM 1 SECTOR, HEAD 0 & VERIFIED THAT THEY ARE DIFFERENT FROM CYL 0 TO SHOW THAT THE HEADS DID ACTUALLY MOVE.

TEST 30 WRITE & READ HEADERS CYLINDER 1, HEAD 0

TEST 31 TEST RECALIBRATE COMMAND & READ HEADERS

THIS TEST DOES A RECALIBRATE & READS HEADERS.
IT VERIFIES THAT WRITING HEADERS ON CYL 1 FROM THE PREVIOUS TEST DID NOT OVERWRITE CYL 0 HEADERS.

AN ERROR IN THIS TEST INDICATES THAT HEADS:

A. MOVED TO A CYL OTHER THAN 1
OR B. DID NOT GET BACK TO CYL 0

TEST 32 SINGLE INCREMENT SEEKS TO CYL 410

THIS TEST DOES SINGLE INCREMENT SEEKS OUT TO CYL 410 WITHOUT ANY WRITING OR READING SO AS NOT TO INADVERTENTLY DESTROY DATA.

TEST 33 READ & SAVE BAD SECTOR INFO & TYPE PACK SERIAL #

THIS TEST VERIFIES THAT CYL 410, TRACK 2 CAN BE READ.
THIS AREA CONTAINS BAD SECTOR INFO WHICH IS WRITTEN BY THE FACTORY DURING MANF. ALL BAD SECTOR INFO (BSE) WILL BE STORED AT THIS TIME TO MASK FUTURE READ HEADER OR DATA ERROR PRINTOUTS.
IF BSE INFO CANNOT BE READ, OR IF AFTER READING THE BSE INFO IT IS DETERMINED THAT AN ALIGNMENT CARTRIDGE IS USED, A MESSAGE WILL BE TYPED INDICATING THAT ALI.
FUTURE FORMAT AND READ-WRITE TESTS WILL BE BYPASSED.
THIS IS DONE SO AS NOT TO DESTROY BSE INFO OR AN ALIGNMENT PACK BY WRITING

THE PACK SERIAL # IS TYPED IN OCTAL & FOR THE FIRST PASS ONLY.

THIS IS THE FIRST TEST WHERE THE READ DATA COMMAND IS PERFORMED

TEST 34 DETECT INNER LIMIT

THIS TEST VERIFIES THAT THE LAST CYLINDER IN THE ABOVE TEST WAS 410 BY DETECTING INNER LIMIT AS THE ADJACENT CYLINDER.

IF THIS TEST FAILS, IT INDICATES THAT HEADS WERE NOT ON CYL 410
& THAT BSE INFO IS NOT VALID. THE FORMAT PACK TEST
& ALL READ-WRITE TESTS ARE BYPASSED
TO AVOID DESTROYING BSE INFO OR AN ALIGNMENT CARTRIDGE
SINCE THERE IS A SEEKING OR LIMIT DETECTION PROBLEM.

TEST 35 FORMAT PACK

THIS TEST FORMATS THE ENTIRE PACK IN 22 SECTOR FORMAT BY
DOING 1 CYLINDER INCREMENTAL SEEKS
FROM 0 TO 410 WITH WRITE HEADER COMMANDS (ALL TRACKS).
HEADERS WILL BE READ IN THE NEXT TEST

TEST 36 DECREMENT FROM CYLINDER 410 TO 0 & READ HEADERS

THIS TEST VERIFIES MOTION IN THE NEGATIVE DIRECTION BY
SINGLE CYLINDER INCREMENTAL SEEKS.

TEST 37 SEEK FROM CYL 0 TO ALL MAJOR CYLINDERS & READ HEADERS

THIS TEST SEEKS FROM CYL 0 TO ALL THE MAJOR CYLINDERS & READS HEADERS.
IT THEN SEEKS CYL 0 & READS HEADERS.

MAJOR CYLINDERS ARE:

1 (DECIMAL)	=	1 (OCTAL)
2		2
4		4
8		10
16		20
32		40
64		100
128		200
256		400

TEST 40 SEEK TO ALL CYLINDERS FROM 0 & READ HEADERS

TEST 41 SEEK TO ALL CYLINDERS FROM CYL 410 & READ HEADERS

TEST 42 SEEK TO ALL KEY INVALID CYLINDERS

THIS TEST VERIFIES THAT 'INV ADDR' & 'SEEK INCOMPLETE' IS
PRODUCED & THAT HEADS DO NOT MOVE OR UNLOAD IF AN ILLEGAL
CYLINDER IS SPECIFIED IN A SEEK.

INVALID CYLINDERS ARE 411 THRU 511 (10) IE. 633 THRU 777 (8)

THIS TEST CHECKS KEY INVALID CYLINDERS 411,412,416,448 & 480
FOR A FULL LOGIC TEST

THE PROGRAM DOES NOT REQUIRE FORMATTED PACKS AS FORMATTING
IS PERFORMED IN ANY CASE.

ANY TEST THAT MODIFIES STANDARD FORMATTING IS FOLLOWED BY A
'CLEAN UP' TEST TO PUT THOSE CYLINDERS BACK TO STANDARD
FORMAT.

6.0 ERROR REPORTING

6.1 ERROR INTERPRETATION

WHENEVER AN ERROR MESSAGE IS PRINTED OUT, ALL REGISTERS
AND OTHER DATA PERTAINING TO THE ERROR ARE ALSO GIVEN.
MSG A(OC), MSG B(01), RKER, RKBA...ETC, INDICATE THE
CONTENTS OF THE CORRESPONDING REGISTERS AT THE TIME OF
ERROR.

EVERY ERROR MESSAGE CONTAINS A PC. THIS PC INDICATES THE
POSITION IN PROGRAM WHERE THE ERROR CALL IS LOCATED. THE
ERROR MESSAGE, BECAUSE OF PRACTICAL CONSIDERATIONS IS MADE
SHORT AND MEANINGFUL. THE USER IS ADVISED TO LOOK UP THE
PC IN THE PROGRAM LISTING, WHERE HE WILL FIND MORE INFORMATION
ABOUT THE ERROR. IN MANY INSTANCES, A SINGLE FAULT WILL
GIVE RISE TO MORE THAN ONE ERROR REPORT. A LITTLE DELIBERATION
AND CAREFUL EXAMINATION OF THE DATA GIVEN WILL BE CERTAINLY
VERY HELPFUL IN PINPOINTING THE FAULT. A BRIEF EXPLANATION
OF WHAT IS BEING CHECKED IN THE TEST IS GIVEN AT THE
BEGINNING OF EVERY TEST. ALL THE NUMBERS GIVEN WITH ERROR
MESSAGES ARE IN OCTAL.

NOTE

NO ERROR LOGGING OR OPERATION HISTORY IS PROVIDED.

6.2 ERROR PRINTOUT EXAMPLE

```
MESSAGE AO ERROR
AT END OF HEAD LOADING

  TEST NO.  PC
  000014  013432
RKMR2   RKMR3   RKER    RKDS    RKCS1   RKCS2
```

```
140141  100000  000000  140101  040200  000101
  SHOULD BE
  150341

MESSAGE B1 ERROR
AFTER LIMIT DETECT

  TEST NO.  PC
  000023  023316
RKMR2   RKMR3   RKER    RKDS    RKCS1   RKCS2
045720  030001  000000  040000  040200  000100
  SHOULD BE
  045720  120001

IN THE FIRST EXAMPLE, RKMR2 (MESSAGE A0) DID NOT READ BACK
CORRECTLY.  THE CORRECT "SHOULD BE" CONTENTS IS UNDER
'RKMR2'.

IN THE 2ND EXAMPLE, RKMR3 (MESSAGE B1) DID NOT READ BACK CORRECTLY
WITH THE CORRECT 'SHOULD BE' CONTENTS UNDER 'RKMR3'.
```

```
                                .NLIST  CND,MC,MD
                                .LIST   ME
                                .ENABL  ABS,AMA

                                ;DEFINE SYSMAC MACROS

                                .MCALL  .HEADER,.SWRHI,.SWRLO,.EQUAT,.SETUP,.$CATCH,.$CMTAG
                                .MCALL  .$EOP,.$SCOPE,.$ERROR,.$TYPE,.$TYPOCT,.$SAVE
                                .MCALL  .$TYPDEC,.$READ,.$RDOCT,.$TRAP,.$DB2O,.$SUFRS
                                .MCALL  .$ACT11,.$APTHDR,.$APTYPE,.$SB2D,.$DB2D
000000  167400                  $SWR=   167400                  ;DEFINE SWITCHES 15,14,13,11,10,9,8
000001  000001                  $TN=    1                       ;SET FIRST TEST NO. TO 1


                                .TITLE  UNIBUS RK06 DRIVE DIAGNOSTIC  PART 1
                                ;*COPYRIGHT (C) 1976
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*PROGRAM BY GARY PAPAZIAN
                                ;*
                                ;*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
                                ;*        12            ABORT DRIVE AFTER 20 ERRORS
                                ;*        11            INHIBIT ITERATIONS
                                ;*        10            BELL ON ERROR
                                ;*         9            LOOP ON ERROR
                                ;*         8            LOOP ON TEST IN SWR<7:0>


                                .SBTTL  SUMMARY OF STARTING LOCATIONS
                                ;*
                                ;*
                                ;*              200     DEFAULT PARAMETERS
                                ;*              204     DEFAULT PARAMETERS & BYPASS TEST 16
                                ;*              220     INPUT PARAMETERS
                                ;*              230     INPUT PARAMETERS & BYPASS TEST 16
                                ;*              240     ODT11
                                ;*              260     RUN MODULE TEST VERSION-DEFAULT MODE ONLY BYPASS
                                ;*                      TESTS 35,36,40 & 41
                                ;*              270     SAME AS 260 START BUT BYPASS TEST 16 ALSO
                                ;*
```

```
                        .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
000200                  PR4=    200             ;;PRIORITY LEVEL 4
000240                  PR5=    240             ;;PRIORITY LEVEL 5
000300                  PR6=    300             ;;PRIORITY LEVEL 6
000340                  PR7=    340             ;;PRIORITY LEVEL 7

                        ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000                  SW15=   100000
040000                  SW14=   40000
020000                  SW13=   20000
010000                  SW12=   10000
004000                  SW11=   4000
002000                  SW10=   2000
001000                  SW09=   1000
000400                  SW08=   400
000200                  SW07=   200
000100                  SW06=   100
000040                  SW05=   40
000020                  SW04=   20
000010                  SW03=   10
000004                  SW02=   4
```

```
                        SW01=   2
                        SW00=   1
                        .EQUIV  SW09,SW9
                        .EQUIV  SW08,SW8
                        .EQUIV  SW07,SW7
                        .EQUIV  SW06,SW6
                        .EQUIV  SW05,SW5
                        .EQUIV  SW04,SW4
                        .EQUIV  SW03,SW3
                        .EQUIV  SW02,SW2
                        .EQUIV  SW01,SW1
                        .EQUIV  SW00,SW0

                        ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
100000                  BIT15=  100000
040000                  BIT14=  40000
020000                  BIT13=  20000
010000                  BIT12=  10000
004000                  BIT11=  4000
002000                  BIT10=  2000
001000                  BIT09=  1000
000400                  BIT08=  400
000200                  BIT07=  200
000100                  BIT06=  100
000040                  BIT05=  40
000020                  BIT04=  20
000010                  BIT03=  10
000004                  BIT02=  4
000002                  BIT01=  2
000001                  BIT00=  1
                        .EQUIV  BIT09,BIT9
                        .EQUIV  BIT08,BIT8
                        .EQUIV  BIT07,BIT7
                        .EQUIV  BIT06,BIT6
                        .EQUIV  BIT05,BIT5
                        .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  RK06 CONTROLLER REGISTER DEFINITION
```

```
                                ;       $BASE=177440

000000                          RKCS1=  0               ;CONTROL AND STATUS REGISTER 1
000002                          RKWC=   2               ;WORD COUNT REGISTER
000004                          RKBA=   4               ;BUS ADDRESS REGISTER
000006                          RKDA=   6               ;DESIRED TRACK SECTOR REGISTER
000010                          RKCS2=  10              ;CONTROL AND STATUS REGISTER 2
000012                          RKDS=   12              ;DRIVE STATUS REGISTER
000014                          RKER=   14              ;ERROR REGISTER
000016                          RKASOF= 16              ;ATTENTION SUMMARY AND OFFSET REGISTER
000020                          RKDC=   20              ;DESIRED CYLINDER REGISTER
000024                          RKDB=   24              ;DATA BUFFER
000026                          RKMR1=  26              ;MAINTENANCE REGISTER 1
000034                          RKMR2=  34              ;MAINTENANCE REGISTER 2 (MESSAGE LINE A)
000036                          RKMR3=  36              ;MAINTENANCE REGISTER 3 (MESSAGE LINE B)
000030                          RKECPS= 30              ;ECC POSITION INFORMATION
000032                          RKECPT= 32              ;ECC PATTERN INFORMATION

                                .SBTTL  CONTROL AND STATUS REGISTER 1 BITS (RKCS1:0)

                                ;       DRIVE COMMANDS

000001                          SELDRV= 1               ;SELECT DRIVE (GET STATUS)
000003                          PACK=   3               ;PACK ACKNOWLEDGE
000005                          CLEAR=  5               ;DRIVE CLEAR
000007                          UNLOAD= 7               ;UNLOAD
000011                          SRTSPL= 11              ;START SPINDLE
000013                          RECAL=  13              ;RECALIBRATE
000015                          OFFSET= 15              ;OFFSET
000017                          SEEK=   17              ;SEEK
000021                          RDDATA= 21              ;READ DATA
000023                          WRDATA= 23              ;WRITE DATA
000025                          RDHEAD= 25              ;READ HEADER
000027                          WRHEAD= 27              ;WRITE HEADER AND DATA
000031                          WRTCHK= 31              ;WRITE CHECK

000001                          GO=     BIT0            ;GO BIT
000100                          IE=     BIT6            ;INTERRUPT ENABLE
000200                          RDY=    BIT7            ;CONTROLLER READY
000400                          BA16=   BIT8            ;BUS ADDRESS BIT 16
001000                          BA17=   BIT9            ;BUS ADDRESS BIT 17
002000                          CDT=    BIT10           ;CONTROLLER DRIVE TYPE (0=RK06)
004000                          CTO=    BIT11           ;CONTROLLER TIMEOUT
010000                          CFMT=   BIT12           ;CONTROLLER DRIVE FORMAT (0=22 SECTOR, 1=20 SECTOR)
020000                          SPAR=   BIT13           ;SERCON PARITY ERROR DETECTED BY CONTROLLER
040000                          DI=     BIT14           ;DRIVE INTERRUPT
100000                          CERR=   BIT15           ;CONTROLLER ERROR
100000                          CCLR=   BIT15           ;CONTROLLER CLEAR

                                .SBTTL  CONTROL AND STATUS REGISTER 2 BITS (RKCS2:10)

000007                          DRVMSK= 7               ;MASK FOR DRIVE SELECTION CODE
000010                          RLS=    BIT3            ;DESELECT OR RELEASE DRIVE IN BITS 0-2
000020                          BAI=    BIT4            ;BUS ADDRESS INCREMENT INHIBIT
000040                          SCLR=   BIT5            ;SUBSYSTEM CLEAR CONTROLLER AND ALL DRIVES
```

```
                        IR=     BIT6            ;INPUT READY
                        OR=     BIT7            ;OUTPUT READY
                        UFE=    BIT8            ;UNIT FIELD ERROR
                        MDS=    BIT9            ;MULTIPLE DRIVE SELECT
                        PGE=    BIT10           ;PROGRAMMING ERROR
                        NEM=    BIT11           ;NON-EXISTENT MEMORY
                        NED=    BIT12           ;NON-EXISTENT DRIVE
                        UPE=    BIT13           ;UNIBUS PARITY ERROR
                        WCE=    BIT14           ;WRITE CHECK ERROR
                        DLT=    BIT15           ;DATA LATE ERROR

                        .SBTTL  ERROR REGISTER BIT DEFINITION (RKER:14)

                        ILF=    BIT0            ;ILLEGAL FUNCTION CODE
                        SKI=    BIT1            ;SEEK INCOMPLETE
                        NXF=    BIT2            ;NON-EXECUTABLE FUNCTION
                        DRPAR=  BIT3            ;DRIVE DETECTED SERCON PARITY ERROR
                        FMTE=   BIT4            ;FORMAT ERROR
                        DTYE=   BIT5            ;DRIVE TYPE ERROR
                        ECH=    BIT6            ;ECC HARD
                        BSE=    BIT7            ;BAD SECTOR ERROR
                        HVRC=   BIT8            ;HEADER VRC ERROR
                        COE=    BIT9            ;CYLINDER ADDRESS OVERFLOW ERROR
                        IDAE=   BIT10           ;INVALID DISK ADDRESS ERROR: HEAD/CYL
                        WLE=    BIT11           ;WRITE LOCK ERROR
                        DTE=    BIT12           ;DRIVE TIMING ERROR
                        OPI=    BIT13           ;OPERATION (SEARCH) INCOMPLETE
                        UNS=    BIT14           ;DRIVE UNSAFE
                        DCK=    BIT15           ;DATA CHECK

                        .SBTTL  STATUS REGISTER BIT DEFINITION (RKDS:12)

                        DRA=    BIT0            ;DRIVE AVAILABLE (CONTROLLER IS SET IF
                                                ;   THIS BIT IS RESET)
                        OFST=   BIT2            ;DRIVE OFFSET
                        ACLO=   BIT3            ;AC LOW
                        DCLO=   BIT4            ;DC LOW
                        DROT=   BIT5            ;DRIVE OFF TRACK
                        VV=     BIT6            ;VOLUME VALID
                        DRDY=   BIT7            ;DRIVE READY
                        DDT=    BIT8            ;DRIVE TYPE (0=RK06)
                        WRL=    BIT11           ;WRITE LOCK
                        PIP=    BIT13           ;POSITIONING IN PROGRESS
                        DSC=    BIT14           ;DRIVE STATUS CHANGE
                        SVAL=   BIT15           ;STATUS VALID

                        .SBTTL  MAINTENANCE REGISTER 1 BIT DEFINITION (RKMR1:22)

                        MESMSK= 17              ;MESSAGE MASK
                        PAT=    BIT4            ;FORCE EVEN PARITY ON SERCON MESSAGE LINES
                        DMD=    BIT5            ;DIAGNOSTIC MODE
                        MSP=    BIT6            ;MAINTENANCE SECTOR PULSE
                        MIND=   BIT7            ;MAINTENANCE INDEX
                        MCLK=   BIT8            ;MAINTENANCE CLOCK
                        MERD=   BIT9            ;MAINTENANCE ENCODED READ DATA
                        MEWD=   BIT10           ;MAINTENANCE ENCODED WRITE DATA
```

```
                000040          PCA=    BIT11           ;PRECOMPENSATION ADVANCE
                010000          PCD=    BIT12           ;PRECOMPENSATION DELAY
                020000          ECCW=   BIT13           ;ECC WORD IS BEING READ OR WRITTEN
                040000          WRTGAT= BIT14           ;WRITE GATE
                100000          RDGATE= BIT15           ;READ GATE

                                .SBTTL  DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE A (RKMR2:34)

                000040          D.DRA=  BIT5            ;DRIVE AVAILABLE
                000100          D.VV=   BIT6            ;VOLUME VALID
                000200          D.DRDY= BIT7            ;DRIVE READY
                000400          D.DDT=  BIT8            ;DRIVE TYPE (0=RK06)
                001000          D.FORM= BIT9            ;DRIVE FORMAT
                002000          D.OFF=  BIT10           ;OFFSET ON
                004000          D.WRL=  BIT11           ;WRITE LOCK
                010000          D.SPIN= BIT12           ;SPINDLE ON
                020000          D.PIP=  BIT13           ;POSITIONING IN PROGRESS
                040000          D.DSC=  BIT14           ;DRIVE STATUS CHANGE

                                .SBTTL  DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE A (RKMR2:34)

                000020          D.TFOK= BIT4            ;TRACK FOLLOWING OK
                000040          D.HOHM= BIT5            ;HEADS HOME
                000100          D.BRHM= BIT6            ;BRUSHES HOME
                000200          D.DOOR= BIT7            ;DOOR INTERLOCKED
                000400          D.CART= BIT8            ;CARTRAGE INTERLOCK
                001000          D.SPOK= BIT9            ;SPEED OK
                002000          D.FWD=  BIT10           ;FORWARD
                004000          D.REV=  BIT11           ;REVERSE
                010000          D.LOAD= BIT12           ;HEADS LOADING
                020000          D.RTZ=  BIT13           ;RETURN TO ZERO
                040000          D.UNLD= BIT14           ;HEADS UNLOADING

                                .SBTTL  DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE B (RKMR3:36)

                000040          D.IDAE= BIT5            ;INVALID DISK ADDRESS ERROR:HEAD/CYL
                000100          D.ACLO= BIT6            ;AC LOW
                000200          D.FLT=  BIT7            ;DRIVE FAULT
                000400          D.ILF=  BIT8            ;ILLEGAL FUNCTION CODE
                001000          D.PAR=  BIT9            ;DRIVE DETECTED SERCON PARITY ERROR
                002000          D.SKI=  BIT10           ;SEEK INCOMPLETE
                004000          D.WLE=  BIT11           ;WRITE LOCK ERROR
                010000          D.SPLS= BIT12           ;SPEED LOSS
                020000          D.DROT= BIT13           ;DRIVE OFF TRACK
                040000          D.UNS=  BIT14           ;R/W UNSAFE

                                .SBTTL  DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE B (RKMR3:36)

                000020          D.SECT= BIT4            ;SECTOR ERROR
                000040          D.WCUR= BIT5            ;WRITE CURRENT AND NO WRITE GATE
                000100          D.WGAT= BIT6            ;WRITE GATE AND NO TRANSISTIONS
                000200          D.HDFL= BIT7            ;HEAD FAULT
                000400          D.MHD=  BIT8            ;MULTIPLE HEAD SELECT
                001000          D.XERR= BIT9            ;INDEX ERROR
                002000          D.TIB=  BIT10           ;TRIBIT ERROR
                004000          D.PLO=  BIT11           ;PLO ERROR
```

```
                        D.NMOV= BIT12                   ;SEEK AND NO MOTION
                        D.LIMD= BIT13                   ;LIMIT DETECT ON SEEK
                        D.SUNS= BIT14                   ;SERVO UNSAFE

                        .SBTTL  COMMON MASKS AND OTHER BITS: MESSAGE A (RKMR2:34)

                        M.DRV=  7                       ;DRIVE CODE, ALL BYTES
                        M.CDIF= 17760                   ;CYLINDER DIFF, BYTE 10
                        M.OFST= 17760                   ;OFFSET VALUE, BYTE 10
                        M.SER=  77770                   ;DRIVE SERIAL #, BYTE 11

                        .SBTTL  COMMON MASKS AND OTHER BITS: MESSAGE B (RKMR3:36)

                        M.ID=   3                       ;BYTE ID, ALL BYTES
                        M.CADD= 17760                   ;CYLINDER ADDRESS, BYTE 10
                        M.ALGN= BIT14                   ;ALIGN SIGN, BYTE 10
                        M.SECT= 760                     ;SECTOR COUNT, BYTE 11
                        M.HEAD= 7000                    ;HEAD DECODE, BYTE 11
                        M.PAR=  BIT15                   ;PARITY, MESS A/B, ALL BYTES
```

```
                                .SBTTL  TRAP CATCHER

                        000000          .=0
                                ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
                                ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
                                ;*LOCATION 0 CONTAIN 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  STARTING ADDRESS(ES)
000200  000137 007750          JMP     @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM
                        000204          .=204
000204  000137 007644          JMP     BYT16           ;BYPASS N-SQUARE TEST IN DEFAULT MODE
                        000220          .=220
000220  000137 007624          JMP     PARSRT          ;INPUT ALL PARAMETERS & START TESTING
                        000230          .=230
000230  000137 007664          JMP     BYT16A  ;BYPASS N-SQUARE TEST IN PARAM MODE

                        000240          .=240
000240  000137 072466          JMP     0.ODT           ;ENTER ODT11
                        000260          .=260
000260  000137 007706          JMP     MDTST           ;MODULE TESTS DEFAULT MODE ONLY
                        000270          .=270           ;BYPASS SEVERAL TESTS
000270  000137 007726          JMP     MDTSTA          ;SAME AS 260 & BYPASS N-SQUARE TEST ALSO

                                .SBTTL  ACT11 HOOKS

                                ;;**************************************************************
                                ;HOOKS REQUIRED BY ACT11
                                        $SVPC=.                 ;SAVE PC
                                        .=46
                                        $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOP
                                        .=52
                                        .WORD   100000          ;;2)SET LOC.52 TO 100000
                                        .=$SVPC                 ;; RESTORE PC
                        001000          .=1000
                                .SBTTL  APT PARAMETER BLOCK

                                ;;**************************************************************
                                ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                                ;;**************************************************************
                                        .$X=.   ;;SAVE CURRENT LOCATION
                                        .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
                                        200     ;;FOR APT START UP
                                        .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
                                        $APTHDR ;;POINT TO APT HEADER BLOCK
                                        .=.$X   ;;RESET LOCATION COUNTER
                                ;;**************************************************************
                                ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
                                ;INTERFACE SPEC.

001000                          $APTHD:
001000  000000        $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
001002  001210        $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
001004  000454        $TSTM:  .WORD   300.    ;;RUN TIM OF LONGEST TEST
```

```
   411  001006  001130                $PASTM: .WORD   600.     ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
   412  001010  001130                $UNITM: .WORD   600.     ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
   413  001012  000052                        .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
   414
```

```
                                .SBTTL  COMMON TAGS

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

                        001100          .=1100
001100                  $CMTAG:                         ;;START OF COMMON TAGS
001100  000000                  .WORD   0
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          $TMP1:  .WORD   0               ;;USER DEFINED
001164  000000          $TMP2:  .WORD   0               ;;USER DEFINED
001166  000000          $TMP3:  .WORD   0               ;;USER DEFINED
001170  000000          $TMP4:  .WORD   0               ;;USER DEFINED
001172  000000          $TMP5:  .WORD   0               ;;USER DEFINED
001174  000000          $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
001176  000000          $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
001200  177607  000377  $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
001204  077             $QUES:  .ASCII  /?/             ;;QUESTION MARK
001205  015             $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
001206  000012          $LF:    .ASCIZ  <12>            ;;LINE FEED
                                ;;**************************************************************
                                .SBTTL  APT MAILBOX-ETABLE

                                ;;**************************************************************
                                .EVEN
001210                  $MAIL:                          ;;APT MAILBOX
001210  000000          $MSGTY: .WORD   AMSGTY          ;;MESSAGE TYPE CODE
```

```
001212  000000        $FATAL:  .WORD   AFATAL   ;;FATAL ERROR NUMBER
001214  000000        $TESTN:  .WORD   ATESTN   ;;TEST NUMBER
001216  000000        $PASS:   .WORD   APASS    ;;PASS COUNT
001220  000000        $DEVCT:  .WORD   ADEVCT   ;;DEVICE COUNT
001222  000000        $UNIT:   .WORD   AUNIT    ;;I/O UNIT NUMBER
001224  000000        $MSGAD:  .WORD   AMSGAD   ;;MESSAGE ADDRESS
001226  000000        $MSGLG:  .WORD   AMSGLG   ;;MESSAGE LENGTH
001230                $ETABLE:                  ;;APT ENVIRONMENT TABLE
001230     000        $ENV:    .BYTE   AENV     ;;ENVIRONMENT BYTE
001231     000        $ENVM:   .BYTE   AENVM    ;;ENVIRONMENT MODE BITS
001232  000000        $SWREG:  .WORD   ASWREG   ;;APT SWITCH REGISTER
001234  000000        $USWR:   .WORD   AUSWR    ;;USER SWITCHES
001236  000000        $CPUOP:  .WORD   ACPUOP   ;;CPU TYPE,OPTIONS
                      ;*                        BITS 15-11=CPU TYPE
                      ;*                                11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
                      ;*                                11/70=06,PDQ=07,Q=10
                      ;*                        BIT 10=REAL TIME CLOCK
                      ;*                        BIT  9=FLOATING POINT PROCESSOR
                      ;*                        BIT  8=MEMORY MANAGEMENT
001240     000        $MAMS1:  .BYTE   AMAMS1   ;;HIGH ADDRESS,M.S. BYTE
001241     000        $MTYP1:  .BYTE   AMTYP1   ;;MEM. TYPE,BLK#1
                      ;*                        MEM.TYPE BYTE   --  (HIGH BYTE)
                      ;*                                900 NSEC CORE=001
                      ;*                                300 NSEC BIPOLAR=002
                      ;*                                500 NSEC MOS=003
001242  000000        $MADR1:  .WORD   AMADR1   ;;HIGH ADDRESS,BLK#1
                      ;*                         MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
001244     000        $MAMS2:  .BYTE   AMAMS2   ;;HIGH ADDRESS,M.S. BYTE
001245     000        $MTYP2:  .BYTE   AMTYP2   ;;MEM.TYPE,BLK#2
001246  000000        $MADR2:  .WORD   AMADR2   ;;MEM.LAST ADDRESS,BLK#2
001250     000        $MAMS3:  .BYTE   AMAMS3   ;;HIGH ADDRESS,M.S.BYTE
001251     000        $MTYP3:  .BYTE   AMTYP3   ;;MEM.TYPE,BLK#3
001252  000000        $MADR3:  .WORD   AMADR3   ;;MEM.LAST ADDRESS,BLK#3
001254     000        $MAMS4:  .BYTE   AMAMS4   ;;HIGH ADDRESS,M.S.BYTE
001255     000        $MTYP4:  .BYTE   AMTYP4   ;;MEM.TYPE,BLK#4
001256  000000        $MADR4:  .WORD   AMADR4   ;;MEM.LAST ADDRESS,BLK#4
001260  000000        $VECT1:  .WORD   AVECT1   ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
001262  000000        $VECT2:  .WORD   AVECT2   ;;INTERRUPT VECTOR#2BUS PRIORITY#2
001264  177440        $BASE:   .WORD   ABASE    ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
001266  000000        $DEVM:   .WORD   ADEVM    ;;DEVICE MAP
001270  000000        $CDW1:   .WORD   ACDW1    ;;CONTROLLER DESCRIPTION WORD#1
001272  000000        $CDW2:   .WORD   ACDW2    ;;CONTROLLER DESCRIPTION WORD#2
001274  000000        $DDW0:   .WORD   ADDW0    ;;DEVICE DESCRIPTOR WORD#0
001276  000000        $DDW1:   .WORD   ADDW1    ;;DEVICE DESCRIPTOR WORD#1
001300  000000        $DDW2:   .WORD   ADDW2    ;;DEVICE DESCRIPTOR WORD#2
001302  000000        $DDW3:   .WORD   ADDW3    ;;DEVICE DESCRIPTOR WORD#3
001304  000000        $DDW4:   .WORD   ADDW4    ;;DEVICE DESCRIPTOR WORD#4
001306  000000        $DDW5:   .WORD   ADDW5    ;;DEVICE DESCRIPTOR WORD#5
001310  000000        $DDW6:   .WORD   ADDW6    ;;DEVICE DESCRIPTOR WORD#6
001312  000000        $DDW7:   .WORD   ADDW7    ;;DEVICE DESCRIPTOR WORD#7
001314  000000        $DDW8:   .WORD   ADDW8    ;;DEVICE DESCRIPTOR WORD#8
001316  000000        $DDW9:   .WORD   ADDW9    ;;DEVICE DESCRIPTOR WORD#9
001320  000000        $DDW10:  .WORD   ADDW10   ;;DEVICE DESCRIPTOR WORD#10
001322  000000        $DDW11:  .WORD   ADDW11   ;;DEVICE DESCRIPTOR WORD#11
001324  000000        $DDW12:  .WORD   ADDW12   ;;DEVICE DESCRIPTOR WORD#12
001326  000000        $DDW13:  .WORD   ADDW13   ;;DEVICE DESCRIPTOR WORD#13
```

```
                        000000          $DDW14: .WORD   ADDW14  ;;DEVICE DESCRIPTOR WORD#14
                        000000          $DDW15: .WORD   ADDW15  ;;DEVICE DESCRIPTOR WORD#15


001334                  $ETEND:

                177440  ABASE=  177440          ;DEFAULT BUSS ADDRESS
001334          000210  RKVEC:  210             ;DEFAULT CONTROLLER INTERRUPT VECTOR
001336          000240  RKPRI:  PR5             ;PRIORITY
001340          172540  PKS:    172540          ;P-CLOCK STATUS REG
001342          172542  PKSB:   172542          ;P-CLOCK SET BUFFER
001344          172544  PKRB:   172544          ;P-CLOCK READ BUFFER
001346          177546  LKS:    177546          ;L-CLOCK STATUS REG.

001350          000100  LCVEC:  100             ;L-CLOCK INTERRUPT VECTOR
001352          000104  PCVEC:  104             ;P-CLOCK INTERRUPT VECTOR.

                000114  MEMVEC= 114             ;MEMORY PARITY VECTOR
                172100  MEMBAS= 172100          ;MEMORY PARITY OPTION
001354          000000  TRAPPC: 0               ;PC FOR MEMORY CHECK ENABLE TRAP

001356          000000  PARAM:  0               ;1 FOR 220 OR 230 START, NO DEFAULT
001360          000000  BYPT16: 0               ;1 FOR 210, 230, 270 START
001362          000000  MODTST: 0               ;1 FOR 260 OR 270 START
001364          000000  FTITLE: 0               ;FLAG FOR PRINTING OUT 1ST PROGRAM TITLE

001366          000000  DRVPTR: 0               ;CONTAINS THE POINTER TO THE DRIVE FLAG
                                                ;(DRIV0-DRIV7) OF THE DRIVE TO BE CHECKED NEXT.
001370          000000  FRCYL:  0               ;FROM CYLINDER
001372          000000  TOCYL:  0               ;TO CYLINDER
001374          000000  CCYL:   0               ;CURRENT CYL, USED IN N SQUARE TEST
001376          000000  PCYL:   0               ;PREV CYL., USED IN N SQUARE TEST
001400          000000  CALDIF: 0               ;CALC CYL DIFF USED IN N SQUARE TEST
001402          000000  CYLDIF: 0               ;CYL DIFF, RIGHT JUSTIFIED FROM RKMR3
001404          000000  CYLADD: 0               ;CYL ADDR, RIGHT JUSTIFIED FROM RKMR3
001406          000000  CALADD: 0               ;CYL ADDR USED IN FHDTAB ROUTINE

001410          000074  HZ:     60.             ;60 FOR 60 CPS
                                                ;50 FOR 50 CPS
001412          000000  COUNT:  0               ;LOADED TO 50 OR 60 TO COUNT TO 1 SEC
                                                ;OR ANY OTHER NUMBER TO COUNT OFF FRACTIONAL SECOND
001414          000000  SEC:    0               ;SECOND COUNTER
001416          000000  TIMUP:  0               ;FLAG TO INDICATE TIME IS UP
001420          000000  SECNT:  0               ;SECTOR COUNT
001422          000000  PSEC:   0               ;PREVIOUS SECTOR
001424          000000  ESEC:   0               ;EXPECTED SECTOR
001426          000000  SECTOR: 0               ;SECTOR COUNT, RIGHT JUSTIFIED FROM RKMR3

001430          000000  LPFLG:  0               ;SET TO 0 TO RETURN TO $LPADR
                                                ;IF SW14 OR SW8 SET
                                                ;SET TO 1 TO RETURN TO $LPERR
                                                ;IF SW9 SET
001432          000001  T1:     1               ;TIMEOUT CONSTANTS
001434          000012  T10:    10.
001436          000144  T100:   100.
001440          000764  T500:   500.
```

```
                                T2500:  2500.
                                T5000:  5000.
                                T50000: 50000.

                                HEAD:   0               ;HEAD NUMBER
                                HD1:    0               ;SHIFTED HEAD# FOR FORMATTER ROUTINE
                                FORMAT: 0               ;FORMAT TYPE
                                FMT1:   0               ;SHIFTED FORMAT FOR FORMATTER ROUTINE
                                WDCNT:  0               ;WORD COUNT

                                DATA0:  0               ;ALL 0'S
                                DATA01: 52525           ;0101 PATT
                                DATA1:  177777          ;ALL 1'S

                                WORD:   0               ;HEADER/DATA WORD
                                HDWD:   0               ;HEADER WORD FROM RKDB

                                BSERR:  0               ;CANNOT READ BSE INFO WHEN SET
                                LIMERR: 0               ;LIMIT DETECT ERROR FLAG

                                HDTAB:  .BLKW 66.       ;CALCULATED HEADER WORD TABLE
                                RHTAB:  .BLKW 66.       ;FILLED AFTER READ HEADER CMD
                                SRTTAB: .BLKW 66.       ;ABOVE RHTAB SORTED STARTING FORM
                                                        ;SECTOR 0 BY SORT ROUTINE
                                BSE20:  .BLKW 256.      ;20 SECTOR BSE INFO
                                BSE22:  .BLKW 256.      ;22 SECTOR BSE INFO.

                                INVCYL: 411.            ;INVALID CYLINDER ADDR
                                        412.
                                        416.
                                        448.
                                        480.


004326      001     002     004 ATTN:   .BYTE 1,2,4,10,20,40,100,200    ;ATN 0-7 RESP.
004331      010     020     040
004334      100     200
                                        .EVEN

                                        .LIST   MD
                                ;
                                ;USE LOOP X TO OMIT JSR PC, SUBCLR
                                ;

                                .MACRO  LOOP    A
                                        SCOP1
                                        MOV     #STACK,SP       ;RESTORE STK PTR

                                .IF     B       A
                                        JSR     PC,SUBCLR
                                        ERROR   24              ;CERR AFTER SCLR

                                .ENDC
                                .ENDM   LOOP

                                ;
```

```
                        ;A=MSG A0 ERROR#, B=MSG B0 ERROR#
                        ;C=MSG A1 ERROR#, D=MSG B1 ERROR#
                        ;
                        ;E=<ERROR CONDITION DESCRIPTION>
                        ;F=D.DSC AFTER ATTN OR 0 AFTER DRIVE CLEAR OR ANY IMPLIED SEEKS
                        ;
                        ;       NOTE: F CAN BE ANY BIT COMBINATION DESIRED
                        ;
                        .MACRO  CKWD12  A,B,C,D,E,F

                                MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
                                JSR     PC,GSTAT        ;GET LATEST STATUS
                                MOV     #<F!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2     ;SHOULD BE VALUE
                                JSR     PC,CKMR2        ;CHECK MR2
                                ERROR   A               ;MSG A0 ERROR E
                                CLR     SBMR3
                                JSR     PC,CKMR3        ;CHECK MR3
                                ERROR   B               ;MSG B0 ERROR E

                                MOV     #CCLR,RKCS1(R5)
                                MOV     #1,RKMR1(R5)    ;SELECT WORD 1
                                JSR     PC,GSTAT
                                MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
                                JSR     PC,CKMR2
                                ERROR   C               ;MSG A1 ERROR E
                                CLR     SBMR3
                                JSR     PC,CKMR3
                                ERROR   D               ;MSG B1 ERROR E

                        .ENDM   CKWD12

                        ;
                        ;A=CYL DIFF/OFFSET ERROR#
                        ;B=CYL ADDR ERROR#
                        ;C=<ERROR CONDITION DESCRIPTION>
                        ;
                        .MACRO  CKWD3   A,B,C,?D,?E

                                MOV     #CCLR,RKCS1(R5)
                                JSR     PC,RDCYLD       ;READ CYL DIFF IN RKMR2
                                TST     CYLDIF
                                BEQ     D
                                ERROR   A               ;CYL DIFF/OFFSET NOT CLEARED C
                        D:      JSR     PC,RDCYLA       ;READ CYL ADDR IN RKMR3
                                TST     CYLADD
                                BEQ     E
                                ERROR   B               ;CYL ADDR NOT CLEARED C
                        E:

                        .ENDM   CKWD3


                        .MACRO  LPCHK   ?A
                                CLR     $ESCAPE
                                TST     LPFLG
                                BEQ     A
```

```
                        JMP     @$LPERR         ;SW 9 WAS SET.
A:                      JMP     @$LPADR         ;SW 14 OR 8 WAS SET
.ENDM   LPCHK

.MACRO  SW814
        JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
        SKIP R,<GO TO NEXT TEST>        ;RETURN HERE IF NEITHER IS SET
                                ;RETURN HERE IF SW 14 IS SET OR
                                ;SW 8 WITH SWR <7:0> APPLY
.ENDM   SW814


;SWR9 (LOOP ON ERROR) TEST      A=BRANCH POINT TO RECONDITION DRIVE
;                               B=JMP POINT TO RE-ENTER MAIN LINE
.MACRO  TSTSW9  A,B
        INC     LPFLG
        BIT     #SW9,@SWR       ;LOOP ON ERROR?
        BNE     A               ;YES, RECONDITION DRIVE
        JMP     B               ;RETURN TO MAINLINE
.ENDM   TSTSW9


; USE DRCLR     X    TO OMIT CKWD12
;
.MACRO  DRCLR   A,?C

        MOV     #CCLR,RKCS1(R5)
        MOV     $UNIT,RKCS2(R5) ;DRIVE#
        MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
        MOV     T10,TEMP1
        JSR     PC,FRDY         ;FIND RDY
        ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
        JSR     PC,TSTATN       ;TEST FOR ATTN
        BR      C
        ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
C:
.IF B   A
        CKWD12  273,265,274,265,<AFTER DRIVE CLEAR CMD>,0
.ENDC

.ENDM   DRCLR


;USE CALIB      X   TO OMIT CKWD12 & CKWD3
;
.MACRO  CALIB   A,?C

        MOV     #CCLR,RKCS1(R5)
        MOV     $UNIT,RKCS2(R5)
        MOV     #RECAL,RKCS1(R5)        ;RECAL CMD
                                ;RESET CYL DIFF/OFFSET & CYL ADDR REG
                                ;IN RKMR2 & RKMR3 RESP.
        MOV     T10,TEMP1
        JSR     PC,FRDY         ;FIND RDY
```

```
		ERROR	124		;RDY NOT SET AFTER RECAL CMD

		MOV	#1,RKMR1(R5)	;SELECT WORD 1
		JSR	PC,GSTAT
		BIT	#D.RTZ,HMR2
		BNE	C
		ERROR	244		;RTZ NOT SET DURING RECAL CMD
	C:	MOV	T1,TEMP2	;SETUP TIMEOUT
		JSR	PC,FATT1	;FIND ATTN
		ERROR	55		;NO ATTN AFTER RECAL CMD
	.IF B	A
		CKWD12	221,275,222,276,<AFTER RECAL CMD>,D.DSC
		CKWD3	47,50,<AFTER RECAL CMD>
	.ENDC
		DRCLR

	.ENDM	CALIB

	;
	;IDAE IS CLEARED ONLY BY RECAL & DRIVE CLEAR
	;
	.MACRO	CIDAE	?A

		MOV	#CCLR,RKCS1(R5)
		MOV	$UNIT,RKCS2(R5)
		MOV	#RECAL,RKCS1(R5)		;RECAL CMD
		MOV	T10,TEMP1
		JSR	PC,FRDY		;FIND RDY
		ERROR	124		;RDY NOT FOUND AFTER RECAL CMD
		DRCLR	X

		JSR	PC,GSTAT
		BIT	#D.IDAE,HMR3	;SEE IF IDAE IS CLEARED
		BEQ	A		;BR IF YES
		ERROR	155		;IDAE NOT CLEARED AFTER RECAL CMD

	A:	MOV	#CCLR,RKCS1(R5)
		MOV	T1,TEMP2	;LOOK FOR ATTN FROM RECAL
		JSR	PC,FATT1
		ERROR	55		;NO ATTN AFTER RECAL CMD

	.ENDM	CIDAE

	;
	; A=D.FWD/D.REV
	;
	.MACRO	SKRDY A

		MOV	#SEEK,RKCS1(R5)	;SEEK CMD
		MOV	T10,TEMP1	;SETUP  TIMEOUT
		JSR	PC,FRDY		;FIND RDY
		ERROR	131		;NO RDY AFTER SEEK CMD
		MOV	#<D.PIP!D.SPIN!D.VV!D.DRA>,SBMR2	;LOAD SHOULD BE DATA
		JSR	PC,CKMR2	;CHECK MR2
		ERROR	203		;MSG AO ERROR DURING SEEK CMD
		CLR	SBMR3		;MR3 SHOULD BE 0
```

```
                        JSR     PC,CKMR3        ;CHECK MR3
                        ERROR   204             ;MSG B0 ERROR DURING SEEK CMD

                        MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
                        MOV     #1,RKMR1(R5)    ;SELECT WORD 1
                        JSR     PC,GSTAT
                        MOV     #<A!D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
                        JSR     PC,CKMR2
                        ERROR   205             ;MSG A1 ERROR DURING SEEK CMD
                        CLR     SBMR3
                        JSR     PC,CKMR3        ;CHECK MR3
                        ERROR   206             ;MSG B1 ERROR DURING SEEK CMD

                .ENDM   SKRDY

                .MACRO  SKATN ?A,?B

                        JSR     PC,FATT2        ;FIND ATTN
                        ERROR   132             ;NO ATTN AFTER SEEK CMD
                        BIT     #CERR,HCS1
                        BEQ     A
                        ERROR   210             ;CERR AFTER SEEK CMD
                A:      CKWL12  133,134,135,136,<AFTER SEEK CMD>,D.DSC

                        JSR     PC,RDCYLD       ;READ CYL DIFF IN MSG A2
                        TST     CYLDIF
                        BEQ     B
                        ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD

                B:      DRCLR
                .ENDM   SKATN

                ;
                ;QUICK START SPINDLE.
                ;
                .MACRO  QKSRT

                        JSR     PC,SUBCLR
                        ERROR   24              ;CERR AFTER SCLR

                        MOV     #SRTSPL,RKCS1(R5)       ;START SPINDLE CMD
                        MOV     T10,TEMP1       ;SET TIMEOUT
                        JSR     PC,FRDY         ;FIND RDY
                        ERROR   121             ;RDY NOT FOUND AFTEF ST SPIN CMD.

                        MOV     T100,TEMP2      ;SETUP  TIMEOUT
                        JSR     PC,FATT1        ;FIND ATTN
                        ERROR   67              ;NO ATTN AFTER ST SPIN CMD.

                        TSTSW9  10$,2$
                .ENDM   QKSRT

                ;
                ;QUICK SEEK.    ENTER WITH CYL # IN RKDC
                ;
                .MACRO  QKSEEK  ?A
```

```
                        MOV     #SEEK,RKCS1(R5) ;SEEK CMD TO RECONDITION DRIVE.
                        MOV     T10,TEMP1       ;SETUP TIMEOUT
                        JSR     PC,FRDY         ;FIND RDY
                        ERROR   131             ;NO RDY AFTER SEEK CMD.

                        MOV     T5000,TEMP1
                        JSR     PC,FATT2        ;FIND ATTN
                        ERROR   132             ;NO ATTN AFTER SEEK CMD
                        BIT     #CERR,HCS1
                        BEQ     A
                        ERROR   210             ;CERR AFTER SEEK CMD.

A:                      JSR     PC,SUBCLR
                        ERROR   24              ;CERR AFTER SCLR

.ENDM   QKSEEK

;
;QUICK REPETITIVE SEEKS
;       A=INC/DEC    CYL#
;       B=FIND VALUE OF CYL# BEFORE EXITING

.MACRO  QKRPSK  A,B,?C

                        JSR     PC,SUBCLR
                        ERROR   24              ;CERR AFTER SCRL

C:                      MOV     TOCYL,RKDC(R5)  ;CYL#
                        QKSEEK

                        A       TOCYL
                        CMP     TOCYL,#B        ;ALL CYL DONE?
                        BNE     C

                        JSR     PC,SUBCLR
                        ERROR   24              ;CERR AFTER SCLR

                        LPCHK

.ENDM   QKRPSK

;
;QUICK UNLOAD
;
.MACRO  QKUNLD  ?A,?B

                        JSR     PC,SUBCLR
                        ERROR   24              ;CERR AFTER SCLR

                        MOV     #UNLOAD,RKCS1(R5)       ;UNLOAD CMD TO RECONDITION DRIVE
                        MOV     T10,TEMP1
                        JSR     PC,FRDY         ;FIND RDY
                        ERROR   11              ;RDY NOT SET AFTER UNLOAD CMD.
                        JSR     PC,TSTATN
                        ERROR   12              ;NO ATTN AFTER UNLOAD CMD
```

```
                        JSR     PC,SUBCLR
                        ERROR   24              ;CERR AFTER SCLR

                        MOV     T50000,$TMPO    ;SETUP TIMEOUT
A:                      MOV     #CCLR,RKCS1(R5)
                        MOV     #1,RKMR1(R5)    ;SELECT WORD 1
                        JSR     PC,GSTAT
                        BIT     #D.SPOK,HMR2    ;SEE IF SPEED OK GONE
                        BEQ     B               ;BR IF YES
                        DEC     $TMPO           ;ELSE TRY AGAIN IF TIME NOT UP
                        BNE     A
                        ERROR   315             ;SPEED NOT DOWN BY TIMEOUT

B:                      JSR     PC,SUBCLR
                        ERROR   24              ;CERR AFTER SCLR

                        LPCHK



.ENDM   QKUNLD

;
; A=WRHEAD/<CFMT!WRHEAD>
; USE WRHDR     <A>,X   TO OMIT CKWD12
;
.MACRO  WRHDR   A,C,?D

                        MOV     #<A>,RKCS1(R5)  ;WRITE HEADER CMD
                        MOV     T50000,TEMP1    ;SETUP TIMEOUT
                        JSR     PC,FRDY         ;FIND RDY
                        ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
                        JSR     PC,GSTAT        ;GET FRESH STATUS
                        BIT     #CERR,HCS1
                        BEQ     D
                        ERROR   201             ;CERR AFTER WRITE HEADER CMD
                        TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
                        JMP     $EOP            ;ABORT DRIVE
D:
.IF B   C
                        CKWD12  277,267,300,270,<AFTER WRITE HEADER CMD>,0
.ENDC

.ENDM   WRHDR

;
; A=RDHEAD/<CFMT!RDHEAD>
; USE RDHDR <A>,X TO OMIT CKWD12
;
.MACRO  RDHDR   A,C,?D,?E

                        MOV     #RHTAB,R0
                        MOV     #<A>,RKCS1(R5)  ;READ HEADER CMD
                        MOV     T50000,TEMP1    ;SETUP  TIMEOUT
                        JSR     PC,FRDY         ;FIND RDY
```

```
                        ERROR   171             ;NO RDY AFTER READ HEADER CMD
                        BIT     #CERR,HCS1
                        BEQ     D
                        ERROR   174             ;CERR AFTER READ HEADER CMD
                        TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
                        JMP     $EOP            ;ABORT DRIVE

                D:      MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
                        MOV     RKDB(R5),(R0)+  ;2'ND WORD
                        MOV     RKDB(R5),(R0)+  ;3'RD WORD


                        BIT     #DLT,RKCS2(R5)
                        BEQ     E
                        JSR     PC,GSTAT
                        ERROR   173             ;DLT AFTER READ HEADER CMD
                        TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
                        JMP     $EOP            ;ABORT DRIVE
                E:
                .IF     B       C
                        CKWD12  301,271,302,272,<AFTER READ HEADER CMD>,0
                .ENDC

                .ENDM   RDHDR

                ;
                ;A=TOCYL/FRCYL
                ;B=310 FOR TOCYL/311 FOR FRCYL
                ;

                .MACRO  HDCHK3  A,B,?C

                        RDHDR   RDHEAD,X
                        CMP     RHTAB,A ;CHECK WORD 0 (CYL#) ONLY
                        BEQ     C               ;BR IF SAME
                        ERROR   B               ;READ CYL WORD HEADER ERROR
                C:

                .ENDM   HDCHK3

                .MACRO  RALLHD  ?A,?B,?C,?D,?E

                        MOV     #RHTAB,R0

                A:      MOV     #RDHEAD,RKCS1(R5)       ;READ HEADER CMD
                        MOV     T500,TEMP1      ;SETUP  TIMEOUT
                        JSR     PC,FRDY         ;FIND RDY
                        ERROR   171             ;NO RDY AFTER READ HEADER CMD
                        BIT     #CERR,HCS1
                        BEQ     B
                        ERROR   174             ;CERR AFTER READ HEADER  CMD
                        TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
                        JMP     $EOP            ;ABORT DRIVE

                B:      MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
                        MOV     RKDB(R5),(R0)+  ;2'ND WORD
```

```
1031                                    MOV     RKDB(R5),(R0)+  ;3'RD WORD
1032
1033                                    BIT     #DLT,RKCS2(R5)  ;SEE IF DATA LATE
1034                                    BEQ     C
1035                                    JSR     PC,GSTAT
1036                                    ERROR   173             ;DATA LATE ON READ HEADER
1037                                    TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
1038                                    JMP     $EOP            ;ABORT DRIVE
1039
1040                            C:      CMP     R0,#RHTAB+132.  ;ALL 66 WORDS DONE?
1041                                    BNE     A               ;BR IF NO
1042
1043                                    JSR     PC,SORT         ;SORT RHTAB INTO SRTTAB SO THAT IT
1044                                                            ;BEGINS WITH SECTOR 0
1045                                    CLR     WDCNT           ;WORD COUNT
1046                                    MOV     #SRTTAB,R0      ;ACTUAL HEADER TABLE
1047                                    MOV     #HDTAB,R1       ;CALC HEADER TABLE
1048
1049                            D:      MOV     (R0)+,HDWD
1050                                    MOV     (R1)+,TEMP1
1051                                    CMP     HDWD,TEMP1      ;COMPARE ACTUAL WITH CALCULATED WORD
1052                                    BEQ     E               ;BR IF COMPARE
1053                                    ERROR   202             ;READ HEADER MISMATCH
1054
1055                            E:      INC     WDCNT
1056                                    CMP     WDCNT,#66.      ;ALL WORDS DONE?
1057                                    BNE     D               ;BR IF NO
1058
1059                            .ENDM   RALLHD
1060
1061
1062
1063                                    ;
1064                                    ; A=TOCYL/FRCYL , B=HEAD#, C = 0 FOR 22 SECTOR, 1 FOR 20 SECTOR
1065                                    ;
1066                            .MACRO  HDTBL   A,B,C
1067
1068                                    MOV     A,CALADD        ;SETUP
1069                                    MOV     #B,HEAD         ;TO FILL
1070                                    MOV     #C,FORMAT       ;HEADER
1071                                    JSR     PC,FHDTAB       ;TABLE
1072
1073                            .ENDM   HDTBL
1074                                    ;
1075                                    ;USE FSECA FS022,RDSEC,22 FOR 22 SECTOR FORMAT
1076                                    ;USE FSECA FS020,R20SEC,20 FOR 20 SECTOR FORMAT.
1077                                    ;
1078                            .MACRO  FSECA   A,B,C
1079
1080                                    ;
1081                                    ;FIND SECTOR 0 IN C SECTOR FORMAT.
1082                                    ;ERROR FLAGGED IF NOT FOUND BY TIMEOUT
1083                                    ;
1084                            A:      MOV     TEMP1,-(SP)     ;SAVE TEMP1
1085                                    MOV     T5000,TEMP1     ;SETUP TIMEOUT
1086                            1$:     JSR     PC,B            ;READ SECTOR
```

```
                        TST     SECTOR          ;LOOK FOR SECTOR 0
                        BNE     2$
                        JSR     PC,B
                        TST     SECTOR
                        BEQ     3$              ;BR IF SAME TWICE
                2$:     DEC     TEMP1
                        BNE     1$              ;TRY AGAIN IF TIMEOUT NOT UP
                        MOV     (SP)+,TEMP1     ;ELSE RESTORE TEMP1
                        RTS     PC              ;EXIT
                3$:     MOV     (SP)+,TEMP1
                        ADD     #2,(SP)         ;SKIP OVER ERROR
                        RTS     PC
                .ENDM
                ;
                ;USE FSECB FNS22,RDSEC,22 FOR 22 SECTOR FORMAT
                ;USE FSECB FNS20,R20SEC,20 FOR 20 SECTOR FORMAT
                ;
                .MACRO  FSECB   A,B,C

                ;
                ;FIND NEXT SECTOR IN C SECTOR FORMAT
                ;ERROR FLAGGED IF NOT FOUND BY TIMEOUT
                ;
                A:      MOV     TEMP1,-(SP)     ;SAVE TEMP 1
                        MOV     T500,TEMP1      ;SETUP TIMEOUT
                1$:     JSR     PC,B            ;READ SECTOR
                        CMP     PSEC,SECTOR
                        BEQ     3$              ;BR IF SAME
                        JSR     PC,B            ;ELSE TRY READ DIFFERENT TWICE
                        CMP     PSEC,SECTOR
                        BNE     2$              ;BR IF DIFFERENT TWICE
                3$:     DEC     TEMP1           ;ELSE TRY AGAIN IF TIME LEFT
                        BNE     1$
                        MOV     (SP)+,TEMP1     ;RESTORE TEMP 1
                        RTS     PC
                2$:     MOV     (SP)+,TEMP1     ;RESTORE TEMP 1
                        ADD     #2,(SP)         ;SKIP OVER ERROR
                        RTS     PC
                .ENDM

                ;
                ;USE SECTST FS022,FNS22,RDSEC FOR 22 SECTOR FORMAT
                ;USE SECTST FS020,FNS20,R20SEC FOR 20 SECTOR FORMAT
                ;
                .MACRO  SECTST  D,E,F?A,?B,?C

                        JSR     PC,D            ;FIND SECTOR 0
                        ERROR   142             ;SECTOR 0 NOT FOUND BY TIMEOUT

                        CLR PSEC                ;PREVIOUS SECTOR
                A:      JSR     PC,E            ;FIND NEXT SECTOR
                        ERROR   143             ;DIFFERENT SECTOR  NOT FOUND BY TIMEOUT
                        MOV     PSEC,ESEC
                        ADD     #1,ESEC         ;SETUP EXPECTED SECTOR
                        MOV     SECTOR,PSEC     ;UPDATE PREV SECTOR
                        JSR     PC,F            ;READ SECTOR
```

```
1143                                    CMP     SECTOR,PSEC
1144                                    BEQ     B               ;BR IF READ SAME TWICE
1145                                    JSR     PC,F
1146                                    CMP     SECTOR,PSEC
1147                                    BEQ     B               ;TRY 1 MORE TIME
1148                                    ERROR   144             ;MSG B3 ERROR, SECTOR REG UNSTABLE
1149                                                            ;MAY BE DURING SECTOR PULSE TIME
1150                                B:  CMP     SECTOR,ESEC
1151                                    BEQ     C
1152                                    ERROR   145             ;MSG B3 ERROR BETWEEN SECTOR COUNTS
1153                                C:  DEC     SECNT
1154                                    BNE     A               ;BR IF SECTOR COUNT NOT DONE
1155
1156                                .ENDM   SECTST
1157
1158
1159                                ;
1160                                ;DETECT OUTER LIMIT: 1.0,D.REV,OUTER
1161                                ;DETECT INNER LIMIT: 409.,410.,D.FWD,INNER
1162                                ;
1163                                .MACRO  LIMIT A,B,C,D
1164                                    JSR     PC,SUBCLR       ;SUBSYS CLEAR & GET STATUS
1165                                    ERROR   24              ;CERR AFTER SCLR
1166
1167                                    CLR     LPFLG
1168
1169                                    MOV     #PAT,RKMR1(R5)  ;PARITY & WORD 0
1170                                    MOV     #A,RKDC(R5)     ;CYL A
1171                                    MOV     #SEEK,RKCS1(R5) ;SEEK CMD
1172                                    MOV     T10,TEMP1
1173                                    JSR     PC,FRDY         ;FIND RDY
1174                                    ERROR   122             ;NO RDY FROM SEEK WITH BAD PARITY
1175                                    JSR     PC,TSTATN       ;TEST FOR ATTN
1176                                    ERROR   125             ;NO ATTN FROM SEEK WITH BAD PARITY
1177                                    MOV     #CCLR,RKCS1(R5)
1178                                    JSR     PC,GSTAT
1179                                    MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2
1180                                    JSR     PC,CKMR2        ;CHECK MR2
1181                                    ERROR   110             ;MSG A0 ERROR AFTER SEEK WITH BAD PARITY
1182                                    MOV     #<D.FLT!D.PAR>,SBMR3
1183                                    JSR     PC,CKMR3        ;CHECK MR3
1184                                    ERROR   111             ;MSG B0 ERROR AFTER SEEK WITH BAD PARITY
1185
1186                                    CLR     RKMR1(R5)       ;REMOVE PARITY & SELECT WORD 0
1187                                    DRCLR
1188                                    MOV     #CCLR,RKCS1(R5)
1189                                    JSR     PC,GSTAT
1190                                    MOV     #<D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2
1191                                    JSR     PC,CKMR2        ;CHECK MR2
1192                                    ERROR   152             ;MSG A0 ERROR AFTER CLEAR CMD
1193                                    CLR     SBMR3           ;MR3 SHOULD BE CLEAR
1194                                    JSR     PC,CKMR3        ;CHECK MR3
1195                                    ERROR   153             ;MSG B0 ERROR AFTER CLEAR CMD
1196
1197                                    MOV     #B,RKDC(R5)     ;CYL B
1198                                    MOV     #SEEK,RKCS1(R5) ;SEEK TO CYL B
```

```
1199                                    MOV     T10,TEMP1
1200                                    JSR     PC,FRDY         ;FIND RDY
1201                                    ERROR   131             ;NO RDY AFTER SEEK CMD
1202                                    MOV     #CCLR,RKCS1(R5)
1203                                    JSR     PC,GSTAT
1204                                    JSR     PC,FLIM         ;FIND LIMIT DETECT
1205                                    ERROR   160             ;LIMIT DETECT NOT FOUND BEFORE TIMEOUT
1206
1207                                    BIT     #D.UNLD,HMR2
1208                                    BNE     1$
1209                                    ERROR   305             ;DRIVE NOT UNLOADING AFTER LIMIT DETECT
1210                                    JMP     30$             ;BYPASS REST OF TEST
1211
1212                            1$:     MOV     #20$,$ESCAPE    ;MUST ESCAPE TO CYCLE UP DRIVE & TEST SWR
1213                                    MOV     #CCLR,RKCS1(R5)
1214                                    CLR     RKMR1(R5)       ;SELECT WORD 0
1215                                    JSR     PC,GSTAT
1216                                    MOV     #<D.DSC!D.PIP!D.SPIN!D.VV!D.DRA>,SBMR2
1217                                    JSR     PC,CKMR2        ;CHECK MR2
1218                                    ERROR   161             ;MSG A0 ERROR AFTER D LIMIT DETECT
1219                                    MOV     #<D.SKI!D.FLT>,SBMR3
1220                                    JSR     PC,CKMR3        ;CHECK MR3
1221                                    ERROR   162             ;MSG B0 ERROR AFTER D LIMIT DETECT
1222                                    MOV     #CCLR,RKCS1(R5)
1223                                    MOV     #1,RKMR1(R5)    ;WORD 1
1224                                    JSR     PC,GSTAT
1225                                    MOV     #<D.UNLD!D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
1226                                    JSR     PC,CKMR2        ;CHECK MR2
1227                                    ERROR   163             ;MSG A1 ERROR AFTER D LIMIT DETECT
1228                                    JSR     PC,TSTATN
1229                                    ERROR   165             ;NO ATTN AFTER D LIMIT DETECT
1230
1231                                    JSR     PC,SUBCLR       ;SUBSYS CLR
1232                                    ERROR   24              ;CERR AFTER SCLR
1233                                    MOV     T10,TEMP2       ;SET UP TIMEOUT
1234                                    JSR     PC,FHDHM        ;FIND HEAD HOME
1235                                    ERROR   166             ;HEAD HOME NOT FOUND BEFORE TIMEOUT
1236                                    JSR     PC,FLOAD        ;FIND LOAD HEADS
1237                                    ERROR   167             ;LOAD HEADS NOT FOUND BEFORE TIMEOUT
1238                                    MOV     T10,TEMP2       ;SETUP TIMEOUT
1239                                    JSR     PC,FATT1        ;FIND ATTN
1240                                    ERROR   67              ;ATTN NOT FOUND BEFORE TIMEOUT
1241                            2$:     CLR     $ESCAPE
1242                                    CKWD12  63,64,65,66,<AT END OF HEAD LOADING>,D.DSC
1243                                    CKWD3   175,176,<AT END OF HEAD LOADING>
1244                                    DRCLR
1245                                    SW814
1246
1247
1248                            .ENDM   LIMIT
1249
1250
1251                            ;
1252                            ; A=CYL#, B=HEAD#
1253                            ;
1254                            .MACRO  HEADER  A,B
```

```
1255
1256                                    NEWTST <<WRITE ? READ HEADERS CYLINDER A, HEAD B>>,1
1257                                    MOV     #STACK,SP       ;RESTORE STK PTR
1258
1259                                    JSR     PC,SUBCLR
1260                                    ERROR   24              ;CERR AFTER SCLR
1261                                    MOV     #HDTAB,RKBA(R5) ;HEADER WORD TABLE
1262                                    MOV     #-66.,RKWC(R5)  ;WORD COUNT.
1263                                    MOV     #A,TOCYL
1264                                    HDTBL   TOCYL,0,0
1265                                    MOV     #A,RKDC(R5)     ;CYL#
1266                                    WRHDR   WRHEAD
1267                                    CLR     SECNT           ;SECTOR COUNT
1268                                    LOOP
1269                                    MOV     #A,RKDC(R5)     ;CYL #
1270                                    RALLHD
1271
1272                                .ENDM HEADER
1273
1274
1275
1276                                ;
1277                                ;SEEK TO MAJOR CYL: 0,1,TEMP3,TEMP4,D.FWD,D.REV,ASL,1000,DEC,-1
1278                                ;SEEK 0 TO ALL CYL: 0,1,TEMP3,TEMP4,D.FWD,D.REV,INC,411.,DEC,-1
1279                                ;SEEK 410 TO ALL CYL: 410.,409.,TEMP4,TEMP3,D.REV,D.FWD,DEC,-1,INC,411.
1280                                ;
1281                                .MACRO  SKOSC   A,B,C,D,E,F,G,H,I,J
1282
1283                                    MOV     #A,FRCYL        ;SETUP FROM CYL
1284                                    MOV     #B,TOCYL        ;SETUP TO CYL
1285
1286                                1$: LOOP
1287                                    MOV     #10$,$ESCAPE
1288                                    MOV     FRCYL,TEMP3     ;SETUP
1289                                    MOV     TOCYL,TEMP4     ;CYL DIFF
1290                                    SUB     C,D     ;FOR
1291                                    MOV     D,CALDIF        ;ERROR PRINTOUT
1292
1293                                    MOV     TOCYL,RKDC(R5)  ;GO TO CYL #
1294                                    SKRDY   E
1295                                2$: MOV     #12$,$ESCAPE
1296                                    MOV     T5000,TEMP1     ;SETUP  TIMEOUT
1297                                    SKATN
1298                                    JSR     PC,RDCYLA       ;READ CYL ADDR IN MSG B2
1299                                    CMP     CYLADD,TOCYL
1300                                    BEQ     3$
1301                                    ERROR   207             ;CYL ADDR IN RKMR3 NOT=RKDC
1302
1303                                3$: LOOP
1304                                    CLR     $ESCAPE
1305                                    MOV     TOCYL,RKDC(R5)  ; YL #
1306                                    HDCHK3  TOCYL,310
1307
1308                                    LOOP
1309                                    MOV     #14$,$ESCAPE
1310                                    MOV     FRCYL,RKDC(R5)  ;RETURN TO CYL #
```

```
                                        MOV     FRCYL,CCYL      ;CURRENT CYL FOR TRUERR ROUTINE
                                        SKRDY   F

                                4$:     MOV     #16$,$ESCAPE
                                        MOV     T5000,TEMP1     ;SETUP  TIMEOUT
                                        SKATN
                                        JSR     PC,RDCYLA       ;READ CYL ADDR IN MSG B2
                                        CMP     CYLADD,FRCYL
                                        BEQ     5$
                                        ERROR   243             ;CYL ADDR IN RKMR3 NOT=RKDC

                                5$:     LOOP
                                        CLR     $ESCAPE
                                        MOV     FRCYL,RKDC(R5)  ;CYL #
                                        HDCHK3  FRCYL,311

                                        G       TOCYL
                                        CMP     TOCYL,#H        ;ALL CYL DONE?
                                        BEQ     6$
                                        JMP     1$
                                6$:     SW814
                                8$:     QKRPSK  I,J
                                10$:    TSTSW9  8$,2$
                                12$:    TSTSW9  8$,3$
                                14$:    TSTSW9  8$,4$
                                16$:    TSTSW9  8$,5$
                                .ENDM   SKOSC

                                .MACRO  EOPGM

                                        SCOPE
                                        MOV     #STACK,SP
                                        INC     $DEVCT          ;INCR COUNT FOR # OF DRIVES THAT ARE CHECKED
                                        CMP     DRIVS,$DEVCT    ;ARE ALL DRIVES PRESINT TESTED?
                                        BEQ     $EOP1+2         ;BR IF YES
                                        JMP     NUDRV           ;IF NOT , TEST NEXT DRIVE PRESENT
                                $EOP1:  SCOPE
                                .ENDM   EOPGM

                                        .NLIST  MD


                                ;
                                ;THE FOLLOWING ARE HOLDING REGISTERS FOR THE RK611 REGISTERS
                                ;THEY ARE LOADED AFTER RDY IS REC'D FROM WRDY ROUTINE.
                                ;

 004336  000000                 HCS1:   0                       ;HOLD RKCS1
 004340  000000                 HCS2:   0                       ;HOLD RKCS2
 004342  000000                 HWC:    0                       ;HOLD RKWC
 004344  000000                 HBA:    0                       ;ETC.
 004346  000000                 HDA:    0
 004350  000000                 HDS:    0
 004352  000000                 HER:    0
 004354  000000                 HASOF:  0
 004356  000000                 HDC:    0
```

```
1367  004360  000000                HDB:    0
1368  004362  000000                HMR1:   0
1369  004364  000000                HMR2:   0
1370  004366  000000                HMR3:   0
1371  004370  000000                HPOS:   0
1372  004372  000000                HPAT:   0
1373
1374                                ;
1375                                ;THE FOLLOWING ARE "SHOULD BE" REGISTERS FOR THE ABOVE REGISTERS
1376                                ;
1377  004374  000000                SBMR2:  0                       ;'SHOULD BE' FOR HMR2
1378  004376  000000                SBMR3:  0                       ;ETC
1379
1380  004400  000000                TEMP1:  0                       ;TEMPORARY STORAGE.
1381  004402  000000                TEMP2:  0
1382  004404  000000                TEMP3:  0
1383  004406  000000                TEMP4:  0
1384  004410  000000                TEMP5:  0
1385
1386                                ;
1387                                ;ALL THE FLAGS BELOW ARE CLEARED INITIALLY BY THE CLRFLG ROUTINE.
1388                                ;
1389
1390  004412  000000                DDUMP:  0                       ;FLAG - SET WHEN IN DDP DUMP MODE
1391  004414  000000                DDPCH:  0                       ;FLAG - SET WHEN IN DDP CHAIN MODE
1392  004416  000000                ACT11:  0                       ;FLAG - SET WHEN IN ACT11 MODE OF OPERATION
1393  004420  000000                PPTP:   0                       ;FLAG - SET WHEN PROGRAM LOADED BY PAPER TAPE
1394  004422  000000                DRIVS:  0                       ;CONTAINS THE NUMBER OF DRIVES PRESENT
1395
1396                                ;THE FLAGS BELOW ARE SET TO 1 TO INDICATE THAT A PARTICULAR DRIVE
1397                                ;IS PRESENT AND IS TO BE TESTED.
1398
1399  004424  000000                DRIV0:  0                       ;FLAG SET TO 1 WHEN DRIVE 0 PRESENT
1400  004426  000000                DRIV1:  0                       ;FOR DRIVE 1
1401  004430  000000                DRIV2:  0                       ;FOR DRIVE 2
1402  004432  000000                DRIV3:  0                       ;FOR DRIVE 3
1403  004434  000000                DRIV4:  0                       ;FOR DRIVE 4
1404  004436  000000                DRIV5:  0                       ;FOR DRIVE 5
1405  004440  000000                DRIV6:  0                       ;FOR DRIVE 6
1406  004442  000000                DRIV7:  0                       ;FOR DRIVE 7
1407
1408  004444  000000                LCLKF:  0                       ;L-CLOCK FLAG PRESENT FLAG
1409  004446  000000                PCLKF:  0                       ;P-CLOCK FLAG PRESENT FLAG
1410  004450  000000                DOTIM:  0                       ;SET IF EITHER CLOCK PRESENT FOR TIMING TESTS.
1411  004452  000000                SIZFLG: 0                       ;SET IF DEFAULT DO SIZING IN TEST 1
```

```
                                .SBTTL  ERROR POINTER TABLE

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

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


004454                          $ERRTB:

                                ;ERROR 1
004454  061335                          EM2                     ;DR # IN RKCS2 CANNOT BE READ BACK CORRECTLY IN RKMR2
004456  066504                          DH1
004460  071260                          DT1
004462  071556                          DF1

                                ;ERROR 2
004464  061565                          EM5                     ;DETECTED MDS
004466  066504                          DH1
004470  071260                          DT1
004472  071556                          DF1

                                ;ERROR 3
004474  061606                          EM6                     ;DETECTED UFE
004476  066504                          DH1
004500  071260                          DT1
004502  071556                          DF1

                                ;ERROR 4
004504  061627                          EM7                     ;DETECTED DRA & NED RESET (WRONG PORT SELECTED?)
004506  066504                          DH1
004510  071260                          DT1
004512  071556                          DF1
                                ;ERROR 5
004514  061716                          EM8                     ;DR PRESENT BUT NOT SPECIFIED BY OPERATOR
004516  066504                          DH1
004520  071260                          DT1
004522  071556                          DF1

                                ;ERROR 6
004524  061772                          EM9                     ;DR NOT PRESENT BUT SPECIFIED BY OPERATOR
004526  066504                          DH1
004530  071260                          DT1
004532  071556                          DF1

                                ;ERROR 7
004534  062046                          EM10                    ;ABORT TEST, COULD NOT REFERENCE CONTROLLER REGISTER
004536  066504                          DH1
004540  071260                          DT1
004542  071556                          DF1
```

```
1468
1469                                    ;ERROR 10
1470  004544  062131                        EM11            ;DRA & NED BOTH SET
1471  004546  066504                        DH1
1472  004550  071260                        DT1
1473  004552  071556                        DF1
1474
1475                                    ;ERROR 11
1476  004554  062175                        EM12            ;CONTROLLER NOT READY
1477  004556  067512                        DH18            ;AFTER UNLOAD COMMAND.
1478  004560  071260                        DT1
1479  004562  071716                        DF10
1480
1481                                    ;ERROR 12
1482  004564  062233                        EM13            ;NO ATTN
1483  004566  067512                        DH18            ;AFTER UNLOAD CMD
1484  004570  071304                        DT2
1485  004572  071732                        DF11
1486                                    ;ERROR 13
1487  004574  062255                        EM14            ;WRONG ATTN
1488  004576  067512                        DH18
1489  004600  071304                        DT2
1490  004602  071732                        DF11
1491                                    ;ERROR 14
1492  004604  062302                        EM15            ;DRDY NOT CLEARED
1493  004606  067512                        DH18
1494  004610  071260                        DT1
1495  004612  071716                        DF10
1496                                    ;ERROR 15
1497  004614  062334                        EM16            ;DSC NOT SET
1498  004616  067512                        DH18
1499  004620  071260                        DT1
1500  004622  071716                        DF10
1501                                    ;ERROR 16
1502  004624  062361                        EM17            ;MSG AC ERROR
1503  004626  067004                        DH8             ;IN UNLD
1504  004630  071260                        DT1
1505  004632  071656                        DF8
1506                                    ;ERROR 17
1507  004634  062402                        EM18            ;MSG BO ERROR
1508  004636  067004                        DH8             ;IN UNLD
1509  004640  071260                        DT1
1510  004642  071676                        DF9
1511                                    ;ERROR 20
1512  004644  062423                        EM19            ;MSG A1 ERROR
1513  004646  067004                        DH8             ;IN UNLD
1514  004650  071260                        DT1
1515  004652  071656                        DF8
1516                                    ;ERROR 21
1517  004654  062444                        EM20            ;MSG B1 ERROR
1518  004656  067004                        DH8             ;IN UNLD
1519  004660  071260                        DT1
1520  004662  071676                        DF9
1521                                    ;ERROR 22
1522  004664  064203                        EM46            ;MSG A2 ERROR
1523  004666  067004                        DH8             ;IN UNLD
```

```
1524  004670  071260                    DT1
1525  004672  071656                    DF8
1526                              ;ERROR 23
1527  004674  064224                    EM47          ;MSG B2 ERROR
1528  004676  067004                    DH8           ;IN UNLD
1529  004700  071260                    DT1
1530  004702  071676                    DF9
1531
1532                              ;ERROR 24
1533  004704  062465                    EM21          ;CERR SET
1534  004706  067615                    DH21          ;AFTER SCLR
1535  004710  071260                    DT1
1536  004712  071716                    DF10
1537                              ;ERROR 25
1538  004714  062507                    EM22          ;RLS DID NOT SET CERR
1539  004716  066504                    DH1
1540  004720  071260                    DT1
1541  004722  071556                    DF1
1542
1543                              ;ERROR 26
1544  004724  062546                    EM23          ;SACK SET AFTER RLS SENT
1545  004726  066504                    DH1
1546  004730  071260                    DT1
1547  004732  071556                    DF1
1548
1549                              ;ERROR 27
1550  004734  062631                    EM24          ;VOL VALID NOT SET
1551  004736  067537                    DH19          ;AFTER PACK CMD
1552  004740  071260                    DT1
1553  004742  071716                    DF10
1554                              ;ERROR 30
1555  004744  062664                    EM25          ;DRIVE TYPE SET IN MR2
1556  004746  066504                    DH1
1557  004750  071260                    DT1
1558  004752  071556                    DF1
1559                              ;ERROR 31
1560  004754  062712                    EM26          ;DDT SET IN RKDS
1561  004756  066504                    DH1
1562  004760  071260                    DT1
1563  004762  071556                    DF1
1564                              ;ERROR  32
1565  004764  062732                    EM27          ;DTYE SET IN RKER
1566  004766  066504                    DH1
1567  004770  071260                    DT1
1568  004772  071556                    DF1
1569                              ;ERROR 33
1570  004774  062753                    EM28          ;DTYE NOT SET IN RKER
1571  004776  066504                    DH1
1572  005000  071260                    DT1
1573  005002  071556                    DF1
1574                              ;ERROR 34
1575  005004  063000                    EM29          ;DTYE DID NOT SET CERR
1576  005006  066504                    DH1
1577  005010  071260                    DT1
1578  005012  071556                    DF1
1579                              ;ERROR 35
```

```
1580  005014  063047                  EM30            ;C-D PARITY ERR SET IN MR3
1581  005016  066504                  DH1
1582  005020  071260                  DT1
1583  005022  071556                  DF1
1584                          ;ERROR 36
1585  005024  063103                  EM31            ;SPAR SET IN CS1
1586  005026  066504                  DH1
1587  005030  071260                  DT1
1588  005032  071556                  DF1
1589                          ;ERROR 37
1590  005034  063125                  EM32            ;FAULT NOT SET IN MR3
1591  005036  066504                  DH1
1592  005040  071260                  DT1
1593  005042  071556                  DF1
1594                          ;ERROR 40
1595  005044  063154                  EM33            ;C-D PARITY ERR NOT SET IN MR3
1596  005046  066504                  DH1
1597  005050  071260                  DT1
1598  005052  071556                  DF1
1599                          ;ERROR 41
1600  005054  063214                  EM34            ;SPAR NOT SET IN CS1
1601  005056  066504                  DH1
1602  005060  071260                  DT1
1603  005062  071556                  DF1
1604                          ;ERROR 42
1605  005064  063242                  EM35            ;SPAR DID NOT SET CERR
1606  005066  066504                  DH1
1607  005070  071260                  DT1
1608  005072  071556                  DF1
1609                          ;ERROR 43
1610  005074  063312                  EM36            ;CYL ADDR IN RKMR3 NOT = RKDC
1611  005076  067313                  DH14            ;AFTER SEEK WITH BAD PARITY
1612  005100  071326                  DT3
1613  005102  071616                  DF5
1614                          ;ERROR 44
1615  005104  063355                  EM37            ;CYL DIFF IN RKMR2 NOT=RKDC
1616  005106  067313                  DH14
1617  005110  071326                  DT3
1618  005112  071616                  DF5
1619                          ;ERROR 45
1620  005114  063312                  EM36            ;CYL ADDR IN RKMR3 NOT=RKDC
1621  005116  067313                  DH14
1622  005120  071350                  DT4
1623  005122  071632                  DF6
1624                          ;ERROR 46
1625  005124  063420                  EM38            ;CYL DIFF IN RKMR2 NOT=CALDIF
1626  005126  067313                  DH14
1627  005130  071350                  DT4
1628  005132  071632                  DF6
1629                          ;ERROR 47
1630  005134  063471                  EM39            ;CYL DIFF/OFFSET IN RKMR2 NOT CLEARED
1631  005136  067466                  DH17            ;AFTER RECAL CMD
1632  005140  071260                  DT1
1633  005142  071716                  DF10
1634                          ;ERROR 50
1635  005144  063540                  EM40            ;CYL ADDR IN RKMR3 NOT CLEARED
```

```
1636  005146  067466                  DH17                    ;AFTER RECAL COMD
1637  005150  071260                  DT1
1638  005152  071716                  DF10
1639                          ;ERROR 51
1640  005154  063576                  EM41                    ;CYL ADDR IN RKMR3 DID NOT REMAIN 0
1641  005156  067346                  DH15                    ;AFTER OFFSET CMD WITH BAD PARITY
1642  005160  071226                  DT3
1643  005162  071616                  DF5
1644                          ;ERROR 52
1645  005164  063647                  EM42                    ;OFFSET IN RKMR2 NOT=RKASOF
1646  005166  066504                  DH1
1647  005170  071304                  DT2
1648  005172  071566                  DF2
1649                          ;ERROR 53
1650  005174  063723                  EM43                    ;HEAD DECODE IN MR3 NOT CLEARED
1651  005176  067466                  DH17                    ;AFTER RECAL CMD
1652  005200  071260                  DT1
1653  005202  071716                  DF10
1654                          ;ERROR 54
1655  005204  063762                  EM44                    ;RKMR3 HEAD DECODE NOT=RKDA
1656  005206  066504                  DH1
1657  005210  071370                  DT5
1658  005212  071646                  DF7
1659                          ;ERROR 55
1660  005214  062233                  EM13                    ;NO ATTN
1661  005216  067466                  DH17                    ;AFTER RECAL CMD
1662  005220  071260                  DT1
1663  005222  071716                  DF10
1664                          ;ERROR 56
1665  005224  065435                  EM64                    ;MSG B3 HEAD REG NOT CLEARED IN UNLOAD
1666  005226  066504                  DH1
1667  005230  071260                  DT1
1668  005232  071556                  DF1
1669                          ;ERROR 57
1670  005234  062361                  EM17                    ;MSG A0 ERROR
1671  005236  067030                  DH9                     ;AFTER START SPINDLE CMD REC'D BY DRIVE
1672  005240  071260                  DT1
1673  005242  071656                  DF8
1674                          ;ERROR 60
1675  005244  062402                  EM18                    ;MSG B0 ERROR
1676  005246  067030                  DH9
1677  005250  071260                  DT1
1678  005252  071676                  DF9
1679                          ;ERROR 61
1680  005254  062423                  EM19                    ;MSG A1 ERROR
1681  005256  067030                  DH9
1682  005260  071260                  DT1
1683  005262  071656                  DF8
1684                          ;ERROR 62
1685  005264  000044                  EM20                    ;MSG B1 ERROR
1686  005266  000030                  DH9
1687  005270  071260                  DT1
1688  005272  071676                  DF9
1689                          ;ERROR 63
1690  005274  062361                  EM17
1691  005276  067103                  DH10                    ;AT END OF HEAD LOADING
```

```
1692  005300  071260                    DF1
1693  005302  071656                    DF8
1694                            ;ERROR 64
1695  005304  062402                    EM18
1696  005306  067103                    DH10
1697  005310  071260                    DT1
1698  005312  071676                    DF9
1699                            ;ERROR 65
1700  005314  062423                    EM19
1701  005316  067103                    DH10
1702  005320  071260                    DT1
1703  005322  071656                    DF8
1704                            ;ERROR 66
1705  005324  062444                    EM20
1706  005326  067103                    DH10
1707  005330  071260                    DT1
1708  005332  071676                    DF9
1709                            ;ERROR 67
1710  005334  062233                    EM13                ;NO ATTN
1711  005336  067103                    DH10                ;AT END OF HEAD LOADING.
1712  005340  071260                    DT1
1713  005342  071716                    DF10
1714                            ;ERROR 70
1715  005344  064403                    EM50                ;FWD NOT SET WITHIN 60 SEC FROM
1716  005346  066504                    DH1                 ;START SPINDLE CMD.
1717  005350  071260                    DT1
1718  005352  071556                    DF1
1719                            ;ERROR 71
1720  005354  062361                    EM17
1721  005356  067132                    DH11                ;AFTER START SPINDLE CMD & FWD SET.
1722  005360  071260                    DT1
1723  005362  071656                    DF8
1724                            ;ERROR 72
1725  005364  062402                    EM18
1726  005366  067132                    DH11
1727  005370  071260                    DT1
1728  005372  071676                    DF9
1729                            ;ERROR 73
1730  005374  062423                    EM19
1731  005376  067132                    DH11
1732  005400  071260                    DT1
1733  005402  071656                    DF8
1734                            ;ERROR 74
1735  005404  062444                    EM20
1736  005406  067132                    DH11
1737  005410  071260                    DT1
1738  005412  071676                    DF9
1739                            ;ERROR 75
1740  005414  064504                    EM51                ;FWD NOT CLEARED WITHIN 5 SEC OF MOTION
1741  005416  066504                    DH1                 ;FROM START SPINDLE CMD.
1742  005420  071260                    DT1
1743  005422  071556                    DF1
1744                            ;ERROR 76
1745  005424  062361                    EM17
1746  005426  067174                    DH12                ;AT INNER LIMIT FROM START SPINDLE CMD.
1747  005430  071260                    DT1
```

```
 1748  005432  071656                        DF8
 1749                                ;ERROR 77
 1750  005434  062402                        EM18
 1751  005436  067174                        DH12
 1752  005440  071260                        DT1
 1753  005442  071676                        DF9
 1754                                ;ERROR 100
 1755  005444  062423                        EM19
 1756  005446  067174                        DH12
 1757  005450  071260                        DT1
 1758  005452  071656                        DF8
 1759                                ;ERROR 101
 1760  005454  062444                        EM20
 1761  005456  067174                        DH12
 1762  005460  071260                        DT1
 1763  005462  071676                        DF9
 1764                                ;ERROR 102
 1765  005464  064266                        EM49                    ;FWD NOT SET WITHIN 4 SEC IN RTZ PORTION
 1766  005466  066504                        DH1                     ;OF START SPIN CMD.
 1767  005470  071260                        DT1
 1768  005472  071556                        DF1
 1769                                ;ERROR 103
 1770  005474  062361                        EM17
 1771  005476  067242                        DH13                    ;FROM OUTER LIMIT TO CYL 0 DURING LOADING
 1772  005500  071260                        DT1
 1773  005502  071656                        DF8
 1774                                ;ERROR 104
 1775  005504  062402                        EM18
 1776  005506  067242                        DH13
 1777  005510  071260                        DT1
 1778  005512  071676                        DF9
 1779                                ;ERROR 105
 1780  005514  062423                        EM19
 1781  005516  067242                        DH13
 1782  005520  071260                        DT1
 1783  005522  071656                        DF8
 1784                                ;ERROR 106
 1785  005524  062444                        EM20
 1786  005526  067242                        DH13
 1787  005530  071260                        DT1
 1788  005532  071676                        DF9
 1789                                ;ERROR 107
 1790  005534  064044                        EM45                    ;DRIVE READY NOT SET WITHIN 1 SEC
 1791  005536  066504                        DH1                     ;FROM FWD IN RTZ PORTION OF START SPIN CMD.
 1792  005540  071260                        DT1
 1793  005542  071556                        DF1
 1794                                ;ERROR 110
 1795  005544  062361                        EM17                    ;MSG A0 ERROR
 1796  005546  067313                        DH14                    ;AFTER SEEK WITH BAD PARITY
 1797  005550  071260                        DT1
 1798  005552  071656                        DF8
 1799                                ;ERROR 111
 1800  005554  062402                        EM18                    ;MSG B0 ERROR
 1801  005556  067313                        DH14                    ;AFTER SEEK WITH BAD PARITY
 1802  005560  071260                        DT1
 1803  005562  071676                        DF9
```

```
 1804                                   ;ERROR 112
 1805  005564  062361                          EM17
 1806  005566  067346                          DH15          ;AFTER OFFSET WITH BAD PARITY
 1807  005570  071260                          DT1
 1808  005572  071656                          DF8
 1809                                   ;ERROR 113
 1810  005574  062402                          EM18
 1811  005576  067346                          DH15
 1812  005600  071260                          DT1
 1813  005602  071676                          DF9
 1814                                   ;ERROR 114
 1815  005604  062361                          EM17
 1816  005606  067413                          DH16          ;AFTER LOADING HEAD REGISTER & SEEK CMD
 1817  005610  071370                          DT5
 1818  005612  072056                          DF18
 1819                                   ;ERROR 115
 1820  005614  062402                          EM18
 1821  005616  067413                          DH16
 1822  005620  071370                          DT5
 1823  005622  072076                          DF19
 1824                                   ;ERROR 116
 1825  005624  062175                          EM12          ;CONT NOT RDY
 1826  005626  067537                          DH19          ;AFTER PACK CMD
 1827  005630  071260                          DT1
 1828  005632  071716                          DF10
 1829                                   ;ERROR 117
 1830  005634  062175                          EM12          ;CONT NOT RDY
 1831  005636  067562                          DH20          ;AFTER SEL DR CMD
 1832  005640  071260                          DT1
 1833  005642  071716                          DF10
 1834                                   ;ERROR 120
 1835  005644  062175                          EM12
 1836  005646  067615                          DH21          ;AFTER SUBSYS CLEAR
 1837  005650  071260                          DT1
 1838  005652  071716                          DF10
 1839                                   ;ERROR 121
 1840  005654  062175                          EM12
 1841  005656  067030                          DH9           ;AFTER START SPINDLE CMD
 1842  005660  071260                          DT1
 1843  005662  071716                          DF10
 1844                                   ;ERROR 122
 1845  005664  062175                          EM12
 1846  005666  067313                          DH14          ;AFTER SEEK WITH BAD PARITY
 1847  005670  071260                          DT1
 1848  005672  071716                          DF10
 1849                                   ;ERROR 123
 1850  005674  062175                          EM12
 1851  005676  067346                          DH15          ;AFTER OFFSET WITH BAD PARITY
 1852  005700  071304                          DT2
 1853  005702  071732                          DF11
 1854                                   ;ERROR 124
 1855  005704  062175                          EM12
 1856  005706  067466                          DH17          ;AFTER RECAL CMD
 1857  005710  071260                          DT1
 1858  005712  071716                          DF10
 1859                                   ;ERROR 125
```

```
1860  005714  062233                  EM13                    ;NO ATTN
1861  005716  067313                  DH14                    ;FROM SEEK WITH BAD PARITY
1862  005720  071304                  DT2
1863  005722  071732                  DF11
1864                          ;ERROR 126
1865  005724  062233                  EM13                    ;NO ATTN
1866  005726  067346                  DH15                    ;AFTER OFFSET WITH BAD PARITY
1867  005730  071304                  DT2
1868  005732  071732                  DF11
1869                          ;ERROR 127
1870  005734  063471                  EM39                    ;CYL DIFF/OFFSET IN RKMR2 NOT CLEARED
1871  005736  067615                  DH21                    ;AFTER SCLR
1872  005740  071260                  DT1
1873  005742  071716                  DF10
1874                          ;ERROR 130
1875  005744  063540                  EM40                    ;CYL ADDR IN RKMR3 NOT CLEARED
1876  005746  067615                  DH21
1877  005750  071260                  DT1
1878  005752  071716                  DF10
1879                          ;ERROR 131
1880  005754  062175                  EM12                    ;NO RDY
1881  005756  067743                  DH25                    ;AFTER SEEK CMD
1882  005760  071260                  DT1
1883  005762  071716                  DF10
1884                          ;ERROR 132
1885  005764  062233                  EM13                    ;NO ATTN
1886  005766  067743                  DH25
1887  005770  071260                  DT1
1888  005772  071716                  DF10
1889                          ;ERROR 133
1890  005774  062361                  EM17                    ;MSG A0 ERROR
1891  005776  067743                  DH25
1892  006000  071260                  DT1
1893  006002  071656                  DF8
1894                          ;ERROR 134
1895  006004  062402                  EM18                    ;MSG B0 ERROR
1896  006006  067743                  DH25
1897  006010  071260                  DT1
1898  006012  071676                  DF9
1899                          ;ERROR 135
1900  006014  062423                  EM19                    ;MSG A1 ERROR
1901  006016  067743                  DH25
1902  006020  071260                  DT1
1903  006022  071656                  DF8
1904                          ;ERROR 136
1905  006024  062444                  EM20                    ;MSG B1 ERROR.
1906  006026  067743                  DH25
1907  006030  071260                  DT1
1908  006032  071676                  DF9
1909                          ;ERROR 137
1910  006034  063471                  EM39                    ;CYL DIFF/OFFSET IN RKMR2 NOT CLEARED
1911  006036  067743                  DH25
1912  006040  071260                  DT1
1913  006042  071716                  DF10
1914                          ;ERROR 140
1915  006044  063540                  EM40                    ;CYL ADDR IN RKMR3 NOT CLEARED
```

```
1916  006046  067743                  DH25
1917  006050  071260                  DT1
1918  006052  071716                  DF10
1919                          ;ERROR 141
1920  006054  064622                  EM52            ;20 SECTOR FORMAT NOT SET IN RKMR2
1921  006056  066504                  DH1
1922  006060  071260                  DT1
1923  006062  071556                  DF1
1924                          ;ERROR 142
1925  006064  064664                  EM53            ;SECTOR 0 NOT FOUND WITHIN 50 MS
1926  006066  066504                  DH1
1927  006070  071260                  DT1
1928  006072  071556                  DF1
1929                          ;ERROR 143
1930  006074  064724                  EM54            ;DIFF SECTOR NOT FOUND WITHIN 3MS
1931  006076  066504                  DH1
1932  006100  071260                  DT1
1933  006102  071556                  DF1
1934                          ;ERROR 144
1935  006104  064245                  EM48            ;MSG B3 ERROR
1936  006106  070167                  DH34            ;SECTOR REG UNSTABLE
1937  006110  071260                  DT1
1938  006112  071716                  DF10
1939                          ;ERROR 145
1940  006114  064245                  EM48
1941  006116  070213                  DH35            ;BETWEEN SECTOR COUNTS
1942  006120  071416                  DT6
1943  006122  071752                  DF12
1944                          ;ERROR 146
1945  006124  062423                  EM19            ;MSG A1 ERROR
1946  006126  067313                  DH14            ;AFTER SEEK WITH BAD PARITY
1947  006130  071260                  DT1
1948  006132  071656                  DF8
1949                          ;ERROR 147
1950  006134  062444                  EM20            ;MSG B1 ERROR
1951  006136  067313                  DH14
1952  006140  071260                  DT1
1953  006142  071676                  DF9
1954                          ;ERROR 150
1955  006144  062423                  EM19            ;MSG A1 ERROR
1956  006146  070323                  DH37
1957  006150  071260                  DT1
1958  006152  071716                  DF10
1959                          ;ERROR 151
1960  006154  062175                  EM12            ;NO RDY
1961  006156  067643                  DH22            ;AFTER CLEAR CMD
1962  006160  071260                  DT1
1963  006162  071716                  DF10
1964                          ;ERROR 152
1965  006164  062361                  EM17            ;MSG A0 ERROR
1966  006166  067643                  DH22
1967  006170  071260                  DT1
1968  006172  071656                  DF8
1969                          ;ERROR 153
1970  006174  062402                  EM18            ;MSG B0 ERROR
1971  006176  067643                  DH22
```

```
1972  006200  071260                    DF1
1973  006202  071676                    DF9
1974                            ;ERROR 154
1975  006204  064766                    EM55                    ;ATTN NOT CLEARED
1976  006206  067643                    DH22
1977  006210  071304                    DT2
1978  006212  071732                    DF11
1979                            ;ERR 155
1980  006214  066374                    EM85                    ;IDAE NOT CLEARED
1981  006216  067466                    DH17                    ;AFTER RECAL CMD
1982  006220  071260                    DT1
1983  006222  071716                    DF10
1984                            ;ERR 156
1985  006224  062175                    EM12                    ;CONT NOT READY
1986  006226  071153                    DH51                    ;AFTER SEEK TO SELF
1987  006230  071260                    DT1
1988  006232  071716                    DF10
1989                            ;ERR 157
1990  006234  062233                    EM13                    ;NO ATTN
1991  006236  071153                    DH51
1992  006240  071304                    DT2
1993  006242  071732                    DF11
1994                            ;ERROR 160
1995  006244  065240                    EM59                    ;LIMIT DETECT NOT FOUND
1996  006246  066504                    DH1
1997  006250  071260                    DT1
1998  006252  071556                    DF1
1999                            ;ERROR 161
2000  006254  062361                    EM17                    ;MSG A0 ERROR
2001  006256  070374                    DH38                    ;AFTER LIMIT DETECT
2002  006260  071260                    DT1
2003  006262  071656                    DF8
2004                            ;ERROR 162
2005  006264  062402                    EM18                    ;MSG B0 ERROR
2006  006266  070374                    DH38
2007  006270  071260                    DT1
2008  006272  071676                    DF9
2009                            ;ERROR 163
2010  006274  062423                    EM19                    ;MSG A1 ERROR
2011  006276  070374                    DH38
2012  006300  071260                    DT1
2013  006302  071656                    DF8
2014                            ;ERROR 164
2015  006304  062444                    EM20                    ;MSG B1 ERROR
2016  006306  070374                    DH38
2017  006310  071260                    DT1
2018  006312  071676                    DF9
2019                            ;ERROR 165
2020  006314  062233                    EM13                    ;NO ATTN
2021  006316  070374                    DH38
2022  006320  071304                    DT2
2023  006322  071732                    DF11
2024                            ;ERROR 166
2025  006324  065300                    EM60                    ;HEADS HOME NOT FOUND
2026  006326  070374                    DH38
2027  006330  071260                    DT1
```

```
 2028  006332  071716                        DF10
 2029                          ;ERROR 167
 2030  006334  065336                        EM61            ;LOAD HEADS NOT FOUND
 2031  006336  070374                        DH38
 2032  006340  071260                        DT1
 2033  006342  071716                        DF10
 2034                          ;ERROR 170
 2035  006344  065731                        EM72            ;FORMAT TEST BYPASSED
 2036  006346  070754                        DH45            ;LIMIT DETECT ERROR ON PREVIOUS TEST
 2037  006350  071260                        DT1
 2038  006352  072026                        DF16
 2039                          ;ERROR 171
 2040  006354  062175                        EM12            ;NO RDY
 2041  006356  070047                        DH30            ;AFTER READ HEADER CMD
 2042  006360  071260                        DT1
 2043  006362  071716                        DF10
 2044                          ;ERROR 172
 2045  006364  063471                        EM39            ;CYL DIFF/OFFSET NOT CLEARED
 2046  006366  070047                        DH30            ;AFTER READ HEADER CMD
 2047  006370  071260                        DT1
 2048  006372  071716                        DF10
 2049                          ;ERROR 173
 2050  006374  065414                        EM63            ;DLT SET
 2051  006376  070047                        DH30
 2052  006400  071370                        DT5
 2053  006402  072012                        DF15
 2054                          ;ERROR 174
 2055  006404  062465                        EM21            ;CERR SET
 2056  006406  070047                        DH30
 2057  006410  071370                        DT5
 2058  006412  072012                        DF15
 2059                          ;ERROR 175
 2060  006414  063471                        EM39            ;CYL DIFF NOT CLEARED
 2061  006416  067103                        DH10            ;AT END OF HEAD LOADING
 2062  006420  071260                        DT1
 2063  006422  071716                        DF10
 2064                          ;ERROR 176
 2065  006424  063540                        EM40            ;CYL ADDR NOT CLEARED.
 2066  006426  067103                        DH10
 2067  006430  071260                        DT1
 2068  006432  071716                        DF10
 2069                          ;ERROR 1
 2070  006434  065731                        EM72            ;FORMAT TEST BYPASSED
 2071  006436  071033                        DH46            ;COULD NOT READ BSE INFO
 2072  006440  071260                        DT1
 2073  006442  072026                        DF16
 2074                          ;ERROR 200
 2075  006444  062175                        EM12            ;NO RDY
 2076  006446  070417                        DH39            ;AFTER WRITE HEADER CMD
 2077  006450  071370                        DT5
 2078  006452  071766                        DF13
 2079                          ;ERROR 201
 2080  006454  062465                        EM21            ;CERR SET
 2081  006456  070417                        DH39
 2082  006460  071370                        DT5
 2083  006462  071766                        DF13
```

```
2084
2085  006464  065503          ;ERROR 202
2086  006466  066504                  EM65                    ;READ HEADER ERROR
2087  006470  071436                  DH1
2088  006472  072002                  DT7
2089                                   DF14
2090  006474  062361          ;ERROR 203
2091  006476  070143                  EM17                    ;MSG A0 ERROR
2092  006500  071260                  DH33                    ;DURING SEEK CMD
2093  006502  071656                  DT1
2094                                   DF8
2095  006504  062402          ;ERROR 204
2096  006506  070143                  EM18                    ;MSG B0 ERROR
2097  006510  071260                  DH33
2098  006512  071676                  DT1
2099                                   DF9
2100  006514  062423          ;ERROR 205
2101  006516  070143                  EM19                    ;MSG A1 ERROR
2102  006520  071260                  DH33
2103  006522  071656                  DT1
2104                                   DF8
2105  006524  062444          ;ERROR 206
2106  006526  070143                  EM20                    ;MSG B1 ERROR
2107  006530  071260                  DH33
2108  006532  071676                  DT1
2109                                   DF9
2110  006534  063312          ;ERROR 207
2111  006536  067743                  EM36                    ;CYL ADDR IN RKMR3 INCORRECT
2112  006540  071350                  DH25                    ;AFTER SEEK CMD
2113  006542  071632                  DT4
2114                                   DF6
2115  006544  062465          ;ERROR 210
2116  006546  067743                  EM21                    ;CERR SET
2117  006550  071260                  DH25
2118  006552  071716                  DT1
2119                                   DF10
2120  006554  065561          ;ERROR 211
2121  006556  067743                  EM67                    ;READ CYL 0 HEADERS ON CYL 1
2122  006560  071260                  DH25
2123  006562  071716                  DT1
2124                                   DF10
2125  006564  063420          ;ERROR 212
2126  006566  070143                  EM38                    ;CYL DIFF IN RKMR2 NOT = CALDIF
2127  006570  071350                  DH33                    ;DURING SEEK CMD
2128  006572  071632                  DT4
2129                                   DF6
2130  006574  062361          ;ERROR 213
2131  006576  070527                  EM17                    ;MSG A0 ERROR
2132  006600  071260                  DH41                    ;DURING RECAL CMD
2133  006602  071656                  DT1
2134                                   DF8
2135  006604  062402          ;ERROR 214
2136  006606  070527                  EM18                    ;MSG B0 ERROR
2137  006610  071260                  DH41
2138  006612  071676                  DT1
2139                                   DF9
                                ;ERROR 215
```

```
 2140  006614  062423          EM19                    ;MSG A1 ERROR
 2141  006616  070527          DH41
 2142  006620  071260          DT1
 2143  006622  071656          DF8
 2144                  ;ERROR 216
 2145  006624  062444          EM20                    ;MSG B1 ERROR
 2146  006626  070527          DH41
 2147  006630  071260          DT1
 2148  006632  071676          DF9
 2149                  ;ERROR 217
 2150  006634  063420          EM38                    ;CYL DIFF IN RKMR2 NOT=CALDIF
 2151  006636  070527          DH41
 2152  006640  071350          DT4
 2153  006642  071632          DF6
 2154                  ;ERROR 220
 2155  006644  062465          EM21                    ;CERR SET
 2156  006646  067466          DH17                    ;AFTER RECAL CMD
 2157  006650  071260          DT1
 2158  006652  071716          DF10
 2159                  ;ERROR 221
 2160  006654  062361          EM17                    ;MSG A0 ERROR
 2161  006656  067466          DH17
 2162  006660  071260          DT1
 2163  006662  071656          DF8
 2164                  ;ERROR 222
 2165  006664  062423          EM19                    ;MSG A1 ERROR
 2166  006666  067466          DH17
 2167  006670  071260          DT1
 2168  006672  071656          DF8
 2169                  ;ERROR 223
 2170  006674  063471          EM39                    ;CYL DIFF/OFFSET IN RKMR2 NOT CLEARED
 2171  006676  067466          DH17
 2172  006700  071260          DT1
 2173  006702  071716          DF10
 2174                  ;ERROR 224
 2175  006704  065525          EM66                    ;CYL ADDR IN RKMR3 INCORRECT
 2176  006706  067466          DH17
 2177  006710  071260          DT1
 2178  006712  071716          DF10
 2179                  ;ERROR 225
 2180  006714  065620          EM68                    ;READING CYL 1 HEADERS ON CYL 0
 2181  006716  067466          DH17
 2182  006720  071260          DT1
 2183  006722  071716          DF10
 2184                  ;ERROR 226
 2185  006724  062175          EM12                    ;NO RDY
 2186  006726  067766          DH26                    ;AFTER READ DATA CMD
 2187  006730  071260          DT1
 2188  006732  071716          DF10
 2189                  ;ERROR 227
 2190  006734  062465          EM21                    ;CERR SET
 2191  006736  067766          DH26
 2192  006740  071370          DT5
 2193  006742  072012          DF15
 2194                  ;ERROR 230
 2195  006744  065374          EM62                    ;DTE SET
```

```
2196  006746  067766                  DH26
2197  006750  071370                  DT5
2198  006752  072012                  DF15
2199                          ;ERROR 231
2200  006754  065414                  EM63                    ;DLT SET
2201  006756  067766                  DH26
2202  006760  071370                  DT5
2203  006762  072012                  DF15
2204                          ;ERROR 232
2205  006764  063312                  EM36                    ;CYL ADDR IN RKMR3 NOT=RKDC
2206  006766  067743                  DH25                    ;AFTER SEEK CMD
2207  006770  071326                  DT3
2208  006772  071616                  DF5
2209                          ;ERROR 233
2210  006774  066441                  EM87                    ;CANNOT READ BSE INFO
2211  006776  070554                  DH42                    ;ON SECT 0,2,4,6,8
2212  007000  071260                  DT1
2213  007002  072036                  DF17
2214                          ;ERROR 234
2215  007004  066441                  EM87
2216  007006  070625                  DH43                    ;ON SECT 1,3,5,7,9
2217  007010  071260                  DT1
2218  007012  072036                  DF17
2219                          ;ERROR 235
2220  007014  065657                  EM69                    ;ALIGN CARTRIDGE USED
2221  007016  070675                  DH44                    ;WILL BYPASS FORMAT & ALL R/W TESTS
2222  007020  071260                  DT1
2223  007022  071716                  DF15
2224                          ;ERROR 236
2225  007024  065021                  EM56                    ;UNEXP MEM PARITY TRAP
2226  007026  067675                  DH23                    ;TEST #, TRAP PC
2227  007030  071532                  DT11
2228  007032  071602                  DF3
2229                          ;ERROR 237
2230  007034  065710                  EM71                    ;DSC SET
2231  007036  067643                  DH22                    ;AFTER DRIVE CLEAR CMD
2232  007040  071304                  DT2
2233  007042  071732                  DF11
2234                          ;ERROR 240
2235  007044  065620                  EM68                    ;READ CYL 1 HEADERS ON CYL 0
2236  007046  067466                  DH17                    ;AFTER RECAL CMD
2237  007050  071260                  DT1
2238  007052  071716                  DF10
2239                          ;ERROR 241
2240  007054  063312                  EM36                    ;RKMR3 NOT = RKDC
2241  007056  067313                  DH14                    ;AFTER SEEK WITH BAD PARITY
2242  007060  071456                  DT8
2243  007062  071632                  DF6
2244                          ;ERROR 242
2245  007064  063420                  EM38                    ;CYL DIFF IN RKMR2 INCORRECT
2246  007066  067313                  DH14
2247  007070  071456                  DT8
2248  007072  071632                  DF6
2249
2250                          ;ERROR 243
2251  007074  063312                  EM36                    ;CYL ADDR IN RKMR3 INCORRECT
```

```
 2252  007076  067743                DH25                    ;AFTER SEEK CMD
 2253  007100  071456                DT8
 2254  007102  071632                DF6
 2255                          ;ERR 244
 2256  007104  065776                EM74                    ;RTZ NOT SET
 2257  007106  070527                DH41                    ;DURING RECAL CMD
 2258  007110  071260                DT1
 2259  007112  071716                DF10
 2260                          ;ERR 245
 2261  007114  062233                EM13                    ;NO ATTN
 2262  007116  071104                DH48                    ;AFTER SEEK TO INVALID CYL
 2263  007120  071304                DT2
 2264  007122  071732                DF11
 2265                          ;ERR 246
 2266  007124  066003                EM75                    ;IDAE NOT SET
 2267  007126  071104                DH48
 2268  007130  071350                DT4
 2269  007132  071632                DF6
 2270                          ;ERR 247
 2271  007134  063125                EM32                    ;FAULT NOT SET
 2272  007136  071104                DH48
 2273  007140  071350                DT4
 2274  007142  071632                DF6
 2275                          ;ERR 250
 2276  007144  066031                EM76                    ;PIP SET
 2277  007146  071104                DH48
 2278  007150  071350                DT4
 2279  007152  071632                DF6
 2280                          ;ERR 251
 2281  007154  062334                EM16                    ;DSC NOT SET
 2282  007156  071104                DH48
 2283  007160  071350                DT4
 2284  007162  071632                DF6
 2285                          ;ERR 252
 2286  007164  062361                EM17                    ;MSG A0 ERROR
 2287  007166  071104                DH48
 2288  007170  071260                DT1
 2289  007172  071656                DF8
 2290                          ;ERR 253
 2291  007174  062402                EM18                    ;MSG B0 ERROR
 2292  007176  071104                DH48
 2293  007200  071260                DT1
 2294  007202  071676                DF9
 2295                          ;ERR 254
 2296  007204  062423                EM19                    ;MSG A1 ERROR
 2297  007206  071104                DH48
 2298  007210  071260                DT1
 2299  007212  071656                DF8
 2300                          ;ERR 255
 2301  007214  062444                EM20                    ;MSG B1 ERROR
 2302  007216  071104                DH48
 2303  007220  071260                DT1
 2304  007222  071676                DF9
 2305                          ;ERR 256
 2306  007224  063420                EM38                    ;CYL DIFF IN RKMR2 NOT='CYL DIF'
 2307  007226  071104                DH48
```

```
 2308  007230  071350                    DF4
 2309  007232  071632                    DF6
 2310                            ;ERR 257
 2311  007234  063312                    EM36                    ;CYL ADDR IN RKMR3 NOT=RKDC
 2312  007236  071104                    DH48
 2313  007240  071350                    DT4
 2314  007242  071632                    DF6
 2315                            ;ERR 260
 2316  007244  062361                    EM17                    ;MSG A0 ERROR
 2317  007246  067716                    DH24                    ;AFTER OFFSET CMD
 2318  007250  071260                    DT1
 2319  007252  071656                    DF8
 2320                            ;ERR 261
 2321  007254  062402                    EM18                    ;MSG B0 ERROR
 2322  007256  067716                    DH24
 2323  007260  071260                    DT1
 2324  007262  071676                    DF9
 2325                            ;ERR 262
 2326  007264  066052                    EM77                    ;FAULT NOT CLEARED
 2327  007266  067643                    DH22                    ;AFTER DRIVE CLEAR CMD
 2328  007270  071260                    DT1
 2329  007272  071716                    DF10
 2330                            ;ERR 263
 2331  007274  066105                    EM78                    ;CYL DIFF IN RKMR2 NOT=1 IN SEEK TO SELF
 2332  007276  067313                    DH14                    ;AFTER SEEK WITH BAD PARITY
 2333  007300  071456                    DT8
 2334  007302  071632                    DF6
 2335                            ;ERR 264
 2336  007304  063576                    EM41                    ;CYL ADDR NOT CLEARED
 2337  007306  070047                    DH30                    ;AFTER READ HEADER CMD
 2338  007310  071260                    DT1
 2339  007312  071716                    DF10
 2340                            ;ERR 265
 2341  007314  062402                    EM18                    ;MSG B0 ERROR
 2342  007316  067643                    DH22                    ;AFTER DRIVE CLEAR CMD
 2343  007320  071260                    DT1
 2344  007322  071676                    DF9
 2345                            ;ERR 266
 2346  007324  062444                    EM20                    ;MSG B1 ERROR
 2347  007326  067643                    DH22
 2348  007330  071260                    DT1
 2349  007332  071676                    DF9
 2350                            ;ERR 267
 2351  007334  062402                    EM18
 2352  007336  070417                    DH39                    ;AFTER WRITE HEADER CMD
 2353  007340  071260                    DT1
 2354  007342  071676                    DF9
 2355                            ;ERR 270
 2356  007344  062444                    EM20
 2357  007346  070417                    DH39
 2358  007350  071260                    DT1
 2359  007352  071676                    DF9
 2360                            ;ERR 271
 2361  007354  062402                    EM18
 2362  007356  070047                    DH30                    ;AFTER READ HEADER CMD
 2363  007360  071260                    DT1
```

```
2364  007362  071676                    DF9
2365                              ;ERR 272
2366  007364  062444                    EM20
2367  007366  070047                    DH30
2368  007370  071260                    DT1
2369  007372  071676                    DF9
2370                              ;ERR 273
2371  007374  062361                    EM17                ;MSG A0 ERROR
2372  007376  067643                    DH22                ;AFTER DRV CLR CMD
2373  007400  071260                    DT1
2374  007402  071656                    DF8
2375                              ;ERR 274
2376  007404  062423                    EM19                ;MSG A1 ERROR
2377  007406  067643                    DH22
2378  007410  071260                    DT1
2379  007412  071656                    DF8
2380                              ;ERR 275
2381  007414  062402                    EM18                ;MSG B0 ERROR
2382  007416  067466                    DH17                ;AFTER RECAL CMD
2383  007420  071260                    DT1
2384  007422  071676                    DF9
2385                              ;ERR 276
2386  007424  062444                    EM20                ;MSG B1 ERROR
2387  007426  067466                    DH17
2388  007430  071260                    DT1
2389  007432  071676                    DF9
2390                              ;ERR 277
2391  007434  062361                    EM17                ;MSG A0 ERROR
2392  007436  070417                    DH39                ;AFTER WRITE HEADER CMD
2393  007440  071260                    DT1
2394  007442  071656                    DF8
2395                              ;ERR 300
2396  007444  062423                    EM19                ;MSG A1 ERROR
2397  007446  070417                    DH39
2398  007450  071260                    DT1
2399  007452  071656                    DF8
2400                              ;ERR 301
2401  007454  062361                    EM17
2402  007456  070047                    DH30                ;AFTER READ HEADER CMD
2403  007460  071260                    DT1
2404  007462  071656                    DF8
2405                              ;ERR 302
2406  007464  062423                    EM19
2407  007466  070047                    DH30
2408  007470  071260                    DT1
2409  007472  071656                    DF8
2410                              ;ERR 303
2411  007474  063471                    EM39                ;CYL DIFF/OFFSET NOT CLEARED
2412  007476  070417                    DH39                ;AFTER WRITE HEADER CMD
2413  007500  071260                    DT1
2414  007502  071716                    DF10
2415                              ;ERR 304
2416  007504  063576                    EM41                ;CYL ADDR NOT CLEARED
2417  007506  070417                    DH39
2418  007510  071260                    DT1
2419  007512  071716                    DF10
```

```
2420
2421                                    ;ERR 305
2422  007514  066172                            EM80            ;UNLD NOT SET
2423  007516  070374                            DH38            ;AFTER LIMIT DETECT
2424  007520  071260                            DT1
2425  007522  071716                            DF10
2426                                    ;ERR 306
2427  007524  066222                            EM81            ;SPIN NOT SET
2428  007526  067030                            DH9             ;AFTER START SPIN CMD.
2429  007530  071260                            DT1
2430  007532  071716                            DF10
2431                                    ;ERR 307
2432  007534  066250                            EM82            ;RTZ NOT SET
2433  007536  070527                            DH41            ;DURING RECAL CMD
2434  007540  071260                            DT1
2435  007542  071716                            DF10
2436                                    ;ERR 310
2437  007544  066275                            EM83            ;READ HEADER ERROR
2438  007546  066504                            DH1
2439  007550  071476                            DT9
2440  007552  072116                            DF21
2441                                    ;ERR 311
2442  007554  066275                            EM83
2443  007556  066504                            DH1
2444  007560  071514                            DT10
2445  007562  072002                            DF14
2446                                    ;ERR 312
2447  007564  066344                            EM84            ;FORMAT NOT SET
2448  007566  070417                            DH39            ;AFTER WRITE HEADER CMD
2449  007570  071260                            DT1
2450  007572  071716                            DF10
2451                                    ;ERR 313
2452  007574  066344                            EM84
2453  007576  070047                            DH30            ;AFTER READ HEADER CMD
2454  007600  071260                            DT1
2455  007602  071716                            DF10
2456                                    ;ERR 314
2457  007604  065057                            EM57            ;WCE AT CYL 411,TRK 2,SEC 21
2458  007606  066504                            DH1
2459  007610  071536                            DT12
2460  007612  071606                            DF4
2461                                    ;ERR 315
2462  007614  065165                            EM58            ;SPOK NOT CLEARED
2463  007616  067512                            DH18            ;AFTER UNLD CMD
2464  007620  071260                            DT1
2465  007622  071716                            DF10
```

```
                                        .SBTTL  PROGRAM SETUP

007624  012737  000001  001356  PARSRT: MOV     #1,PARAM        ;SET FLAG FOR 220 START
007632  005037  001360                  CLR     BYPT16
007636  005037  001362                  CLR     MODTST
007642  000450                          BR      PRGSRT          ;START PROGRAM

007644  005037  001356          BYT16:  CLR     PARAM
007650  012737  000001  001360        MOV     #1,BYPT16       ;SET FLAG TO BYPASS TEST 16
007656  005037  001362                  CLR     MODTST
007662  000440                          BR      PRGSRT

007664  012737  000001  001356  BYT16A: MOV     #1,PARAM
007672  012737  000001  001360        MOV     #1,BYPT16
007700  005037  001362                  CLR     MODTST
007704  000427                          BR      PRGSRT

007706  005037  001356          MDTST:  CLR     PARAM
007712  005037  001360                  CLR     BYPT16
007716  012737  000001  001362        MOV     #1,MODTST
007724  000417                          BR      PRGSRT

007726  005037  001356          MDT A:  CLR     PARAM
007732  012737  000001  001360        MOV     #1,BYPT16
007740  012737  000001  001362        MOV     #1,MODTST
007746  000406                          BR      PRGSRT
007750  105037  001356          START:  CLRB    PARAM           ;CLEAR FOR 200 START
007754  005037  001360                  CLR     BYPT16
007760  005037  001362                  CLR     MODTST
007764  000005          PRGSRT: RESET                           ;CLEAR ALL INT ENABLE & INIT
007766  012706  001100                  MOV     #STACK,SP       ;SETUP STACK POINTER
007772  012746  000340                  MOV     #PR7,-(SP)      ;PSW LOADED TO BE
007776  012746  010004                  MOV     #1$,-(SP)       ;LSI-11 COMPATABLE
010002  000002                          RTI                     ;LOCKOUT ALL INTERRUPTS

010004                          1$:
                                        .SBTTL  INITIALIZE THE COMMON TAGS
                                ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
010004  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
010010  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
010012  022706  001140                  CMP     #SWR,R6 ;;DONE?
010016  001374                          BNE     .-6             ;;LOOP BACK IF NO
010020  012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                                ;;INITIALIZE A FEW VECTORS
010024  012737  053400  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
010032  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
010040  012737  053660  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
010046  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
010054  012737  057042  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
010062  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
010070  012737  053312  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
010076  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
010104  013737  047246  047240          MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
010112  005037  001174                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
010116  005037  001176                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
```

```
2522  010122  112737  000001  001115          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
2523  010130  012737  010130  001106          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
2524  010136  012737  010136  001110          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
2525                                  ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
2526                                  ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
2527  010144  013746  000004          MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
2528  010150  012737  010204  000004  MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
2529  010156  012737  177570  001140  MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
2530  010164  012737  177570  001142  MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
2531  010172  022777  177777  170740  CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
2532  010200  001012                  BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
2533                                                          ;;AND  THE HARDWARE SWR IS NOT = -1
2534  010202  000403                  BR      65$             ;;BRANCH IF NO TIMEOUT
2535  010204  012716  010212          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
2536  010210  000002                  RTI
2537  010212  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
2538  010220  012737  000174  001142          MOV     #DISPREG,DISPLAY
2539  010226  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
2540
2541  010232  005037  001216          CLR     $PASS           ;;CLEAR PASS COUNT
2542  010236  132737  000200  001231  BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
2543  010244  001403                  BEQ     67$             ;;YES,USE NON-APT SWITCH
2544  010246  012737  001232  001140  MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
2545  010254                          67$:
2546  010254  012737  010314  000004  MEMPAR: MOV     #1$,ERRVEC      ;SET TIMEOUT VECTOR
2547  010262  012737  000340  000006          MOV     #PR7,ERRVEC+2
2548
2549  010270  012737  000001  172100          MOV     #1,MEMBAS       ;LOAD REG TO DETERMINE IF
2550                                                          ;MEM CHECK ENABLE AVAIL
2551  010276  012737  053214  000114          MOV     #MEMERR,MEMVEC  ;LOAD MEMORY CHECK VECTOR IF NO TIMEOUT
2552  010304  012737  000340  000116          MOV     #PR7,MEMVEC+2
2553  010312  000401                  BR      2$
2554
2555  010314  022626                  1$:     CMP     (SP)+,(SP)+     ;ADJ STACK
2556  010316  012737  000006  000004  2$:     MOV     #ERRVEC+2,ERRVEC        ;RESTORE TRAP CATCHER
2557  010324  005037  000006          CLR     ERRVEC+2
2558
2559  010330  004737  047334          JSR     PC,CLRFLG       ;CLEAR DDUMP THRU SIZFLG
2560  010334  005037  001220          CLR     $DEVCT
2561  010340  005037  001222          CLR     $UNIT
2562
2563                                  ;
2564                                  ;FIND OUT IF XXDP, ACT, APT; CHAIN OR DUMP MODE
2565                                  ;
2566
2567  010344  005737  000042          START1: TST     42
2568  010350  001014                  BNE     1$              ;BR IF AUTO
2569  010352  004737  047354          JSR     PC,TITLE        ;MANUAL, TYPE PROG ID
2570  010356  123727  000041  000013  CMPB    41,#13          ;13=LOADED BY XXDP
2571  010364  001010                  BNE     2$
2572  010366  005237  004412          INC     DDUMP           ;SET RK06 DUMP MODE FLAG
2573  010372  104401  060003          TYPE    ,MSG2           ;REPLACE DRO PACK W/SCRATCH & DO<CR>
2574  010376  000137  010412          JMP     ST2
2575  010402  000137  010456          1$:     JMP     ST3
2576  010406  005237  004420          2$:     INC     PPTP            ;SET ACT/APT/PTP DUMP MODE FLAG
2577
```

```
                                        ;
                                        ;CHECK IF ALL PARAMETERS DEFAULTED.  IF NOT, BEGIN INPUT DIALOGUE
                                        ;WITH OPERATOR.  THE REPLY TO 'DRIVES TO BE TESTED' SHOULD BE
                                        ;DRIVE NOS.  SEPERATED BY COMMAS & TERMINATED BY <CR>
                                        ;       EX:     DRIVES TO BE TESTED:  1,2,4<CR>
                                        ;
010412  005737  001356          ST2:    TST     PARAM
010416  001002                          BNE     1$              ;BR IF 220 START
010420  000137  010510                  JMP     ST4             ;200 START, DEFAULT & SIZE THE BUSS
010424  104401  060054          1$:     TYPE    ,MSG3           ;DRIVES TO BE TESTED
010430  004737  047374                  JSR     PC,GDRVS        ;GET DR NOS.
010434  104401  060106                  TYPE    ,MSG4           ;BUSS ADDR
010440  004737  047534                  JSR     PC,GBA          ;GET BA
010444  104401  060151                  TYPE    ,MSG5           ;CONT INT VECTOR
010450  004737  047562                  JSR     PC,GINT         ;GET INT VECTOR
010454  000427                          BR      ST5

                                        ;
                                        ;AUTO MODE
                                        ;CHECK IF LOADED BY XXDP OR OTHER.  SET FLAGS & NO INPUT DIALOGUE.
                                        ;DEFAULT ALL PARAMETERS.  TEST ONLY THOSE DRIVES THAT ARE READY
                                        ;ON THE BUSS
                                        ;
010456  123727  000041  000013  ST3:    CMPB    41,#13          ;13=LOADED BY XXDP
010464  001007                          BNE     1$
010466  005237  004414                  INC     DDPCH           ;SET RK06 CHAIN MODE FLAG
010472  004737  047354                  JSR     PC,TITLE
010476  104401  060264                  TYPE    ,MSG7           ;DRO NOT TSTD
010502  000402                          BR      ST4
010504  005237  004416          1$:     INC     ACT11           ;SET ACT AUTO FLAG.

010510  012737  177440  001264  ST4:    MOV     #177440,$BASE   ;DEFAULT VALUE
010516  012737  000210  001334          MOV     #210,RKVEC      ;DEFAULT VALUE
010524  004737  047614                  JSR     PC,SETINT
010530  005237  004452                  INC     SIZFLG          ;DO "SIZE THE BUSS" TEST

010534  012706  001100          ST5:    MOV     #STACK,SP       ;INIT STACK
010540  012746  000340                  MOV     #PR7,-(SP)      ;PSW LOADED TO BE
010544  012746  010552                  MOV     #5$,-(SP)       ;LSI-11 COMPATABLE
010550  000002                          RTI                     ;LOCKOUT ALL INTERRUPTS
010552  012737  004424  001366  5$:     MOV     #DRIV0,DRVPTR   ;SETUP
010560  005037  001220                  CLR     $DEVCT          ;NO. OF DRVS DONE
010564  005037  001222                  CLR     $UNIT           ;CURRENT DRV UNDER TEST
010570  012737  010636  000004          MOV     #1$,ERRVEC      ;SETUP TIMEOUT ERROR VECTOR
010576  005777  170544                  TST     @LKS            ;SEE IF L-CLOCK THERE
010602  005237  004444                  INC     LCLKF           ;PRESENT, SET FLAG.
010606  013700  001350                  MOV     LCVEC,R0        ;VECTOR ADDR
010612  012737  010700  000004          MOV     #2$,ERRVEC
010620  005777  170514                  TST     @P1             ;SEE IF P-CLOCK THERE
010624  005237  004446                  INC     PCLKF           ;PRESENT, SET FLAG
010630  013700  001352                  MOV     PCVEC,R0        ;VECTOR ADDR
010634  000412                          BR      3$

010636  022626                  1$:     CMP     (SP)+,(SP)+     ;L-CLOCK NOT THERE, CLEAR STACK
```

```
2634  010640  012737  010704  000004          MOV     #4$,ERRVEC
2635  010646  005777  170466                  TST     @PKS            ;SEE IF P-CLOCK THERE
2636  010652  005237  004446                  INC     PCLKF           ;PRESENT, SET FLAG
2637  010656  013700  001352                  MOV     PCVEC,R0        ;VECTOR ADDR
2638  010662  005237  004450          3$:     INC     DOTIM           ;INDICATES TIMING TESTS CAN BE DONE
2639  010666  012720  052414                  MOV     #CLOCK,(R0)+    ;SERVICE ROUTINE FOR CLOCKS
2640  010672  012710  000340                  MOV     #PR7,(R0)
2641  010676  000407                          BR      TST1            ;;GO TO NEXT TEST
2642
2643  010700  022626                  2$:     CMP     (SP)+,(SP)+     ;P-CLOCK NOT THERE, CLEAR STACK
2644  010702  000767                          BR      3$
2645
2646  010704  022626                  4$:     CMP     (SP)+,(SP)+     ;NEITHER CLOCK THERE, CLEAR STACK
2647  010706  005C37  004450                  CLR     DOTIM           ;TIMING TESTS CANNOT BE DONE.
2648  010712  104401  060521                  TYPE    ,MSG13          ;ALL TIMING TESTS BYPASSED
2649
2650
```

```
                                    .SBTTL BASIC CONTROLLER TESTS, SIZING & SETUP

                                    ;;*****************************************************************
                                    ;*TEST 1        REFERENCE ALL CONTROLLER REGISTERS
                                    ;*
                                    ;*      THIS TEST VERIFIES THAT ALL THE CONTROLLER REGISTERS
                                    ;*      CAN BE ACCESSED.  THE INABILITY  TO BE ACCESSED WILL
                                    ;*      RESULT IN A TIMEOUT TRAP WITH  AN ERROR MESSAGE.  ANY
                                    ;*      ERROR IN THIS TEST WILL RESULT IN ABORTING ALL OTHER
                                    ;*      TESTS AND JUMPING TO  'END OF PASS'
                                    ;*
                                    ;;*****************************************************************
010716  000004                    TST1:   SCOPE
010720  012737  000001  001174            MOV     #1,$TIMES       ;;DO 1 ITERATION
010726  012706  001100                    MOV     #STACK,SP       ;RESTORE STK PTR

010732  012737  011050  000004            MOV     #1$,ERRVEC      ;SETUP TIMOUT ERROR VECTOR
010740  013705  001264                    MOV     $BASE,R5        ;SETUP INDEX REG.
010744  005765  000000                    TST     RKCS1(R5)       ;REFERENCE ALL THE
010750  005765  000010                    TST     RKCS2(R5)       ;CONTROLLER REGISTERS
010754  005765  000002                    TST     RKWC(R5)
010760  005765  000004                    TST     RKBA(R5)
010764  005765  000006                    TST     RKDA(R5)
010770  005765  000012                    TST     RKDS(R5)        ;TIMEOUTS IN THIS SECTION
010774  005765  000014                    TST     RKER(R5)        ;INDICATE THAT THE CONTROLLER
011000  005765  000016                    TST     RKASOF(R5)      ;REGISTERS CANNOT BE READ.
011004  005765  000020                    TST     RKDC(R5)        ;TESTING SHOULD NOT PROCEED
011010  005765  000024                    TST     RKDB(R5)        ;UNTIL THIS IS REMEDIED.
011014  005765  000026                    TST     RKMR1(R5)
011020  005765  000034                    TST     RKMR2(R5)
011024  005765  000036                    TST     RKMR3(R5)
011030  005765  000030                    TST     RKECPS(R5)
011034  005765  000032                    TST     RKECPT(R5)

011040  012737  053040  000004            MOV     #BADTMO,ERRVEC  ;SETUP TIMEOUT HANDLER
011046  000404                          BR      TST2            ;;GO TO NEXT TEST

011050  022626                    1$:     CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
011052  104007                            ERROR   7               ;ABORT-COULD NOT REFERENCE CONTROLLER REGISTER
011054  000137  047212                    JMP     $EOP1

                                    ;;*****************************************************************
                                    ;*TEST 2        SIZE THE BUSS
                                    ;*
                                    ;*      THIS TEST IS ENTERED ONLY IF 'DRIVE SELECTION' IS DEFAULTED
                                    ;*      EITHER BY RUNNING IN THE AUTO MODE OR A 200 START IN THE
                                    ;*      MANUAL MODE.
                                    ;*      EVERY DRIVE FROM 0 THRU 7 IS ADDRESSED.
                                    ;*      CONTROLLER ERROR (CERR) IS EXAMINED AND IF NOT SET, THE
                                    ;*      DRIVE WILL BE TESTED.  IF SET, THE PROGRAM WILL BYPASS
                                    ;*      TESTING THAT DRIVE ONLY IF THE ERROR WAS A RESULT OF
                                    ;*      MDS, UFE OR NED BEING SET; OR BOTH NED & DRA RESET IN-
                                    ;*      DICATING THE OTHER PORT IS ACCESSED.
                                    ;*
                                    ;;*****************************************************************
```

```
2707  011060  000004                  TST2:   SCOPE
2708  011062  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
2709  011070  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
2710
2711  011074  012765  000040  000010          MOV     #SCLR,RKCS2(R5) ;SUBSYSTEM CLEAR
2712  011102  013737  001434  004400          MOV     T10,TEMP1
2713  011110  004737  047632                  JSR     PC,FRDY         ;FIND RDY
2714  011114  104120                          ERROR   120             ;RDY NOT SET BY END OF SCLR
2715  011116  005737  004452                  TST     SIZFLG
2716  011122  001562                          BEQ     TST3            ;;DO NOT SIZE, GOTO NEXT TEST
2717  011124  104401  060425                  TYPE    ,MSG10          ;WILL TEST DRIVES
2718  011130  005037  004422                  CLR     DRIVS           ;# OF DRIVES PRESENT
2719  011134  005000                          CLR     R0              ;DRV ADDR
2720  011136  012701  004424                  MOV     #DRIV0,R1       ;DRV FLAG
2721  011142                          1$:
2722  011142  104415                          SCOP1
2723  011144  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
2724
2725  011150  012765  000040  0[    ]          MOV     #SCLR,RKCS2(R5) ;SUBSYS CLEAR
2726  011156  013737  001434  004[JO          MOV     T10,TEMP1
2727  011164  004737  047632                  JSR     PC,FRDY         ;FIND RDY
2728  011170  104120                          ERROR   120             ;RDY NOT SET BY END OF SCLR
2729  011172  010065  000010                  MOV     R0,RKCS2(R5)    ;SELECT THE DRIVE ADDR
2730  011176  012765  000001  000000          MOV     #SELDRV,RKCS1(R5)
2731  011204  013737  001434  004400          MOV     T10,TEMP1
2732  011212  004737  047632                  JSR     PC,FRDY         ;FIND RDY
2733  011216  104117                          ERROR   117             ;NO RDY AFTER SELECT DR. CMD
2734  011220  032737  100000  004336          BIT     #CERR,HCS1
2735  011226  001046                          BNE     2$
2736  011230  013737  004364  004400          MOV     HMR2,TEMP1
2737  011236  042737  177770  004400          BIC     #^C<DRVMSK>,TEMP1
2738  011244  020037  004400                  CMP     R0,TEMP1        ;S/B SAME
2739  011250  001016                          BNE     3$
2740  011252  005700                          TST     R0
2741  011254  001003                          BNE     4$
2742  011256  005737  004414                  TST     DDPCH           ;SEE IF XXDP CHAIN MODE
2743  011262  001014                          BNE     5$
2744  011264  005237  004422          4$:     INC     DRIVS           ;INC DRIVE COUNT.
2745  011270  005211                          INC     (R1)            ;SET DRIVE PRESENT FLAG
2746  011272  104401  001205                  TYPE    ,$CRLF
2747  011276  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
2748                                                                  ;;TYPE DR #
2749  011300  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
2750  011302     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
2751  011303     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
2752  011304  000403                          BR      5$
2753
2754  011306  004737  050242          3$:     JSR     PC,BYP          ;TYPE BYPASS DR #
2755  011312  104001                          ERROR   1               ;SELECTED DR # DOES NOT MATCH RKMR2 DR #
2756
2757  011314  005721                  5$:     TST     (R1)+           ;SHIFT PTR TO NEXT DR. FLAG
2758  011316  005200                          INC     R0              ;INC DR #
2759  011320  022700  000010                  CMP     #8.,R0
2760  011324  001306                          BNE     1$              ;MORE LEFT.
2761  011326  005737  004422                  TST     DRIVS
2762  011332  001054                          BNE     10$
```

```
2763  011334  104401  060512                  TYPE    ,MSG12          ;NONE
2764  011340  000137  047212                  JMP     $EOP1
2765
2766  011344  032737  001000  004340  2$:     BIT     #MDS,HCS2
2767  011352  001015                          BNE     6$
2768  011354  032737  000400  004340          BIT     #UFE,HCS2
2769  011362  001015                          BNE     7$
2770  011364  032737  000001  004350          BIT     #DRA,HDS
2771  011372  001015                          BNE     8$
2772  011374  032737  010000  004340          BIT     #NED,HCS2
2773  011402  001424                          BEQ     9$
2774  011404  000743                          BR      5$
2775
2776  011406  004737  050242          6$:     JSR     PC,BYP          ;TYPE BYP DR #
2777  011412  104002                          ERROR   2               ;MDS DETECTED
2778  011414  000737                          BR      5$
2779
2780  011416  004737  050242          7$:     JSR     PC,BYP
2781  011422  104003                          ERROR   3               ;UFE DETECTED
2782  011424  000733                          BR      5$
2783
2784  011426  032737  010000  004340  8$:     BIT     #NED,HCS2
2785  011434  001713                          BEQ     4$
2786  011436  104401  060632                  TYPE    ,MSG15          ;DRV#
2787  011442  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
2788                                                                  ;;TYPE DR#
2789  011444  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
2790  011446     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
2791  011447     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
2792  011450  104010                          ERROR   10              ;DRA & NED BOTH SET
2793  011452  000720                          BR      5$
2794
2795  011454  004737  050242          9$:     JSR     PC,BYP
2796  011460  104004                          ERROR   4               ;NO DRA & NO NED = OTHER PORT SELECTED
2797  011462  000714                          BR      5$
2798  011464  000137  012012          10$:    JMP     UNLDRV
2799
2800                                          ;;*****************************************************************
2801                                          ;*TEST 3        VERIFY OPERATOR DRIVE SELECTIONS
2802                                          ;*
2803                                          ;*      THIS TEST IS ENTERED ONLY IF DRIVE SELECTION IS NOT
2804                                          ;*      DEFAULTED.  EVERY DRIVE FROM 0 TO 7 IS ADDRESSED &
2805                                          ;*      CONTROLLER ERROR (CERR) IS EXAMINED. IF NOT SET, THE
2806                                          ;*      PROGRAM WILL ASSUME THE DRIVE IS PRESENT.  IT WILL THEN CHECK
2807                                          ;*      TO SEE THAT THE DRIVE WAS INPUTTED FOR TESTING.  IF NOT, IT WILL
2808                                          ;*      BE AN ERROR.  IF CERR WAS SET, THAT DRIVE WILL BE BYPASSED
2809                                          ;*      ONLY IF THE ERROR WAS A RESULT OF MDS OR UFE SET OR BOTH
2810                                          ;*      NED & DRA RESET (WRONG PORT).  IF CERR IS  A RESULT OF
2811                                          ;*      NED ONLY, IT IS CHECKED AGAINT THE INPUTTED INFOR TO
2812                                          ;*      VERIFY IT WAS NOT SPECIFIED.
2813                                          ;*
2814                                          ;*****************************************************************
2815  011470  000004                  TST3:   SCOPE
2816  011472  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
2817  011500  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
2818  011504  005000                          CLR     R0              ;DRIVE ADDR
```

```
2819  011506  012701  004424                  MOV     #DRIV0,R1       ;DRIVE FLAG
2820  011512                          1$:
2821  011512  104415                          SCOP1
2822  011514  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
2823
2824  011520  012765  000040  000010          MOV     #SCLR,RKCS2(R5)
2825  011526  013737  001434  004400          MOV     T10,TEMP1
2826  011534  004737  047632                  JSR     PC,FRDY         ;FIND RDY
2827  011540  104120                          ERROR   120             ;NO RDY AFTER SCLR
2828  011542  010065  000010                  MOV     R0,RKCS2(R5)    ;DRV ADDR
2829  011546  012765  000001  000000          MOV     #SELDRV,RKCS1(R5)
2830  011554  013737  001434  004400          MOV     T10,TEMP1
2831  011562  004737  047632                  JSR     PC,FRDY         ;FIND RDY
2832  011566  104117                          ERROR   117             ;NO RDY AFTER SELDRV CMD
2833  011570  032737  100000  004336          BIT     #CERR,HCS1
2834  011576  001036                          BNE     2$
2835  011600  013737  004364  004400          MOV     HMR2,TEMP1
2836  011606  042737  177770  004400          BIC     #^C<DRVMSK>,TEMP1
2837  011614  020037  004400                  CMP     R0,TEMP1        ;S/B SAME
2838  011620  001010                          BNE     3$
2839  011622  005711                  11$:    TST     (R1)
2840  011624  001417                          BEQ     5$
2841  011626  005721                  4$:     TST     (R1)+           ;SHIFT PTR TO NEXT DR FLAG
2842  011630  005200                          INC     R0              ;INC DR#
2843  011632  022700  000010                  CMP     #8.,R0
2844  011636  001325                          BNE     1$              ;MORE LEFT
2845  011640  000464                          BR      TST4            ;;GO TO NEXT TEST
2846
2847  011642  004737  050242          3$:     JSR     PC,BYP          ;TRY BYPASS DRIVE#
2848  011646  104001                          ERROR   1               ;WRITTEN DR# DOES NOT MATCH RKMR2 DR#
2849  011650  005711                          TST     (R1)
2850  011652  001765                          BEQ     4$              ;BRANCH IF NOT SPEC BY INPUT
2851  011654  005337  004422          12$:    DEC     DRIVS           ;DECREMENT TOTAL DRIVS
2852  011660  005011                          CLR     (R1)            ;CLEAR DRIVE FLAG
2853  011662  000761                          BR      4$
2854
2855  011664  004737  050242          5$:     JSR     PC,BYP
2856  011670  104005                          ERROR   5               ;DR PRESENT BUT NOT SPECIFIED BY OPERATOR
2857  011672  000755                          BR      4$
2858
2859  011674  032737  001000  004340  2$:     BIT     #MDS,HCS2
2860  011702  001027                          BNE     6$
2861  011704  032737  000400  004340          BIT     #UFE,HCS2
2862  011712  001027                          BNE     7$
2863  011714  032737  000001  004350          BIT     #DRA,HDS
2864  011722  001005                          BNE     8$
2865  011724  032737  010000  004340          BIT     #NED,HCS2
2866  011732  001423                          BEQ     9$
2867  011734  000404                          BR      10$
2868  011736  032737  010000  004340  8$:     BIT     #NED,HCS2
2869  011744  001726                          BEQ     11$
2870  011746  005711                  10$:    TST     (R1)
2871  011750  001726                          BEQ     4$
2872
2873  011752  004737  050242                  JSR     PC,BYP          ;TYPE BYPASS DRIVE#
2874  011756  104006                          ERROR   6
```

```
 2875  011760  000735                          BR      12$
 2876
 2877  011762  004737  050242          6$:     JSR     PC,BYP          ;TYPE BYPASS DRIVE#
 2878  011766  104002                          ERROR   2               ;MDS DETECTED
 2879  011770  000762                          BR      8$
 2880
 2881  011772  004737  050242          7$:     JSR     PC,BYP
 2882  011776  104003                          ERROR   3               ;UFE DETECTED
 2883  012000  000756                          BR      8$
 2884
 2885  012002  004737  050242          9$:     JSR     PC,BYP
 2886  012006  104004                          ERROR   4               ;DRA & NED RESET - OTHER PORT SELECTED
 2887  012010  000752                          BR      8$
 2888
 2889  012012                          UNLDRV:
 2890
 2891                                          ;;*****************************************************************
 2892                                          ;*TEST 4        UNLOAD ALL DRIVES TO BE TESTED
 2893                                          ;*
 2894                                          ;*      THIS TEST UNLOADS THE DRIVES TO BE TESTED, WAITS FOR
 2895                                          ;*      ATTN & VERIFIES IT CAME FROM THE CORRECT DRIVE WITHIN
 2896                                          ;*      APPROX 100 USEC.  DRDY & DSC IN RKMR2 ARE CHECKED TO BE
 2897                                          ;*      CORRECT.
 2898                                          ;*
 2899                                          ;;*****************************************************************
 2900  012012  000004                  TST4:   SCOPE
 2901  012014  012737  000001  001174          MOV     #1,STIMES       ;;DO 1 ITERATION
 2902  012022  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
 2903
 2904  012026  005000                          CLR     R0              ;DRV ADDR
 2905  012030  012701  004424                  MOV     #DRIVD,R1       ;DRV FLAG
 2906
 2907  012034                          1$:
 2908  012034  104415                          SCOP1
 2909  012036  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
 2910
 2911  012042  012765  000040  000010          MOV     #SCLR,RKCS2(R5)
 2912  012050  013737  001434  004420          MOV     TIO,TEMP1
 2913  012056  004737  047632                  JSR     PC,FRDY         ;FIND RDY
 2914  012062  104120                          ERROR   120             ;NO RDY AFTER SCLR
 2915  012064  005721                          TST     (R1)+
 2916  012066  001436                          BEQ     4$
 2917  012070  010065  000010                  MOV     R0,RKCS2(R5)    ;SELECT DRV ADDR.
 2918  012074  012765  000007  000000          MOV     #UNLOAD,RKCS1(R5)  ;UNLOAD CMD
 2919  012102  013737  001434  004400          MOV     TIO,TEMP1
 2920  012110  004737  047632                  JSR     PC,FRDY         ;FIND RDY
 2921  012114  104011                          ERROR   11              ;RDY NOT SET AFTER UNLOAD CMD
 2922  012116  105737  004355                  TSTB    HASOF+1         ;TEST FOR ANY ATTN BITS
 2923  012122  001001                          BNE     5$              ;BR IF ANY
 2924  012124  104012                          ERROR   12              ;NO ATTN AFTER UNLOAD CMD
 2925  012126  136037  004326  004355  5$:     BITB    ATTN(R0),HASOF+1  ;TEST FOR SPECIFIC ATTN
 2926  012134  001001                          BNE     2$              ;BR IF THERE
 2927  012136  104013                          ERROR   13              ;WRONG ATTN AFTER UNLOAD CMD
 2928  012140  032737  000200  004364  2$:     BIT     #D.DRDY,HMR2
 2929  012146  001401                          BEQ     3$
 2930  012150  104014                          ERROR   14              ;D.DRDY SET AFTER UNLOAD CMD
```

```
                012152  032737  040000  004364  3$:     BIT     #D.DSC,HMR2
                012160  001001                          BNE     4$
                012162  104015                          ERROR   15              ;D.DSC NOT SET AFTER UNLOAD CMD
                012164  005200                  4$:     INC     R0              ;INCR DRIVE ADDR.
                012166  020027  000010                  CMP     R0,#8.
                012172  001320                          BNE     1$              ;BRANCH IF MORE LEFT.
                                                        ;
                                                        ;THIS PART OF THE PROGRAM WILL BE REPEATED FOR EACH
                                                        ;DRIVE PRESENT
                                                        ;
                                                        ;'SUNIT' CONTAINS THE ADDRESS OF THE DRIVE CURRENTLY
                                                        ;UNDER TEST
                012174                          NUDRV:                          ;ENTER HERE FROM LAST TEST
                                                        ;;*****************************************************************
                                                        ;*TEST 5        FIND NEXT DRIVE TO BE TESTED
                                                        ;*
                                                        ;*      THIS TEST FINDS THE NEXT DRIVE PRESENT & PUTS THAT
                                                        ;*      ADDRESS IN 'SUNIT'.
                                                        ;*      THROUGHOUT THE FOLLOWING TESTS, THE DRIVE TESTED IS
                                                        ;*      THE DRIVE WHOSE ADDRESS IS IN 'SUNIT'.
                                                        ;*
                                                        ;;*****************************************************************
                012174  000004                  TST5:   SCOPE
                012176  012737  000001  001174          MOV     #1,STIMES       ;;DO 1 ITERATION
                012204  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
                012210  012737  000005  001214          MOV     #STN-1,STESTN
                012216  012737  000005  001102          MOV     #STN-1,STSTNM
                012224  005737  004422                  TST     DRIVS           ;ANY DRIVES PRESENT?
                012230  001004                          BNE     4$              ;YES BRANCH
                012232  104401  060760                  TYPE    ,MSG19          ;ALL DRIVES TESTED
                012236  000137  047212                  JMP     $EOP1           ;NO, GO TO END
                012242  013701  001366          4$:     MOV     DRVPTR,R1       ;ADDR OF NEXT DRIVE FLAG
                012246  005737  001220                  TST     $DEVCT          ;IS FIRST DRIVE BEING CHECKED
                012252  001402                          BEQ     2$              ;YES, BRANCH
                012254  005237  001222          1$:     INC     $UNIT           ;INCR DRIVE ADDR TO NEXT DRIVE
                012260  005721                  2$:     TST     (R1)+           ;IS DRIVE PRESENT?
                012262  001774                          BEQ     1$              ;NO,FIND NEXT DRIVE PRESENT
                012264  005737  004414                  TST     DDPCH           ;DDP CHAIN MODE?
                012270  001403                          BEQ     3$              ;NO, BRANCH
                012272  005737  001222                  TST     $UNIT           ;YES, IS IT DRIVE 0?
                012276  001766                          BEQ     1$              ;IF YES, DON'T TEST DR 0
                012300  010137  001366          3$:     MOV     R1,DRVPTR       ;STORE POINTER TO THE NEXT DR. FLAG
                012304  104401  060632                  TYPE    ,MSG15          ;"DRIVE"
                012310  013700  001222                  MOV     $UNIT,R0
                012314  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
                                                        ;;DRIVE #
                012316  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
                012320     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
                012321     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
```

```
                                TYPE    ,$CRLF

                        PFSRT:                          ;ENTER HERE FOR POWER FAIL RESTART
                        .SBTTL STATIC & CYCLE UP TESTS

                        ;;*****************************************************************
                        ;*TEST 6        REFERENCE & CHECK ALL STATUS BYTES IN RKMR2 & RKMR3
                        ;*
                        ;*      THIS TEST WAITS FOR APPROX 10 SEC FOR THE HEADS TO BE
                        ;*      IN THE HOME POSITION & SPEED OK TO GO LOW FROM THE 'UNLOAD' GIVEN
                        ;*      IN A PREVIOUS TEST.  IT THEN CHECKS THE ABILITY TO REFERENCE ALL
                        ;*      DRIVE REGISTERS AND THAT THEY CONTAIN CORRECT STATUS.
                        ;*
                        ;;*****************************************************************
012326  000004          TST6:   SCOPE
012330  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
012336  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR

012342  004737  050320                  JSR     PC,SUBCLR       ;SUBSYS CLEAR
012346  104024                          ERROR   24              ;CERR AFTER SCLR

012350  012737  000012  004402          MOV     #10.,TEMP2      ;SETUP FOR 10 SEC. DELAY
012356  012737  177777  004400  1$:     MOV     #-1,TEMP1       ;ALL 1'S
012364  012765  000001  000026          MOV     #1,RKMR1(R5)    ;WORD 1
012372  004737  050256          2$:     JSR     PC,GSTAT
012376  032737  001000  004364          BIT     #D.SPOK,HMR2    ;CHECK SPEED OK
012404  001407                          BEQ     3$
012406  005337  004400                  DEC     TEMP1
012412  001367                          BNE     2$
012414  005337  004402                  DEC     TEMP2
012420  001356                          BNE     1$
012422  104150                          ERROR   150             ;SPEED OK NOT CLEARED 10 SEC AFT UNLOAD
012424                          3$:
012424  104415                          SCOP1
012426  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR

012432  004737  050320                  JSR     PC,SUBCLR
012436  104024                          ERROR   24              ;CERR AFTER SCLR

012440  004737  050256                  JSR     PC,GSTAT        ;GET STATUS
012444  032737  000100  004364          BIT     #D.VV,HMR2      ;SEE IF VOL VALID SET
012452  001007                          BNE     4$              ;BR IF SET
012454  012737  000040  004374          MOV     #D.DRA,SBMR2    ;LOAD SHOULD BE VALUE
012462  004737  052504                  JSR     PC,CKMR2        ;CHECK RKMR2
012466  104016                          ERROR   16              ;MSG AO ERROR FOR DRIVE UNLOADED
012470  000406                          BR      5$

012472  012737  000140  004374  4$:     MOV     #<D.DRA!D.VV>,SBMR2
012500  004737  052504                  JSR     PC,CKMR2        ;CHECK RKMR2
012504  104016                          ERROR   16              ;MSG AO ERROR FOR DRIVE UNLOADED

012506  005037  004376          5$.     CLR     SBMR3           ;RKMR3 SHOULD BE 0
012512  004737  052566                  JSR     PC,CKMR3        ;CHECK RKMR3
012516  104017                          ERROR   17              ;MSG BO ERROR FOR DRIVE UNLOADED
```

```
3043
3044  012520  104415                          SCOP1
3045  012522  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
3046
3047  012526  004737  050320                  JSR     PC,SUBCLR
3048  012532  104024                          ERROR   24              ;CERR AFTER SCLR
3049
3050  012534  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
3051  012542  004737  050256                  JSR     PC,GSTAT        ;GET STATUS
3052  012546  012737  000740  004374          MOV     #<D.HDHM!D.BRHM!D.DOOR!D.CART>,SBMR2
3053  012554  004737  052504                  JSR     PC,CKMR2        ;CHECK RKMR2
3054  012560  104020                          ERROR   20              ;MSG A1 ERROR FOR DRIVE UNLOADED
3055
3056  012562  005037  004376                  CLR     SBMR3           ;RKMR3 SHOULD BE 0
3057  012566  004737  052566                  JSR     PC,CKMR3        ;CHECK RKMR3
3058  012572  104021                          ERROR   21              ;MSG B1 ERROR FOR DRIVE UNLOADED
3059
3060  012574  104415                          SCOP1
3061  012576  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
3062
3063  012602  004737  050320                  JSR     PC,SUBCLR
3064  012606  104024                          ERROR   24              ;CERR AFTER SCLR
3065
3066  012610  012765  000002  000026          MOV     #2,RKMR1(R5)    ;SELECT WORD 2
3067  012616  004737  050256                  JSR     PC,GSTAT        ;GET STATUS
3068  012622  005037  004374                  CLR     SBMR2           ;RKMR2 SHOULD BE 0
3069  012626  004737  052504                  JSR     PC,CKMR2        ;CHECK RKMR2
3070  012632  104022                          ERROR   22              ;MSG A2 ERROR FOR DRIVE UNLOADED.
3071
3072  012634  005037  004376                  CLR     SBMR3           ;RKMR3 SHOULD BE 0
3073  012640  004737  052566                  JSR     PC,CKMR3        ;CHECK RKMR3
3074  012644  104023                          ERROR   23              ;MSG B2 ERROR FOR DRIVE UNLOADED
3075
3076  012646  104415                          SCOP1
3077  012650  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
3078
3079  012654  004737  050320                  JSR     PC,SUBCLR
3080  012660  104024                          ERROR   24              ;CERR AFTER SCLR
3081
3082  012662  012765  000003  000026          MOV     #3,RKMR1(R5)    ;SELECT WORD 3
3083  012670  004737  050256                  JSR     PC,GSTAT        ;GET STATUS
3084  012674  013737  004364  004374          MOV     HMR2,SBMR2
3085  012702  042737  100007  004374          BIC     #<M.PAR!7>,SBMR2  ;PARITY & DRV ADDR
3086  012710  004737  052504                  JSR     PC,CKMR2        ;CHECK RKMR2
3087  012714  104001                          ERROR   1               ;SUNIT IN CS2 DOES NOT MATCH RKMR2
3088
3089  012716  013737  004366  004400          MOV     HMR3,TEMP1
3090  012724  042737  170777  004400          BIC     #^C<M.HEAD>,TEMP1
3091  012732  023727  004400  001000          CMP     TEMP1,#BIT9     ;WHEN RKDA=HEAD 0, HEAD=1 IN RKMR3
3092  012740  001401                          BEQ     TST7            ;;GO TO NEXT TST IF COMPARE
3093  012742  104056                          ERROR   56              ;HEAD REG IN MR3 NOT CLEARED IN UNLD
3094
3095
3096                                  ;;*****************************************************************
3097                                  ;*TEST 7        PRINT DRIVE SERIAL NUMBER
3098                                  ;*
```

```
                                        ;*      THIS TEST READS & PRINTS THE DRIVE SERIAL # FROM MSG A, WORD 11
                                        ;*      IN BCD & IS PERFORMED ON THE 1ST PASS ONLY
                                        ;*
                                        ;;*****************************************************************
012744  000004                        TST7:   SCOPE
012746  012737  000001  001174                MOV     #1,$TIMES       ;;DO 1 ITERATION
012754  012706  001100                        MOV     #STACK,SP       ;RESTORE STK PTR

012760  005737  001216                        TST     $PASS
012764  001046                        BNE     TST10                   ;;GO TO NEXT IF NOT FIRST PASS
012766  004737  050320                        JSR     PC,SUBCLR       ;DO SUBSYS CLEAR
012772  104024                        ERROR   24                      ;CERR AFTER SCLR

012774  104401  060644                        TYPE    ,MSG16          ;DRIVE SERIAL NO.
013000  012765  000003  000026                MOV     #3,RKMR1(R5)    ;SELECT BYTE 3
013006  004737  050256                        JSR     PC,GSTAT        ;GET STATUS
013012  013701  004364                        MOV     HMR2,R1         ;GET SERIAL #
013016  012704  056440                        MOV     #$OCTVL,R4      ;GET ADDR CHAR BUFF
013022  010446                        MOV     R4,-(SP)                ;STORE ON STACK FOR $SUPRS
013024  012703  000003                        MOV     #3,R3           ;SETUP CHAR COUNT
013030  006101                        ROL     R1                      ;INITIALIZE BIT POSITIONS
013032  006101                        ROL     R1
013034  006101                    1$: ROL     R1                      ;GET NEXT 4 BITS
013036  006101                        ROL     R1
013040  006101                        ROL     R1
013042  006101                        ROL     R1
013044  010100                        MOV     R1,R0                   ;GET WORKING COPY
013046  042700  177760                BIC     #177760,R0              ;CLEAR ALL BUT LOW 4 BITS
013052  052700  000060                BIS     #60,R0                  ;CONVERT TO ASCII DIGIT
013056  110024                        MOVB    R0,(R4)+                ;PUT ASCII DIGIT INTO CHAR BUFF
013060  005303                        DEC     R3
013062  001364                        BNE     1$                      ;BR IF ALL 3 CHARS NOT DONE
013064  105014                        CLRB    (R4)                    ;ELSE INSERT NULL TERMINATOR
013066  004737  056706                JSR     PC,$SUPRS               ;TYPE
013072  104401  001205                TYPE    ,$CRLF
013076  104401  001205                TYPE    ,$CRLF

                                    ;;*****************************************************************
                                    ;*TEST 10        SET VV WITH PACK COMMAND
                                    ;*
                                    ;*      IF VV IS RESET, THE PACK COMMAND IS USED TO SET IT.
                                    ;*
                                    ;;*****************************************************************
013102  000004                    TST10:  SCOPE
013104  012737  000001  001174            MOV     #1,$TIMES       ;;DO 1 ITERATION
013112  012706  001100                    MOV     #STACK,SP       ;RESTORE STK PTR

013116  005065  000026                    CLR     RKMR1(R5)       ;SELECT BYTE 0
013122  004737  050256                    JSR     PC,GSTAT        ;GET STATUS
013126  032737  000100  004364            BIT     #D.VV,HMR2
013134  001024                    BNE     TST11                   ;;GO TO NEXT TEST IF VV SET

013136  104415                    SCOP1
013140  012706  001100                    MOV     #STACK,SP       ;RESTORE STK PTR

013144  004737  050320                    JSR     PC,SUBCLR
```

```
3155  013150  104024                          ERROR   24              ;CERR AFTER SCLR
3156
3157  013152  012765  000003  000000          MOV     #PACK,RKCS1(R5) ;CMD TO SET VV
3158  013160  013737  001434  004400          MOV     T10,TEMP1
3159  013166  004737  047632                  JSR     PC,FRDY         ;FIND RDY
3160  013172  104116                          ERROR   116             ;RDY NOT SET AFTER PACK CMD
3161
3162  013174  032737  000100  004364          BIT     #D.VV,HMR2
3163  013202  001001                          BNE     TST11           ;;GO TO NEXT TEST IF VV NOW SET
3164  013204  104027                          ERROR   27              ;PACK DID NOT SET V.V.
3165
3166                                  ;;**************************************************************
3167                                  ;*TEST 11       RELEASE DRIVE
3168                                  ;*
3169                                  ;*      TESTS THE ABILITY TO RECOGNIZE THE RLS BIT AND NOT RAISE SACK
3170                                  ;*
3171                                  ;;**************************************************************
3172  013206  000004                  TST11:  SCOPE
3173  013210  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
3174  013216  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
3175
3176  013222  004737  050320                  JSR     PC,SUBCLR       ;DO SUBSYS CLEAR & GET STATUS
3177  013226  104024                          ERROR   24              ;CONTR ERROR SET AFTER SCLR
3178  013230  032737  000400  004340          BIT     #UFE,HCS2
3179  013236  001401                          BEQ     1$
3180  013240  104003                          ERROR   3               ;UFE SET AFTER SCLR
3181
3182  013242                          1$:
3183  013242  104415                          SCOP1
3184  013244  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
3185
3186  013250  004737  050320                  JSR     PC,SUBCLR
3187  013254  104024                          ERROR   24              ;CERR AFTER SCLR
3188
3189  013256  062765  000010  000010          ADD     #RLS,RKCS2(R5)  ;ADD RELEASE BIT TO $UNIT
3190  013264  004737  050256                  JSR     PC,GSTAT        ;GET STATUS
3191
3192  013270  032737  100000  004336          BIT     #CERR,HCS1      ;CHECK FOR CONTR ERROR
3193  013276  001401                          BEQ     2$
3194  013300  104025                          ERROR   25              ;RLS SET CERR
3195  013302  032737  000400  004340  2$:     BIT     #UFE,HCS2
3196  013310  001401                          BEQ     TST12           ;;GO TO NEXT TEST IF SET
3197  013312  104026                          ERROR   26              ;SACK SET AFTER RLS SENT
3198
3199                                  ;;**************************************************************
3200                                  ;*TEST 12       DRIVE TYPE TEST
3201                                  ;*
3202                                  ;*      THIS TEST COMPARES DRIVE TYPE IN MESSAGE A AGAINST 'DDT' IN RKDS.
3203                                  ;*      WRONG CDT IN RKCS1 IS SENT & ERRORS ARE VERIFIED.
3204                                  ;*
3205                                  ;;**************************************************************
3206  013314  000004                  TST12:  SCOPE
3207  013316  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
3208  013324  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
3209
3210  013330  004737  050320                  JSR     PC,SUBCLR       ;SUBSYS CLEAR & GET STATUS
```

```
3211  013334  104024                          ERROR   24              ;CONT ERR SET AFT SUBSYS CLEAR
3212  013336  032737  000400  004364          BIT     #D.DDT,HMR2
3213  013344  001401                          BEQ     2$
3214  013346  104030                          ERROR   30              ;DR TYPE SET IN MR2
3215  013350  032737  000400  004350  2$:     BIT     #DDT,HDS
3216  013356  001401                          BEQ     3$
3217  013360  104031                          ERROR   31              ;DDT SET IN RKDS
3218  013362  032737  000040  004352  3$:     BIT     #DTYE,HER
3219  013370  001401                          BEQ     4$
3220  013372  104032                          ERROR   32              ;DTYE SET IN RKER
3221
3222  013374                          4$:
3223  013374  104415                          SCOP1
3224  013376  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
3225
3226  013402  004737  050320                  JSR     PC,SUBCLR
3227  013406  104024                          ERROR   24              ;CERR AFTER SCLR
3228
3229  013410  012765  002001  000000          MOV     #<CDT!SELDRV>,RKCS1(R5)  ;GET STATUS WITH CDT SET
3230  013416  013737  001434  004400          MOV     T10,TEMP1
3231  013424  004737  047632                  JSR     PC,FRDY         ;FIND RDY
3232  013430  104117                          ERROR   117             ;RDY NOT SET BY END OF SEL DRV CMD
3233  013432  032737  000400  004364          BIT     #D.DDT,HMR2
3234  013440  001401                          BEQ     5$
3235  013442  104030                          ERROR   30              ;DR TYPE SET IN MR2
3236  013444  032737  000400  004350  5$:     BIT     #DDT,HDS
3237  013452  001401                          BEQ     6$
3238  013454  104031                          ERROR   31              ;DDT SET IN RKDS
3239  013456  032737  000040  004352  6$:     BIT     #DTYE,HER       ;DTYE=DDT(NOT)*CDT
3240  013464  001001                          BNE     7$
3241  013466  104033                          ERROR   33              ;DTYE NOT SET AFT WRITING CDT=1
3242  013470  032737  100000  004336  7$:     BIT     #CERR,HCS1
3243  013476  001001                          BNE     TST13           ;;GO TO NEXT TEST
3244  013500  104034                          ERROR   34              ;CERR NOT SET AFT WRITING CDT=1
3245
3246                                  ;;*****************************************************************
3247                                  ;*TEST 13       C-D PARITY ERROR DETECTION
3248                                  ;*
3249                                  ;*      TESTS THE ABILITY OF THE DRIVE TO DETECT EVEN PARITY SENT BY
3250                                  ;*      THE CONTROLLER BY SETTING 'PAT' ON RKMR1.
3251                                  ;*      THE DRIVE SHOULD RESPOND WITH 'C-D PARITY ERROR'
3252                                  ;*      THE DRIVE STILL SENDS ODD PARITY TO THE CONTROLLER WHICH IS NOW
3253                                  ;*      CHECKING FOR EVEN PARITY THEREFORE THE CONTROLLER SHOULD DETECT
3254                                  ;*      AN ERROR AND SET SPAR.
3255                                  ;*      THE ERROR CONDITION IS RESET WITH THE CLEAR COMMAND
3256                                  ;*
3257                                  ;*****************************************************************
3258  013502  000004                  TST13:  SCOPE
3259  013504  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
3260  013512  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
3261
3262  013516  004737  050320                  JSR     PC,SUBCLR       ;SUBSYS CLEAR & GET STATUS
3263  013522  104024                          ERROR   24              ;CO'IT ERR AFTER SUBSYS CLR
3264  013524  032737  001000  004366          BIT     #D.PAR,HMR3
3265  013532  001401                          BEQ     2$
3266  013534  104035                          ERROR   35              ;C-D PARITY ERROR SET IN MR3
```

```
3267  013536  032737  020000  004336  2$:     BIT     #SPAR,HCS1
3268  013544  001401                          BEQ     3$
3269  013546  104036                          ERROR   36              ;SPAR SET IN CS1
3270
3271  013550                          3$:
3272  013550  104415                          SCOP1
3273  013552  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
3274
3275  013556  004737  050320                  JSR     PC,SUBCLR
3276  013562  104024                          ERROR   24              ;CERR AFTER SCLR
3277
3278  013564  012765  000020  000026          MOV     #PAT,RKMR1(R5)  ;SELECT BYTE 0 & EVEN PARITY
3279  013572  004737  050256                  JSR     PC,GSTAT        ;GET STATUS
3280  013576  032737  000200  004366          BIT     #D.FLT,HMR3
3281  013604  001001                          BNE     4$
3282  013606  104037                          ERROR   37              ;FAULT NOT SET IN MR3
3283  013610  032737  001000  004366  4$:     BIT     #D.PAR,HMR3
3284  013616  001001                          BNE     5$
3285  013620  104040                          ERROR   40              ;C-D PARITY ERROR NOT SET IN MR3
3286  013622  032737  020000  004336  5$:     BIT     #SPAR,HCS1
3287  013630  001001                          BNE     6$
3288  013632  104041                          ERROR   41              ;SPAR NOT SET AFT WRITING PAT IN MR1
3289  013634  032737  100000  004336  6$:     BIT     #CERR,HCS1
3290  013642  001001                          BNE     TST14           ;;GO TO NEXT TEST
3291  013644  104042                          ERROR   42              ;CERR NOT SET BY WRITING PAT IN MR1
3292
3293                                  ;;**************************************************************
3294                                  ;*TEST 14       VERIFY START SPINDLE COMMAND
3295                                  ;*
3296                                  ;*      THE PROGRAM CHECKS THE ENTIRE STARTUP SEQUENCE, IE:
3297                                  ;*      BRUSH CYCLE, HEADS HOME, FWD, REV ETC.
3298                                  ;*      BY VERIFYING ALL APPROPRIATE STATUS BITS FOR PROPER SEQUENCING.
3299                                  ;*      THE CYLINDER ADDRESS & CYLINDER DIFFERENCE REGS ARE CHECKED
3300                                  ;*      TO BE ZERO AT THE END OF THE SEQUENCE.
3301                                  ;*
3302                                  ;**************************************************************
3303  013646  000004                  TST14:  SCOPE
3304  013650  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
3305  013656  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
3306
3307  013662  004737  050320                  JSR     PC,SUBCLR       ;SUBSYS CLEAR & GET STATUS
3308  013666  104024                          ERROR   24              ;CERR AFTER SCLR
3309
3310  013670  012765  000011  000000          MOV     #SRTSPL,RKCS1(R5) ;START SPINDLE COMMAND
3311  013676  013737  001434  004400          MOV     T10,TEMP1       ;SETUP TIMEOUT
3312  013704  004737  047632                  JSR     PC,FRDY
3313  013710  104121                          ERROR   121             ;RDY NOT SET AFTER START SPIN CMD
3314
3315  013712  004737  050256                  JSR     PC,GSTAT        ;WORD 0
3316  013716  032737  010000  004364          BIT     #D.SPIN,HMR2
3317  013724  001003                          BNE     13$
3318  013726  104306                          ERROR   306             ;SPIN NOT SET AFTER START SPIN CMD
3319  013730  000137  047164                  JMP     $EOP            ;ABORT DRIVE
3320
3321  013734  012737  015472  001176  13$:     MOV     #25$,$ESCAPE
3322  013742  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
```

```
3323  013750  012737  010140  004374          MOV   #<D.SPIN!D.VV!D.DRA>,SBMR2 ;LOAD SHOULD BE VALUE
3324  013756  004737  052504                  JSR   PC,CKMR2          ;CHECK RKMR2
3325  013762  104057                          ERROR 57                ;MSG AO ERROR AFTER START SPIN CMD
3326                                                                  ;REC'D BY DRIVE
3327
3328  013764  005037  004376                  CLR   SBMR3             ;SHOULD BE 0 FOR REMAINDER OF TEST
3329  013770  004737  052566                  JSR   PC,CKMR3          ;CHECK RKMR3
3330  013774  104060                          ERROR 60                ;MSG BO ERROR AFTER START SPIN CMD
3331                                                                  ;REC'D BY DRIVE
3332  013776  012765  000001  000026          MOV   #1,RKMR1(R5)      ;SELECT WORD 1
3333  014004  004737  050256                  JSR   PC,GSTAT          ;GET STATUS
3334  014010  012737  000740  004374          MOV   #<D.CART!D.DOOR!D.HDHM!D.BRHM>,SBMR2 ;LOAD SHOULD BE VALUE
3335  014016  004737  052504                  JSR   PC,CKMR2          ;CHECK MR2
3336  014022  104061                          ERROR 61                ;MSG A1 ERROR AFTER START SPIN CMD
3337                                                                  ;REC'D BY DRIVE
3338  014024  005037  004376                  CLR   SBMR3
3339  014030  004737  052566                  JSR   PC,CKMR3          ;CHECK MR3
3340  014034  104062                          ERROR 62                ;MSG B1 ERROR AFTER START SPIN CMD
3341                                                                  ;REC'D BY DRIVE
3342  014036  005737  004450                  TST   DOTIM
3343  014042  001166                          BNE   3$                ;BRANCH IF P OR L CLOCK PRESENT
3344  014044  012737  015516  001176  1$:     MOV   #30$,$ESCAPE
3345  014052  012765  100000  000000          MOV   #CCLR,RKCS1(R5)
3346  014060  013737  001436  004402          MOV   T100,TEMP2        ;SETUP TIMEOUT
3347  014066  004737  050050                  JSR   PC,FATT1          ;FIND ATTN
3348  014072  104067                          ERROR 67                ;NO ATTN AFTER HEAD LOADING
3349  014074                          2$:
3350
3351  014074  012765  100000  000000          MOV   #CCLR,RKCS1(R5) ;CONTR CLEAR
3352  014102  004737  050256                  JSR   PC,GSTAT          ;GET LATEST STATUS
3353  014106  012737  050340  004374          MOV   #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2 ;SHOULD BE VALUE
3354  014114  004737  052504                  JSR   PC,CKMR2          ;CHECK MR2
3355  014120  104063                          ERROR 63                ;MSG AO ERROR AT END OF HEAD LOAD
3356  014122  005037  004376                  CLR   SBMR3
3357  014126  004737  052566                  JSR   PC,CKMR3          ;CHECK MR3
3358  014132  104064                          ERROR 64                ;MSG BO ERROR AT END OF HEAD LOAD
3359
3360  014134  012765  100000  000000          MOV   #CCLR,RKCS1(R5)
3361  014142  012765  000001  000026          MOV   #1,RKMR1(R5)      ;SELECT WORD 1
3362  014150  004737  050256                  JSR   PC,GSTAT
3363  014154  012737  001720  004374          MOV   #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
3364  014162  004737  052504                  JSR   PC,CKMR2
3365  014166  104065                          ERROR 65                ;MSG A1 ERROR AT END OF HEAD LOAD
3366  014170  005037  004376                  CLR   SBMR3
3367  014174  004737  052566                  JSR   PC,CKMR3
3368  014200  104066                          ERROR 66                ;MSG B1 ERROR AT END OF HEAD LOAD
3369
3370
3371  014202  012765  100000  000000          MOV   #CCLR,RKCS1(R5)
3372  014210  004737  051124                  JSR   PC,RDCYLD         ;READ CYL DIFF IN RKMR2
3373  014214  005737  001402                  TST   CYLDIF
3374  014220  001401                          BEQ   64$
3375  014222  104175                          ERROR 175               ;CYL DIFF/OFFSET NOT CLEARED AT END OF HEAD LOAD
3376  014224  004737  051210          64$:    JSR   PC,RDCYLA         ;READ CYL ADDR IN RKMR3
3377  014230  005737  001404                  TST   CYLADD
3378  014234  001401                          BEQ   65$
```

```
3379  014236  104176                          ERROR   176             ;CYL ADDR NOT CLEARED AT END OF HEAD LOAD
3380  014240                          65$:
3381
3382
3383  014240  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
3384  014246  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
3385  014254  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
3386  014262  013737  001434  004400          MOV     T10,TEMP1
3387  014270  004737  047632                  JSR     PC,FRDY         ;FIND RDY
3388  014274  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
3389  014276  004737  050016                  JSR     PC,TSTATN       ;TEST FOR ATTN
3390  014302  000401                          BR      66$
3391  014304  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
3392  014306                          66$:
3393
3394  014306  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
3395  014314  004737  050256                  JSR     PC,GSTAT        ;GET LATEST STATUS
3396  014320  012737  010340  004374          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2      ;SHOULD BE VALUE
3397  014326  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
3398  014332  104273                          ERROR   273             ;MSG AO ERROR AFTER DRIVE CLEAR CMD
3399  014334  005037  004376                  CLR     SBMR3
3400  014340  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
3401  014344  104265                          ERROR   265             ;MSG BO ERROR AFTER DRIVE CLEAR CMD
3402
3403  014346  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
3404  014354  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
3405  014362  004737  050256                  JSR     PC,GSTAT
3406  014366  012737  001720  004374          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
3407  014374  004737  052504                  JSR     PC,CKMR2
3408  014400  104274                          ERROR   274             ;MSG A1 ERROR AFTER DRIVE CLEAR CMD
3409  014402  005037  004376                  CLR     SBMR3
3410  014406  004737  052566                  JSR     PC,CKMR3
3411  014412  104266                          ERROR   266             ;MSG B1 ERROR AFTER DRIVE CLEAR CMD
3412
3413
3414  014414  000137  015262                  JMP     12$
3415
3416  014420  012765  100000  000000  3$:     MOV     #CCLR,RKCS1(R5)
3417  014426  013737  001410  001412          MOV     HZ,COUNT
3418  014434  012737  000074  001414          MOV     #60.,SEC
3419  014442  004737  052342                  JSR     PC,CLKON        ;TURN CLK INTR ON FOR 60 SEC MAX
3420  014446  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
3421  014454  004737  050256          4$:     JSR     PC,GSTAT
3422  014460  032737  002000  004364          BIT     #D.FWD,HMR2
3423  014466  001004                          BNE     5$
3424  014470  005737  001416                  TST     TIMUP           ;IS 60 SEC DELAY UP?
3425  014474  001767                          BEQ     4$              ;BRANCH IF NO & REPEAT
3426  014476  104070                          ERROR   70              ;FWD NOT SET WITHIN 60 SEC FROM
3427                                                                  ;START SPINDLE CMD.
3428  014500  004737  052450          5$:     JSR     PC,CLKOF        ;TURN OFF CLOCK INTERRUPT
3429  014504  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
3430  014512  012737  011610  004400          MOV     #5000.,TEMP1
3431  014520  004737  050224                  JSR     PC,DLY          ;WAIT APPROX 100MS FOR PIP
3432  014524  005065  000026                  CLR     RKMR1(R5)       ;SELECT WORD 0
3433  014530  004737  050256                  JSR     PC,GSTAT
3434  014534  012737  030140  004374          MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,SBMR2
```

```
3435  014542  004737  052504                    JSR     PC,CKMR2        ;CHECK MR2
3436  014546  104071                            ERROR   71              ;MSG AO ERROR AFTER ST SPIN & FWD DETECTED
3437
3438  014550  005037  004376                    CLR     SBMR3
3439  014554  004737  052566                    JSR     PC,CKMR3        ;CHECK MR3
3440  014560  104072                            ERROR   72              ;MSG BO ERROR AFTER ST SPIN & FWD DETECTED
3441
3442  014562  012765  000001  000026            MOV     #1,RKMR1(R5)    ;SELECT WORD 1
3443  014570  004737  050256                    JSR     PC,GSTAT
3444  014574  012737  013700  004374            MOV     #<D.LOAD!D.FWD!D.SPOK!D.CART!D.DOOR!D.BRHM>,SBMR2       ;LOAD SHOULD BE
3445  014602  004737  052504                    JSR     PC,CKMR2        ;CHECK MR2
3446  014606  104073                            ERROR   73              ;MSG A1 ERROR AFTER ST SPIN & FWD DETECTED
3447
3448  014610  005037  004376                    CLR     SBMR3
3449  014614  004737  052566                    JSR     PC,CKMR3        ;CHECK MR3
3450  014620  104074                            ERROR   74              ;MSG B1 ERROR AFT ST SPIN CMD & FWD DETECTED
3451
3452  014622  013737  001410  001412            MOV     HZ,COUNT
3453  014630  012737  000005  001414            MOV     #5,SEC
3454  014636  004737  052342                    JSR     PC,CLKON        ;TURN CLK INTR ON FOR 5 SEC MAX
3455  014642  012765  000001  000026  6$:       MOV     #1,RKMR1(R5)    ;WORD 1
3456  014650  004737  050256                    JSR     PC,GSTAT
3457  014654  032737  002000  004364            BIT     #D.FWD,HMR2
3458  014662  001404                            BEQ     7$
3459  014664  005737  001416                    TST     TIMUP
3460  014670  001764                            BEQ     6$
3461  014672  104075                            ERROR   75              ;FWD NOT CLEARED WITHIN 5 SEC OF MOTION
3462                                                                    ;FROM START SPINDLE CMD.
3463  014674  004737  052450            7$:     JSR     PC,CLKOF        ;TURN OFF CLK INTERRUPT
3464  014700  012765  100000  000000            MOV     #CCLR,RKCS1(R5)
3465  014706  005065  000026                    CLR     RKMR1(R5)       ;SELECT WORD 0
3466  014712  004737  050256                    JSR     PC,GSTAT
3467  014716  012737  030140  004374            MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,SBMR2
3468  014724  004737  052504                    JSR     PC,CKMR2        ;CHECK MR2
3469  014730  104076                            ERROR   76              ;MSG AO ERROR AT INNER LIM DETECT
3470
3471  014732  005037  004376                    CLR     SBMR3
3472  014736  004737  052566                    JSR     PC,CKMR3        ;CHECK MR3
3473  014742  104077                            ERROR   77              ;MSG BO ERROR AT INNER LIM DETECT
3474
3475  014744  012765  000001  000026            MOV     #1,RKMR1(R5)    ;SELECT WORD 1
3476  014752  004737  050256                    JSR     PC,GSTAT
3477  014756  012737  025720  004374            MOV     #<D.RTZ!D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
3478  014764  004737  052504                    JSR     PC,CKMR2        ;CHECK MR2
3479  014770  104100                            ERROR   100             ;MSG A1 ERROR AT INNER LIMIT DETECT
3480
3481  014772  005037  004376                    CLR     SBMR3
3482  014776  004737  052566                    JSR     PC,CKMR3        ;CHECK MR3
3483  015002  104101                            ERROR   101             ;MSG B1 ERROR AT INNER LIMIT DETECT.
3484
3485  015004  013737  001410  001412            MOV     HZ,COUNT
3486  015012  012737  000004  001414            MOV     #4,SEC
3487  015020  004737  052342                    JSR     PC,CLKON        ;TURN CLK INTR ON FOR 4 SEC MAX
3488  015024  012765  000001  000026  8$:       MOV     #1,RKMR1(R5)    ;WORD 1
3489  015032  004737  050256                    JSR     PC,GSTAT
3490  015036  032737  002000  004364            BIT     #D.FWD,HMR2
```

```
3491  015044  001004                        BNE     9$
3492  015046  005737  001416                TST     TIMUP
3493  015052  001764                        BEQ     8$
3494  015054  104102                        ERROR   102             ;FWD NOT DETECTED WITHIN 4 SEC IN RTZ PORTION OF
3495                                                                 ;START SPINDLE CMD.
3496  015056  004737  052450          9$:   JSR     PC,CLKOF        ;TURN CLOCK INTR OFF.
3497  015062  012765  100000  000000        MOV     #CCLR,RKCS1(R5)
3498  015070  005065  000026                CLR     RKMR1(R5)
3499  015074  004737  050256                JSR     PC,GSTAT
3500  015100  012737  030140  004374        MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,SBMR2
3501  015106  004737  052504                JSR     PC,CKMR2        ;CHECK MR2
3502  015112  104103                        ERROR   103             ;MSG A0 ERROR FROM OUTER LIMIT TO CYL 0
3503                                                                 ;DURING LOADING
3504  015114  005037  004376                CLR     SBMR3
3505  015120  004737  052566                JSR     PC,CKMR3        ;CHECK MR3
3506  015124  104104                        ERROR   104             ;MSG B0 ERROR FROM OUTER LIMIT TO CYL 0
3507                                                                 ;DURING LOADING
3508  015126  012765  000001  000026        MOV     #1,RKMR1(R5)    ;SELECT WORD 1
3509  015134  004737  050256                JSR     PC,GSTAT
3510  015140  012737  023720  004374        MOV     #<D.RTZ!D.FWD!D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
3511  015146  004737  052504                JSR     PC,CKMR2        ;CHECK MR2
3512  015152  104105                        ERROR   105             ;MSG A1 ERROR FROM OUTER LIMIT TO CYL 0
3513                                                                 ;DURING LOADING
3514  015154  005037  004376                CLR     SBMR3
3515  015160  004737  052566                JSR     PC,CKMR3        ;CHECK MR3
3516  015164  104106                        ERROR   106             ;MSG B1 ERROR FROM OUTER LIMIT TO CYL 0
3517                                                                 ;DURING LOADING
3518  015166  013737  001410  001412        MOV     HZ,COUNT
3519  015174  012737  000001  001414        MOV     #1,SEC
3520  015202  004737  052342                JSR     PC,CLKON        ;TURN CLK INTR ON FOR 1 SEC MAX
3521  015206  005065  000026          10$:  CLR     RKMR1(R5)       ;WORD 0
3522  015212  004737  050256                JSR     PC,GSTAT
3523  015216  032737  000200  004364        BIT     #D.DRDY,HMR2    ;SEE IF DRIVE READY
3524  015224  001004                        BNE     11$
3525  015226  005737  001416                TST     TIMUP
3526  015232  001765                        BEQ     10$
3527  015234  104107                        ERROR   107             ;DRIVE READY NOT SET WITHIN 1 SEC FROM
3528                                                                 ;FWD IN RTZ PORTION OF START SPIN CMD
3529  015236  004737  052450          11$:  JSR     PC,CLKOF        ;TURN CLOCKS OFF
3530  015242  004737  050016                JSR     PC,TSTATN       ;TEST ATTN
3531  015246  104067                        ERROR   67              ;NO ATTN AFTER START SPIN CMD
3532  015250  012737  015516  001176        MOV     #30$,$ESCAPE
3533  015256  000137  014074                JMP     2$              ;CHECK RKMR 2 & 3 WORDS 0 & 1
3534
3535  015262  005037  001176          12$:  CLR     $ESCAPE
3536  015266  012765  100000  000000        MOV     #CCLR,RKCS1(R5)
3537  015274  005065  000026                CLR     RKMR1(R5)
3538  015300  004737  050256                JSR     PC,GSTAT
3539  015304  032737  010000  004364        BIT     #D.SPIN,HMR2
3540  015312  001003                        BNE     14$
3541  015314  104306                        ERROR   306             ;SPIN NOT SET AFTER ST. SPIN CMD
3542  015316  000137  047164                JMP     $EOP            ;ABORT DRIVE
3543  015322                          14$:
3544  015322  004737  052770                JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
3545  015326  000505                        BR      TST15           ;;GO TO NEXT TEST
3546                                                                 ;RETURN HERE IF SW 14 IS SET OR
```

```
3547
3548  015330                          20$:                                    ;SW 8 WITH SWR <7:0> APPLY
3549
3550  015330  004737  050320                  JSR     PC,SUBCLR
3551  015334  104024                          ERROR   24                      ;CERR AFTER SCLR
3552
3553  015336  012765  000007  000000          MOV     #UNLOAD,RKCS1(R5)       ;UNLOAD CMD TO RECONDITION DRIVE
3554  015344  013737  001434  004400          MOV     T10,TEMP1
3555  015352  004737  047632                  JSR     PC,FRDY                 ;FIND RDY
3556  015356  104011                          ERROR   11                      ;RDY NOT SET AFTER UNLOAD CMD.
3557  015360  004737  050016                  JSR     PC,TSTATN
3558  015364  104012                          ERROR   12                      ;NO ATTN AFTER UNLOAD CMD
3559
3560  015366  004737  050320                  JSR     PC,SUBCLR
3561  015372  104024                          ERROR   24                      ;CERR AFTER SCLR
3562
3563  015374  013737  001446  001160          MOV     T50000,$TMPO    ;SETUP TIMEOUT
3564  015402  012765  100000  000000  67$:     MOV     #CCLR,RKCS1(R5)
3565  015410  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
3566  015416  004737  050256                  JSR     PC,GSTAT
3567  015422  032737  001000  004364          BIT     #D.SPOK,HMR2    ;SEE IF SPEED OK GONE
3568  015430  001404                          BEQ     68$             ;BR IF YES
3569  015432  005337  001160                  DEC     $TMPO           ;ELSE TRY AGAIN IF TIME NOT UP
3570  015436  001361                          BNE     67$
3571  015440  104315                          ERROR   315             ;SPEED NOT DOWN BY TIMEOUT
3572
3573  015442  004737  050320          68$:    JSR     PC,SUBCLR
3574  015446  104024                          ERROR   24                      ;CERR AFTER SCLR
3575
3576  015450  005037  001176                  CLR     $ESCAPE
3577  015454  005737  001430                  TST     LPFLG
3578  015460  001402                          BEQ     69$
3579  015462  000177  163422                  JMP     @$LPERR         ;SW 9 WAS SET.
3580  015466  000177  163414          69$:    JMP     @$LPADR         ;SW 14 OR 8 WAS SET
3581
3582
3583
3584  015472  004737  052450          25$:    JSR     PC,CLKOF
3585  015476  005237  001430                  INC     LPFLG
3586  015502  032777  001000  163430          BIT     #SW9,@SWR       ;LOOP ON ERROR?
3587  015510  001307                          BNE     20$             ;YES, RECONDITION DRIVE
3588  015512  000137  014044                  JMP     1$              ;RETURN TO MAINLINE
3589  015516  004737  052450          30$:    JSR     PC,CLKOF
3590  015522  005237  001430                  INC     LPFLG
3591  015526  032777  001000  163404          BIT     #SW9,@SWR       ;LOOP ON ERROR?
3592  015534  001275                          BNE     20$             ;YES, RECONDITION DRIVE
3593  015536  000137  015262                  JMP     12$             ;RETURN TO MAINLINE
3594                                  .SBTTL SEEK/READ HEADER/WRITE HEADER TESTS
3595
3596                                  ;;*****************************************************************
3597                                  ;*TEST 15       STATIC CYLINDER DIFF AND CYLINDER ADDR REG TEST; PART 1
3598                                  ;*
3599                                  ;*      THIS TEST CHECKS EACH BIT OF THE CYLINDER DIFFERENCE
3600                                  ;*      AND CYLINDER ADDRESS REGISTERS BY PERFORMING SEEKS TO ALL
3601                                  ;*      MAJOR CYLINDERS (0,1,2,4,8,16,32,64,128,256) WITH EVEN PARITY SET.
3602                                  ;*      THIS FREEZES THE INFORMATION IN THE ABOVE REGISTERS & ALLOWS FOR CHECKING.
```

```
                                        ;*      THIS TEST VERIFIES C-D PARITY ERROR BIT SET, THAT HEADS DID
                                        ;*      NOT MOVE & ALL OTHER APPLICABLE STATUS BITS & REGS.
                                        ;*
                                        ;;*****************************************************************
015542  000004                TST15:  SCOPE
015544  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
015552  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR

015556  005000                          CLR     R0              ;CYL # REGISTER
015560  012737  100000  004410          MOV     #BIT15,TEMP5

015566                          1$:
015566  104415                          SCOP1
015570  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR

015574  004737  050320                  JSR     PC,SUBCLR
015600  104024                          ERROR   24              ;CERR AFTER SCLR


015602  012765  100000  000030          MOV     #CCLR,RKCS1(R5)
015610  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
015616  012765  000013  000000          MOV     #RECAL,RKCS1(R5)        ;RECAL CMD
                                                                ;RESET CYL DIFF/OFFSET & CYL ADDR REG
                                                                ;IN RKMR2 & RKMR3 RESP.
015624  013737  001434  004400          MOV     T10,TEMP1
015632  004737  047632                  JSR     PC,FRDY         ;FIND RDY
015636  104124                          ERROR   124             ;RDY NOT SET AFTER RECAL CMD

015640  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
015646  004737  050256                  JSR     PC,GSTAT
015652  032737  020000  004364          BIT     #D.RTZ,HMR2
015660  001001                          BNE     64$
015662  104244                          ERROR   244             ;RTZ NOT SET DURING RECAL CMD
015664  013737  001432  004402  64$:    MOV     T1,TEMP2        ;SETUP TIMEOUT
015672  004737  050050                  JSR     PC,FATT1        ;FIND ATTN
015676  104055                          ERROR   55              ;NO ATTN AFTER RECAL CMD

015700  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
015706  004737  050256                  JSR     PC,GSTAT        ;GET LATEST STATUS
015712  012737  050340  004374          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2 ;SHOULD BE VALUE
015720  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
015724  104221                          ERROR   221             ;MSG AO ERROR AFTER RECAL CMD
015726  005037  004376                  CLR     SBMR3
015732  004737  052566                  JSR     PC CKMR3        ;CHECK MR3
015736  104275                          ERROR   27              ;MSG BO ERROR AFTER RECAL CMD

015740  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
015746  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
015754  004737  050256                  JSR     PC,GSTAT
015760  012737  001720  004374          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
015766  004737  052504                  JSR     PC,CKMR2
015772  104222                          ERROR   222             ;MSG A1 ERROR AFTER RECAL CMD
015774  005037  004376                  CLR     SBMR3
016000  004737  052566                  JSR     PC,CKMR3
016004  104276                          ERROR   276             ;MSG B1 ERROR AFTER RECAL CMD
```

```
3659
3660  016006  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
3661  016014  004737  051124                  JSR     PC,RDCYLD       ;READ CYL DIFF IN RKMR2
3662  016020  005737  001402                  TST     CYLDIF
3663  016024  001401                          BEQ     65$
3664  016026  104047                          ERROR   47              ;CYL DIFF/OFFSET NOT CLEARED AFTER RECAL CMD
3665  016030  004737  051210          65$:    JSR     PC,RDCYLA       ;READ CYL ADDR IN RKMR3
3666  016034  005737  001404                  TST     CYLADD
3667  016040  001401                          BEQ     66$
3668  016042  104050                          ERROR   50              ;CYL ADDR NOT CLEARED AFTER RECAL CMD
3669  016044                          66$:
3670
3671
3672  016044  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
3673  016052  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
3674  016060  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
3675  016066  013737  001434  004400          MOV     T10,TEMP1
3676  016074  004737  047632                  JSR     PC,FRDY         ;FIND RDY
3677  016100  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
3678  016102  004737  050016                  JSR     PC,TSTATN       ;TEST FOR ATTN
3679  016106  000401                          BR      67$
3680  016110  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
3681  016112                          67$:
3682
3683  016112  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
3684  016120  004737  050256                  JSR     PC,GSTAT        ;GET LATEST STATUS
3685  016124  012737  010340  004374          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2     ;SHOULD BE VALUE
3686  016132  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
3687  016136  104273                          ERROR   273             ;MSG AO ERROR AFTER DRIVE CLEAR CMD
3688  016140  005037  004376                  CLR     SBMR3
3689  016144  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
3690  016150  104265                          ERROR   265             ;MSG BO ERROR AFTER DRIVE CLEAR CMD
3691
3692  016152  012765  000000  000000          MOV     #CCLR,RKCS1(R5)
3693  016160  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
3694  016166  004737  050256                  JSR     PC,GSTAT
3695  016172  012737  001720  004374          MOV     #<0.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
3696  016200  004737  052504                  JSR     PC,CKMR2
3697  016204  104274                          ERROR   274             ;MSG A1 ERROR AFTER DRIVE CLEAR CMD
3698  016206  005037  004376                  CLR     SBMR3
3699  016212  004737  052566                  JSR     PC,CKMR3
3700  016216  104266                          ERROR   266             ;MSG B1 ERROR AFTER DRIVE CLEAR CMD
3701
3702
3703
3704  016220  104415                          SCOP1
3705  016222  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
3706
3707  016226  004737  050320                  JSR     PC,SUBCLR
3708  016232  104024                          ERROR   24              ;CERR AFTER SCLR
3709
3710  016234  012765  000020  000026          MOV     #PAT,RKMR1(R5)  ;EVEN PARITY
3711  016242  010065  000020                  MOV     R0,RKDC(R5)     ;CYL ADDR
3712  016246  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD.
3713  016254  013737  001434  004400          MOV     T10,TEMP1
3714  016262  004737  047632                  JSR     PC,FRDY         ;FIND RDY
```

```
3715  016266  104122                          ERROR   122             ;NO RDY FROM SEEK WITH BAD PARITY
3716  016270  004737  050016                  JSR     PC,TSTATN       ;TEST FOR ATTN
3717  016274  104125                          ERROR   125             ;NO ATTN FROM SEEK & BAD PARITY
3718  016276  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
3719  016304  010065  000020                  MOV     R0,RKDC(R5)     ;RESTORE CYL ADDR IN RKDS AFT CLR
3720  016310  004737  051210                  JSR     PC,RDCYLA       ;READ CYL ADDR
3721  016314  020037  001404                  CMP     R0,CYLADD
3722  016320  001401                          BEQ     2$
3723  016322  104043                          ERROR   43              ;CYL ADDR IN MR3 NOT=RKDC
3724  016324  004737  051124          2$:     JSR     PC,RDCYLD       ;READ CYL DIFF
3725  016330  020037  001402                  CMP     R0,CYLDIF
3726  016334  001401                          BEQ     3$
3727  016336  104044                          ERROR   44              ;CYL DIFF IN MR2 NOT=RKDC
3728
3729  016340  012765  100000  000000  3$:     MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
3730  016346  005065  000026                  CLR     RKMR1(R5)       ;SELECT WORD 0
3731  016352  004737  050256                  JSR     PC,GSTAT
3732  016356  012737  050340  004374          MOV     #<0.DSC!0.SPIN!0.DRDY!0.VV!0.DRA>,SBMR2
3733  016364  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
3734  016370  104110                          ERROR   110             ;MSG A0 ERROR AFTER SEEK WITH BAD PARITY
3735  016372  012737  001200  004376          MOV     #<0.FLT!0.PAR>,SBMR3
3736  016400  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
3737  016404  104111                          ERROR   111             ;MSG B0 ERROR AFTER SEEK WITH BAD PARITY.
3738  016406  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
3739  016414  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
3740  016422  004737  050256                  JSR     PC,GSTAT
3741  016426  012737  001720  004374          MOV     #<0.SPOK!0.CART!0.DOOR!0.BRHM!0.TFOK>,SBMR2
3742  016434  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
3743  016440  104146                          ERROR   146             ;MSG A1 ERROR AFTER SEEK WITH BAD PARITY
3744  016442  005037  004376                  CLR     SBMR3
3745  016446  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
3746  016452  104147                          ERROR   147             ;MSG B1 ERROR AFTER SEEK WITH BAD PARITY
3747  016454  006137  004410                  ROL     TEMP5           ;SET CARRY ONLY ONCE
3748  016460  006100                          ROL     R0              ;SELECT NEXT MAJOR CYL
3749  016462  020027  001000                  CMP     R0,#1000        ;ALL MAJOR CYL DONE?
3750  016466  001001                          BNE     4$              ;BRANCH IF NO
3751  016470  000402                          BR      TST16           ;;GO TO  NEXT TST
3752  016472  000137  015566          45:     JMP     1$
3753
3754                                          ;;*****************************************************************
3755                                          ;*TEST 16       STATIC CYLINDER DIFF & CYLINDER ADDR REG TEST-PART 2
3756                                          ;*
3757                                          ;*      THIS TEST CHECKS THE ABILITY OF THE DRIVE TO PROPERLY SET THE CYL
3758                                          ;*      DIFF. & CYL ADDR REGS FOR ALL COMBINATIONS BY SEEKING TO
3759                                          ;*      ALL CYLINDERS FROM EVERY OTHER CYLINDER. (N SQUARE SEEKS).
3760                                          ;*      IT IS PERFORMED IN THE SAME MANNER AS THE ABOVE TEST.
3761                                          ;*
3762                                          ;;*****************************************************************
3763  016476  000004                  TST16:  SCOPE
3764  016500  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
3765  016506  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
3766
3767  016512  005737  001360                  TST     BYPT16
3768  016516  001404                          BEQ     13$
3769  016520  104401  060377                  TYPE    ,MSG9           ;BYPASSING TEST 16
3770  016524  000137  020056                  JMP     12$
```

```
3771
3772  016530  005737  001216          13$:    TST     $PASS           ;TYPE MSG ONLY FOR FIRST PASS
3773  016534  001002                          BNE     11$
3774  016536  104401  060321                  TYPE    ,MSG8           ;PLEASE WAIT, LONG TEST
3775
3776  016542  005037  001370          11$:    CLR     FRCYL           ;FROM CYL
3777  016546  005037  001372                  CLR     TOCYL           ;TO CYL
3778  016552  005037  001374                  CLR     CCYL            ;CURRENT CYL
3779  016556  005037  001376                  CLR     PCYL            ;PREV CYL
3780
3781  016562  104415                          SCOPI
3782  016564  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
3783
3784  016570  004737  050320                  JSR     PC,SUBCLR
3785  016574  104024                          ERROR   24              ;CERR AFTER SCLR
3786
3787
3788  016576  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
3789  016604  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
3790  016612  012765  000013  000000          MOV     #RECAL,RKCS1(R5)        ;RECAL CMD
3791                                                                  ;RESET CYL DIFF/OFFSET & CYL ADDR REG
3792                                                                  ;IN RKMR2 & RKMR3 RESP.
3793  016620  013737  001434  004400          MOV     T10,TEMP1
3794  016626  004737  047632                  JSR     PC,FRDY         ;FIND RDY
3795  016632  104124                          ERROR   124             ;RDY NOT SET AFTER RECAL CMD
3796
3797  016634  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
3798  016642  004737  050256                  JSR     PC,GSTAT
3799  016646  032737  020000  004364          BIT     #D.RTZ,HMR2
3800  016654  001001                          BNE     64$
3801  016656  104244                          ERROR   244             ;RTZ NOT SET DURING RECAL CMD
3802  016660  013737  001432  004402  64$:    MOV     T1,TEMP2        ;SETUP TIMEOUT
3803  016666  004737  050050                  JSR     PC,FATT1        ;FIND ATTN
3804  016672  104055                          ERROR   55              ;NO ATTN AFTER RECAL CMD
3805
3806  016674  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
3807  016702  004737  050256                  JSR     PC,GSTAT        ;GET LATEST STATUS
3808  016706  012737  050340  004374          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2 ,SHOULD BE VALUE
3809  016714  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
3810  016720  104221                          ERROR   221             ;MSG AO ERROR AFTER RECAL CMD
3811  016722  005037  004376                  CLR     SBMR3
3812  016726  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
3813  016732  104275                          ERROR   275             ;MSG BO ERROR AFTER RECAL CMD
3814
3815  016734  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
3816  016742  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
3817  016750  004737  050256                  JSR     PC,GSTAT
3818  016754  012737  001720  004374          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
3819  016762  004737  052504                  JSR     PC,CKMR2
3820  016766  104222                          ERROR   222             ;MSG A1 ERROR AFTER RECAL CMD
3821  016770  005037  004376                  CLR     SBMR3
3822  016774  004737  052566                  JSR     PC,CKMR3
3823  017000  104276                          ERROR   276             ;MSG B1 ERROR AFTER RECAL CMD
3824
3825
3826  017002  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
```

```
3827  017010  004737  051124                JSR     PC,RDCYLD       ;READ CYL DIFF IN RKMR2
3828  017014  005737  001402                TST     CYLDIF
3829  017020  001401                       BEQ     65$
3830  017022  104047                       ERROR   47              ;CYL DIFF/OFFSET NOT CLEARED AFTER RECAL CMD
3831  017024  004737  051210          65$: JSR     PC,RDCYLA       ;READ CYL ADDR IN RKMR3
3832  017030  005737  001404                TST     CYLADD
3833  017034  001401                       BEQ     66$
3834  017036  104050                       ERROR   50              ;CYL ADDR NOT CLEARED AFTER RECAL CMD
3835  017040                          66$:
3836
3837
3838  017040  012765  100000  000000        MOV     #CCLR,RKCS1(R5)
3839  017046  013765  001222  000010        MOV     $UNIT,RKCS2(R5) ;DRIVE#
3840  017054  012765  000005  000000        MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
3841  017062  013737  001434  004400        MOV     T10,TEMP1
3842  017070  004737  047632                JSR     PC,FRDY         ;FIND RDY
3843  017074  104151                       ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
3844  017076  004737  050016                JSR     PC,TSTATN       ;TEST FOR ATTN
3845  017102  000401                       BR      67$
3846  017104  104154                       ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
3847  017106                          67$:
3848
3849  017106  012765  100000  000000        MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
3850  017114  004737  050256                JSR     PC,GSTAT        ;GET LATEST STATUS
3851  017120  012737  010340  004374        MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2     ;SHOULD BE VALUE
3852  017126  004737  052504                JSR     PC,CKMR2        ;CHECK MR2
3853  017132  104273                       ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
3854  017134  005037  004376                CLR     SBMR3
3855  017140  004737  052566                JSR     PC,CKMR3        ;CHECK MR3
3856  017144  104265                       ERROR   265             ;MSG B0 ERROR AFTER DRIVE CLEAR CMD
3857
3858  017146  012765  100000  000000        MOV     #CCLR,RKCS1(R5)
3859  017154  012765  000001  000026        MOV     #1,RKMR1(R5)    ;SELECT WORD 1
3860  017162  004737  050256                JSR     PC,GSTAT
3861  017166  012737  001720  004374        MOV     #<0.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
3862  017174  004737  052504                JSR     PC,CKMR2
3863  017200  104274                       ERROR   274             ;MSG A1 ERROR AFTER DRIVE CLEAR CMD
3864  017202  005037  004376                CLR     SBMR3
3865  017206  004737  052566                JSR     PC,CKMR3
3866  017212  104266                       ERROR   266             ;MSG B1 ERROR AFTER DRIVE CLEAR CMD
3867
3868
3869
3870  017214  104415                       SCOP1
3871  017216  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR
3872
3873  017222  004737  050320                JSR     PC,SUBCLR
3874  017226  104024                       ERROR   24              ;CERR AFTER SCLR
3875
3876
3877  017230  012765  000020  000026  1$:   MOV     #PAT,RKMR1(R5)  ;EVEN PARITY
3878  017236  013765  001372  000020        MOV     TOCYL,RKDC(R5)  ;SET TO CYL ADDR
3879  017244  013737  001372  001374        MOV     TOCYL,CCYL      ;CURRENT CYL
3880  017252  013737  001374  004404        MOV     CCYL,TEMP3
3881  017260  013737  001376  004406        MOV     PCYL,TEMP4      ;PREV CYL
3882  017266  163737  004404  004406        SUB     TEMP3,TEMP4
```

```
3883  017274  100002                          BPL     2$              ;BR IF TEMP4 IS POS
3884  017276  005437  004406                  NEG     TEMP4
3885  017302  013737  004406  001400  2$:     MOV     TEMP4,CALDIF
3886  017310  013737  001374  001376          MOV     CCYL,PCYL
3887  017316  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD.
3888  017324  013737  001434  004400          MOV     T10,TEMP1
3889  017332  004737  047632                  JSR     PC,FRDY         ;FIND RDY
3890  017336  104122                          ERROR   122             ;NO RDY AFTER SEEK WITH BAD PARITY
3891  017340  004737  050016                  JSR     PC,TSTATN       ;TEST FOR ATTN
3892  017344  104125                          ERROR   125             ;NO ATTN FROM SEEK & BAD PARITY
3893  017346  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CLEAR ERROR
3894  017354  013765  001372  000020          MOV     TOCYL,RKDC(R5)  ;RESTOR RKDC AFT CCLR
3895  017362  004737  051210                  JSR     PC,RDCYLA       ;READ CYL ADDR
3896  017366  023737  001372  001404          CMP     TOCYL,CYLADD    ;SEE IF TO CYL ECHOED OK
3897  017374  001401                          BEQ     3$
3898  017376  104045                          ERROR   45              ;MR3 NOT=RKDC
3899
3900  017400  004737  051124          3$:     JSR     PC,RDCYLD       ;READ CYL DIFF
3901  017404  023737  001400  001402          CMP     CALDIF,CYLDIF   ;SEE IF CYL DIFF CORRECT
3902  017412  001401                          BEQ     4$
3903  017414  104046                          ERROR   46              ;CYL DIFF IN RKMR2 INCORRECT
3904
3905  017416                          4$:
3906
3907  017416  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
3908  017424  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
3909  017432  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
3910  017440  013737  001434  004400          MOV     T10,TEMP1
3911  017446  004737  047632                  JSR     PC,FRDY         ;FIND RDY
3912  017452  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
3913  017454  004737  050016                  JSR     PC,TSTATN       ;TEST FOR ATTN
3914  017460  000401                          BR      68$
3915  017462  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
3916  017464                          68$:
3917
3918
3919  017464  104415                          SCOP1
3920  017466  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
3921
3922  017472  004737  050320                  JSR     PC,SUBCLR
3923  017476  104024                          ERROR   24              ;CERR AFTER SCLR
3924
3925  017500  012765  000020  000026          MOV     #PAT,RKMR1(R5)  ;EVEN PARITY
3926  017506  013765  001370  000020          MOV     FRCYL,RKDC(R5)  ;SET RETURN CYL ADDR
3927  017514  013737  001370  001374          MOV     FRCYL,CCYL
3928  017522  013737  001374  004404          MOV     CCYL,TEMP3
3929  017530  013737  001376  004406          MOV     PCYL,TEMP4
3930  017536  163737  004404  004406          SUB     TEMP3,TEMP4
3931  017544  100002                          BPL     5$              ;BR IF TEMP4 IS POS
3932  017546  005437  004406                  NEG     TEMP4
3933  017552  013737  004406  001400  5$:     MOV     TEMP4,CALDIF
3934  017560  013737  001374  001376          MOV     CCYL,PCYL
3935  017566  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD
3936  017574  013737  001434  004400          MOV     T10,TEMP1
3937  017602  004737  047632                  JSR     PC,FRDY         ;FIND RDY
3938  017606  104122                          ERROR   122             ;NO RDY AFTER SEEK WITH BAD PARITY
```

```
3939  017610  004737  050016                  JSR     PC,TSTATN       ;TEST FOR ATTN
3940  017614  104125                          ERROR   125             ;NO ATTN FROM SEEK & BAD PARITY
3941  017616  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CLEAR ERROR
3942  017624  013765  001370  000020          MOV     FRCYL,RKDC(R5)  ;RESTOR RKDC AFT CCLR
3943  017632  004737  051210                  JSR     PC,RDCYLA       ;READ CYL ADDR
3944  017636  023737  001370  001404          CMP     FRCYL,CYLADD    ;SEE IF RETURN CYL ECHOED OK
3945  017644  001401                          BEQ     6$
3946  017646  104241                          ERROR   241             ;MR3 NOT=RKDC
3947
3948  017650  023737  001372  001370  6$:     CMP     TOCYL,FRCYL     ;SEE IF TO=FROM
3949  017656  001022                          BNE     10$             ;DO NORMAL TEST IF NO
3950  017660  005737  001372                  TST     TOCYL           ;SEE IF=0
3951  017664  001007                          BNE     9$
3952  017666  004737  051124                  JSR     PC,RDCYLD       ;CYL DIFF S/B 0 ON CYL 0
3953  017672  005737  001402                  TST     CYLDIF
3954  017676  001421                          BEQ     7$
3955  017700  104242                          ERROR   242             ;CYL DIFF IN RKMR2 INCORRECT
3956  017702  000417                          BR      7$
3957
3958  017704  004737  051124          9$:     JSR     PC,RDCYLD       ;CYL DIFF/OFFSET SHOULD NOT
3959  017710  023727  001402  000001          CMP     CYLDIF,#1       ;CHANGE IN SEEK TO SELF
3960  017716  001411                          BEQ     7$              ;SHOULD = 1 IN THIS TEST
3961  017720  104263                          ERROR   263             ;CYL DIFF IN RKMR2 DID NOT REMAIN = 0
3962  017722  000407                          BR      7$
3963  017724  004737  051124          10$:    JSR     PC,RDCYLD       ;READ CYL DIFF
3964  017730  023737  001400  001402          CMP     CALDIF,CYLDIF   ;SEE IF CYL DIFF OK
3965  017736  001401                          BEQ     7$
3966  017740  104242                          ERROR   242             ;CYL DIFF IN RKMR2 INCORRECT
3967
3968  017742                          7$:
3969
3970  017742  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
3971  017750  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
3972  017756  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
3973  017764  013737  001434  004400          MOV     T10,TEMP1
3974  017772  004737  047632                  JSR     PC,FRDY         ;FIND RDY
3975  017776  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
3976  020000  004737  050016                  JSR     PC,TSTATN       ;TEST FOR ATTN
3977  020004  000401                          BR      69$
3978  020006  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
3979  020010                          69$:
3980
3981
3982  020010  005237  001372                  INC     TOCYL
3983  020014  023727  001372  000633          CMP     TOCYL,#411.     ;SEE IF SCANNED ALL CYLINDERS
3984  020022  001402                          BEQ     8$              ;BR IF YES
3985  020024  000137  017230                  JMP     1$              ;ELSE REPEAT
3986
3987  020030  005237  001370          8$:     INC     FRCYL
3988  020034  023727  001370  000633          CMP     FRCYL,#411.     ;SEE IF ALL DONE
3989  020042  001405                          BEQ     TST17           ;;GO TO NEXT TST
3990  020044  013737  001370  001372          MOV     FRCYL,TOCYL     ;FRCYL ALWAYS = OR > TOCYL
3991  020052  000137  017230                  JMP     1$              ;ELSE REPEAT
3992  020056                          12$:
3993
3994
```

```
                                        ;;*****************************************************************
                                        ;*TEST 17       HEAD REGISTER TEST
                                        ;*
                                        ;*      THIS TEST CHECKS THE ABILITY TO SELECT ALL HEADS (0,1,2)
                                        ;*      VIA RKDA & READING BACK FROM MSG B3 BY THE SELECT DRIVE CMD.
                                        ;*      HEAD 3 IS CHECKED TO PRODUCE INV. ADDR.
                                        ;*
                                        ;*      SINCE CHANGING HEAD ADDRESSES ARE TIED TO SEEK COMMANDS,
                                        ;*      SELECTING HEAD 3 MUST RESULT IN A SEEK INCOMPLETE ALONG WITH
                                        ;*      ILLEGAL ADDRESS. IF NOT,THIS MEANS THAT CHANGING HEAD ADDRESSES
                                        ;*      ARE NOT TIED TO SEEK COMMANDS
                                        ;*
                                        ;;*****************************************************************
020056  000004                        TST17:  SCOPE
020060  012737  000001  001174                MOV     #1,$TIMES       ;;DO 1 ITERATION
020066  012706  001100                        MOV     #STACK,SP       ;RESTORE STK PTR

020072  005000                                CLR     R0              ;HEAD #
020074                                  1$:
020074  104415                                SCOP1
020076  012706  001100                        MOV     #STACK,SP       ;RESTORE STK PTR

020102  004737  050320                        JSR     PC,SUBCLR
020106  104024                                ERROR   24              ;CERR AFTER SCLR


020110  012765  100000  000000                MOV     #CCLR,RKCS1(R5)
020116  013765  001222  000010                MOV     $UNIT,RKCS2(R5)
020124  012765  000013  000000                MOV     #RECAL,RKCS1(R5)        ;RECAL CMD
                                                                        ;RESET CYL DIFF/OFFSET & CYL ADDR REG
                                                                        ;IN RKMR2 & RKMR3 RESP.
020132  013737  001434  004400                MOV     T10,TEMP1
020140  004737  047632                        JSR     PC,FRDY         ;FIND RDY
020144  104124                                ERROR   124             ;RDY NOT SET AFTER RECAL CMD

020146  012765  000001  000026                MOV     #1,RKMR1(R5)    ;SELECT WORD 1
020154  004737  050256                        JSR     PC,GSTAT
020160  032737  020000  004364                BIT     #D.RTZ,HMR2
020166  001001                                BNE     64$
020170  104244                                ERROR   244             ;RTZ NOT SET DURING RECAL CMD
020172  013737  001432  004402          64$:  MOV     T1,TEMP2        ;SETUP TIMEOUT
020200  004737  050050                        JSR     PC,FATT1        ;FIND ATTN
020204  104055                                ERROR   55              ;NO ATTN AFTER RECAL CMD

020206  012765  100000  000000                MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
020214  004737  050256                        JSR     PC,GSTAT        ;GET LATEST STATUS
020220  012737  050340  004374                MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2 ;SHOULD BE VALUE
020226  004737  052504                        JSR     PC,CKMR2        ;CHECK MR2
020232  104221                                ERROR   221             ;MSG AO ERROR AFTER RECAL CMD
020234  005037  004376                        CLR     SBMR3
020240  004737  052566                        JSR     PC,CKMR3        ;CHECK MR3
020244  104275                                ERROR   275             ;MSG BO ERROR AFTER RECAL CMD

020246  012765  100000  000000                MOV     #CCLR,RKCS1(R5)
020254  012765  000001  000026                MOV     #1,RKMR1(R5)    ;SELECT WORD 1
```

```
4051  020262  004737  050256                  JSR     PC,GSTAT
4052  020266  012737  001720  004374          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
4053  020274  004737  052504                  JSR     PC,CKMR2
4054  020300  104222                          ERROR   222             ;MSG A1 ERROR AFTER RECAL CMD
4055  020302  005037  004376                  CLR     SBMR3
4056  020306  004737  052566                  JSR     PC,CKMR3
4057  020312  104276                          ERROR   276             ;MSG B1 ERROR AFTER RECAL CMD
4058
4059
4060  020314  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
4061  020322  004737  051124                  JSR     PC,RDCYLD       ;READ CYL DIFF IN RKMR2
4062  020326  005737  001402                  TST     CYLDIF
4063  020332  001401                          BEQ     65$
4064  020334  104047                          ERROR   47              ;CYL DIFF/OFFSET NOT CLEARED AFTER RECAL CMD
4065  020336  004737  051210          65$:    JSR     PC,RDCYLA       ;READ CYL ADDR IN RKMR3
4066  020342  005737  001404                  TST     CYLADD
4067  020346  001401                          BEQ     66$
4068  020350  104050                          ERROR   50              ;CYL ADDR NOT CLEARED AFTER RECAL CMD
4069  020352                          66$:
4070
4071
4072  020352  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
4073  020360  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
4074  020366  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
4075  020374  013737  001434  004400          MOV     T10,TEMP1
4076  020402  004737  047632                  JSR     PC,FRDY         ;FIND RDY
4077  020406  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
4078  020410  004737  050016                  JSR     PC,TSTATN       ;TEST FOR ATTN
4079  020414  000401                          BR      67$
4080  020416  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
4081  020420                          67$:
4082
4083  020420  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
4084  020426  004737  050256                  JSR     PC,GSTAT        ;GET LATEST STATUS
4085  020432  012737  010340  004374          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2     ;SHOULD BE VALUE
4086  020440  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
4087  020444  104273                          ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
4088  020446  005037  004375                  CLR     SBMR3
4089  020452  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
4090  020456  104265                          ERROR   265             ;MSG B0 ERROR AFTER DRIVE CLEAR CMD
4091
4092  020460  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
4093  020466  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
4094  020474  004737  050256                  JSR     PC,GSTAT
4095  020500  012737  001720  004374          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
4096  020506  004737  052504                  JSR     PC,CKMR2
4097  020512  104274                          ERROR   274             ;MSG A1 ERROR AFTER DRIVE CLEAR CMD
4098  020514  005037  004376                  CLR     SBMR3
4099  020520  004737  052566                  JSR     PC,CKMR3
4100  020524  104266                          ERROR   266             ;MSG B1 ERROR AFTER DRIVE CLEAR CMD
4101
4102
4103
4104  020526  012765  000003  000026          MOV     #3,RKMR1(R5)    ;SELECT WORD 3
4105  020534  004737  050256                  JSR     PC,GSTAT        ;GET STATUS
4106  020540  013737  004366  004400          MOV     HMR3,TEMP1
```

```
4107  020546  042737  170777  004400          BIC     #^C<M.HEAD>,TEMP1
4108  020554  023727  004400  001000          CMP     TEMP1,#BIT9     ;HEAD 0=BIT 9
4109  020562  001401                          BEQ     3$
4110  020564  104053                          ERROR   53              ;RECAL DID NOT RESET HEAD REG IN RKMR3.
4111
4112  020566                          3$:
4113  020566  104415                          SCOP1
4114  020570  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
4115
4116  020574  004737  050320                  JSR     PC,SUBCLR
4117  020600  104024                          ERROR   24              ;CERR AFTER SCLR
4118
4119  020602  000300                          SWAB    R0
4120  020604  010065  000006                  MOV     R0,RKDA(R5)     ;HEAD #
4121  020610  000300                          SWAB    R0
4122
4123  020612  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD
4124  020620  013737  001434  004400          MOV     T10,TEMP1
4125  020626  004737  047632                  JSR     PC,FRDY         ;FIND RDY
4126  020632  104156                          ERROR   156             ;NO RDY AFTER SEEK TO SELF
4127  020634  004737  050016                  JSR     PC,TSTATN
4128  020640  104157                          ERROR   157             ;NO ATTN AFTER SEEK TO SELF
4129  020642  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CLEAR CONTR ERROR
4130  020650  000300                          SWAB    R0
4131  020652  010065  000006                  MOV     R0,RKDA(R5)     ;RESTOR RKDA AFTER CCLR
4132  020656  000300                          SWAB    R0
4133  020660  020027  000003                  CMP     R0,#3
4134  020664  002031                          BGE     4$              ;BRANCH FOR HEAD 3
4135
4136  020666  012765  000003  000026          MOV     #3,RKMR1(R5)    ;SELECT WORD 3
4137  020674  004737  050256                  JSR     PC,GSTAT
4138  020700  013737  004366  004402          MOV     HMR3,TEMP2      ;HERE FOR HEADS 0,1,2
4139  020706  042737  170777  004402          BIC     #^C<M.HEAD>,TEMP2
4140  020714  006237  004402                  ASR     TEMP2           ;RIGHT JUSTIFY
4141  020720  000337  004402                  SWAB    TEMP2           ;HEAD DECODE
4142  020724  005037  004400                  CLR     TEMP1
4143  020730  116037  004326  004400          MOVB    ATTN(R0),TEMP1
4144  020736  023737  004400  004402          CMP     TEMP1,TEMP2     ;FOR RKDA=HEAD 0, HEAD=1 IN RKMR3
4145                                                                  ;FOR RKDA=HEAD 1, HEAD=2 IN RKMR3
4146                                                                  ;FOR RKDA=HEAD 2, HEAD=4 IN RKMR3
4147  020744  001401                          BEQ     4$
4148  020746  104054                          ERROR   54              ;HEAD DECODE IN RKMR3 NOT=RKDA
4149
4150  020750  005065  000026          4$:     CLR     RKMR1(R5)       ;SELECT WORD 0
4151  020754  004737  050256                  JSR     PC,GSTAT
4152  020760  012737  050340  004374          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2
4153  020766  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
4154  020772  104114                          ERROR   114             ;MSG A0 ERROR AFTER LOADING HEAD REG
4155                                                                  ;& SEEK CMD
4156  020774  020027  000003                  CMP     R0,#3
4157  021000  002003                          BGE     5$
4158
4159  021002  005037  004376                  CLR     SBMR3           ;HEADS 0,1,2
4160  021006  000403                          BR      6$
4161  021010  012737  002240  004376  5$:     MOV     #<D.SKI!D.FLT!D.IDAE>,SBMR3  ;HEAD 3
4162  021016  004737  052566          6$:     JSR     PC,CKMR3        ;CHECK MR3
```

```
4163  021022  104115                          ERROR   115             ;MSG 80 ERROR AFTER LOADING HEAD REG
4164                                                                  ;& SEEK CMD
4165
4166  021024  005200                          INC     R0
4167  021026  020027  000004                  CMP     R0,#4           ;0 THRU 3 DONE?
4168  021032  001402                          BEQ     7$              ;BR IF YES
4169  021034  000137  020074                  JMP     1$              ;ELSE REPEAT
4170
4171  021040                          7$:
4172
4173  021040  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
4174  021046  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
4175  021054  012765  000013  000000          MOV     #RECAL,RKCS1(R5)        ;RECAL CMD
4176  021062  013737  001434  004400          MOV     T10,TEMP1
4177  021070  004737  047632                  JSR     PC,FRDY         ;FIND RDY
4178  021074  104124                          ERROR   124             ;RDY NOT FOUND AFTER RECAL CMD
4179
4180  021076  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
4181  021104  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
4182  021112  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
4183  021120  013737  001434  004400          MOV     T10,TEMP1
4184  021126  004737  047632                  JSR     PC,FRDY         ;FIND RDY
4185  021132  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
4186  021134  004737  050016                  JSR     PC,TSTATN       ;TEST FOR ATTN
4187  021140  000401                          BR      69$
4188  021142  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
4189  021144                          69$:
4190
4191
4192  021144  004737  050256                  JSR     PC,GSTAT
4193  021150  032737  000040  004366          BIT     #D.IDAE,HMR3    ;SEE IF IDAE IS CLEARED
4194  021156  001401                          BEQ     68$             ;BR IF YES
4195  021160  104155                          ERROR   155             ;IDAE NOT CLEARED AFTER RECAL CMD
4196
4197  021162  012765  100000  000000  68$:    MOV     #CCLR,RKCS1(R5)
4198  021170  013737  001432  004402          MOV     T1,TEMP2        ;LOOK FOR ATTN FROM RECAL
4199  021176  004737  050050                  JSR     PC,FATT1
4200  021202  104055                          ERROR   55              ;NO ATTN AFTER RECAL CMD
4201
4202
4203                                  ;;*****************************************************************
4204                                  ;*TEST 20       SEEK TO CYLINDER 0
4205                                  ;*
4206                                  ;*      TESTS THE ABILITY TO DO A SEEK COMMAND.
4207                                  ;*      VERIFIES THERE WAS NO MOVEMENT BY CHECKING ALL APPROPRIATE
4208                                  ;*      STATUS BITS.  VERIFIES COMMAND COMPLETION BETWEEN 10-15USEC.
4209                                  ;*      READ HEADER IS NOT PERFORMED AS THE PACK MAY NOT BE FORMATTED.
4210                                  ;*
4211                                  ;;*****************************************************************
4212  021204  000004                  TST20:  SCOPE
4213  021206  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
4214  021214  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
4215
4216  021220  004737  050320                  JSR     PC,SUBCLR       ;SUBSYS CLEAR & GET STATUS
4217  021224  104024                          ERROR   24              ;CERR AFTER SCLR
4218  021226  004737  051210                  JSR     PC,RDCYLA       ;READ CYL ADDR IN RKMR3
```

```
4219  021232  005737  001404          TST     CYLADD
4220  021236  001401                  BEQ     1$
4221  021240  104130                  ERROR   130             ;CYL ADDR NOT CLEARED AFTER SCLR
4222  021242                  1$:
4223  021242  104415                  SCOP1
4224  021244  012706  001100          MOV     #STACK,SP       ;RESTORE STK PTR
4225
4226  021250  004737  050320          JSR     PC,SUBCLR
4227  021254  104024                  ERROR   24              ;CERR AFTER SCLR
4228
4229  021256  012765  000017  000000  MOV     #SEEK,RKCS1(R5) ;SEEK CMD: SEEK TO SELF
4230  021264  012737  000005  004400  MOV     #5,TEMP1        ;SETUP 100US TIMEOUT
4231  021272  004737  047632          JSR     PC,FRDY         ;FIND RDY & GET STATUS
4232  021276  104131                  ERROR   131             ;RDY NOT SET AFTER SEEK CMD
4233  021300  012737  000005  004400  MOV     #5,TEMP1        ;SETUP 100US TIMEOUT
4234
4235  021306  004737  050144          JSR     PC,FATT2        ;FIND ATTN
4236  021312  104132                  ERROR   132             ;NO ATTN AFTER SEEK CMD
4237  021314  032737  100000  004336  BIT     #CERR,HCS1
4238  021322  001401                  BEQ     64$
4239  021324  104210                  ERROR   210             ;CERR AFTER SEEK CMD
4240  021326                  64$:
4241
4242  021326  012765  100000  000000  MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
4243  021334  004737  050256          JSR     PC,GSTAT        ;GET LATEST STATUS
4244  021340  012737  050340  004374  MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2 ;SHOULD BE VALUE
4245  021346  004737  052504          JSR     PC,CKMR2        ;CHECK MR2
4246  021352  104133                  ERROR   133             ;MSG AO ERROR AFTER SEEK CMD
4247  021354  005037  004376          CLR     SBMR3
4248  021360  004737  052566          JSR     PC,CKMR3        ;CHECK MR3
4249  021364  104134                  ERROR   134             ;MSG BO ERROR AFTER SEEK CMD
4250
4251  021366  012765  100000  000000  MOV     #CCLR,RKCS1(R5)
4252  021374  012765  000001  000026  MOV     #1,RKMR1(R5)    ;SELECT WORD 1
4253  021402  004737  050256          JSR     PC,GSTAT
4254  021406  012737  001720  004374  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
4255  021414  004737  052504          JSR     PC,CKMR2
4256  021420  104135                  ERROR   135             ;MSG A1 ERROR AFTER SEEK CMD
4257  021422  005037  004376          CLR     SBMR3
4258  021426  004737  052566          JSR     PC,CKMR3
4259  021432  104136                  ERROR   136             ;MSG B1 ERROR AFTER SEEK CMD
4260
4261
4262  021434  004737  051124          JSR     PC,RDCYLD       ;READ CYL DIFF IN MSG A2
4263  021440  005737  001402          TST     CYLDIF
4264  021444  001401                  BEQ     65$
4265  021446  104137                  ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
4266
4267  021450                  65$:
4268
4269  021450  012765  100000  000000  MOV     #CCLR,RKCS1(R5)
4270  021456  013765  001222  000010  MOV     $UNIT,RKCS2(R5) ;DRIVE#
4271  021464  012765  000005  000000  MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
4272  021472  013737  001434  004400  MOV     T10,TEMP1
4273  021500  004737  047632          JSR     PC,FRDY         ;FIND RDY
4274  021504  104151                  ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
```

```
 4275  021506  004737  050016                    JSR     PC,TSTATN       ;TEST FOR ATTN
 4276  021512  000401                            BR      66$
 4277  021514  104154                            ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
 4278  021516                            66$:
 4279
 4280  021516  012765  100000  000000            MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
 4281  021524  004737  050256                    JSR     PC,GSTAT        ;GET LATEST STATUS
 4282  021530  012737  010340  004374            MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2     ;SHOULD BE VALUE
 4283  021536  004737  052504                    JSR     PC,CKMR2        ;CHECK MR2
 4284  021542  104273                            ERROR   273             ;MSG AO ERROR AFTER DRIVE CLEAR CMD
 4285  021544  005037  004376                    CLR     SBMR3
 4286  021550  004737  052566                    JSR     PC,CKMR3        ;CHECK MR3
 4287  021554  104265                            ERROR   265             ;MSG BO ERROR AFTER DRIVE CLEAR CMD
 4288
 4289  021556  012765  100000  000000            MOV     #CCLR,RKCS1(R5)
 4290  021564  012765  000001  000026            MOV     #1,RKMR1(R5)    ;SELECT WORD 1
 4291  021572  004737  050256                    JSR     PC,GSTAT
 4292  021576  012737  001720  004374            MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
 4293  021604  004737  052504                    JSR     PC,CKMR2
 4294  021610  104274                            ERROR   274             ;MSG A1 ERROR AFTER DRIVE CLEAR CMD
 4295  021612  005037  004376                    CLR     SBMR3
 4296  021616  004737  052566                    JSR     PC,CKMR3
 4297  021622  104266                            ERROR   266             ;MSG B1 ERROR AFTER DRIVE CLEAR CMD
 4298
 4299
 4300  021624  004737  051210                    JSR     PC,RDCYLA       ;READ CYL ADDR IN MR3
 4301  021630  005737  001404                    TST     CYLADD
 4302  021634  001401                            BEQ     TST21           ;;GO TO NEXT TEST
 4303  021636  104140                            ERROR   140             ;CYL ADDR IN MR3 NOT CLEARED AFT SEEK CMD.
 4304
 4305                            ;;**************************************************************
 4306                            ;*TEST 21       TEST SECTOR COUNT REG. IN MSG B3
 4307                            ;;**************************************************************
 4308  021640  000004            TST21:  SCOPE
 4309  021642  012737  000001  001174            MOV     #1,$TIMES       ;;DO 1 ITERATION
 4310  021650  012706  001100                    MOV     #STACK,SP       ;RESTORE STK PTR
 4311
 4312  021654  004737  050320                    JSR     PC,SUBCLR       ;SUBSYS CLEAR & GET STATUS
 4313  021660  104024                            ERROR   24              ;CERR AFTER SCLR
 4314  021662  012737  022012  001176            MOV     #2$,$ESCAPE     ;GO TO NEXT TEST IF ANY ERROR DETECTED
 4315
 4316  021670  012737  000025  001420            MOV     #21.,SECNT      ;22 SECTOR FORMAT TEST
 4317
 4318  021676  004737  050574                    JSR     PC,FSO22                ;FIND SECTOR 0
 4319  021702  104142                            ERROR   142             ;SECTOR 0 NOT FOUND BY TIMEOUT
 4320
 4321  021704  005037  001422                    CLR PSEC                ;PREVIOUS SECTOR
 4322  021710  004737  050744            64$:    JSR     PC,FNS22                ;FIND NEXT SECTOR
 4323  021714  104143                            ERROR   143             ;DIFFERENT SECTOR  NOT FOUND BY TIMEOUT
 4324  021716  013737  001422  001424            MOV     PSEC,ESEC
 4325  021724  062737  000001  001424            ADD     #1,ESEC         ;SETUP EXPECTED SECTOR
 4326  021732  013737  001426  001422            MOV     SECTOR,PSEC     ;UPDATE PREV SECTOR
 4327  021740  004737  050400                    JSR     PC,RDSEC                ;READ SECTOR
 4328  021744  023737  001426  001422            CMP     SECTOR,PSEC
 4329  021752  001407                            BEQ     65$             ;BR IF READ SAME TWICE
 4330  021754  004737  050400                    JSR     PC,RDSEC
```

```
4331  021760  023737  001426  001422          CMP     SECTOR,PSEC
4332  021766  001401                          BEQ     65$                 ;TRY 1 MORE TIME
4333  021770  104144                          ERROR   144                 ;MSG B3 ERROR, SECTOR REG UNSTABLE
4334                                                                      ;MAY BE DURING SECTOR PULSE TIME
4335  021772  023737  001426  001424  65$:    CMP     SECTOR,ESEC
4336  022000  001401                          BEQ     66$
4337  022002  104145                          ERROR   145                 ;MSG B3 ERROR BETWEEN SECTOR COUNTS
4338  022004  005337  001420          66$:    DEC     SECNT
4339  022010  001337                          BNE     64$                 ;BR IF SECTOR COUNT NOT DONE
4340
4341
4342  022012  005037  001176          2$:     CLR     $ESCAPE
4343
4344                                  ;;*****************************************************************
4345                                  ;*TEST 22        DETECT OUTER LIMIT
4346                                  ;*
4347                                  ;*      THIS TEST VERIFIES THAT THE ABOVE TEST DID ACTUALLY POSITION ON CYLINDER 0
4348                                  ;*      BY DETECTING OUTER LIMIT AS THE ADJACENT CYLINDER.
4349                                  ;*      AN ERROR IN THIS TEST INDICATES:
4350                                  ;*
4351                                  ;*              A.  HEADS WERE NOT ON CYLINDER 0
4352                                  ;*      AND/OR  B.  COULD NOT SEEK IN REVERSE DIRECTION.
4353                                  ;*
4354                                  ;;*****************************************************************
4355  022016  000004                  TST22:  SCOPE
4356  022020  012737  000001  001174          MOV     #1,$TIMES           ;;DO 1 ITERATION
4357  022026  012706  001100                  MOV     #STACK,SP           ;RESTORE STK PTR
4358  022032  004737  050320                  JSR     PC,SUBCLR           ;SUBSYS CLEAR & GET STATUS
4359  022036  104024                          ERROR   24                  ;CERR AFTER SCLR
4360
4361  022040  005037  001430                  CLR     LPFLG
4362
4363  022044  012765  000020  000026          MOV     #PAT,RKMR1(R5)  ;PARITY & WORD 0
4364  022052  012765  000001  000020          MOV     #1,RKDC(R5)     ;CYL 1
4365  022060  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD
4366  022066  013737  001434  004400          MOV     TIO,TEMP1
4367  022074  004737  047632                  JSR     PC,FRDY         ;FIND RDY
4368  022100  104122                          ERROR   122             ;NO RDY FROM SEEK WITH BAD PARITY
4369  022102  004737  050016                  JSR     PC,TSTATN       ;TEST FOR ATTN
4370  022106  104125                          ERROR   125             ;NO ATTN FROM SEEK WITH BAD PARITY
4371  022110  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
4372  022116  004737  050256                  JSR     PC,GSTAT
4373  022122  012737  050340  004374          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2
4374  022130  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
4375  022134  104110                          ERROR   110             ;MSG A0 ERROR AFTER SEEK WITH BAD PARITY
4376  022136  012737  001200  004376          MOV     #<D.FLT!D.PAR>,SBMR3
4377  022144  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
4378  022150  104111                          ERROR   111             ;MSG B0 ERROR AFTER SEEK WITH BAD PARITY
4379
4380  022152  005065  000026                  CLR     RKMR1(R5)       ;REMOVE PARITY & SELECT WORD 0
4381
4382  022156  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
4383  022164  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
4384  022172  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
4385  022200  013737  001434  004400          MOV     TIO,TEMP1
4386  022206  004737  047632                  JSR     PC,FRDY         ;FIND RDY
```

```
4387  022212  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
4388  022214  004737  050016                  JSR     PC,TSTATN       ;TEST FOR ATTN
4389  022220  000401                          BR      64$
4390  022222  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
4391  022224                          64$:
4392
4393  022224  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
4394  022232  004737  050256                  JSR     PC,GSTAT        ;GET LATEST STATUS
4395  022236  012737  010340  004374          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2    ;SHOULD BE VALUE
4396  022244  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
4397  022250  104273                          ERROR   273             ;MSG AO ERROR AFTER DRIVE CLEAR CMD
4398  022252  005037  004376                  CLR     SBMR3
4399  022256  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
4400  022262  104265                          ERROR   265             ;MSG BO ERROR AFTER DRIVE CLEAR CMD
4401
4402  022264  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
4403  022272  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
4404  022300  004737  050256                  JSR     PC,GSTAT
4405  022304  012737  001720  004374          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
4406  022312  004737  052504                  JSR     PC,CKMR2
4407  022316  104274                          ERROR   274             ;MSG A1 ERROR AFTER DRIVE CLEAR CMD
4408  022320  005037  004376                  CLR     SBMR3
4409  022324  004737  052566                  JSR     PC,CKMR3
4410  022330  104266                          ERROR   266             ;MSG B1 ERROR AFTER DRIVE CLEAR CMD
4411
4412
4413  022332  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
4414  022340  004737  050256                  JSR     PC,GSTAT
4415  022344  012737  010340  004374          MOV     #<D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2
4416  022352  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
4417  022356  104152                          ERROR   152             ;MSG AC ERROR AFTER CLEAR CMD
4418  022360  005037  004376                  CLR     SBMR3           ;MR3 SHOULD BE CLEAR
4419  022364  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
4420  022370  104153                          ERROR   153             ;MSG BO ERROR AFTER CLEAR CMD
4421
4422  022372  012765  000000  000020          MOV     #0,RKDC(R5)     ;CYL 0
4423  022400  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK TO CYL 0
4424  022406  013737  001434  004400          MOV     T10,TEMP1
4425  022414  004737  047632                  JSR     PC,FRDY         ;FIND RDY
4426  022420  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD
4427  022422  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
4428  022430  004737  050256                  JSR     PC,GSTAT
4429  022434  004737  051260                  JSR     PC,FLIM         ;FIND LIMIT DETECT
4430  022440  104160                          ERROR   160             ;LIMIT DETECT NOT FOUND BEFORE TIMEOUT
4431
4432  022442  032737  040000  004364          BIT     #D.UNLD,HMR2
4433  022450  001003                          BNE     1$
4434  022452  104305                          ERROR   305             ;DRIVE NOT UNLOADING AFTER LIMIT DETECT
4435  022454  000137  023270                  JMP     30$             ;BYPASS REST OF TEST
4436
4437  022460  012737  023204  001176  1$:     MOV     #20$,$ESCAPE    ;MUST ESCAPE TO CYCLE UP DRIVE & TEST SWR
4438  022466  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
4439  022474  005065  000026                  CLR     RKMR1(R5)       ;SELECT WORD 0
4440  022500  004737  050256                  JSR     PC,GSTAT
4441  022504  012737  070140  004374          MOV     #<D.DSC!D.PIP!D.SPIN!D.VV!D.DRA>,SBMR2
4442  022512  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
```

```
4443  022516  104161                          ERROR   161             ;MSG AO ERROR AFTER OUTER LIMIT DETECT
4444  022520  012737  002200  004376          MOV     #<D.SKI!D.FLT>,SBMR3
4445  022526  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
4446  022532  104162                          ERROR   162             ;MSG BO ERROR AFTER OUTER LIMIT DETECT
4447  022534  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
4448  022542  012765  000001  000026          MOV     #1,RKMR1(R5)    ;WORD 1
4449  022550  004737  050256                  JSR     PC,GSTAT
4450  022554  012737  045720  004374          MOV     #<D.UNLD!D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
4451  022562  004737  052504                  JSR     PC,CKMR2        ;CHECK MF2
4452  022566  104163                          ERROR   163             ;MSG A1 ERROR AFTER OUTER LIMIT DETECT
4453  022570  004737  050016                  JSR     PC,TSTATN
4454  022574  104165                          ERROR   165             ;NO ATTN AFTER OUTER LIMIT DETECT
4455
4456  022576  004737  050320                  JSR     PC,SUBCLR       ;SUBSYS CLR
4457  022602  104024                          ERROR   24              ;CERR AFTER SCLR
4458  022604  013737  001434  004402          MOV     T10,TEMP2       ;SET UP TIMEOUT
4459  022612  004737  051336                  JSR     PC,FHDHM        ;FIND HEAD HOME
4460  022616  104166                          ERROR   166             ;HEAD HOME NOT FOUND BEFORE TIMEOUT
4461  022620  004737  051412                  JSR     PC,FLOAD        ;FIND LOAD HEADS
4462  022624  104167                          ERROR   167             ;LOAD HEADS NOT FOUND BEFORE TIMEOUT
4463  022626  013737  001434  004402          MOV     T10,TEMP2       ;SETUP TIMEOUT
4464  022634  004737  050050                  JSR     PC,FATT1        ;FIND ATTN
4465  022640  104067                          ERROR   67              ;ATTN NOT FOUND BEFORE TIMEOUT
4466  022642  005037  001176          25:     CLR     $ESCAPE
4467
4468  022646  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
4469  022654  004737  050256                  JSR     PC,GSTAT        ;GET LATEST STATUS
4470  022660  012737  050340  004374          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2 ;SHOULD BE VALUE
4471  022666  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
4472  022672  104063                          ERROR   63              ;MSG AO ERROR AT END OF HEAD LOADING
4473  022674  005037  004376                  CLR     SBMR3
4474  022700  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
4475  022704  104064                          ERROR   64              ;MSG BO ERROR AT END OF HEAD LOADING
4476
4477  022706  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
4478  022714  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
4479  022722  004737  050256                  JSR     PC,GSTAT
4480  022726  012737  001720  004374          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
4481  022734  004737  052504                  JSR     PC,CKMR2
4482  022740  104065                          ERROR   65              ;MSG A1 ERROR AT END OF HEAD LOADING
4483  022742  005037  004376                  CLR     SBMR3
4484  022746  004737  052566                  JSR     PC,CKMR3
4485  022752  104066                          ERROR   66              ;MSG B1 ERROR AT END OF HEAD LOADING
4486
4487
4488  022754  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
4489  022762  004737  051124                  JSR     PC,RDCYLD       ;READ CYL DIFF IN RKMR2
4490  022766  005737  001402                  TST     CYLDIF
4491  022772  001401                          BEQ     65$
4492  022774  104175                          ERROR   175             ;CYL DIFF/OFFSET NOT CLEARED AT END OF HEAD LOADING
4493  022776  004737  051210          65$:    JSR     PC,RDCYLA       ;READ CYL ADDR IN RKMR3
4494  023002  005737  001404                  TST     CYLADD
4495  023006  001401                          BEQ     66$
4496  023010  104176                          ERROR   176             ;CYL ADDR NOT CLEARED AT END OF HEAD LOADING
4497  023012                          66$:
4498
```

```
4499
4500  023012  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
4501  023020  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
4502  023026  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
4503  023034  013737  001434  004400          MOV     T10,TEMP1
4504  023042  004737  047632                  JSR     PC,FRDY         ;FIND RDY
4505  023046  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
4506  023050  004737  050016                  JSR     PC,TSTATN       ;TEST FOR ATTN
4507  023054  000401                          BR      67$
4508  023056  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
4509  023060                          67$:
4510
4511  023060  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
4512  023066  004737  050256                  JSR     PC,GSTAT        ;GET LATEST STATUS
4513  023072  012737  010340  004374          MOV     #<C!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2     ;SHOULD BE VALUE
4514  023100  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
4515  023104  104273                          ERROR   273             ;MSG AO ERROR AFTER DRIVE CLEAR CMD
4516  023106  005037  004376                  CLR     SBMR3
4517  023112  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
4518  023116  104265                          ERROR   265             ;MSG BO ERROR AFTER DRIVE CLEAR CMD
4519
4520  023120  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
4521  023126  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
4522  023134  004737  050256                  JSR     PC,GSTAT
4523  023140  012737  001720  004374          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
4524  023146  004737  052504                  JSR     PC,CKMR2
4525  023152  104274                          ERROR   274             ;MSG A1 ERROR AFTER DRIVE CLEAR CMD
4526  023154  005037  004376                  CLR     SBMR3
4527  023160  004737  052566                  JSR     PC,CKMR3
4528  023164  104266                          ERROR   266             ;MSG B1 ERROR AFTER DRIVE CLEAR CMD
4529
4530
4531  023166  004737  052770                  JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
4532  023172  000436                          BR      TST23           ;;GO TO NEXT TEST
4533                                                                  ;RETURN HERE IF SW 14 IS SET OR
4534                                                                  ;SW 8 WITH SWR <7:0> APPLY
4535
4536
4537  023174  005037  001176          10$:    CLR     $ESCAPE
4538  023200  000177  155702                  JMP     @$LPADR
4539  023204                          20$:
4540
4541  023204  004737  050320                  JSR     PC,SUBCLR
4542  023210  104024                          ERROR   24              ;CERR AFTER SCLR
4543
4544  023212  012765  000011  000000          MOV     #SRTSPL,RKCS1(R5)       ;START SPINDLE CMD
4545  023220  013737  001434  004400          MOV     T10,TEMP1       ;SET TIMEOUT
4546  023226  004737  047632                  JSR     PC,FRDY         ;FIND RDY
4547  023232  104121                          ERROR   121             ;RDY NOT FOUND AFTER ST SPIN CMD.
4548
4549  023234  013737  001436  004402          MOV     T100,TEMP2      ;SETUP  TIMEOUT
4550  023242  004737  050050                  JSR     PC,FATT1        ;FIND ATTN
4551  023246  104067                          ERROR   67              ;NO ATTN AFTER ST SPIN CMD.
4552
4553  023250  005237  001430                  INC     LPFLG
4554  023254  032777  001000  155656          BIT     #SW9,@SWR       ;LOOP ON ERROR?
```

```
4555  023262  001344                          BNE     10$             ;YES, RECONDITION DRIVE
4556  023264  000137  022642                  JMP     2$              ;RETURN TO MAINLINE
4557  023270                          30$:
4558
4559
4560
4561                                  ;;*****************************************************************
4562                                  ;*TEST 23       READ HEADERS, ALL TRACKS, CYLINDER 0
4563                                  ;*
4564                                  ;*      THIS TEST DOES NOT CHECK THE HEADERS, BUT ONLY THE FACT
4565                                  ;*      THAT HEADERS CAN BE READ CORRECTLY BY CHECKING CERR & DTE.
4566                                  ;*
4567                                  ;*      THIS TEST IS VALID ONLY IF THE PACK HAS NOT BEEN ZERO'D OUT.
4568                                  ;*
4569                                  ;*****************************************************************
4570  023270  000004                  TST23:  SCOPE
4571  023272  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
4572  023300  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
4573
4574  023304  005037  001450                  CLR     HEAD            ;HEAD COUNTER
4575
4576  023310                          1$:
4577  023310  104415                          SCOP1
4578  023312  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
4579
4580  023316  004737  050320                  JSR     PC,SUBCLR
4581  023322  104024                          ERROR   24              ;CERR AFTER SCLR
4582
4583  023324  000337  001450                  SWAB    HEAD
4584  023330  013765  001450  000006          MOV     HEAD,RKDA(R5)   ;LOAD HEAD REG
4585  023336  000337  001450                  SWAB    HEAD
4586
4587  023342  012700  001704                  MOV     #RHTAB,R0
4588  023346  012765  000025  000000          MOV     #<RDHEAD>,RKCS1(R5)     ;READ HEADER CMD
4589  023354  013737  001446  004400          MOV     T50000,TEMP1    ;SETUP  TIMEOUT
4590  023362  004737  047632                  JSR     PC,FRDY         ;FIND RDY
4591  023366  104171                          ERROR   171             ;NO RDY AFTER READ HEADER CMD
4592  023370  032737  100000  004336          BIT     #CERR,HCS1
4593  023376  001405                          BEQ     64$
4594  023400  104174                          ERROR   174             ;CERR AFTER READ HEADER CMD
4595  023402  104401  060720                  TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
4596  023406  000137  047164                  JMP     $EOP            ;ABORT DRIVE
4597
4598  023412  016520  000024          64$:    MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
4599  023416  016520  000024                  MOV     RKDB(R5),(R0)+  ;2'ND WORD
4600  023422  016520  000024                  MOV     RKDB(R5),(R0)+  ;3'RD WORD
4601
4602
4603  023426  032765  100000  000010          BIT     #DLT,RKCS2(R5)
4604  023434  001407                          BEQ     65$
4605  023436  004737  050256                  JSR     PC,GSTAT
4606  023442  104173                          ERROR   173             ;DLT AFTER READ HEADER CMD
4607  023444  104401  060720                  TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
4608  023450  000137  047164                  JMP     $EOP            ;ABORT DRIVE
4609  023454                          65$:
4610
```

```
4611  023454  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
4612  023462  004737  050256                  JSR     PC,GSTAT        ;GET LATEST STATUS
4613  023466  012737  010340  004374          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2     ;SHOULD BE VALLE
4614  023474  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
4615  023500  104301                          ERROR   301             ;MSG AO ERROR AFTER READ HEADER CMD
4616  023502  005037  004376                  CLR     SBMR3
4617  023506  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
4618  023512  104271                          ERROR   271             ;MSG BO ERROR AFTER READ HEADER CMD
4619
4620  023514  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
4621  023522  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
4622  023530  004737  050256                  JSR     PC,GSTAT
4623  023534  012737  001720  004374          MOV     #<0.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
4624  023542  004737  052504                  JSR     PC,CKMR2
4625  023546  104302                          ERROR   302             ;MSG A1 ERROR AFTER READ HEADER CMD
4626  023550  005037  004376                  CLR     SBMR3
4627  023554  004737  052566                  JSR     PC,CKMR3
4628  023560  104272                          ERROR   272             ;MSG B1 ERROR AFTER READ HEADER CMD
4629
4630
4631
4632  023562  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
4633  023570  004737  051124                  JSR     PC,RDCYLD       ;READ CYL DIFF IN RKMR2
4634  023574  005737  001402                  TST     CYLDIF
4635  023600  001401                          BEQ     66$
4636  023602  104172                          ERROR   172             ;CYL DIFF/OFFSET NOT CLEARED AFTER READ HEADER CMD
4637  023604  004737  051210          66$:    JSR     PC,RDCYLA       ;READ CYL ADDR IN RKMR3
4638  023610  005737  001404                  TST     CYLADD
4639  023614  001401                          BEQ     67$
4640  023616  104264                          ERROR   264             ;CYL ADDR NOT CLEARED AFTER READ HEADER CMD
4641  023620                          67$:
4642
4643  023620  005237  001450                  INC     HEAD
4644  023624  023727  001450  000003          CMP     HEAD,#3         ;ALL HEADS DONE?
4645  023632  001226                          BNE     1$              ;REPEAT IF NO
4646
4647
4648                                  ;;*****************************************************************
4649                                  ;*TEST 24       BASIC WRITE/READ HEADER & HEAD SWITCHING TEST
4650                                  ;*
4651                                  ;*      THIS TEST CHECKS HEAD SWITCHING BY WRITING UNIQUE HEADERS
4652                                  ;*      ON EACH TRACK OF CYL 0, READING BACK & VERIFYING THEY REMAINED
4653                                  ;*      UNIQUE. 22 SECTOR FORMAT IS USED
4654                                  ;*
4655                                  ;*      I.E.    TRACK 0: ALL 0'S FOR ALL SECTOR HEADERS
4656                                  ;*              TRACK 1: 0101 FOR ALL SECTOR HEADERS
4657                                  ;*              TRACK 2: ALL 1'S FOR ALL SECTOR HEADERS
4658                                  ;*
4659                                  ;;*****************************************************************
4660  023634  000004                  $ST24:  SCOPE
4661  023636  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
4662  023644  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
4663
4664
4665  023650  005037  001450                  CLR     HEAD            ;HEAD CTR
4666
```

```
4667  023654  104415                          SCOP1
4668  023656  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
4669
4670  023662  004737  050320                  JSR     PC,SUBCLR
4671  023666  104024                          ERROR   24              ;CERR AFTER SCLR
4672
4673  023670  052765  000020  000010  1$:     BIS     #BAI,RKCS2(R5)  ;SET BUSS ADDR INCR INHIBIT
4674  023676  012765  001500  000004          MOV     #HDTAB,RKBA(R5) ;HEADER WORD TABLE
4675  023704  012765  177676  000002          MOV     #-66.,RKWC(R5)  ;WORD COUNT.
4676  023712  000337  001450                  SWAB    HEAD
4677  023716  013765  001450  000006          MOV     HEAD,RKDA(R5)   ;SETUP HEAD ADDR
4678  023724  000337  001450                  SWAB    HEAD
4679
4680  023730  013700  001450                  MOV     HEAD,R0
4681  023734  006300                          ASL     R0              ;DOUBLE R0
4682  023736  016037  001462  001500          MOV     DATA0(R0),HDTAB ;SETUP HEADER WORD FOR RKBA
4683
4684  023744  012765  000027  000000          MOV     #<WRHEAD>,RKCS1(R5)     ;WRITE HEADER CMD
4685  023752  013737  001446  004400          MOV     T50000,TEMP1    ;SETUP TIMEOUT
4686  023760  004737  047632                  JSR     PC,FRDY         ;FIND RDY
4687  023764  104200                          ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
4688  023766  004737  050256                  JSR     PC,GSTAT        ;GET FRESH STATUS
4689  023772  032737  100000  004336          BIT     #CERR,HCS1
4690  024000  001405                          BEQ     64$
4691  024002  104201                          ERROR   201             ;CERR AFTER WRITE HEADER CMD
4692  024004  104401  060720                  TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
4693  024010  000137  047164                  JMP     $EOP            ;ABORT DRIVE
4694  024014                          64$:
4695
4696  024014  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
4697  024022  004737  050256                  JSR     PC,GSTAT        ;GET LATEST STATUS
4698  024026  012737  010340  004374          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2     ;SHOULD BE VALUE
4699  024034  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
4700  024040  104277                          ERROR   277             ;MSG AO ERROR AFTER WRITE HEADER CMD
4701  024042  005037  004376                  CLR     SBMR3
4702  024046  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
4703  024052  104267                          ERROR   267             ;MSG BO ERROR AFTER WRITE HEADER CMD
4704
4705  024054  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
4706  024062  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
4707  024070  004737  050256                  JSR     PC,GSTAT
4708  024074  012737  001720  004374          MOV     #<0.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
4709  024102  004737  052504                  JSR     PC,CKMR2
4710  024106  104300                          ERROR   300             ;MSG A1 ERROR AFTER WRITE HEADER CMD
4711  024110  005037  004376                  CLR     SBMR3
4712  024114  004737  052566                  JSR     PC,CKMR3
4713  024120  104270                          ERROR   270             ;MSG B1 ERROR AFTER WRITE HEADER CMD
4714
4715
4716
4717  024122  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
4718  024130  004737  051124                  JSR     PC,RDCYLD       ;READ CYL DIFF IN RKMR2
4719  024134  005737  001402                  TST     CYLDIF
4720  024140  001401                          BEQ     65$
4721  024142  104303                          ERROR   303             ;CYL DIFF/OFFSET NOT CLEARED AFTER WRITE HEADER CMD
4722  024144  004737  051210          65$:    JSR     PC,RDCYLA       ;READ CYL ADDR IN RKMR3
```

```
4723  024150  005737  001404                TST     CYLADD
4724  024154  001401                        BEQ     66$
4725  024156  104304                        ERROR   304             ;CYL ADDR NOT CLEARED AFTER WRITE HEADER CMD
4726  024160                          66$:
4727
4728
4729  024160  005237  001450                INC     HEAD
4730  024164  023727  001450  000003        CMP     HEAD,#3
4731  024172  001236                        BNE     1$
4732
4733  024174  005037  001450                CLR     HEAD            ;HEAD CTR
4734  024200  104415                        SCOP1
4735  024202  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR
4736
4737  024206  004737  050320                JSR     PC,SUBCLR
4738  024212  104024                        ERROR   24              ;CERR AFTER SCLR
4739
4740
4741
4742  024214  000337  001450          2$:   SWAB    HEAD
4743  024220  013765  001450  000006        MOV     HEAD,RKDA(R5)   ;SETUP HEAD ADDR
4744  024226  000337  001450                SWAB    HEAD
4745
4746  024232  012700  001704                MOV     #RHTAB,R0
4747  024236  012765  000025  000000        MOV     #<RDHEAD>,RKCS1(R5)     ;READ HEADER CMD
4748  024244  013737  001446  004400        MOV     T50000,TEMP1    ;SETUP  TIMEOUT
4749  024252  004737  047632                JSR     PC,FRDY         ;FIND RDY
4750  024256  104171                        ERROR   171             ;NO RDY AFTER READ HEADER CMD
4751  024260  032737  100000  004336        BIT     #CERR,HCS1
4752  024266  001405                        BEQ     67$
4753  024270  104174                        ERROR   174             ;CERR AFTER READ HEADER CMD
4754  024272  104401  060720                TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
4755  024276  000137  047164                JMP     $EOP            ;ABORT DRIVE
4756
4757  024302  016520  000024          67$:  MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
4758  024306  016520  000024                MOV     RKDB(R5),(R0)+  ;2'ND WORD
4759  024312  016520  000024                MOV     RKDB(R5),(R0)+  ;3'RD WORD
4760
4761
4762  024316  032765  100000  000010        BIT     #DLT,RKCS2(R5)
4763  024324  001407                        BEQ     68$
4764  024326  004737  050256                JSR     PC,GSTAT
4765  024332  104173                        ERROR   173             ;DLT AFTER READ HEADER CMD
4766  024334  104401  060720                TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
4767  024340  000137  047164                JMP     $EOP            ;ABORT DRIVE
4768  024344                          68$:
4769
4770  024344  012765  100000  000000        MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
4771  024352  004737  050256                JSR     PC,GSTAT        ;GET LATEST STATUS
4772  024356  012737  010340  004374        MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2     ;SHOULD BE VALUE
4773  024364  004737  052504                JSR     PC,CKMR2        ;CHECK MR2
4774  024370  104301                        ERROR   301             ;MSG AO ERROR AFTER READ HEADER CMD
4775  024372  005037  004376                CLR     SBMR3
4776  024376  004737  052566                JSR     PC,CKMR3        ;CHECK MR3
4777  024402  104271                        ERROR   271             ;MSG BO ERROR AFTER READ HEADER CMD
4778
```

```
 4779  024404  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
 4780  024412  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
 4781  024420  004737  050256                  JSR     PC,GSTAT
 4782  024424  012737  001720  004374          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
 4783  024432  004737  052504                  JSR     PC,CKMR2
 4784  024436  104302                          ERROR   302             ;MSG A1 ERROR AFTER READ HEADER CMD
 4785  024440  005037  004376                  CLR     SBMR3
 4786  024444  004737  052566                  JSR     PC,CKMR3
 4787  024450  104272                          ERROR   272             ;MSG B1 ERROR AFTER READ HEADER CMD
 4788
 4789
 4790
 4791  024452  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
 4792  024460  004737  051124                  JSR     PC,RDCYLD       ;READ CYL DIFF IN RKMR2
 4793  024464  005737  001402                  TST     CYLDIF
 4794  024470  001401                          BEQ     69$
 4795  024472  104172                          ERROR   172             ;CYL DIFF/OFFSET NOT CLEARED AFTER READ HEADER CMD
 4796  024474  004737  051210          69$:    JSR     PC,RDCYLA       ;READ CYL ADDR IN RKMR3
 4797  024500  005737  001404                  TST     CYLADD
 4798  024504  001401                          BEQ     70$
 4799  024506  104264                          ERROR   264             ;CYL ADDR NOT CLEARED AFTER READ HEADER CMD
 4800  024510                          70$:
 4801
 4802  024510  000337  001450                  SWAB    HEAD
 4803  024514  013765  001450  000006          MOV     HEAD,RKDA(R5)   ;RESTORE RKDA
 4804  024522  000337  001450                  SWAB    HEAD
 4805
 4806  024526  012701  001704                  MOV     #RHTAB,R1
 4807
 4808  024532  005037  001460                  CLR     WDCNT           ;HEADER WORD COUNT
 4809  024536  013700  001450                  MOV     HEAD,R0
 4810  024542  006300                          ASL     R0              ;DOUBLE R0
 4811  024544  016037  001462  004400          MOV     DATA0(R0),TEMP1 ;GET THE 'SHOULD BE' DATA
 4812  024552  012137  001472          3$:     MOV     (R1)+,HDWD      ;READ HEADER WORD
 4813  024556  023737  001472  004400          CMP     HDWD,TEMP1
 4814  024564  001401                          BEQ     4$
 4815  024566  104202                          ERROR   202             ;READ HEADER MISMATCH
 4816  024570  005237  001460          4$:     INC     WDCNT
 4817  024574  023727  001460  000003          CMP     WDCNT,#3        ;DO ONLY 1 SECTOR
 4818  024602  001363                          BNE     3$
 4819
 4820  024604  005237  001450                  INC     HEAD
 4821  024610  023727  001450  000003          CMP     HEAD,#3         ;ALL 3 HEADS DONE?
 4822  024616  001402                          BEQ     TST25           ;;GO TO NXT TST IF YES
 4823  024620  000137  024214                  JMP     2$              ;ELSE REPEAT
 4824
 4825                                  ;;*****************************************************************
 4826                                  ;*TEST 25       BASIC WRITE/READ HEADER TEST; ALL 1'S, 20 SECTORS
 4827                                  ;*
 4828                                  ;*      USING HEAD 0, WRITE & READ 20 SECTOR HEADERS BY WRITING ALL
 4829                                  ;*      1'S AS HEADERS.  ATTEMPT TO FIND SECTORS 20 & 21.  VERIFY
 4830                                  ;*      THEY ARE NO LONGER THERE BY READING 22 SECTORS AND NOT
 4831                                  ;*      FINDING 0'S AS DATA FROM THE PREVIOUS TEST.
 4832                                  ;*
 4833                                  ;;*****************************************************************
 4834  024624  000004                  TST25:  SCOPE
```

```
4835  024626  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
4836  024634  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
4837
4838  024640  004737  050320                  JSR     PC,SUBCLR
4839  024644  104024                          ERROR   24              ;CERR AFTER SCLR
4840  024646  052765  000020  000010          BIS     #BAI,RKCS2(R5)  ;SET BUSS ADDR INCR INHIBIT
4841  024654  012765  001466  000004          MOV     #DATA1,RKBA(R5) ;XFER 1'S ONLY
4842  024662  012765  177704  000002          MOV     #-60.,RKWC(R5)  ;WORD COUNT
4843
4844
4845  024670  012765  010027  000000          MOV     #<CFMT!WRHEAD>,RKCS1(R5)        ;WRITE HEADER CMD
4846  024676  013737  001446  004400          MOV     T50000,TEMP1    ;SETUP TIMEOUT
4847  024704  004737  047632                  JSR     PC,FRDY         ;FIND RDY
4848  024710  104200                          ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
4849  024712  004737  050256                  JSR     PC,GSTAT        ;GET FRESH STATUS
4850  024716  032737  100000  004336          BIT     #CERR,HCS1
4851  024724  001405                          BEQ     64$
4852  024726  104201                          ERROR   201             ;CERR AFTER WRITE HEADER CMD
4853  024730  104401  060720                  TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
4854  024734  000137  047164                  JMP     $EOP            ;ABORT DRIVE
4855  024740                          64$:
4856
4857  024740  012765  010001  000000          MOV     #<CFMT!SELDRV>,RKCS1(R5)        ;GET 20 SECTOR STATUS
4858  024746  013737  001434  004400          MOV     T10,TEMP1
4859  024754  004737  047632                  JSR     PC,FRDY         ;FIND RDY
4860  024760  104117                          ERROR   117             ;NO RDY AFTER SELDRV CMD
4861  024762  032737  001000  004364          BIT     #D.FORM,HMR2
4862  024770  001001                          BNE     1$
4863  024772  104312                          ERROR   312             ;FORMAT NOT SET AFTER WRITE HDR CMD
4864
4865  024774                          1$:
4866
4867  024774  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
4868  025002  004737  050256                  JSR     PC,GSTAT        ;GET LATEST STATUS
4869  025006  012737  010340  004374          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2     ;SHOULD BE VALUE
4870  025014  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
4871  025020  104277                          ERROR   277             ;MSG AO ERROR AFTER WRITE HEADER CMD
4872  025022  005037  004376                  CLR     SBMR3
4873  025026  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
4874  025032  104267                          ERROR   267             ;MSG BO ERROR AFTER WRITE HEADER CMD
4875
4876  025034  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
4877  025042  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
4878  025050  004737  050256                  JSR     PC,GSTAT
4879  025054  012737  001720  004374          MOV     #<0.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
4880  025062  004737  052504                  JSR     PC,CKMR2
4881  025066  104300                          ERROR   300             ;MSG A1 ERROR AFTER WRITE HEADER CMD
4882  025070  005037  004376                  CLR     SBMR3
4883  025074  004737  052566                  JSR     PC,CKMR3
4884  025100  104270                          ERROR   270             ;MSG B1 ERROR AFTER WRITE HEADER CMD
4885
4886
4887  025102  005037  001420                  CLR     SECNT           ;SECTOR COUNT
4888  025106                          2$:
4889  025106  104415                          SCOP1
4890  025110  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
```

```
4891
4892  025114  004737  050320                  JSR     PC,SUBCLR
4893  025120  104024                          ERROR   24                  ;CERR AFTER SCLR
4894
4895
4896  025122  012700  001704                  MOV     #RHTAB,R0
4897  025126  012765  010025  000000          MOV     #<CFMT!RDHEAD>,RKCS1(R5)        ;READ HEADER CMD
4898  025134  013737  001446  004400          MOV     T50000,TEMP1        ;SETUP  TIMEOUT
4899  025142  004737  047632                  JSR     PC,FRDY             ;FIND RDY
4900  025146  104171                          ERROR   171                 ;NO RDY AFTER READ HEADER CMD
4901  025150  032737  100000  004336          BIT     #CERR,HCS1
4902  025156  001405                          BEQ     65$
4903  025160  104174                          ERROR   174                 ;CERR AFTER READ HEADER CMD
4904  025162  104401  060720                  TYPE    ,MSG18              ;ABORT BALANCE OF TESTS
4905  025166  000137  047164                  JMP     $EOP                ;ABORT DRIVE
4906
4907  025172  016520  000024          65$:    MOV     RKDB(R5),(R0)+      ;1'ST WORD FROM SILO TO RHTAB
4908  025176  016520  000024                  MOV     RKDB(R5),(R0)+      ;2'ND WORD
4909  025202  016520  000024                  MOV     RKDB(R5),(R0)+      ;3'RD WORD
4910
4911
4912  025206  032765  100000  000010          BIT     #DLT,RKCS2(R5)
4913  025214  001407                          BEQ     66$
4914  025216  004737  050256                  JSR     PC,GSTAT
4915  025222  104173                          ERROR   173                 ;DLT AFTER READ HEADER CMD
4916  025224  104401  060720                  TYPE    ,MSG18              ;ABORTING BALANCE OF TESTS
4917  025230  000137  047164                  JMP     $EOP                ;ABORT DRIVE
4918  025234                          66$:
4919
4920  025234  012765  010001  000000          MOV     #<CFMT!SELDRV>,RKCS1(R5)
4921  025242  013737  001434  004400          MOV     T10,TEMP1
4922  025250  004737  047632                  JSR     PC,FRDY             ;FIND RDY
4923  025254  104117                          ERROR   117                 ;NO RDY AFTER SELDRV CMD
4924  025256  032737  001000  004364          BIT     #D.FORM,HMR2
4925  025264  001001                          BNE     6$
4926  025266  104313                          ERROR   313                 ;FORMAT NOT SET AFTER READ HDR CMD
4927
4928  025270                          6$:
4929
4930  025270  012765  100000  000000          MOV     #CCLR,RKCS1(R5)     ;CONTR CLEAR
4931  025276  004737  050256                  JSR     PC,GSTAT            ;GET LATEST STATUS
4932  025302  012737  010340  004374          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2     ;SHOULD BE VALUE
4933  025310  004737  052504                  JSR     PC,CKMR2            ;CHECK MR2
4934  025314  104301                          ERROR   301                 ;MSG A0 ERROR AFTER READ HEADER CMD
4935  025316  005037  004376                  CLR     SBMR3
4936  025322  004737  052566                  JSR     PC,CKMR3            ;CHECK MR3
4937  025326  104271                          ERROR   271                 ;MSG B0 ERROR AFTER READ HEADER CMD
4938
4939  025330  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
4940  025336  012765  000001  000026          MOV     #1,RKMR1(R5)        ;SELECT WORD 1
4941  025344  004737  050256                  JSR     PC,GSTAT
4942  025350  012737  001720  0043            MOV     #<0.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
4943  025356  004737  052504                  JSR     PC,CKMR2
4944  025362  104302                          ERROR   302                 ;MSG A1 ERROR AFTER READ HEADER CMD
4945  025364  005037  004376                  CLR     SBMR3
4946  025370  004737  052566                  JSR     PC,CKMR3
```

```
                                        ERROR   272             ;MSG B1 ERROR AFTER READ HEADER CMD
                                        MOV     #CCLR,RKCS1(R5)
                                        JSR     PC,RDCYLD       ;READ CYL DIFF IN RKMR2
                                        TST     CYLDIF
                                        BEQ     67$
                                        ERROR   172             ;CYL DIFF/OFFSET NOT CLEARED AFTER READ HEADER CMD
                                67$:    JSR     PC,RDCYLA       ;READ CYL ADDR IN RKMR3
                                        TST     CYLADD
                                        BEQ     68$
                                        ERROR   264             ;CYL ADDR NOT CLEARED AFTER READ HEADER CMD
                                68$:
                                        MOV     #RHTAB,R1
                                3$:     CLR     WDCNT           ;HEADER WORD COUNT
                                        MOV     DATA1,TEMP1     ;GET 'SHOULD BE' DATA
                                4$:     MOV     (R1)+,HDWD      ;READ HEADER WORD
                                        CMP     HDWD,TEMP1      ;MATCH OK?
                                        BEQ     5$              ;BR IF YES
                                        ERROR   202             ;READ HEADER MISMATCH
                                5$:     INC     WDCNT
                                        CMP     WDCNT,#3        ;JUST 1 SECTOR AND 1 HEAD
                                        BNE     4$

                        ;;**************************************************************
                        ;*TEST 26       WRITE & READ HEADERS CYLINDER 0, HEAD 0
                        ;;**************************************************************
                        TST26:  SCOPE
                                MOV     #1,$TIMES       ;;DO 1 ITERATION
                                MOV     #STACK,SP       ;RESTORE STK PTR

                                JSR     PC,SUBCLR
                                ERROR   24              ;CERR AFTER CLR
                                MOV     #HDTAB,RKBA(R5) ;HEADER WORD TABLE
                                MOV     #-66.,RKWC(R5)  ;WORD COUNT.
                                MOV     #0,TOCYL

                                MOV     TOCYL,CALADD    ;SETUP
                                MOV     #0,HEAD         ;TO FILL
                                MOV     #0,FORMAT       ;HEADER
                                JSR     PC,FHDTAB       ;TABLE

                                MOV     #0,RKDC(R5)     ;CYL#

                                MOV     #<WRHEAD>,RKCS1(R5)     ;WRITE HEADER CMD
                                MOV     T50000,TEMP1    ;SETUP TIMEOUT
                                JSR     PC,FRDY         ;FIND RDY
                                ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
                                JSR     PC,GSTAT        ;GET FRESH STATUS
                                BIT     #CERR,HCS1
                                BEQ     64$
```

```
025642  104201                              ERROR   201                     ;CERR AFTER WRITE HEADER CMD
025644  104401  060720                      TYPE    ,MSG18                  ;ABORTING BALANCE OF TESTS
025650  000137  047164                      JMP     $EOP                    ;ABORT DRIVE
025654                              64$:

025654  012765  100000  000000              MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
025662  004737  050256                      JSR     PC,GSTAT        ;GET LATEST STATUS
025666  012737  010340  004374              MOV     #<O!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2     ;SHOULD BE VALUE
025674  004737  052504                      JSR     PC,CKMR2        ;CHECK MR2
025700  104277                              ERROR   277             ;MSG A0 ERROR AFTER WRITE HEADER CMD
025702  005037  004376                      CLR     SBMR3
025706  004737  052566                      JSR     PC,CKMR3        ;CHECK MR3
025712  104267                              ERROR   267             ;MSG B0 ERROR AFTER WRITE HEADER CMD

025714  012765  100000  000000              MOV     #CCLR,RKCS1(R5)
025722  012765  000001  000026              MOV     #1,RKMR1(R5)    ;SELECT WORD 1
025730  004737  050256                      JSR     PC,GSTAT
025734  012737  001720  004374              MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
025742  004737  052504                      JSR     PC,CKMR2
025746  104300                              ERROR   300             ;MSG A1 ERROR AFTER WRITE HEADER CMD
025750  005037  004376                      CLR     SBMR3
025754  004737  052566                      JSR     PC,CKMR3
025760  104270                              ERROR   270             ;MSG B1 ERROR AFTER WRITE HEADER CMD


025762  005037  001420                      CLR     SECNT           ;SECTOR COUNT
025766  104415                              SCOP1
025770  012706  001100                      MOV     #STACK,SP       ;RESTORE STK PTR

025774  004737  050320                      JSR     PC,SUBCLR
026000  104024                              ERROR   24              ;CERR AFTER SCLR

026002  012765  000000  000020              MOV     #0,RKDC(R5)     ;CYL #

026010  012700  001704                      MOV     #RHTAB,R0

026014  012765  000025  000000      65$:    MOV     #RDHEAD,RKCS1(R5)       ;READ HEADER CMD
026022  013737  001440  004400              MOV     T500,TEMP1      ;SETUP  TIMEOUT
026030  004737  047632                      JSR     PC,FRDY         ;FIND RDY
026034  104171                              ERROR   171             ;NO RDY AFTER READ HEADER CMD
026036  032737  100000  004336              BIT     #CERR,HCS1
026044  001405                              BEQ     66$
026046  104174                              ERROR   174             ;CERR AFTER READ HEADER  CMD
026050  104401  060720                      TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
026054  000137  047164                      JMP     $EOP            ;ABORT DRIVE

026060  016520  000024              66$:    MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
026064  016520  000024                      MOV     RKDB(R5),(R0)+  ;2'ND WORD
026070  016520  000024                      MOV     RKDB(R5),(R0)+  ;3'RD WORD

026074  032765  100000  000010              BIT     #DLT,RKCS2(R5)  ;SEE IF DATA LATE
026102  001407                              BEQ     67$
026104  004737  050256                      JSR     PC,GSTAT
026110  104173                              ERROR   173             ;DATA LATE ON READ HEADER
026112  104401  060720                      TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
026116  000137  047164                      JMP     $EOP            ;ABORT DRIVE
```

```
5059
5060  026122  020027  002110          67$:    CMP     R0,#RHTAB+132.  ;ALL 66 WORDS DONE?
5061  026126  001332                          BNE     65$             ;BR IF NO
5062
5063  026130  004737  051640                  JSR     PC,SORT         ;SORT RHTAB INTO SRTTAB SO THAT IT
5064                                                                    ;BEGINS WITH SECTOR 0
5065  026134  005037  001460                  CLR     WDCNT           ;WORD COUNT
5066  026140  012700  002110                  MOV     #SRTTAB,R0      ;ACTUAL HEADER TABLE
5067  026144  012701  001500                  MOV     #HDTAB,R1       ;CALC HEADER TABLE
5068
5069  026150  012037  001472          68$:    MOV     (R0)+,HDWD
5070  026154  012137  004400                  MOV     (R1)+,TEMP1
5071  026160  023737  001472  004400          CMP     HDWD,TEMP1      ;COMPARE ACTUAL WITH CALCULATED WORD
5072  026166  001401                          BEQ     69$             ;BR IF COMPARE
5073  026170  104202                          ERROR   202             ;READ HEADER MISMATCH
5074
5075  026172  005237  001460          69$:    INC     WDCNT
5076  026176  023727  001460  000102          CMP     WDCNT,#66.      ;ALL WORDS DONE?
5077  026204  001361                          BNE     68$             ;BR IF NO
5078
5079
5080
5081
5082
5083                                  ;;*****************************************************************
5084                                  ;*TEST 27       SEEK FROM CYL 0 TO 1 & READ HEADERS
5085                                  ;*
5086                                  ;*      THIS TEST CHECKS MSG A & B WORDS 0,1,2 FOR CORRECT STATUS AFTER RDY
5087                                  ;*      IS RECEIVED FROM A SEEK COMMAND TO DETERMINE
5088                                  ;*      THAT THE HEADS ARE ACTUALLY MOVING & THE CYL DIFF IS 1.
5089                                  ;*      AFTER ATTN IS RECEIVED, CERR IS EXAMINED FOR ANY ERRORS.
5090                                  ;*      CYL DIFFERENCE IN MSG A2 IS VERIFIED TO BE 0 & CYL ADDR
5091                                  ;*      IN MSG B2 IS VERIFIED TO BE 1.
5092                                  ;*
5093                                  ;*      HEADERS ARE READ FROM 1 SECTOR, HEAD 0 & VERIFIED THAT THEY ARE
5094                                  ;*      DIFFERENT FROM CYL 0 TO SHOW THAT THE HEADS DID ACTUALLY MOVE.
5095                                  ;*
5096                                  ;*****************************************************************
5097  026206  000004          TST27:  SCOPE
5098  026210  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
5099  026216  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
5100  026222  004737  050320                  JSR     PC,SUBCLR
5101  026226  104024                          ERROR   24              ;CERR AFTER SCLR
5102  026230  005037  001370                  CLR     FRCYL
5103  026234  012737  000001  001372          MOV     #1,TOCYL
5104  026242  012737  000001  001400          MOV     #1,CALDIF
5105  026250  012765  000001  000020          MOV     #1,RKDC(R5)     ;SET FOR CYL 1
5106  026256  012737  027322  001176          MOV     #10$,$ESCAPE
5107
5108  026264  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD
5109  026272  013737  001434  004400          MOV     T10,TEMP1       ;SETUP  TIMEOUT
5110  026300  004737  047632                  JSR     PC,FRDY         ;FIND RDY
5111  026304  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD
5112  026306  012737  030140  004374          MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,SBMR2        ;LOAD SHOULD BE DATA
5113  026314  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
5114  026320  104203                          ERROR   203             ;MSG A0 ERROR DURING SEEK CMD
```

```
5115  026322  005037  004376                  CLR     SBMR3           ;MR3 SHOULD BE 0
5116  026326  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
5117  026332  104204                          ERROR   204             ;MSG B0 ERROR DURING SEEK CMD
5118
5119  026334  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
5120  026342  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
5121  026350  004737  050256                  JSR     PC,GSTAT
5122  026354  012737  003720  004374          MOV     #<D.FWD!D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
5123  026362  004737  052504                  JSR     PC,CKMR2
5124  026366  104205                          ERROR   205             ;MSG A1 ERROR DURING SEEK CMD
5125  026370  005037  004376                  CLR     SBMR3
5126  026374  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
5127  026400  104206                          ERROR   206             ;MSG B1 ERROR DURING SEEK CMD
5128
5129  026402  004737  051124                  JSR     PC,RDCYLD       ;READ CYL DIFF IN MSG A2
5130  026406  023727  001402  000001          CMP     CYLDIF,#1
5131  026414  001401                          BEQ     1$
5132  026416  104212                          ERROR   212             ;CYL DIFF INCORRECT DURING SEEK CMD.
5133
5134
5135  026420  012737  027342  001176  1$:     MOV     #12$,$ESCAPE
5136  026426  013737  001442  004400          MOV     T2500,TEMP1     ;SETUP TIMEOUT
5137
5138
5139  026434  004737  050144                  JSR     PC,FATT2        ;FIND ATTN
5140  026440  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
5141  026442  032737  100000  004336          BIT     #CERR,HCS1
5142  026450  001401                          BEQ     64$
5143  026452  104210                          ERROR   210             ;CERR AFTER SEEK CMD
5144  026454                          64$:
5145
5146  026454  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
5147  026462  004737  050256                  JSR     PC,GSTAT        ;GET LATEST STATUS
5148  026466  012737  050340  004374          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2 ;SHOULD BE VALUE
5149  026474  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
5150  026500  104133                          ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
5151  026502  005037  004376                  CLR     SBMR3
5152  026506  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
5153  026512  104134                          ERROR   134             ;MSG B0 ERROR AFTER SEEK CMD
5154
5155  026514  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
5156  026522  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
5157  026530  004737  050256                  JSR     PC,GSTAT
5158  026534  012737  001720  004374          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
5159  026542  004737  052504                  JSR     PC,CKMR2
5160  026546  104135                          ERROR   135             ;MSG A1 ERROR AFTER SEEK CMD
5161  026550  005037  004376                  CLR     SBMR3
5162  026554  004737  052566                  JSR     PC,CKMR3
5163  026560  104136                          ERROR   136             ;MSG B1 ERROR AFTER SEEK CMD
5164
5165
5166  026562  004737  051124                  JSR     PC,RDCYLD       ;READ CYL DIFF IN MSG A2
5167  026566  005737  001402                  TST     CYLDIF
5168  026572  001401                          BEQ     65$
5169  026574  104137                          ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
5170
```

```
5171  026576                                  65$:
5172
5173  026576  012765  100000  000000            MOV     #CCLR,RKCS1(R5)
5174  026604  013765  001222  000010            MOV     $UNIT,RKCS2(R5) ;DRIVE#
5175  026612  012765  000005  000000            MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
5176  026620  013737  001434  004400            MOV     T10,TEMP1
5177  026626  004737  047632                    JSR     PC,FRDY         ;FIND RDY
5178  026632  104151                            ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
5179  026634  004737  050016                    JSR     PC,TSTATN       ;TEST FOR ATTN
5180  026640  000401                            BR      66$
5181  026642  104154                            ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5182  026644                                  66$:
5183
5184  026644  012765  100000  000000            MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
5185  026652  004737  050256                    JSR     PC,GSTAT        ;GET LATEST STATUS
5186  026656  012737  010340  004374            MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2     ;SHOULD BE VALUE
5187  026664  004737  052504                    JSR     PC,CKMR2        ;CHECK MR2
5188  026670  104273                            ERROR   273             ;MSG AO ERROR AFTER DRIVE CLEAR CMD
5189  026672  005037  004376                    CLR     SBMR3
5190  026676  004737  052566                    JSR     PC,CKMR3        ;CHECK MR3
5191  026702  104265                            ERROR   265             ;MSG BO ERROR AFTER DRIVE CLEAR CMD
5192
5193  026704  012765  100000  000000            MOV     #CCLR,RKCS1(R5)
5194  026712  012765  000001  000026            MOV     #1,RKMR1(R5)    ;SELECT WORD 1
5195  026720  004737  050256                    JSR     PC,GSTAT
5196  026724  012737  001720  004374            MOV     #<0.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
5197  026732  004737  052504                    JSR     PC,CKMR2
5198  026736  104274                            ERROR   274             ;MSG A1 ERROR AFTER DRIVE CLEAR CM
5199  026740  005037  004376                    CLR     SBMR3
5200  026744  004737  052566                    JSR     PC,CKMR3
5201  026750  104266                            ERROR   266             ;MSG B1 ERROR AFTER DRIVE CLEAR CMD
5202
5203
5204  026752  012765  100000  000000            MOV     #CCLR,RKCS1(R5)
5205  026760  013765  001222  000010            MOV     $UNIT,RKCS2(R5)
5206  026766  004737  051210                    JSR     PC,RDCYLA       ;READ CYL ADDR IN MSG B2
5207  026772  005737  001404                    TST     CYLADD
5208  026776  023727  001404  000001            CMP     CYLADD,#1
5209  027004  001401                            BEQ     2$
5210  027006  104207                            ERROR   207             ;CYL ADDR INCORRECT AFTER SEEK CMD
5211
5212
5213  027010                                  2$:
5214  027010  104415                            SCOP1
5215  027012  012706  001100                    MOV     #STACK,SP       ;RESTORE STK PTR
5216
5217  027016  004737  050320                    JSR     PC,SUBCLR
5218  027022  104024                            ERROR   24              ;CERR AFTER SCLR
5219
5220  027024  005037  001176                    CLR     $ESCAPE
5221  027030  012765  000001  000020            MOV     #1,RKDC(R5)     ;CYL #
5222
5223  027036  012700  001704                    MOV     #RHTAB,R0
5224  027042  012765  000025  000000            MOV     #<RDHEAD>,RKCS1(R5)     ;READ HEADER CMD
5225  027050  013737  001446  004400            MOV     T50000,TEMP1    ;SETUP  TIMEOUT
5226  027056  004737  047632                    JSR     PC,FRDY         ;FIND RDY
```

```
5227  027062  104171                          ERROR   171             ;NO RDY AFTER READ HEADER CMD
5228  027064  032737  100000  004336          BIT     #CERR,HCS1
5229  027072  001405                          BEQ     67$
5230  027074  104174                          ERROR   174             ;CERR AFTER READ HEADER CMD
5231  027076  104401  060720                  TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
5232  027102  000137  047164                  JMP     $EOP            ;ABORT DRIVE
5233
5234  027106  016520  000024          67$:    MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
5235  027112  016520  000024                  MOV     RKDB(R5),(R0)+  ;2'ND WORD
5236  027116  016520  000024                  MOV     RKDB(R5),(R0)+  ;3'RD WORD
5237
5238
5239  027122  032765  100000  000010          BIT     #DLT,RKCS2(R5)
5240  027130  001407                          BEQ     68$
5241  027132  004737  050256                  JSR     PC,GSTAT
5242  027136  104173                          ERROR   173             ;DLT AFTER READ HEADER CMD
5243  027140  104401  060720                  TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
5244  027144  000137  047164                  JMP     $EOP            ;ABORT DRIVE
5245  027150                          68$:
5246
5247
5248  027150  005737  001704                  TST     RHTAB           ;CHECK 1'ST WORD ONLY:CYL#
5249  027154  001001                          BNE     3$
5250  027156  104211                          ERROR   211             ;CYL 0 HEADER ON CYL 1
5251
5252  027160  023727  001704  000001  3$:     CMP     RHTAB,#1
5253  027166  001401                          BEQ     4$
5254  027170  104202                          ERROR   202             ;READ CYL WORD HEADER ERROR
5255  027172                          4$:
5256  027172  004737  052770                  JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
5257  027176  000471                          BR      TST30           ;;GO TO NEXT TEST
5258                                                                  ;RETURN HERE IF SW 14 IS SET OR
5259                                                                  ;SW 8 WITH SWR <7:0> APPLY
5260  027200  004737  050320          6$:     JSR     PC,SUBCLR
5261  027204  104024                          ERROR   24              ;CERR AFTER SCLR
5262
5263  027206  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD TO RECONDITION DRIVE.
5264  027214  013737  001434  004400          MOV     T10,TEMP1       ;SETUP TIMEOUT
5265  027222  004737  047632                  JSR     PC,FRDY         ;FIND RDY
5266  027226  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD.
5267
5268  027230  013737  001444  004400          MOV     T5000,TEMP1
5269  027236  004737  050144                  JSR     PC,FATT2        ;FIND ATTN
5270  027242  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
5271  027244  032737  100000  004336          BIT     #CERR,HCS1
5272  027252  001401                          BEQ     69$
5273  027254  104210                          ERROR   210             ;CERR AFTER SEEK CMD.
5274
5275  027256  004737  050320          69$:    JSR     PC,SUBCLR
5276  027262  104024                          ERROR   24              ;CERR AFTER SCLR
5277
5278  027264  004737  0?1210                  JSR     PC,RDCYLA
5279  027270  005737  001404                  TST     CYLADD
5280  027274  001401                          BEQ     7$
5281  027276  104043                          ERROR   43              ;CYL ADDR IN RKMR3 NOT=RKDC
5282
```

```
5283  027300                          7$:
5284  027300  005037  001176                  CLR     $ESCAPE
5285  027304  005737  001430                  TST     LPFLG
5286  027310  001402                          BEQ     70$
5287  027312  000177  151572                  JMP     @$LPERR         ;SW 9 WAS SET.
5288  027316  000177  151564          70$:    JMP     @$LPADR         ;SW 14 OR 8 WAS SET
5289
5290  027322                          10$:
5291  027322  005237  001430                  INC     LPFLG
5292  027326  032777  001000  151604          BIT     #SW9,@SWR       ;LOOP ON ERROR?
5293  027334  001321                          BNE     6$              ;YES, RECONDITION DRIVE
5294  027336  000137  026420                  JMP     1$              ;RETURN TO MAINLINE
5295  027342                          12$:
5296  027342  005237  001430                  INC     LPFLG
5297  027346  032777  001000  151564          BIT     #SW9,@SWR       ;LOOP ON ERROR?
5298  027354  001311                          BNE     6$              ;YES, RECONDITION DRIVE
5299  027356  000137  027010                  JMP     2$              ;RETURN TO MAINLINE
5300
5301
5302
5303
5304                                  ;;*****************************************************************
5305                                  ;*TEST 30       WRITE & READ HEADERS CYLINDER 1, HEAD 0
5306                                  ;;*****************************************************************
5307  027362  000004                  TST30:  SCOPE
5308  027364  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
5309  027372  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
5310
5311  027376  004737  050320                  JSR     PC,SUBCLR
5312  027402  104024                          ERROR   24              ;CERR AFTER SCLR
5313  027404  012765  001500  000004          MOV     #HDTAB,RKBA(R5) ;HEADER WORD TABLE
5314  027412  012765  177676  000002          MOV     #-66.,RKWC(R5)  ;WORD COUNT.
5315  027420  012737  000001  001372          MOV     #1,TOCYL
5316
5317  027426  013737  001372  001406          MOV     TOCYL,CALADD    ;SETUP
5318  027434  012737  000000  001450          MOV     #0,HEAD         ;TO FILL
5319  027442  012737  000000  001454          MOV     #0,FORMAT       ;HEADER
5320  027450  004737  051460                  JSR     PC,FHDTAB       ;TABLE
5321
5322  027454  012765  000001  000020          MOV     #1,RKDC(R5)     ;CYL#
5323
5324  027462  012765  000027  000000          MOV     #<WRHEAD>,RKCS1(R5)     ;WRITE HEADER CMD
5325  027470  013737  001446  004400          MOV     TS0000,TEMP1    ;SETUP TIMEOUT
5326  027476  004737  047632                  JSR     PC,FRDY         ;FIND RDY
5327  027502  104200                          ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
5328  027504  004737  050256                  JSR     PC,GSTAT        ;GET FRESH STATUS
5329  027510  032737  100000  004336          BIT     #CERR,HCS1
5330  027516  001405                          BEQ     64$
5331  027520  104201                          ERROR   201             ;CERR AFTER WRITE HEADER CMD
5332  027522  104401  060720                  TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
5333  027526  000137  047164                  JMP     $EOP            ;ABORT DRIVE
5334  027532                          64$:
5335
5336  027532  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
5337  027540  004737  050256                  JSR     PC,GSTAT        ;GET LATEST STATUS
5338  027544  012737  010340  004374          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2     ;SHOULD BE VALUE
```

```
5339  027552  004737  052504          JSR     PC,CKMR2        ;CHECK MR2
5340  027556  104277                  ERROR   277             ;MSG AO ERROR AFTER WRITE HEADER CMD
5341  027560  005037  004376          CLR     SBMR3
5342  027564  004737  052566          JSR     PC,CKMR3        ;CHECK MR3
5343  027570  104267                  ERROR   267             ;MSG BO ERROR AFTER WRITE HEADER CMD
5344
5345  027572  012765  100000  000000  MOV     #CCLR,RKCS1(R5)
5346  027600  012765  000001  000026  MOV     #1,RKMR1(R5)    ;SELECT WORD 1
5347  027606  004737  050256          JSR     PC,GSTAT
5348  027612  012737  001720  004374  MOV     #<D.SPOK!D.CART!D.DCOR!D.BRHM!D.TFOK>,SBMR2
5349  027620  004737  052504          JSR     PC,CKMR2
5350  027624  104300                  ERROR   300             ;MSG A1 ERROR AFTER WRITE HEADER CMD
5351  027626  005037  004376          CLR     SBMR3
5352  027632  004737  052566          JSR     PC,CKMR3
5353  027636  104270                  ERROR   270             ;MSG B1 ERROR AFTER WRITE HEADER CMD
5354
5355
5356  027640  005037  001420          CLR     SECNT           ;SECTOR COUNT
5357  027644  104415                  SCOP1
5358  027646  012706  001100          MOV     #STACK,SP       ;RESTORE STK PTR
5359
5360  027652  004737  050320          JSR     PC,SUBCLR
5361  027656  104024                  ERROR   24              ;CERR AFTER SCLR
5362
5363  027660  012765  000001  000020  MOV     #1,RKDC(R5)     ;CYL #
5364
5365  027666  012700  001704          MOV     #RHTAB,R0
5366
5367  027672  012765  000025  000000  65$:    MOV     #RDHEAD,RKCS1(R5)       ;READ HEADER CMD
5368  027700  013737  001440  004400          MOV     T500,TEMP1      ;SETUP  TIMEOUT
5369  027706  004737  047632          JSR     PC,FRDY         ;FIND RDY
5370  027712  104171                  ERROR   171             ;NO RDY AFTER READ HEADER CMD
5371  027714  032737  100000  004336  BIT     #CERR,HCS1
5372  027722  001405                  BEQ     66$
5373  027724  104174                  ERROR   174             ;CERR AFTER READ HEADER  CMD
5374  027726  104401  060720          TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
5375  027732  000137  047164          JMP     $EOP            ;ABORT DRIVE
5376
5377  027736  016520  000024  66$:    MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
5378  027742  016520  000024          MOV     RKDB(R5),(R0)+  ;2'ND WORD
5379  027746  016520  000024          MOV     RKDB(R5),(R0)+  ;3'RD WORD
5380
5381  027752  032765  100000  000010  BIT     #DLT,RKCS2(R5)  ;SEE IF DATA LATE
5382  027760  001407                  BEQ     67$
5383  027762  004737  050256          JSR     PC,GSTAT
5384  027766  104173                  ERROR   173             ;DATA LATE ON READ HEADER
5385  027770  104401  060720          TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
5386  027774  000137  047164          JMP     $EOP            ;ABORT DRIVE
5387
5388  030000  020027  002110  67$:    CMP     R0,#RHTAB+132.  ;ALL 66 WORDS DONE?
5389  030004  001332                  BNE     65$             ;BR IF NO
5390
5391  030006  004737  051640          JSR     PC,SORT         ;SORT RHTAB INTO SRTTAB SO THAT IT
5392                                                          ;BEGINS WITH SECTOR 0
5393  030012  005037  001460          CLR     WDCNT           ;WORD COUNT
5394  030016  012700  002110          MOV     #SRTTAB,R0      ;ACTUAL HEADER TABLE
```

```
                030022  012701  001500                  MOV     #HDTAB,R1       ;CALC HEADER TABLE
                                                        
                030026  012037  001472          68$:    MOV     (R0)+,HDWD
                030032  012137  004400                  MOV     (R1)+,TEMP1
                030036  023737  001472  004400          CMP     HDWD,TEMP1      ;COMPARE ACTUAL WITH CALCULATED WORD
                030044  001401                          BEQ     69$             ;BR IF COMPARE
                030046  104202                          ERROR   202             ;READ HEADER MISMATCH
                                                        
                030050  005237  001460          69$:    INC     WDCNT
                030054  023727  001460  000102          CMP     WDCNT,#56.      ;ALL WORDS DONE?
                030062  001361                          BNE     68$             ;BR IF NO
                                                        
                                                        
                                                        
                                                        
                                                ;;**********************************************************************
                                                ;*TEST 31       TEST RECALIBRATE COMMAND & READ HEADERS
                                                ;*
                                                ;*      THIS TEST DOES A RECALIBRATE & READS HEADERS.
                                                ;*      IT VERIFIES THAT WRITING HEADERS ON CYL 1 FROM THE PREVIOUS
                                                ;*      TEST DID NOT OVERWRITE CYL 0 HEADERS.
                                                ;*      AN ERROR IN THIS TEST INDICATES THAT HEADS:
                                                ;*
                                                ;*
                                                ;*              A. MOVED TO A CYL OTHER THAN 1
                                                ;*      OR      B. DID NOT GET BACK TO CYL 0
                                                ;*
                                                ;**********************************************************************
                030064  000004                  TST31:  SCOPE
                030066  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
                030074  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
                                                        
                030100  004737  050320                  JSR     PC,SUBCLR
                030104  104024                          ERROR   24              ;CERR AFTER SCLR
                030106  012737  000001  001370          MOV     #1,FRCYL        ;PARAMETERS
                030114  005037  001372                  CLR     TOCYL           ;FOR
                030120  012737  000001  001400          MOV     #1,CALDIF       ;ERROR TYPEOUTS
                030126  012737  031212  001176          MOV     #10$,$ESCAPE
                030134  012765  000013  000000          MOV     #RECAL,RKCS1(R5)        ;RECAL CMD
                030142  013737  001434  004400          MOV     T10,TEMP1       ;SETUP TIMEOUT
                030150  004737  047632                  JSR     PC,FRDY         ;FIND RDY
                030154  104124                          ERROR   124             ;NO RDY AFTER RECAL CMD
                030156  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
                030164  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
                030172  004737  050256                  JSR     PC,GSTAT
                030176  032737  020000  004364          BIT     #D.RTZ,HMR2
                030204  001001                          BNE     1$
                030206  104307                          ERROR   307             ;RTZ NOT SET DURING RECAL CMD
                                                        
                030210  005065  000026          1$:     CLR     RKMR1(R5)       ;SELECT WORD 0
                030214  004737  050256                  JSR     PC,GSTAT
                                                        
                030220  012737  030140  004374          MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,SBMR2        ;LOAD SHOULD BE DATA
                030226  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
                030232  104213                          ERROR   213             ;MSG AO ERROR DURING RECAL CMD
```

```
5451  030234  005037  004376                  CLR     SBMR3           ;MR3 SHOULD BE 0
5452  030240  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
5453  030244  104214                          ERROR   214             ;MSG BO ERROR DURING RECAL CMD
5454
5455  030246  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
5456  030254  004737  050256                  JSR     PC,GSTAT
5457  030260  012737  025720  004374          MOV     #<D.RTZ!D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
5458  030266  004737  052504                  JSR     PC,CKMR2
5459  030272  104215                          ERROR   215             ;MSG A1 ERROR DURING RECAL CMD
5460  030274  005037  004376                  CLR     SBMR3
5461  030300  004737  052566                  JSR     PC,CKMR3
5462  030304  104216                          ERROR   216             ;MSG B1 ERROR DURING RECAL CMD
5463
5464  030306  004737  051124                  JSR     PC,RDCYLD       ;READ CYL DIFF IN MSG A2
5465  030312  005737  001402                  TST     CYLDIF
5466  030316  001401                          BEQ     2$
5467  030320  104217                          ERROR   217             ;CYL DIFF INCORRECT DURING RECAL CMD.
5468
5469  030322  012737  031232  001176  2$:     MOV     #12$,$ESCAPE
5470  030330  012737  177777  004400          MOV     #-1,TEMP1       ;SETUP TIMEOUT
5471  030336  004737  050144                  JSR     PC,FATT2        ;FIND ATTN
5472  030342  104055                          ERROR   55              ;NO ATTN AFTER RECAL CMD
5473  030344  032737  100000  004336          BIT     #CERR,HCS1
5474  030352  001401                          BEQ     3$
5475  030354  104220                          ERROR   220             ;CERR AFTER RECAL CML
5476  030356                          3$:
5477
5478  030356  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
5479  030364  004737  050256                  JSR     PC,GSTAT        ;GET LATEST STATUS
5480  030370  012737  050340  004374          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2 ;SHOULD BE VALUE
5481  030376  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
5482  030402  104221                          ERROR   221             ;MSG AO ERROR AFTER RECAL CMD
5483  030404  005037  004376                  CLR     SBMR3
5484  030410  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
5485  030414  104275                          ERROR   275             ;MSG 30 ERROR AFTER RECAL CMD
5486
5487  030416  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
5488  030424  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
5489  030432  004737  050256                  JSR     PC,GSTAT
5490  030436  012737  001720  004374          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
5491  030444  004737  052504                  JSR     PC,CKMR2
5492  030450  104222                          ERROR   222             ;MSG A1 ERROR AFTER RECAL CMD
5493  030452  005037  004376                  CLR     SBMR3
5494  030456  004737  052566                  JSR     PC,CKMR3
5495  030462  104276                          ERROR   276             ;MSG B1 ERROR AFTER RECAL CMD
5496
5497
5498  030464  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
5499  030472  004737  051124                  JSR     PC,RDCYLD       ;READ CYL DIFF IN RKMR2
5500  030476  005737  001402                  TST     CYLDIF
5501  030502  001401                          BEQ     64$
5502  030504  104047                          ERROR   47              ;CYL DIFF/OFFSET NOT CLEARED AFTER RECAL CMD
5503  030506  004737  051210          64$:    JSR     PC,RDCYLA       ;READ CYL ADDR IN RKMR3
5504  030512  005737  001404                  TST     CYLADD
5505  030516  001401                          BEQ     65$
5506  030520  104050                          ERROR   50              ;CYL ADDR NOT CLEARED AFTER RECAL CMD
```

```
5507  030522                          65$:
5508
5509
5510  030522  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
5511  030530  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
5512  030536  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
5513  030544  013737  001434  004400          MOV     T10,TEMP1
5514  030552  004737  047632                  JSR     PC,FRDY         ;FIND RDY
5515  030556  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
5516  030560  004737  050016                  JSR     PC,TSTATN       ;TEST FOR ATTN
5517  030564  000401                          BR      66$
5518  030566  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5519  030570                          66$:
5520
5521  030570  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
5522  030576  004737  050256                  JSR     PC,GSTAT        ;GET LATEST STATUS
5523  030602  012737  01.340  004374          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2     ;SHOULD BE VALUE
552.  030610  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
5525  030614  104273                          ERROR   273             ;MSG AO ERROR AFTER DRIVE CLEAR CMD
5526  030616  005037  004376                  CLR     SBMR3
5527  030622  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
5528  030626  104265                          ERROR   265             ;MSG BO ERROR AFTER DRIVE CLEAR CMD
5529
5530  030630  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
5531  030636  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
5532  030644  004737  050256                  JSR     PC,GSTAT
5533  030650  012737  001720  004374          MOV     #<0.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
5534  030656  004737  052504                  JSR     PC,CKMR2
5535  030662  104274                          ERROR   274             ;MSG A1 ERROR AFTER DRIVE CLEAR CMD
5536  030664  005037  004376                  CLR     SBMR3
5537  030670  004737  052566                  JSR     PC,CKMR3
5538  030674  104266                          ERROR   266             ;MSG B1 ERROR AFTER DRIVE CLEAR CMD
5539
5540
5541  030676                          4$:
5542  030676  104415                          SCOP1
5543  030700  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
5544
5545  030704  004737  050320                  JSR     PC,SUBCLR
5546  030710  104024                          ERROR   24              ;CERR AFTER SCLR
5547
5548  030712  005037  001176                  CLR     $ESCAPE
5549
5550  030716  012700  001704                  MOV     #RHTAB,R0
5551  030722  012765  000025  000000          MOV     #<RDHEAD>,RKCS1(R5)     ;READ HEADER CMD
5552  030730  013737  001446  004400          MOV     T50000,TEMP1    ;SETUP  TIMEOUT
5553  030736  004737  047632                  JSR     PC,FRDY         ;FIND RDY
5554  030742  104171                          ERROR   171             ;NO RDY AFTER READ HEADER CMD
5555  030744  032737  100000  004336          BIT     #CERR,HCS1
5556  030752  001405                          BEQ     67$
5557  030754  104174                          ERROR   174             ;CERR AFTER READ HEADER CMD
5558  030756  104401  060720                  TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
5559  030762  000137  047164                  JMP     $EOP            ;ABORT DRIVE
5560
5561  030766  016520  000024          67$:    MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
5562  030772  016520  000024                  MOV     RKDB(R5),(R0)+  ;2'ND WORD
```

```
5563  030776  016520  000024                  MOV     RKDB(R5),(R0)+  ;3'RD WORD
5564
5565
5566  031002  032765  100000  000010          BIT     #DLT,RKCS2(R5)
5567  031010  001407                          BEQ     68$
5568  031012  004737  050256                  JSR     PC,GSTAT
5569  031016  104173                          ERROR   173             ;DLT AFTER READ HEADER CMD
5570  031020  104401  060720                  TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
5571  031024  000137  047164                  JMP     $EOP            ;ABORT DRIVE
5572  031030                          68$:
5573
5574  031030  023727  001704  000001          CMP     RHTAB,#1        ;CHECK WORD 0, CYL # ONLY
5575  031036  001001                          BNE     5$
5576  031040  104240                          ERROR   240             ;CYL 1 HEADERS ON CYL 0
5577
5578  031042  005737  001704          5$:     TST     RHTAB
5579  031046  001401                          BEQ     6$
5580  031050  104202                          ERROR   202             ;READ CYL WORD HEADER ERROR
5581  031052                          6$:
5582  031052  004737  052770                  JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
5583  031056  000475                          BR      TST32           ;;GO TO NEXT TEST
5584                                                                  ;RETURN HERE IF SW 14 IS SET OR
5585                                                                  ;SW 8 WITH SWR <7:0> APPLY
5586  031060  004737  050320          8$:     JSR     PC,SUBCLR
5587  031064  104024                          ERROR   24              ;CERR AFTER SCLR
5588  031066  012765  000001  000020          MOV     #1,RKDC(R5)     ;RECONDITION BACK TO CYL 1
5589
5590  031074  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD TO RECONDITION DRIVE.
5591  031102  013737  001434  004400          MOV     T10,TEMP1       ;SETUP TIMEOUT
5592  031110  004737  047632                  JSR     PC,FRDY         ;FIND RDY
5593  031114  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD.
5594
5595  031116  013737  001444  004400          MOV     T5000,TEMP1
5596  031124  004737  050144                  JSR     PC,FATT2        ;FIND ATTN
5597  031130  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
5598  031132  032737  100000  004336          BIT     #CERR,HCS1
5599  031140  001401                          BEQ     69$
5600  031142  104210                          ERROR   210             ;CERR AFTER SEEK CMD.
5601
5602  031144  004737  050320          69$:    JSR     PC,SUBCLR
5603  031150  104024                          ERROR   24              ;CERR AFTER SCLR
5604
5605
5606  031152  004737  051210                  JSR     PC,RDCYLA
5607  031156  023727  001404  000001          CMP     CYLADD,#1
5608  031164  001401                          BEQ     9$
5609  031166  104043                          ERROR   43              ;CYL ADDR IN RKMR3 NOT=RKDC
5610  031170                          9$:
5611  031170  005037  001176                  CLR     $ESCAPE
5612  031174  005737  001430                  TST     LPFLG
5613  031200  001402                          BEQ     70$
5614  031202  000177  147702                  JMP     @$LPERR         ;SW 9 WAS SET.
5615  031206  000177  147674          70$:    JMP     @$LPADR         ;SW 14 OR 8 WAS SET
5616  031212                          10$:
5617  031212  005237  001430                  INC     LPFLG
5618  031216  032777  001000  147714          BIT     #SW9,@SWR       ;LOOP ON ERROR?
```

```
5619  031224  001315                          BNE     8$              ;YES, RECONDITION DRIVE
5620  031226  000137  030322                  JMP     2$              ;RETURN TO MAINLINE
5621  031232                          12$:
5622  031232  005237  001430                  INC     LPFLG
5623  031236  032777  001000  147674          BIT     #SW9,@SWR       ;LOOP ON ERROR?
5624  031244  001305                          BNE     8$              ;YES, RECONDITION DRIVE
5625  031246  000137  030676                  JMP     4$              ;RETURN TO MAINLINE
5626                          ;;*************************************************************
5627                          ;*TEST 32       SINGLE INCREMENT SEEKS TO CYL 410
5628                          ;*
5629                          ;*      THIS TEST DOES SINGLE INCREMENT SEEKS OUT TO CYL 410
5630                          ;*      WITHOUT ANY WRITING OR READING SO AS NOT TO INADVERTENTLY
5631                          ;*      DESTROY DATA.
5632                          ;*
5633                          ;;*************************************************************
5634  031252  000004          TST32:  SCOPE
5635  031254  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
5636  031262  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
5637
5638  031266  004737  050320                  JSR     PC,SUBCLR
5639  031272  104024                          ERROR   24              ;CERR AFTER SCLR
5640  031274  005037  001370                  CLR     FRCYL           ;FROM CYL
5641  031300  012737  000001  001372          MOV     #1,TOCYL        ;TO CYL
5642  031306  012737  000001  001400          MOV     #1,CALDIF       ;CALCULATED DIFF.
5643
5644  031314                          1$:
5645  031314  104415                          SCOP1
5646  031316  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
5647
5648  031322  004737  050320                  JSR     PC,SUBCLR
5649  031326  104024                          ERROR   24              ;CERR AFTER SCLR
5650                                  ,,
5651  031330  012737  032240  001176          MOV     #10$,$ESCAPE
5652  031336  013765  001372  000020          MOV     TOCYL,RKDC(R5)  ;CYL TO SEEK TO
5653
5654  031344  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD
5655  031352  013737  001434  004400          MOV     T10,TEMP1       ;SETUP  TIMEOUT
5656  031360  004737  047632                  JSR     PC,FRDY         ;FIND RDY
5657  031364  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD
5658  031366  012737  030140  004374          MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,SBMR2        ;LOAD SHOULD BE DATA
5659  031374  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
5660  031400  104203                          ERROR   203             ;MSG A0 ERROR DURING SEEK CMD
5661  031402  005037  004376                  CLR     SBMR3           ;MR3 SHOULD BE 0
5662  031406  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
5663  031412  104204                          ERROR   204             ;MSG B0 ERROR DURING SEEK CMD
5664
5665  031414  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
5666  031422  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
5667  031430  004737  050256                  JSR     PC,GSTAT
5668  031434  012737  003720  004374          MOV     #<D.FWD!D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
5669  031442  004737  052504                  JSR     PC,CKMR2
5670  031446  104205                          ERROR   205             ;MSG A1 ERROR DURING SEEK CMD
5671  031450  005037  004376                  CLR     SBMR3
5672  031454  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
5673  031460  104206                          ERROR   206             ;MSG B1 ERROR DURING SEEK CMD
5674
```

```
5675  031462  004737  051124                    JSR     PC,RDCYLD
5676  031466  023727  001402  000001            CMP     CYLDIF,#1
5677  031474  001401                            BEQ     2$
5678  031476  104212                            ERROR   212             ;CYL DIFF INCORRECT DURING SEEK
5679
5680  031500  012737  032260  001176  2$:       MOV     #12$,$ESCAPE
5681  031506  013737  001442  004400            MOV     T2500,TEMP1     ;SETUP TIMEOUT
5682
5683  031514  004737  050144                    JSR     PC,FATT2        ;FIND ATTN
5684  031520  104132                            ERROR   132             ;NO ATTN AFTER SEEK CMD
5685  031522  032737  100000  004336            BIT     #CERR,HCS1
5686  031530  001401                            BEQ     64$
5687  031532  104210                            ERROR   210             ;CERR AFTER SEEK CMD
5688  031534                          64$:
5689
5690  031534  012765  100000  000000            MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
5691  031542  004737  050256                    JSR     PC,GSTAT        ;GET LATEST STATUS
5692  031546  012737  050340  004374            MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2 ;SHOULD BE VALUE
5693  031554  004737  052504                    JSR     PC,CKMR2        ;CHECK MR2
5694  031560  104133                            ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
5695  031562  005037  004376                    CLR     SBMR3
5696  031566  004737  052566                    JSR     PC,CKMR3        ;CHECK MR3
5697  031572  104134                            ERROR   134             ;MSG B0 ERROR AFTER SEEK CMD
5698
5699  031574  012765  100000  000000            MOV     #CCLR,RKCS1(R5)
5700  031602  012765  000001  000026            MOV     #1,RKMR1(R5)    ;SELECT WORD 1
5701  031610  004737  050256                    JSR     PC,GSTAT
5702  031614  012737  001720  004374            MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
5703  031622  004737  052504                    JSR     PC,CKMR2
5704  031626  104135                            ERROR   135             ;MSG A1 ERROR AFTER SEEK CMD
5705  031630  005037  004376                    CLR     SBMR3
5706  031634  004737  052566                    JSR     PC,CKMR3
5707  031640  104136                            ERROR   136             ;MSG B1 ERROR AFTER SEEK CMD
5708
5709
5710  031642  004737  051124                    JSR     PC,RDCYLD       ;READ CYL DIFF IN MSG A2
5711  031646  005737  001402                    TST     CYLDIF
5712  031652  001401                            BEQ     65$
5713  031654  104137                            ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
5714
5715  031656                          65$:
5716
5717  031656  012765  100000  000000            MOV     #CCLR,RKCS1(R5)
5718  031664  013765  001222  000010            MOV     $UNIT,RKCS2(R5) ;DRIVE#
5719  031672  012765  000005  000000            MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
5720  031700  013737  001434  004400            MOV     T10,TEMP1
5721  031706  004737  047632                    JSR     PC,FRDY         ;FIND RDY
5722  031712  104151                            ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
5723  031714  004737  050016                    JSR     PC,TSTATN       ;TEST FOR ATTN
5724  031720  000401                            BR      66$
5725  031722  104154                            ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5726  031724                          66$:
5727
5728  031724  012765  100000  000000            MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
5729  031732  004737  050256                    JSR     PC,GSTAT        ;GET LATEST STATUS
5730  031736  012737  010340  004374            MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2     ;SHOULD BE VALUE
```

```
5731  031744  004737  052504              JSR     PC,CKMR2        ;CHECK MR2
5732  031750  104273                      ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
5733  031752  005037  004376              CLR     SBMR3
5734  031756  004737  052566              JSR     PC,CKMR3        ;CHECK MR3
5735  031762  104265                      ERROR   265             ;MSG B0 ERROR AFTER DRIVE CLEAR CMD
5736
5737  031764  012765  100000  000000      MOV     #CCLR,RKCS1(R5)
5738  031772  012765  000001  000026      MOV     #1,RKMR1(R5)    ;SELECT WORD 1
5739  032000  004737  050256              JSR     PC,GSTAT
5740  032004  012737  001720  004374      MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
5741  032012  004737  052504              JSR     PC,CKMR2
5742  032016  104274                      ERROR   274             ;MSG A1 ERROR AFTER DRIVE CLEAR CMD
5743  032020  005037  004376              CLR     SBMR3
5744  032024  004737  052566              JSR     PC,CKMR3
5745  032030  104266                      ERROR   266             ;MSG B1 ERROR AFTER DRIVE CLEAR CMD
5746
5747
5748  032032  004737  051210              JSR     PC,RDCYLA       ;READ CYL ADDR IN MSG B2
5749  032036  023737  001404  001372      CMP     CYLADD,TOCYL
5750  032044  001401                      BEQ     3$
5751  032046  104207                      ERROR   207             ;CYL ADDR IN RKMR2 NOT=RKDC
5752
5753  032050  005237  001370      3$:     INC     FRCYL
5754  032054  005237  001372              INC     TOCYL
5755  032060  023727  001372  000633      CMP     TOCYL,#411.     ;DONE CYL 410?
5756  032066  001402                      BEQ     4$              ;BR IF YES
5757  032070  000137  031314              JMP     1$
5758
5759  032074                      4$:
5760  032074  004737  052770              JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
5761  032100  000477                      BR      TST33           ;;GO TO NEXT TEST
5762                                                              ;RETURN HERE IF SW 14 IS SET OR
5763                                                              ;SW 8 WITH SWR <7:0> APPLY
5764
5765
5766
5767  032102                      6$:
5768
5769  032102  004737  050320              JSR     PC,SUBCLR
5770  032106  104024                      ERROR   24              ;CERR AFTER SCRL
5771
5772  032110  013765  001372  000020  67$:  MOV     TOCYL,RKDC(R5)  ;CYL#
5773
5774  032116  012765  000017  000000      MOV     #SEEK,RKCS1(R5) ;SEEK CMD TO RECONDITION DRIVE.
5775  032124  013737  001434  004400      MOV     T10,TEMP1       ;SETUP TIMEOUT
5776  032132  004737  047632              JSR     PC,FRDY         ;FIND RDY
5777  032136  104131                      ERROR   131             ;NO RDY AFTER SEEK CMD.
5778
5779  032140  013737  001444  004400      MOV     T500C,TEMP1
5780  032146  004737  050144              JSR     PC,FA           ;FIND ATTN
5781  032152  104132                      ERROR   132             ;NO ATTN AFTER SEEK CMD
5782  032154  032737  100000  004336      BIT     #CERR,HCS1
5783  032162  001401                      BEQ     68$
5784  032164  104210                      ERROR   210             ;CERR AFTER SEEK CMD.
5785
5786  032166  004737  050320      68$:    JSR     PC,SUBCLR
```

```
5787  032172  104024                          ERROR   24              ;CERR AFTER SCLR
5788
5789
5790  032174  005337  001372                  DEC     TOCYL
5791  032200  023727  001372  177777          CMP     TOCYL,#-1       ;ALL CYL DONE?
5792  032206  001340                          BNE     67$
5793
5794  032210  004737  050320                  JSR     PC,SUBCLR
5795  032214  104024                          ERROR   24              ;CERR AFTER SCLR
5796
5797  032216  005037  001176                  CLR     $ESCAPE
5798  032222  005737  001430                  TST     LPFLG
5799  032226  001402                          BEQ     69$
5800  032230  000177  146654                  JMP     @SLPERR         ;SW 9 WAS SET.
5801  032234  000177  146646          69$:    JMP     @SLPADR         ;SW 14 OR 8 WAS SET
5802
5803
5804
5805  032240                          10$:
5806  032240  005237  001430                  INC     LPFLG
5807  032244  032777  001000  146666          BIT     #SW9,@SWR       ;LOOP ON ERROR?
5808  032252  001313                          BNE     6$              ;YES, RECONDITION DRIVE
5809  032254  000137  031500                  JMP     2$              ;RETURN TO MAINLINE
5810
5811  032260                          12$:
5812  032260  005237  001430                  INC     LPFLG
5813  032264  032777  001000  146646          BIT     #SW9,@SWR       ;LOOP ON ERROR?
5814  032272  001303                          BNE     6$              ;YES, RECONDITION DRIVE
5815  032274  000137  032074                  JMP     4$              ;RETURN TO MAINLINE
5816
5817                                  ;;*****************************************************************
5818                                  ;*TEST 33       READ & SAVE BAD SECTOR INFO & TYPE PACK SERIAL #
5819                                  ;*
5820                                  ;*      THIS TEST VERIFIES THAT CYL 410, TRACK 2 CAN BE READ.
5821                                  ;*      THIS AREA CONTAINS BAD SECTOR INFO WHICH IS WRITTEN BY THE
5822                                  ;*      FACTORY DURING MANF.  ALL BAD SECTOR INFO (BSE) WILL BE STORED
5823                                  ;*      AT THIS TIME TO MASK FUTURE READ HEADER OR DATA ERROR PRINTOUTS.
5824                                  ;*      IF BSE INFO CANNOT BE READ, OR IF AFTER READING THE BSE INFO
5825                                  ;*      IT IS DETERMINED THAT AN ALIGNMENT CARTRIDGE IS USED,
5826                                  ;*      A MESSAGE WILL BE TYPED INDICATING THAT ALL
5827                                  ;*      FUTURE FORMAT AND READ-WRITE TESTS WILL BE BYPASSED.
5828                                  ;*      THIS IS DONE SO AS NOT TO DESTROY BSE INFO OR AN ALIGNMENT PACK BY WRITING
5829                                  ;*
5830                                  ;*      THE PACK SERIAL # IS TYPED IN OCTAL 3 FOR THE FIRST PASS ONLY.
5831                                  ;*
5832                                  ;*       THIS IS THE FIRST TEST WHERE THE READ DATA COMMAND IS PERFORMED
5833                                  ;*
5834                                  ;*****************************************************************
5835  032300  000004                  TST33:  SCOPE
5836  032302  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
5837  032310  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
5838
5839  032314  004737  050320                  JSR     PC,SUBCLR
5840  032320  104024                          ERROR   24              ;CERR AFTER SCLR
5841  032322  005037  004402                  CLR     TEMP2           ;SECTOR CTR
5842  032326  005037  004404                  CLR     TEMP3           ;0=22 SEC, 1=20 SEC, 2=DONE
```

```
5843  032332  005037  001474          .           CLR     BSERR           ;BSE INFO NO GOOD IF SET
5844
5845  032336  012765  003314  000004              MOV     #BSE22,RKBA(R5) ;BSE TABLE FOR 22 SECTOR FORMAT
5846  032344  012765  001000  000006              MOV     #1000,RKDA(R5)  ;HEAD 2, SECTOR 0
5847  032352  012765  000632  000020              MOV     #410.,RKDC(R5)  ;CYL 410
5848
5849  032360  012765  177400  000002  1$:         MOV     #-256.,RKWC(R5) ;LOAD WORD CT
5850  032366  012765  000021  000000              MOV     #RDDATA,RKCS1(R5);READ DATA COMMAND
5851  032374  013737  001442  004400              MOV     T2500,TEMP1     ;SETUP TIMEOUT
5852  032402  004737  047632                      JSR     PC,FRDY         ;FIND RDY
5853  032406  104226                              ERROR   226             ;NO RDY AFTER READ DATA CMD
5854  032410  004737  050256                      JSR     PC,GSTAT        ;GET FRESH STATUS
5855  032414  032737  100000  004336              BIT     #CERR,HCS1
5856  032422  001416                              BEQ     3$
5857  032424  104227                              ERROR   227             ;CERR AFTER READ DATA CMD
5858  032426  005237  001474                      INC     BSERR           ;SET BSE ERROR FLAG
5859  032432  032737  010000  004352              BIT     #DTE,HER
5860  032440  001401                              BEQ     2$
5861  032442  104230                              ERROR   230             ;DTE AFTER READ DATA CMD
5862  032444  032737  100000  004340  2$:         BIT     #DLT,HCS2
5863  032452  001404                              BEQ     4$
5864  032454  104231                              ERROR   231             ;DLT AFTER READ DATA CMD
5865  032456  000402                              BR      4$
5866
5867  032460  005037  001474          3$:         CLR     BSERR           ;BSE INFO OK
5868  032464  005737  001474          4$:         TST     BSERR           ;BSE READ OK?
5869  032470  001420                              BEQ     8$              ;BR IF YES
5870  032472  062765  000002  000006              ADD     #2,RKDA(R5)     ;TRY NEXT SECTOR
5871  032500  005237  004402                      INC     TEMP2
5872  032504  023727  004402  000005              CMP     TEMP2,#5        ;READ ALL 5 SECTORS?
5873  032512  001322                              BNE     1$              ;BR IF NO
5874  032514  005737  004404                      TST     TEMP3
5875  032520  001002                              BNE     6$
5876  032522  104233                              ERROR   233             ;CANT READ BSE ON SECTORS 0,2,4,6,8
5877  032524  000452                              BR      TST34           ;;GO TO NEXT TEST
5878  032526  104234                  6$:         ERROR   234             ;CANT READ BSE ON SECTORS 1,3,9,7,9
5879  032530  000450                              BR      TST34           ;;GO TO NEXT TEST
5880
5881  032532  012700  000006          8$:         MOV     #6,R0           ;SETUP FOR WORD 3 OF BSE INFO
5882  032536  016037  003314  004406              MOV     BSE22(R0),TEMP4 ;PULL OUT CARTRIDGE TYPE INFO.
5883  032544  001404                              BEQ     9$              ;BRANCH IF DATA CARTRIDGE
5884  032546  104235                              ERROR   235             ;ALIGNMENT CARTRIDGE USED
5885  032550  005237  001474                      INC     BSERR           ;SET BSE ERROR FLAG
5886  032554  000417                              BR      10$
5887
5888  032556  005237  004404          9$:         INC     TEMP3
5889  032562  023727  004404  000002              CMP     TEMP3,#2
5890  032570  001411                              BEQ     10$
5891
5892  032572  005037  004402                      CLR     TEMP2
5893  032576  012765  002314  000004              MOV     #BSE20,RKBA(R5) ;BSE TABLE FOR 20 SECTOR FORMAT
5894  032604  012765  001001  000006              MOV     #1001,RKDA(R5)  ;HEAD 2, SECTOR 1
5895  032612  000662                              BR      1$
5896
5897  032614  005737  001216          10$:        TST     $PASS
5898  032620  001014                              BNE     TST34           ;;GO TO NEXT TST IF NOT 1'ST PASS
```

```
                                TYPE    ,MSG17          ;CART SERIAL #
                                MOV     #BSE22,-(SP)
                                JSR     PC,$DB2O        ;CONVERT DBL BINARY WORD TO OCTAL
                                JSR     PC,$SUPRS       ;TYPE SERIAL #
                                TYPE    ,$CRLF
                                TYPE    ,$CRLF

                        ;;*****************************************************************
                        ;*TEST 34       DETECT INNER LIMIT
                        ;*
                        ;*      THIS TEST VERIF.ES THAT THE LAST CYLINDER IN THE ABOVE
                        ;*      TEST WAS 410 B: DETECTING INNER LIMIT AS THE ADJACENT CYLINDER.
                        ;*      IF THIS TEST FAILS, IT INDICATES THAT HEADS WERE NOT ON CYL 410
                        ;*      & THAT BSE INFO IS NOT VALID.  THE FORMAT PACK TEST
                        ;*      & ALL READ-WRITE TESTS ARE BYPASSED
                        ;*      TO AVOID DESTROYING BSE INFO OR AN ALIGNMENT CARTRIDGE
                        ;*      SINCE THERE IS A SEEKING OR LIMIT DETECTION PROBLEM.
                        ;*
                        ;;*****************************************************************
000004                  TST34:  SCOPE
012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
004737  050320                  JSR     PC,SUBCLR       ;SUBSYS CLEAR & GET STATUS
104024                          ERROR   24              ;CERR AFTER SCLR

005037  001430                  CLR     LPFLG

012765  000020  000026          MOV     #PAT,RKMR1(R5)  ;PARITY & WORD 0
012765  000631  000020          MOV     #409.,RKDC(R5)  ;CYL 409.
012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD
013737  001434  004400          MOV     T10,TEMP1
004737  047632                  JSR     PC,FRDY         ;FIND RDY
104122                          ERROR   122             ;NO RDY FROM SEEK WITH BAD PARITY
004737  050016                  JSR     PC,TSTATN       ;TEST FOR ATTN
104125                          ERROR   125             ;NO ATTN FROM SEEK WITH BAD PARITY
012765  100000  000000          MOV     #CCLR,RKCS1(R5)
004737  050256                  JSR     PC,GSTAT
012737  050340  004374          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2
004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
104110                          ERROR   110             ;MSG AO ERROR AFTER SEEK WITH BAD PARITY
012737  001200  004376          MOV     #<D.FLT!D.PAR>,SBMR3
004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
104111                          ERROR   111             ;MSG BO ERROR AFTER SEEK WITH BAD PARITY

005065  000026                  CLR     RKMR1(R5)       ;REMOVE PARITY & SELECT WORD 0

012765  100000  000000          MOV     #CCLR,RKCS1(R5)
013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
013737  001434  004400          MOV     T10,TEMP1
004737  047632                  JSR     PC,FRDY         ;FIND RDY
104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
004737  050016                  JSR     PC,TSTATN       ;TEST FOR ATTN
000401                          BR      64$
104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
                        64$:
```

```
5955
5956  033060  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
5957  033066  004737  050256                  JSR     PC,GSTAT        ;GET LATEST STATUS
5958  033072  012737  010340  004374          MOV     #<D!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2     ;SHOULD BE VALUE
5959  033100  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
5960  033104  104273                          ERROR   273             ;MSG AO ERROR AFTER DRIVE CLEAR CMD
5961  033106  005037  004376                  CLR     SBMR3
5962  033112  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
5963  033116  104265                          ERROR   265             ;MSG BO ERROR AFTER DRIVE CLEAR CMD
5964
5965  033120  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
5966  033126  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
5967  033134  004737  050256                  JSR     PC,GSTAT
5968  033140  012737  001720  004374          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
5969  033146  004737  052504                  JSR     PC,CKMR2
5970  033152  104274                          ERROR   274             ;MSG A1 ERROR AFTER DRIVE CLEAR CMD
5971  033154  005037  004376                  CLR     SBMR3
5972  033160  004737  052566                  JSR     PC,CKMR3
5973  033164  104266                          ERROR   266             ;MSG B1 ERROR AFTER DRIVE CLEAR CMD
5974
5975
5976  033166  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
5977  033174  004737  050256                  JSR     PC,GSTAT
5978  033200  012737  010340  004374          MOV     #<D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2
5979  033206  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
5980  033212  104152                          ERROR   152             ;MSG AO ERROR AFTER CLEAR CMD
5981  033214  005037  004376                  CLR     SBMR3           ;MR3 SHOULD BE CLEAR
5982  033220  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
5983  033224  104153                          ERROR   153             ;MSG BO ERROR AFTER CLEAR CMD
5984
5985  033226  012765  000632  000020          MOV     #410.,RKDC(R5)  ;CYL 410.
5986  033234  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK TO CYL 410.
5987  033242  013737  001434  004400          MOV     T10,TEMP1
5988  033250  004737  047632                  JSR     PC,FRDY         ;FIND RDY
5989  033254  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD
5990  033256  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
5991  033264  004737  050256                  JSR     PC,GSTAT
5992  033270  004737  051260                  JSR     PC,FLIM         ;FIND LIMIT DETECT
5993  033274  104160                          ERROR   160             ;LIMIT DETECT NOT FOUND BEFORE TIMEOUT
5994
5995  033276  032737  040000  004364          BIT     #D.UNLD,HMR2
5996  033304  001003                          BNE     1$
5997  033306  104305                          ERROR   305             ;DRIVE NOT UNLOADING AFTER LIMIT DETECT
5998  033310  000137  034252                  JMP     30$             ;BYPASS REST OF TEST
5999
6000  033314  012737  034166  001176  1$:     MOV     #20$,$ESCAPE    ;MUST ESCAPE TO CYCLE UP DRIVE & TEST SWR
6001  033322  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
6002  033330  005065  000026                  CLR     RKMR1(R5)       ;SELECT WORD 0
6003  033334  004737  050256                  JSR     PC,GSTAT
6004  033340  012737  070140  004374          MOV     #<D.DSC!D.PIP!D.SPIN!D.VV!D.DRA>,SBMR2
6005  033346  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
6006  033352  104161                          ERROR   161             ;MSG AO ERROR AFTER INNER LIMIT DETECT
6007  033354  012737  002200  004376          MOV     #<D.SKI!D.FLT>,SBMR3
6008  033362  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
6009  033366  104162                          ERROR   162             ;MSG BO ERROR AFTER INNER LIMIT DETECT
6010  033370  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
```

```
6011  033376  012765  000001  000026          MOV     #1,RKMR1(R5)     ;WORD 1
6012  033404  004737  050256                  JSR     PC,GSTAT
6013  033410  012737  045720  004374          MOV     #<D.UNLD!D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
6014  033416  004737  052504                  JSR     PC,CKMR2         ;CHECK MR2
6015  033422  104163                          ERROR   163              ;MSG A1 ERROR AFTER INNER LIMIT DETECT
6016  033424  004737  050016                  JSR     PC,TSTATN
6017  033430  104165                          ERROR   165              ;NO ATTN AFTER INNER LIMIT DETECT
6018
6019  033432  004737  050320                  JSR     PC,SUBCLR        ;SUBSYS CLR
6020  033436  104024                          ERROR   24               ;CERR AFTER SCLR
6021  033440  013737  001434  004402          MOV     T10,TEMP2        ;SET UP TIMEOUT
6022  033446  004737  051336                  JSR     PC,FHDHM         ;FIND HEAD HOME
6023  033452  104166                          ERROR   166              ;HEAD HOME NOT FOUND BEFORE TIMEOUT
6024  033454  004737  051412                  JSR     PC,FLOAD         ;FIND LOAD HEADS
6025  033460  104167                          ERROR   167              ;LOAD HEADS NOT FOUND BEFORE TIMEOUT
6026  033462  013737  001434  004402          MOV     T10,TEMP2        ;SETUP TIMEOUT
6027  033470  004737  050050                  JSR     PC,FATT1         ;FIND ATTN
6028  033474  104067                          ERROR   67               ;ATTN NOT FOUND BEFORE TIMEOUT
6029  033476  005037  001176          2$:     CLR     $ESCAPE
6030
6031  033502  012765  100000  000000          MOV     #CCLR,RKCS1(R5)  ;CONTR CLEAR
6032  033510  004737  050256                  JSR     PC,GSTAT         ;GET LATEST STATUS
6033  033514  012737  050340  004374          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2 ;SHOULD BE VALUE
6034  033522  004737  052504                  JSR     PC,CKMR2         ;CHECK MR2
6035  033526  104063                          ERROR   63               ;MSG AO ERROR AT END OF HEAD LOADING
6036  033530  005037  004376                  CLR     SBMR3
6037  033534  004737  052566                  JSR     PC,CKMR3         ;CHECK MR3
6038  033540  104064                          ERROR   64               ;MSG BO ERROR AT END OF HEAD LOADING
6039
6040  033542  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
6041  033550  012765  000001  000026          MOV     #1,RKMR1(R5)     ;SELECT WORD 1
6042  033556  004737  050256                  JSR     PC,GSTAT
6043  033562  012737  001720  004374          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
6044  033570  004737  052504                  JSR     PC,CKMR2
6045  033574  104065                          ERROR   65               ;MSG A1 ERROR AT END OF HEAD LOADING
6046  033576  005037  004376                  CLR     SBMR3
6047  033602  004737  052566                  JSR     PC,CKMR3
6048  033606  104066                          ERROR   66               ;MSG B1 ERROR AT END OF HEAD LOADING
6049
6050
6051  033610  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
6052  033616  004737  051124                  JSR     PC,RDCYLD        ;READ CYL DIFF IN RKMR2
6053  033622  005737  001402                  TST     CYLDIF
6054  033626  001401                          BEQ     65$
6055  033630  104175                          ERROR   175              ;CYL DIFF/OFFSET NOT CLEARED AT END OF HEAD LOADING
6056  033632  004737  051210          65$:    JSR     PC,RDCYLA        ;READ CYL ADDR IN RKMR3
6057  033636  005737  001404                  TST     CYLADD
6058  033642  001401                          BEQ     66$
6059  033644  104176                          ERROR   176              ;CYL ADDR NOT CLEARED AT END OF HEAD LOADING
6060  033646                          66$:
6061
6062
6063  033646  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
6064  033654  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
6065  033662  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)         ;DRIVE CLEAR CMD
6066  033670  013737  001434  004400          MOV     T10,TEMP1
```

```
6067  033676  004737  047632                      JSR     PC,FRDY         ;FIND RDY
6068  033702  104151                              ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
6069  033704  004737  050016                      JSR     PC,TSTATN       ;TEST FOR ATTN
6070  033710  000401                              BR      67$
6071  033712  104154                              ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
6072  033714                            67$:
6073
6074  033714  012765  100000  000000            MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
6075  033722  004737  050256                      JSR     PC,GSTAT        ;GET LATEST STATUS
6076  033726  012737  010340  004374              MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2     ;SHOULD BE VALUE
6077  033734  004737  052504                      JSR     PC,CKMR2        ;CHECK MR2
6078  033740  104273                              ERROR   273             ;MSG AO ERROR AFTER DRIVE CLEAR CMD
6079  033742  005037  004376                      CLR     SBMR3
6080  033746  004737  052566                      JSR     PC,CKMR3        ;CHECK MR3
6081  033752  104265                              ERROR   265             ;MSG BO ERROR AFTER DRIVE CLEAR CMD
6082
6083  033754  012765  100000  000000            MOV     #CCLR,RKCS1(R5)
6084  033762  012765  000001  000026              MOV     #1,RKMR1(R5)    ;SELECT WORD 1
6085  033770  004737  050256                      JSR     PC,GSTAT
6086  033774  012737  001720  004374              MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
6087  034002  004737  052504                      JSR     PC,CKMR2
6088  034006  104274                              ERROR   274             ;MSG A1 ERROR AFTER DRIVE CLEAR CMD
6089  034010  005037  004376                      CLR     SBMR3
6090  034014  004737  052566                      JSR     PC,CKMR3
6091  034020  104266                              ERROR   266             ;MSG B1 ERROR AFTER DRIVE CLEAR CMD
6092
6093
6094  034022  004737  052770                      JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
6095  034026  000511                              BR      TST35           ;;GO TO NEXT TEST
6096                                                                        ;RETURN HERE IF SW 14 IS SET OR
6097                                                                        ;SW 8 WITH SWR <7:0> APPLY
6098
6099
6100  034030                            10$:
6101
6102  034030  004737  050320                      JSR     PC,SUBCLR
6103  034034  104024                              ERROR   24              ;CERR AFTER SCRL
6104
6105  034036  013765  001372  000020      68$:    MOV     TOCYL,RKDC(R5)  ;CYL#
6106
6107  034044  012765  000017  000000              MOV     #SEEK,RKCS1(R5) ;SEEK CMD TO RECONDITION DRIVE.
6108  034052  013737  001434  004400              MOV     TIO,TEMP1       ;SETUP TIMEOUT
6109  034060  004737  047632                      JSR     PC,FRDY         ;FIND RDY
6110  034064  104131                              ERROR   131             ;NO RDY AFTER SEEK CMD.
6111
6112  034066  013737  001444  004400              MOV     T5000,TEMP1
6113  034074  004737  050144                      JSR     PC,FATT2        ;FIND ATTN
6114  034100  104132                              ERROR   132             ;NO ATTN AFTER SEEK CMD
6115  034102  032737  100000  004336              BIT     #CERR,HCS1
6116  034110  001401                              BEQ     69$
6117  034112  104210                              ERROR   210             ;CERR AFTER SEEK CMD.
6118
6119  034114  004737  050320            69$:    JSR     PC,SUBCLR
6120  034120  104024                              ERROR   24              ;CERR AFTER SCLR
6121
6122
```

```
 6123  034122  005237  001372                  INC     TOCYL
 6124  034126  023727  001372  000633          CMP     TOCYL,#411.     ;ALL CYL DONE?
 6125  034134  001340                          BNE     68$
 6126
 6127  034136  004737  050320                  JSR     PC,SUBCLR
 6128  034142  104024                          ERROR   24              ;CERR AFTER SCLR
 6129
 6130  034144  005037  001176                  CLR     $ESCAPE
 6131  034150  005737  001430                  TST     LPFLG
 6132  034154  001402                          BEQ     70$
 6133  034156  000177  144726                  JMP     @$LPERR         ;SW 9 WAS SET.
 6134  034162  000177  144720          70$:    JMP     @$LPADR         ;SW 14 OR 8 WAS SET
 6135
 6136  034166                          20$:
 6137
 6138  034166  004737  050320                  JSR     PC,SUBCLR
 6139  034172  104024                          ERROR   24              ;CERR AFTER SCLR
 6140
 6141  034174  012765  000011  000000          MOV     #SRTSPL,RKCS1(R5)       ;START SPINDLE CMD
 6142  034202  013737  001434  004400          MOV     T10,TEMP1       ;SET TIMEOUT
 6143  034210  004737  047632                  JSR     PC,FRDY         ;FIND RDY
 6144  034214  104121                          ERROR   121             ;RDY NOT FOUND AFTER ST SPIN CMD.
 6145
 6146  034216  013737  001436  004402          MOV     T100,TEMP2      ;SETUP  TIMEOUT
 6147  034224  004737  050050                  JSR     PC,FATT1        ;FIND ATTN
 6148  034230  104067                          ERROR   67              ;NO ATTN AFTER ST SPIN CMD.
 6149
 6150  034232  005237  001430                  INC     LPFLG
 6151  034236  032777  001000  144674          BIT     #SW9,@SWR       ;LOOP ON ERROR?
 6152  034244  001271                          BNE     10$             ;YES, RECONDITION DRIVE
 6153  034246  000137  033476                  JMP     2$              ;RETURN TO MAINLINE
 6154  034252                          30$:
 6155
 6156
 6157                                  ;;*****************************************************************
 6158                                  ;*TEST 35       FORMAT PACK
 6159                                  ;*
 6160                                  ;*      THIS TEST FORMATS THE ENTIRE PACK IN 22 SECTOR FORMAT BY
 6161                                  ;*      DOING 1 CYLINDER INCREMENTAL SEEKS
 6162                                  ;*      FROM 0 TO 410 WITH WRITE HEADER COMMANDS (ALL TRACKS).
 6163                                  ;*      HEADERS WILL BE READ IN THE NEXT TEST
 6164                                  ;*
 6165                                  ;;*****************************************************************
 6166  034252  000004                  TST35:  SCOPE
 6167  034254  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
 6168
 6169
 6170  034262  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
 6171  034266  005737  001362                  TST     MODTST          ;SEE IF MODULE TESTING
 6172  034272  001402                          BEQ     22$             ;BR IF NO
 6173  034274  104401  061010                  TYPE    ,MSG20          ;RUNNING MODIFIED VERSION OF TEST
 6174
 6175  034300  005737  001476          22$:    TST     LIMERR          ;CHECK IF FOUND LIMIT DETECT ERROR
 6176  034304  001403                          BEQ     1$
 6177  034306  104170                          ERROR   170             ;FORMAT TEST BYPASSED, LIMIT DETECT ERROR
 6178  034310  000137  036442                  JMP     13$
```

```
6179  034314  005737  001474          1$:     TST     BSERR           ;CHECK IF FOUND BSE INFO OK
6180  034320  001403                          BEQ     2$
6181  034322  104177                          ERROR   177             ;FORMAT TEST BYPASSED-BSE ERROR
6182  034324  000137  036442                  JMP     13$
6183  034330  004737  050320          2$:     JSR     PC,SUBCLR
6184  034334  104024                          ERROR   24              ;CERR AFTER SCLR
6185
6186  034336  005037  001372                  CLR     TOCYL
6187
6188  034342  013737  001372  001406          MOV     TOCYL,CALADD    ;SETUP
6189  034350  012737  000001  001450          MOV     #1,HEAD         ;TO FILL
6190  034356  012737  000000  001454          MOV     #0,FORMAT       ;HEADER
6191  034364  004737  051460                  JSR     PC,FHDTAB       ;TABLE
6192
6193
6194  034370  012765  001500  000004  9$:     MOV     #HDTAB,RKBA(R5) ;THIS SECTION
6195  034376  012765  177676  000002          MOV     #-66.,RKWC(R5)  ;OF CODE
6196  034404  000337  001450                  SWAB    HEAD            ;IS TO RESTORE STANDARD FORMAT
6197  034410  013765  001450  000006          MOV     HEAD,RKDA(R5)   ;TO CYL 0
6198  034416  000337  001450                  SWAB    HEAD            ;HEAD 1 AND 2
6199
6200  034422  012765  000027  000000          MOV     #<WRHEAD),RKCS1(R5)     ;WRITE HEADER CMD
6201  034430  013737  001446  004400          MOV     T50000,TEMP1    ;SETUP TIMEOUT
6202  034436  004737  047632                  JSR     PC,FRDY         ;FIND RDY
6203  034442  104200                          ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
6204  034444  004737  050256                  JSR     PC,GSTAT        ;GET FRESH STATUS
6205  034450  032737  100000  004336          BIT     #CERR,HCS1
6206  034456  001405                          BEQ     64$
6207  034460  104201                          ERROR   201             ;CERR AFTER WRITE HEADER CMD
6208  034462  104401  060720                  TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
6209  034466  000137  047164                  JMP     $EOP            ;ABORT DRIVE
6210  034472                          64$:
6211
6212  034472  023727  001450  000002          CMP     HEAD,#2
6213  034500  001406                          BEQ     11$
6214
6215  034502  012737  000002  001450          MOV     #2,HEAD
6216  034510  004737  051460                  JSR     PC,FHDTAB
6217  034514  000725                          BR      9$
6218
6219  034516  012737  000001  001406  11$:    MOV     #1,CALADD       ;SETUP
6220  034524  005037  001450                  CLR     HEAD            ;FOR
6221  034530  005037  001454                  CLR     FORMAT          ;FHDTAB ROUTINE
6222
6223  034534  012737  000001  001400          MOV     #1,CALDIF       ;SETUP
6224  034542  005037  001370                  CLR     FRCYL           ;FOR
6225  034546  012737  000001  001372          MOV     #1,TOCYL        ;ERROR REPORT
6226
6227
6228  034554                          3$:
6229  034554  104415                          SCOP1
6230  034556  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
6231
6232  034562  004737  050320                  JSR     PC,SUBCLR
6233  034566  104024                          ERROR   24              ;CERR AFTER SCLR
6234
```

```
6235  034570  005737  001362              TST     MODTST          ;SEE IF MODULE TESTING
6236  034574  001404                      BEQ     18$             ;BR IF NO
6237  034576  012737  036402  001176      MOV     #16$,$ESCAPE
6238  034604  000403                      BR      19$
6239  034606  012737  036102  001176  18$:MOV     #10$,$ESCAPE
6240  034614  013765  001406  000020  19$:MOV     CALADD,RKDC(R5) ;CYL #
6241  034622  000337  001450              SWAB    HEAD
6242  034626  013765  001450  000006      MOV     HEAD,RKDA(R5)   ;HEAD #
6243  034634  000337  001450              SWAB    HEAD
6244
6245  034640  012765  000017  000000      MOV     #SEEK,RKCS1(R5) ;SEEK CMD
6246  034646  013737  001434  004400      MOV     T10,TEMP1       ;SETUP  TIMEOUT
6247  034654  004737  047632              JSR     PC,FRDY         ;FIND RDY
6248  034660  104131                      ERROR   131             ;NO RDY AFTER SEEK CMD
6249  034662  012737  030140  004374      MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,SBMR2        ;LOAD SHOULD BE DATA
6250  034670  004737  052504              JSR     PC,CKMR2        ;CHECK MR2
6251  034674  104203                      ERROR   203             ;MSG AD ERROR DURING SEEK CMD
6252  034676  005037  004376              CLR     SBMR3           ;MR3 SHOULD BE 0
6253  034702  004737  052566              JSR     PC,CKMR3        ;CHECK MR3
6254  034706  104204                      ERROR   204             ;MSG BD ERROR DURING SEEK CMD
6255
6256  034710  012765  100000  000000      MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
6257  034716  012765  000001  000026      MOV     #1,RKMR1(R5)    ;SELECT WORD 1
6258  034724  004737  050256              JSR     PC,GSTAT
6259  034730  012737  003720  004374      MOV     #<D.FWD!D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
6260  034736  004737  052504              JSR     PC,CKMR2
6261  034742  104205                      ERROR   205             ;MSG A1 ERROR DURING SEEK CMD
6262  034744  005037  004376              CLR     SBMR3
6263  034750  004737  052566              JSR     PC,CKMR3        ;CHECK MR3
6264  034754  104206                      ERROR   206             ;MSG B1 ERROR DURING SEEK CMD
6265
6266  034756  004737  051124              JSR     PC,RDCYLD
6267  034762  023727  001402  000001      CMP     CYLDIF,#1
6268  034770  001401                      BEQ     4$
6269  034772  104212                      ERROR   212             ;CYL DIFF INCORRECT DURING SEEK
6270
6271  034774  005737  001362          4$: TST     MODTST          ;SEE IF MODULE TESTING
6272  035000  001404                      BEQ     20$             ;BR IF NO
6273  035002  012737  036422  001176      MOV     #17$,$ESCAPE
6274  035010  000403                      BR      21$
6275
6276  035012  012737  036122  001176  20$:MOV     #12$,$ESCAPE
6277  035020  012737  004704  004400  21$:MOV     #2500.,TEMP1    ;SETUP TIMEOUT
6278
6279  035026  004737  050144              JSR     PC,FATT2        ;FIND ATTN
6280  035032  104132                      ERROR   132             ;NO ATTN AFTER SEEK CMD
6281  035034  032737  100000  004336      BIT     #CERR,HCS1
6282  035042  001401                      BEQ     65$
6283  035044  104210                      ERROR   210             ;CERR AFTER SEEK CMD
6284  035046                          65$:
6285
6286  035046  012765  100000  000000      MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
6287  035054  004737  050256              JSR     PC,GSTAT        ;GET LATEST STATUS
6288  035060  012737  050340  004374      MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2 ;SHOULD BE VALUE
6289  035066  004737  052504              JSR     PC,CKMR2        ;CHECK MR2
6290  035072  104133                      ERROR   133             ;MSG AD ERROR AFTER SEEK CMD
```

```
6291  035074  005037  004376                  CLR     SBMR3
6292  035100  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
6293  035104  104134                          ERROR   134             ;MSG B0 ERROR AFTER SEEK CMD
6294
6295  035106  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
6296  035114  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
6297  035122  004737  050256                  JSR     PC,GSTAT
6298  035126  012737  001720  004374          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
6299  035134  004737  052504                  JSR     PC,CKMR2
6300  035140  104135                          ERROR   135             ;MSG A1 ERROR AFTER SEEK CMD
6301  035142  005037  004376                  CLR     SBMR3
6302  035146  004737  052566                  JSR     PC,CKMR3
6303  035152  104136                          ERROR   136             ;MSG B1 ERROR AFTER SEEK CMD
6304
6305
6306  035154  004737  051124                  JSR     PC,RDCYLD       ;READ CYL DIFF IN MSG A2
6307  035160  005737  001402                  TST     CYLDIF
6308  035164  001401                          BEQ     66$
6309  035166  104137                          ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
6310
6311  035170                          66$:
6312
6313  035170  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
6314  035176  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
6315  035204  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
6316  035212  013737  001434  004400          MOV     T10,TEMP1
6317  035220  004737  047632                  JSR     PC,FRDY         ;FIND RDY
6318  035224  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
6319  035226  004737  050016                  JSR     PC,TSTATN       ;TEST FOR ATTN
6320  035232  000401                          BR      67$
6321  035234  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
6322  035236                          67$:
6323
6324  035236  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
6325  035244  004737  050256                  JSR     PC,GSTAT        ;GET LATEST STATUS
6326  035250  012737  010340  004374          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2     ;SHOULD BE VALUE
6327  035256  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
6328  035262  104273                          ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
6329  035264  005037  004376                  CLR     SBMR3
6330  035270  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
6331  035274  104265                          ERROR   265             ;MSG B0 ERROR AFTER DRIVE CLEAR CMD
6332
6333  035276  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
6334  035304  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
6335  035312  004737  050256                  JSR     PC,GSTAT
6336  035316  012737  001720  004374          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
6337  035324  004737  052504                  JSR     PC,CKMR2
6338  035330  104274                          ERROR   274             ;MSG A1 ERROR AFTER DRIVE CLEAR CMD
6339  035332  005037  004376                  CLR     SBMR3
6340  035336  004737  052566                  JSR     PC,CKMR3
6341  035342  104266                          ERROR   266             ;MSG B1 ERROR AFTER DRIVE CLEAR CMD
6342
6343
6344  035344  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CONTR CLR
6345  035352  013765  001406  000020          MOV     CALADD,RKDC(R5) ;RESTORE RKDS AFTER CLEAR
6346
```

```
6347  035360  004737  051210          JSR     PC,RDCYLA       ;READ CYL ADDR IN MSG B2
6348  035364  023737  001404  001406  CMP     CYLADD,CALADD
6349  035372  001401                  BEQ     5$
6350  035374  104232                  ERROR   232             ;CYL ADDR IN RKMR2 NOT=RKDC
6351
6352  035376                  5$:
6353  035376  104415                  SCOP1
6354  035400  012706  001100          MOV     #STACK,SP       ;RESTORE STK PTR
6355
6356  035404  004737  050320          JSR     PC,SUBCLR
6357  035410  104024                  ERROR   24              ;CERR AFTER SCLR
6358
6359  035412  005037  001176          CLR     $ESCAPE
6360  035416  004737  051460          JSR     PC,FHDTAB       ;FILL HEADER TABLE
6361  035422  000337  001450          SWAB    HEAD
6362  035426  013765  001450  000006  MOV     HEAD,RKDA(R5)   ;SET TRACK #
6363  035434  000337  001450          SWAB    HEAD
6364  035440  012765  001500  000004  MOV     #HDTAB,RKBA(R5) ;HEADER WORD TABLE
6365  035446  012765  177676  000002  MOV     #-66.,RKWC(R5)  ;WORD CT
6366  035454  013765  001406  000020  MOV     CALADD,RKDC(R5) ;CYL #
6367
6368
6369  035462  012765  000027  000000  MOV     #<WRHEAD>,RKCS1(R5)     ;WRITE HEADER CMD
6370  035470  013737  001446  004400  MOV     T50000,TEMP1    ;SETUP TIMEOUT
6371  035476  004737  047632          JSR     PC,FRDY         ;FIND RDY
6372  035502  104200                  ERROR   200             ;NO RDY AFTER WRITE HEADER CMD
6373  035504  004737  050256          JSR     PC,GSTAT        ;GET FRESH STATUS
6374  035510  032737  100000  004336  BIT     #CERR,HCS1
6375  035516  001405                  BEQ     68$
6376  035520  104201                  ERROR   201             ;CERR AFTER WRITE HEADER CMD
6377  035522  104401  060720          TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
6378  035526  000137  047164          JMP     $EOP            ;ABORT DRIVE
6379  035532                  68$:
6380
6381  035532  012765  100000  000000  MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
6382  035540  004737  050256          JSR     PC,GSTAT        ;GET LATEST STATUS
6383  035544  012737  010340  004374  MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2     ;SHOULD BE VALUE
6384  035552  004737  052504          JSR     PC,CKMR2        ;CHECK MR2
6385  035556  104277                  ERROR   277             ;MSG A0 ERROR AFTER WRITE HEADER CMD
6386  035560  005037  004376          CLR     SBMR3
6387  035564  004737  052566          JSR     PC,CKMR3        ;CHECK MR3
6388  035570  104267                  ERROR   267             ;MSG B0 ERROR AFTER WRITE HEADER CMD
6389
6390  035572  012765  100000  000000  MOV     #CCLR,RKCS1(R5)
6391  035600  012765  000001  000026  MOV     #1,RKMR1(R5)    ;SELECT WORD 1
6392  035606  004737  050256          JSR     PC,GSTAT
6393  035612  012737  001720  004374  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
6394  035620  004737  052504          JSR     PC,CKMR2
6395  035624  104300                  ERROR   300             ;MSG A1 ERROR AFTER WRITE HEADER CMD
6396  035626  005037  004376          CLR     SBMR3
6397  035632  004737  052566          JSR     PC,CKMR3
6398  035636  104270                  ERROR   270             ;MSG B1 ERROR AFTER WRITE HEADER CMD
6399
6400
6401
6402  035640  005737  001362          TST     MODTST          ;SEE IF MODULE TESTING
```

```
6403  035644  001402                        BEQ     23$             ;BR IF NO
6404  035646  000137  036142                JMP     14$             ;ELSE RESTORE HEADERS ONLY
6405
6406  035652  005237  001450        23$:    INC     HEAD
6407  035656  023727  001450  000002        CMP     HEAD,#2
6408  035664  001006                        BNE     6$
6409  035666  023727  001406  000632        CMP     CALADD,#410.    ;HEAD 2, SEE IF CYL 410
6410  035674  001002                        BNE     6$              ;DO NOT WRITE ON CYL 410 HEAD 2
6411  035676  000137  036074                JMP     7$
6412
6413  035702  023727  001450  000003  6$:   CMP     HEAD,#3         ;ALL HEADS DONE?
6414  035710  001232                        BNE     5$              ;BR IF NO
6415  035712  005037  001450                CLR     HEAD
6416  035716  005237  001406                INC     CALADD
6417  035722  005237  001370                INC     FRCYL           ;FOR ERR REPORT
6418  035726  005237  001372                INC     TOCYL           ;FOR ERR REPORT
6419  035732  000137  034554                JMP     3$
6420
6421
6422
6423  035736                        8$:
6424
6425  035736  004737  050320                JSR     PC,SUBCLR
6426  035742  104024                        ERROR   24              ;CERR AFTER SCRL
6427
6428  035744  013765  001372  000020  69$:   MOV     TOCYL,RKDC(R5)  ;CYL#
6429
6430  035752  012765  000017  000000        MOV     #SEEK,RKCS1(R5) ;SEEK CMD TO RECONDITION DRIVE.
6431  035760  013737  001434  004400        MOV     T10,TEMP1       ;SETUP TIMEOUT
6432  035766  004737  047632                JSR     PC,FRDY         ;FIND RDY
6433  035772  104131                        ERROR   131             ;NO RDY AFTER SEEK CMD.
6434
6435  035774  013737  001444  004400        MOV     T5000,TEMP1
6436  036002  004737  050144                JSR     PC,FATT2        ;FIND ATTN
6437  036006  104132                        ERROR   132             ;NO ATTN AFTER SEEK CMD
6438  036010  032737  100000  004336        BIT     #CERR,HCS1
6439  036016  001401                        BEQ     70$
6440  036020  104210                        ERROR   210             ;CERR AFTER SEEK CMD.
6441
6442  036022  004737  050320        70$:    JSR     PC,SUBCLR
6443  036026  104024                        ERROR   24              ;CERR AFTER SCLR
6444
6445
6446  036030  005337  001372                DEC     TOCYL
6447  036034  023727  001372  177777        CMP     TOCYL,#-1       ;ALL CYL DONE?
6448  036042  001340                        BNE     69$
6449
6450  036044  004737  050320                JSR     PC,SUBCLR
6451  036050  104024                        ERROR   24              ;CERR AFTER SCLR
6452
6453  036052  005037  001176                CLR     $ESCAPE
6454  036056  005737  001430                TST     LPFLG
6455  036062  001402                        BEQ     71$
6456  036064  000177  143020                JMP     @$LPERR         ;SW 9 WAS SET.
6457  036070  000177  143012        71$:    JMP     @$LPADR         ;SW 14 OR 8 WAS SET
6458
```

```

036074                          7$:
036074  004737  052770                  JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
036100  000560                          BR      TST36           ;;GO TO NEXT TEST
                                                                ;RETURN HERE IF SW 14 IS SET OR
                                                                ;SW 8 WITH SWR <7:0> APPLY

036102                          10$:
036102  005237  001430                  INC     LPFLG
036106  032777  001000  143024          BIT     #SW9,@SWR       ;LOOP ON ERROR?
036114  001310                          BNE     8$              ;YES, RECONDITION DRIVE
036116  000137  034774                  JMP     4$              ;RETURN TO MAINLINE

036122                          12$:
036122  005237  001430                  INC     LPFLG
036126  032777  001000  143004          BIT     #SW9,@SWR       ;LOOP ON ERROR?
036134  001300                          BNE     8$              ;YES, RECONDITION DRIVE
036136  000137  035376                  JMP     5$              ;RETURN TO MAINLINE


036142  005237  001450          14$:    INC     HEAD
036146  023727  001450  000003          CMP     HEAD,#3         ;SEE IF ALL HEADS DONE
036154  001402                          BEQ     15$             ;BR IF YES TO GO BACK TO CYL 0
036156  000137  035376                  JMP     5$              ;ELSE REPEAT FOR NEXT HEAD

036162  005065  000006          15$:    CLR     RKDA(R5)        ;SEEK TO CYL 0 & READ HEADERS
036166  005037  001372                  CLR     TOCYL           ;TO RECONDITION DRIVE

036172  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD TO RECONDITION DRIVE.
036200  013737  001434  004400          MOV     T10,TEMP1       ;SETUP TIMEOUT
036206  004737  047632                  JSR     PC,FRDY         ;FIND RDY
036212  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD.

036214  013737  001444  004400          MOV     T5000,TEMP1
036222  004737  050144                  JSR     PC,FATT2        ;FIND ATTN
036226  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
036230  032737  100000  004336          BIT     #CERR,HCS1
036236  001401                          BEQ     72$
036240  104210                          ERROR   210             ;CERR AFTER SEEK CMD.

036242  004737  050320          72$:    JSR     PC,SUBCLR
036246  104024                          ERROR   24              ;CERR AFTER SCLR



036250  012700  001704                  MOV     #RHTAB,R0
036254  012765  000025  000000          MOV     #<RDHEAD>,RKCS1(R5)     ;READ HEADER CMD
036262  013737  001446  004400          MOV     T50000,TEMP1    ;SETUP  TIMEOUT
036270  004737  047632                  JSR     PC,FRDY         ;FIND RDY
036274  104171                          ERROR   171             ;NO RDY AFTER READ HEADER CMD
036276  032737  100000  004336          BIT     #CERR,HCS1
036304  001405                          BEQ     74$
036306  104174                          ERROR   174             ;CERR AFTER READ HEADER CMD
036310  104401  060720                  TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
036314  000137  047164                  JMP     $EOP            ;ABORT DRIVE

```

```
6515  036320  016520  000024          74$:    MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
6516  036324  016520  000024                  MOV     RKDB(R5),(R0)+  ;2'ND WORD
6517  036330  016520  000024                  MOV     RKDB(R5),(R0)+  ;3'RD WORD
6518
6519
6520  036334  032765  100000  000010          BIT     #DLT,RKCS2(R5)
6521  036342  001407                          BEQ     75$
6522  036344  004737  050256                  JSR     PC,GSTAT
6523  036350  104173                          ERROR   173             ;DLT AFTER READ HEADER CMD
6524  036352  104401  060720                  TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
6525  036356  000137  047164                  JMP     $EOP            ;ABORT DRIVE
6526  036362                          75$:
6527
6528  036362  023737  001704  001372          CMP     RHTAB,TOCYL     ;CHECK WORD 0 (CYL#) ONLY
6529  036370  001401                          BEQ     73$             ;BR IF SAME
6530  036372  104310                          ERROR   310             ;READ CYL WORD HEADER ERROR
6531  036374                          73$:
6532
6533  036374  004737  052770                  JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
6534  036400  000420                          BR      TST36           ;;GO TO NEXT TEST
6535                                                                  ;RETURN HERE IF SW 14 IS SET OR
6536                                                                  ;SW 8 WITH SWR <7:0> APPLY
6537  036402                          16$:
6538  036402  005237  001430                  INC     LPFLG
6539  036406  032777  001000  142524          BIT     #SW9,@SWR       ;LOOP ON ERROR?
6540  036414  001262                          BNE     15$             ;YES, RECONDITION DRIVE
6541  036416  000137  034774                  JMP     4$              ;RETURN TO MAINLINE
6542  036422                          17$:
6543  036422  005237  001430                  INC     LPFLG
6544  036426  032777  001000  142504          BIT     #SW9,@SWR       ;LOOP ON ERROR?
6545  036434  001252                          BNE     15$             ;YES, RECONDITION DRIVE
6546  036436  000137  035376                  JMP     5$              ;RETURN TO MAINLINE
6547
6548
6549  036442                          13$:
6550
6551
6552
6553                                          ;;*****************************************************************
6554                                          ;*TEST 36       DECREMENT FROM CYLINDER 410 TO 0 & READ HEADERS
6555                                          ;*
6556                                          ;*      THIS TEST VERIFIES MOTION IN THE NEGATIVE DIRECTION BY
6557                                          ;*      SINGLE CYLINDER INCREMENTAL SEEKS.
6558                                          ;*
6559                                          ;*****************************************************************
6560  036442  000004                  TST36:  SCOPE
6561  036444  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
6562  036452  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
6563
6564  036456  005737  001362                  TST     MODTST          ;SEE IF MODULE TESTING
6565  036462  001404                          BEQ     5$              ;BR IF NO
6566  036464  104401  061177                  TYPE    ,MSG21          ;BYP TESTS 36,40,41
6567  036470  000137  037654                  JMP     13$
6568  036474  012737  000632  001370  5$:     MOV     #410.,FRCYL     ;FROM CYL
6569  036502  012737  000631  001372          MOV     #409.,TOCYL     ;TO CYL
6570
```

```
6571  036510                                  1$:
6572  036510  104415                                  SCOP1
6573  036512  012706  001100                          MOV     #STACK,SP       ;RESTORE STK PTR
6574
6575  036516  004737  050320                          JSR     PC,SUBCLR
6576  036522  104024                                  ERROR   24              ;CERR AFTER SCLR
6577
6578  036524  012737  037614  001176                  MOV     #10$,$ESCAPE
6579  036532  013765  001372  000020                  MOV     TOCYL,RKDC(R5)  ;CYL #
6580
6581  036540  012765  000017  000000                  MOV     #SEEK,RKCS1(R5) ;SEEK CMD
6582  036546  013737  001434  004400                  MOV     T10,TEMP1       ;SETUP  TIMEOUT
6583  036554  004737  047632                          JSR     PC,FRDY         ;FIND RDY
6584  036560  104131                                  ERROR   131             ;NO RDY AFTER SEEK CMD
6585  036562  012737  030140  004374                  MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,SBMR2        ;LOAD SHOULD BE DATA
6586  036570  004737  052504                          JSR     PC,CKMR2        ;CHECK MR2
6587  036574  104203                                  ERROR   203             ;MSG A0 ERROR DURING SEEK CMD
6588  036576  005037  004376                          CLR     SBMR3           ;MR3 SHOULD BE 0
6589  036602  004737  052566                          JSR     PC,CKMR3        ;CHECK MR3
6590  036606  104204                                  ERROR   204             ;MSG B0 ERROR DURING SEEK CMD
6591
6592  036610  012765  100000  000000                  MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
6593  036616  012765  000001  000026                  MOV     #1,RKMR1(R5)    ;SELECT WORD 1
6594  036624  004737  050256                          JSR     PC,GSTAT
6595  036630  012737  005720  004374                  MOV     #<D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
6596  036636  004737  052504                          JSR     PC,CKMR2
6597  036642  104205                                  ERROR   205             ;MSG A1 ERROR DURING SEEK CMD
6598  036644  005037  004376                          CLR     SBMR3
6599  036650  004737  052566                          JSR     PC,CKMR3        ;CHECK MR3
6600  036654  104206                                  ERROR   206             ;MSG B1 ERROR DURING SEEK CMD
6601
6602  036656  004737  051124                          JSR     PC,RDCYLD
6603  036662  023727  001402  000001                  CMP     CYLDIF,#1
6604  036670  001401                                  BEQ     2$
6605  036672  104212                                  ERROR   212             ;CYL DIFF INCORRECT DURING SEEK
6606
6607  036674  012737  037634  001176  2$:             MOV     #12$,$ESCAPE
6608  036702  012737  004704  004400                  MOV     #2500.,TEMP1    ;SETUP TIMEOUT
6609
6610  036710  004737  050144                          JSR     PC,FATT2        ;FIND ATTN
6611  036714  104132                                  ERROR   132             ;NO ATTN AFTER SEEK CMD
6612  036716  032737  100000  004336                  BIT     #CERR,HCS1
6613  036724  001401                                  BEQ     64$
6614  036726  104210                                  ERROR   210             ;CERR AFTER SEEK CMD
6615  036730                                  64$:
6616
6617  036730  012765  100000  000000                  MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
6618  036736  004737  050256                          JSR     PC,GSTAT        ;GET LATEST STATUS
6619  036742  012737  050340  004374                  MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2 ;SHOULD BE VALUE
6620  036750  004737  052504                          JSR     PC,CKMR2        ;CHECK MR2
6621  036754  104133                                  ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
6622  036756  005037  004376                          CLR     SBMR3
6623  036762  004737  052566                          JSR     PC,CKMR3        ;CHECK MR3
6624  036766  104134                                  ERROR   134             ;MSG B0 ERROR AFTER SEEK CMD
6625
6626  036770  012765  100000  000000                  MOV     #CCLR,RKCS1(R5)
```

```
6627  036776  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
6628  037004  004737  050256                  JSR     PC,GSTAT
6629  037010  012737  001720  004374          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
6630  037016  004737  052504                  JSR     PC,CKMR2
6631  037022  104135                          ERROR   135             ;MSG A1 ERROR AFTER SEEK CMD
6632  037024  005037  004376                  CLR     SBMR3
6633  037030  004737  052566                  JSR     PC,CKMR3
6634  037034  104136                          ERROR   136             ;MSG B1 ERROR AFTER SEEK CMD
6635
6636
6637  037036  004737  051124                  JSR     PC,RDCYLD       ;READ CYL DIFF IN MSG A2
6638  037042  005737  001402                  TST     CYLDIF
6639  037046  001401                          BEQ     65$
6640  037050  104137                          ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
6641
6642  037052                          65$:
6643
6644  037052  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
6645  037060  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
6646  037066  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
6647  037074  013737  001434  004400          MOV     T10,TEMP1
6648  037102  004737  047632                  JSR     PC,FRDY         ;FIND RDY
6649  037106  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
6650  037110  004737  050016                  JSR     PC,TSTATN       ;TEST FOR ATTN
6651  037114  000401                          BR      66$
6652  037116  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
6653  037120                          66$:
6654
6655  037120  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
6656  037126  004737  050256                  JSR     PC,GSTAT        ;GET LATEST STATUS
6657  037132  012737  010340  004374          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2     ;SHOULD BE VALUE
6658  037140  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
6659  037144  104273                          ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
6660  037146  005037  004376                  CLR     SBMR3
6661  037152  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
6662  037156  104265                          ERROR   265             ;MSG B0 ERROR AFTER DRIVE CLEAR CMD
6663
6664  037160  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
6665  037166  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
6666  037174  004737  050256                  JSR     PC,GSTAT
6667  037200  012737  001720  004374          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
6668  037206  004737  052504                  JSR     PC,CKMR2
6669  037212  104274                          ERROR   274             ;MSG A1 ERROR AFTER DRIVE CLEAR CMD
6670  037214  005037  004376                  CLR     SBMR3
6671  037220  004737  052566                  JSR     PC,CKMR3
6672  037224  104266                          ERROR   266             ;MSG B1 ERROR AFTER DRIVE CLEAR CMD
6673
6674
6675  037226  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
6676  037234  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
6677  037242  004737  051210                  JSR     PC,RDCYLA       ;READ CYL ADDR IN MSG B2
6678  037246  023737  001404  001372          CMP     CYLADD,TOCYL
6679  037254  001401                          BEQ     3$
6680  037256  104207                          ERROR   207             ;CYL ADDR IN RKMR3 NOT = RKDC
6681
6682  037260                          3$:
```

```
6683  037260  104415                          SCOPI
6684  037262  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
6685
6686  037266  004737  050320                  JSR     PC,SUBCLR
6687  037272  104024                          ERROR   24              ;CERR AFTER SCLR
6688
6689  037274  005037  001176                  CLR     $ESCAPE
6690
6691  037300  013765  001372  000020          MOV     TOCYL,RKDC(R5)  ;CYL #
6692
6693
6694  037306  012700  001704                  MOV     #RHTAB,R0
6695  037312  012765  000025  000000          MOV     #<RDHEAD>,RKCS1(R5)     ;READ HEADER CMD
6696  037320  013737  001446  004400          MOV     TS0000,TEMP1    ;SETUP  TIMEOUT
6697  037326  004737  047632                  JSR     PC,FRDY         ;FIND RDY
6698  037332  104171                          ERROR   171             ;NO RDY AFTER READ HEADER CMD
6699  037334  032737  100000  004336          BIT     #CERR,HCS1
6700  037342  001405                          BEQ     68$
6701  037344  104174                          ERROR   174             ;CERR AFTER READ HEADER CMD
6702  037346  104401  060720                  TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
6703  037352  000137  047164                  JMP     $EOP            ;ABORT DRIVE
6704
6705  037356  016520  000024          68$:    MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
6706  037362  016520  000024                  MOV     RKDB(R5),(R0)+  ;2'ND WORD
6707  037366  016520  000024                  MOV     RKDB(R5),(R0)+  ;3'RD WORD
6708
6709
6710  037372  032765  100000  000010          BIT     #DLT,RKCS2(R5)
6711  037400  001407                          BEQ     69$
6712  037402  004737  050256                  JSR     PC,GSTAT
6713  037406  104173                          ERROR   173             ;DLT AFTER READ HEADER CMD
6714  037410  104401  060720                  TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
6715  037414  000137  047164                  JMP     $EOF            ;ABORT DRIVE
6716  037420                          69$:
6717
6718  037420  023737  001704  001372          CMP     RHTAB,TOCYL     ;CHECK WORD 0 (CYL#) ONLY
6719  037426  001401                          BEQ     67$             ;BR IF SAME
6720  037430  104310                          ERROR   310             ;READ CYL WORD HEADER ERROR
6721  037432                          67$:
6722
6723  037432  005337  001370                  DEC     FRCYL
6724  037436  001404                          BEQ     4$
6725  037440  005337  001372                  DEC     TOCYL
6726  037444  000137  036510                  JMP     1$
6727
6728  037450                          4$:
6729  037450  004737  052770                  JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
6730  037454  000477                          BR      TST37           ;;GO TO NEXT TEST
6731                                                                  ;RETURN HERE IF SW 14 IS SET OR
6732                                                                  ;SW 8 WITH SWR <7:0> APPLY
6733
6734
6735  037456                          6$:
6736
6737  037456  004737  050320                  JSR     PC,SUBCLR
6738  037462  104024                          ERROR   24              ;CERR AFTER SCRL
```

```
6739
6740  037464  013765  001372  000020  70$:    MOV     TOCYL,RKDC(R5)  ;CYL#
6741
6742  037472  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD TO RECONDITION DRIVE.
6743  037500  013737  001434  004400          MOV     T10,TEMP1       ;SETUP TIMEOUT
6744  037506  004737  047632                  JSR     PC,FRDY         ;FIND RDY
6745  037512  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD.
6746
6747  037514  013737  001444  004400          MOV     T5000,TEMP1
6748  037522  004737  050144                  JSR     PC,FATT2        ;FIND ATTN
6749  037526  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
6750  037530  032737  100000  004336          BIT     #CERR,HCS1
6751  037536  001401                          BEQ     71$
6752  037540  104210                          ERROR   210             ;CERR AFTER SEEK CMD.
6753
6754  037542  004737  050320          71$:    JSR     PC,SUBCLR
6755  037546  104024                          ERROR   24              ;CERR AFTER SCLR
6756
6757
6758  037550  005237  001372                  INC     TOCYL
6759  037554  023727  001372  000633          CMP     TOCYL,#411.     ;ALL CYL DONE?
6760  037562  001340                          BNE     70$
6761
6762  037564  004737  050320                  JSR     PC,SUBCLR
6763  037570  104024                          ERROR   24              ;CERR AFTER SCLR
6764
6765  037572  005037  001176                  CLR     $ESCAPE
6766  037576  005737  001430                  TST     LPFLG
6767  037602  001402                          BEQ     72$
6768  037604  000177  141300                  JMP     @$LPERR         ;SW 9 WAS SET.
6769  037610  000177  141272          72$:    JMP     @$LPADR         ;SW 14 OR 8 WAS SET
6770
6771
6772
6773  037614                          10$:
6774  037614  005237  001430                  INC     LPFLG
6775  037620  032777  001000  141312          BIT     #SW9,@SWR       ;LOOP ON ERROR?
6776  037626  001313                          BNE     6$              ;YES, RECONDITION DRIVE
6777  037630  000137  036674                  JMP     2$              ;RETURN TO MAINLINE
6778
6779  037634                          12$:
6780  037634  005237  001430                  INC     LPFLG
6781  037640  032777  001000  141272          BIT     #SW9,@SWR       ;LOOP ON ERROR?
6782  037646  001303                          BNE     6$              ;YES, RECONDITION DRIVE
6783  037650  000137  037260                  JMP     3$              ;RETURN TO MAINLINE
6784
6785  037654                          13$:
6786
6787
6788                                  ;;*****************************************************************
6789                                  ;*TEST 37       SEEK FROM CYL 0 TO ALL MAJOR CYLINDERS & READ HEADERS
6790                                  ;*
6791                                  ;*      THIS TEST SEEKS FROM CYL 0 TO ALL THE MAJOR CYLINDERS & READS HEADERS.
6792                                  ;*      IT THEN SEEKS CYL 0 & READS HEADERS.
6793                                  ;*
6794                                  ;*      MAJOR CYLINDERS ARE:    1  (DECIMAL)  =  1 (OCTAL)
```

```
6795                                    ;*                              2               2
6796                                    ;*                              4               4
6797                                    ;*                              8               10
6798                                    ;*                              16              20
6799                                    ;*                              32              40
6800                                    ;*                              64              100
6801                                    ;*                              128             200
6802                                    ;*                              256             400
6803                                    ;*
6804                                    ;*****************************************************************
6805  037654  000004                    TST37:  SCOPE
6806  037656  012737  000001  001174            MOV     #1,$TIMES       ;;DO 1 ITERATION
6807  037664  012706  001100                    MOV     #STACK,SP       ;RESTORE STK PTR
6808
6809  037670  012737  000000  001370            MOV     #0,FRCYL        ;SETUP FROM CYL
6810  037676  012737  000001  001372            MOV     #1,TOCYL        ;SETUP TO CYL
6811
6812  037704                            1$:
6813  037704  104415                            SCOP1
6814  037706  012706  001100                    MOV     #STACK,SP       ;RESTORE STK PTR
6815
6816  037712  004737  050320                    JSR     PC,SUBCLR
6817  037716  104024                            ERROR   24              ;CERR AFTER SCLR
6818
6819  037720  012737  041706  001176            MOV     #10$,$ESCAPE
6820  037726  013737  001370  004404            MOV     FRCYL,TEMP3     ;SETUP
6821  037734  013737  001372  004406            MOV     TOCYL,TEMP4     ;CYL DIFF
6822  037742  163737  004404  004406            SUB     TEMP3,TEMP4     ;FOR
6823  037750  013737  004406  001400            MOV     TEMP4,CALDIF    ;ERROR PRINTOUT
6824
6825  037756  013765  001372  000020            MOV     TOCYL,RKDC(R5)  ;GO TO CYL #
6826
6827  037764  012765  000017  000000            MOV     #SEEK,RKCS1(R5) ;SEEK CMD
6828  037772  013737  001434  004400            MOV     T10,TEMP1       ;SETUP  TIMEOUT
6829  040000  004737  047632                    JSR     PC,FRDY         ;FIND RDY
6830  040004  104131                            ERROR   131             ;NO RDY AFTER SEEK CMD
6831  040006  012737  030140  004374            MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,SBMR2        ;LOAD SHOULD BE DATA
6832  040014  004737  052504                    JSR     PC,CKMR2        ;CHECK MR2
6833  040020  104203                            ERROR   203             ;MSG AO ERROR DURING SEEK CMD
6834  040022  005037  004376                    CLR     SBMR3           ;MR3 SHOULD BE 0
6835  040026  004737  052566                    JSR     PC,CKMR3        ;CHECK MR3
6836  040032  104204                            ERROR   204             ;MSG BO ERROR DURING SEEK CMD
6837
6838  040034  012765  100000  000000            MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
6839  040042  012765  000001  000026            MOV     #1,RKMR1(R5)    ;SELECT WORD 1
6840  040050  004737  050256                    JSR     PC,GSTAT
6841  040054  012737  003720  004374            MOV     #<D.FWD!D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
6842  040062  004737  052504                    JSR     PC,CKMR2
6843  040066  104205                            ERROR   205             ;MSG A1 ERROR DURING SEEK CMD
6844  040070  005037  004376                    CLR     SBMR3
6845  040074  004737  052566                    JSR     PC,CKMR3        ;CHECK MR3
6846  040100  104206                            ERROR   206             ;MSG B1 ERROR DURING SEEK CMD
6847
6848  040102  012737  041726  001176  2$:       MOV     #12$,$ESCAPE
6849  040110  013737  001444  004400            MOV     T5000,TEMP1     ;SETUP  TIMEOUT
6850
```

```
                                                JSR     PC,FATT2        ;FIND ATTN
                                                ERROR   132             ;NO ATTN AFTER SEEK CMD
                                                BIT     #CERR,HCS1
                                                BEQ     64$
                                                ERROR   210             ;CERR AFTER SEEK CMD
                                        64$:
                                                MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
                                                JSR     PC,GSTAT        ;GET LATEST STATUS
                                                MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2 ;SHOULD BE VALUE
                                                JSR     PC,CKMR2        ;CHECK MR2
                                                ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
                                                CLR     SBMR3
                                                JSR     PC,CKMR3        ;CHECK MR3
                                                ERROR   134             ;MSG B0 ERROR AFTER SEEK CMD

                                                MOV     #CCLR,RKCS1(R5)
                                                MOV     #1,RKMR1(R5)    ;SELECT WORD 1
                                                JSR     PC,GSTAT
                                                MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
                                                JSR     PC,CKMR2
                                                ERROR   135             ;MSG A1 ERROR AFTER SEEK CMD
                                                CLR     SBMR3
                                                JSR     PC,CKMR3
                                                ERROR   136             ;MSG B1 ERROR AFTER SEEK CMD


                                                JSR     PC,RDCYLD       ;READ CYL DIFF IN MSG A2
                                                TST     CYLDIF
                                                BEQ     65$
                                                ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD

                                        65$:

                                                MOV     #CCLR,RKCS1(R5)
                                                MOV     $UNIT,RKCS2(R5) ;DRIVE#
                                                MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
                                                MOV     T10,TEMP1
                                                JSR     PC,FRDY         ;FIND RDY
                                                ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
                                                JSR     PC,TSTATN       ;TEST FOR ATTN
                                                BR      66$
                                                ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
                                        66$:

                                                MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
                                                JSR     PC,GSTAT        ;GET LATEST STATUS
                                                MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2     ;SHOULD BE VALUE
                                                JSR     PC,CKMR2        ;CHECK MR2
                                                ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
                                                CLR     SBMR3
                                                JSR     PC,CKMR3        ;CHECK MR3
                                                ERROR   265             ;MSG B0 ERROR AFTER DRIVE CLEAR CMD

                                                MOV     #CCLR,RKCS1(R5)
                                                MOV     #1,RKMR1(R5)    ;SELECT WORD 1
```

```
6907  040402  004737  050256                  JSR     PC,GSTAT
6908  040406  012737  001720  004374          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
6909  040414  004737  052504                  JSR     PC,CKMR2
6910  040420  104274                          ERROR   274             ;MSG A1 ERROR AFTER DRIVE CLEAR CMD
6911  040422  005037  004376                  CLR     SBMR3
6912  040426  004737  052566                  JSR     PC,CKMR3
6913  040432  104266                          ERROR   266             ;MSG B1 ERROR AFTER DRIVE CLEAR CMD
6914
6915
6916  040434  004737  051210                  JSR     PC,RDCYLA       ;READ CYL ADDR IN MSG B2
6917  040440  023737  001404  001372          CMP     CYLADD,TOCYL
6918  040446  001401                          BEQ     3$
6919  040450  104207                          ERROR   207             ;CYL ADDR IN RKMR3 NOT=RKDC
6920
6921  040452                          3$:
6922  040452  104415                          SCOP1
6923  040454  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
6924
6925  040460  004737  050320                  JSR     PC,SUBCLR
6926  040464  104024                          ERROR   24              ;CERR AFTER SCLR
6927
6928  040466  005037  001176                  CLR     $ESCAPE
6929  040472  013765  001372  000020          MOV     TOCYL,RKDC(R5)  ;CYL #
6930
6931
6932  040500  012700  001704                  MOV     #RHTAB,R0
6933  040504  012765  000025  000000          MOV     #<RDHEAD>,RKCS1(R5)     ;READ HEADER CMD
6934  040512  013737  001446  004400          MOV     T50000,TEMP1    ;SETUP TIMEOUT
6935  040520  004737  047632                  JSR     PC,FRDY         ;FIND RDY
6936  040524  104171                          ERROR   171             ;NO RDY AFTER READ HEADER CMD
6937  040526  032737  100000  004336          BIT     #CERR,HCS1
6938  040534  001405                          BEQ     68$
6939  040536  104174                          ERROR   174             ;CERR AFTER READ HEADER CMD
6940  040540  104401  060720                  TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
6941  040544  000137  047164                  JMP     $EOP            ;ABORT DRIVE
6942
6943  040550  016520  000024          68$:    MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
6944  040554  016520  000024                  MOV     RKDB(R5),(R0)+  ;2'ND WORD
6945  040560  016520  000024                  MOV     RKDB(R5),(R0)+  ;3'RD WORD
6946
6947
6948  040564  032765  100000  000010          BIT     #DLT,RKCS2(R5)
6949  040572  001407                          BEQ     69$
6950  040574  004737  050256                  JSR     PC,GSTAT
6951  040600  104173                          ERROR   173             ;DLT AFTER READ HEADER CMD
6952  040602  104401  060720                  TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
6953  040606  000137  047164                  JMP     $EOP            ;ABORT DRIVE
6954  040612                          69$:
6955
6956  040612  023737  001704  001372          CMP     RHTAB,TOCYL     ;CHECK WORD 0 (CYL#) ONLY
6957  040620  001401                          BEQ     67$             ;BR IF SAME
6958  040622  104310                          ERROR   310             ;READ CYL WORD HEADER ERROR
6959  040624                          67$:
6960
6961
6962  040624  104415                          SCOP1
```

```
 6963  040626  012706  001100                    MOV     #STACK,SP       ;RESTORE STK PTR
 6964
 6965  040632  004737  050320                    JSR     PC,SUBCLR
 6966  040636  104024                            ERROR   24              ;CERR AFTER SCLR
 6967
 6968  040640  012737  041746  001176            MOV     #14$,$ESCAPE
 6969  040646  013765  001370  000020            MOV     FRCYL,RKDC(R5)  ;RETURN TO CYL #
 6970  040654  013737  001370  001374            MOV     FRCYL,CCYL      ;CURRENT CYL FOR TRUERR ROUTINE
 6971
 6972  040662  012765  000017  000000            MOV     #SEEK,RKCS1(R5) ;SEEK CMD
 6973  040670  013737  001434  004400            MOV     T10,TEMP1       ;SETUP  TIMEOUT
 6974  040676  004737  047632                    JSR     PC,FRDY         ;FIND RDY
 6975  040702  104131                            ERROR   131             ;NO RDY AFTER SEEK CMD
 6976  040704  012737  030140  004374            MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,SBMR2        ;LOAD SHOULD BE DATA
 6977  040712  004737  052504                    JSR     PC,CKMR2        ;CHECK MR2
 6978  040716  104203                            ERROR   203             ;MSG A0 ERROR DURING SEEK CMD
 6979  040720  005037  004376                    CLR     SBMR3           ;MR3 SHOULD BE 0
 6980  040724  004737  052566                    JSR     PC,CKMR3        ;CHECK MR3
 6981  040730  104204                            ERROR   204             ;MSG B0 ERROR DURING SEEK CMD
 6982
 6983  040732  012765  100000  000000            MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
 6984  040740  012765  000001  000026            MOV     #1,RKMR1(R5)    ;SELECT WORD 1
 6985  040746  004737  050256                    JSR     PC,GSTAT
 6986  040752  012737  005720  004374            MOV     #<D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
 6987  040760  004737  052504                    JSR     PC,CKMR2
 6988  040764  104205                            ERROR   205             ;MSG A1 ERROR DURING SEEK CMD
 6989  040766  005037  004376                    CLR     SBMR3
 6990  040772  004737  052566                    JSR     PC,CKMR3        ;CHECK MR3
 6991  040776  104206                            ERROR   206             ;MSG B1 ERROR DURING SEEK CMD
 6992
 6993
 6994  041000  012737  041766  001176  4$:       MOV     #16$,$ESCAPE
 6995  041006  013737  001444  004400            MOV     T5000,TEMP1     ;SETUP  TIMEOUT
 6996
 6997  041014  004737  050144                    JSR     PC,FATT2        ;FIND ATTN
 6998  041020  104132                            ERROR   132             ;NO ATTN AFTER SEEK CMD
 6999  041022  032737  100000  004336            BIT     #CERR,HCS1
 7000  041030  001401                            BEQ     70$
 7001  041032  104210                            ERROR   210             ;CERR AFTER SEEK CMD
 7002  041034                          70$:
 7003
 7004  041034  012765  100000  000000            MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
 7005  041042  004737  050256                    JSR     PC,GSTAT        ;GET LATEST STATUS
 7006  041046  012737  050340  004374            MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2 ;SHOULD BE VALUE
 7007  041054  004737  052504                    JSR     PC,CKMR2        ;CHECK MR2
 7008  041060  104133                            ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
 7009  041062  005037  004376                    CLR     SBMR3
 7010  041066  004737  052566                    JSR     PC,CKMR3        ;CHECK MR3
 7011  041072  104134                            ERROR   134             ;MSG B0 ERROR AFTER SEEK CMD
 7012
 7013  041074  012765  100000  000000            MOV     #CCLR,RKCS1(R5)
 7014  041102  012765  000001  000026            MOV     #1,RKMR1(R5)    ;SELECT WORD 1
 7015  041110  004737  050256                    JSR     PC,GSTAT
 7016  041114  012737  001720  004374            MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
 7017  041122  004737  052504                    JSR     PC,CKMR2
 7018  041126  104135                            ERROR   135             ;MSG A1 ERROR AFTER SEEK CMD
```

```
7019  041130  005037  004376                  CLR     SBMR3
7020  041134  004737  052566                  JSR     PC,CKMR3
7021  041140  104136                          ERROR   136             ;MSG B1 ERROR AFTER SEEK CMD
7022
7023
7024  041142  004737  051124                  JSR     PC,RDCYLD       ;READ CYL DIFF IN MSG A2
7025  041146  005737  001402                  TST     CYLDIF
7026  041152  001401                          BEQ     71$
7027  041154  104137                          ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
7028
7029  041156                          71$:
7030
7031  041156  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
7032  041164  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
7033  041172  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
7034  041200  013737  001434  004400          MOV     T10,TEMP1
7035  041206  004737  047632                  JSR     PC,FRDY         ;FIND RDY
7036  041212  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
7037  041214  004737  050016                  JSR     PC,TSTATN       ;TEST FOR ATTN
7038  041220  000401                          BR      72$
7039  041222  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
7040  041224                          72$:
7041
7042  041224  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
7043  041232  004737  050256                  JSR     PC,GSTAT        ;GET LATEST STATUS
7044  041236  012737  010340  004374          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2     ;SHOULD BE VALUE
7045  041244  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
7046  041250  104273                          ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
7047  041252  005037  004376                  CLR     SBMR3
7048  041256  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
7049  041262  104265                          ERROR   265             ;MSG B0 ERROR AFTER DRIVE CLEAR CMD
7050
7051  041264  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
7052  041272  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
7053  041300  004737  050256                  JSR     PC,GSTAT
7054  041304  012737  001720  004374          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
7055  041312  004737  052504                  JSR     PC,CKMR2
7056  041316  104274                          ERROR   274             ;MSG A1 ERROR AFTER DRIVE CLEAR CMD
7057  041320  005037  004376                  CLR     SBMR3
7058  041324  004737  052566                  JSR     PC,CKMR3
7059  041330  104266                          ERROR   266             ;MSG B1 ERROR AFTER DRIVE CLEAR CMD
7060
7061
7062  041332  004737  051210                  JSR     PC,RDCYLA       ;READ CYL ADDR IN MSG B2
7063  041336  023737  001404  001370          CMP     CYLADD,FRCYL
7064  041344  001401                          BEQ     5$
7065  041346  104243                          ERROR   243             ;CYL ADDR IN RKMR3 NOT=RKDC
7066
7067  041350                          5$:
7068  041350  104415                          SCOP1
7069  041352  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
7070
7071  041356  004737  050320                  JSR     PC,SUBCLR
7072  041362  104024                          ERROR   24              ;CERR AFTER SCLR
7073
7074  041364  005037  001176                  CLR     $ESCAPE
```

```
7075  041370  013765  001370  000020          MOV     FRCYL,RKDC(R5)  ;CYL #
7076
7077
7078  041376  012700  001704                  MOV     #RHTAB,R0
7079  041402  012765  000025  000000          MOV     #<RDHEAD>,RKCS1(R5)     ;READ HEADER CMD
7080  041410  013737  001446  004400          MOV     T50000,TEMP1    ;SETUP TIMEOUT
7081  041416  004737  047632                  JSR     PC,FRDY         ;FIND RDY
7082  041422  104171                          ERROR   171             ;NO RDY AFTER READ HEADER CMD
7083  041424  032737  100000  004336          BIT     #CERR,HCS1
7084  041432  001405                          BEQ     74$
7085  041434  104174                          ERROR   174             ;CERR AFTER READ HEADER CMD
7086  041436  104401  060720                  TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
7087  041442  000137  047164                  JMP     $EOP            ;ABORT DRIVE
7088
7089  041446  016520  000024          74$:    MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
7090  041452  016520  000024                  MOV     RKDB(R5),(R0)+  ;2'ND WORD
7091  041456  016520  000024                  MOV     RKDB(R5),(R0)+  ;3'RD WORD
7092
7093
7094  041462  032765  100000  000010          BIT     #DLT,RKCS2(R5)
7095  041470  001407                          BEQ     75$
7096  041472  004737  050256                  JSR     PC,GSTAT
7097  041476  104173                          ERROR   173             ;DLT AFTER READ HEADER CMD
7098  041500  104401  060720                  TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
7099  041504  000137  047164                  JMP     $EOP            ;ABORT DRIVE
7100  041510                          75$:
7101
7102  041510  023737  001704  001370          CMP     RHTAB,FRCYL     ;CHECK WORD 0 (CYL#) ONLY
7103  041516  001401                          BEQ     73$             ;BR IF SAME
7104  041520  104311                          ERROR   311             ;READ CYL WORD HEADER ERROR
7105  041522                          73$:
7106
7107
7108  041522  006337  001372                  ASL     TOCYL
7109  041526  023727  001372  001000          CMP     TOCYL,#1000     ;ALL CYL DONE?
7110  041534  001402                          BEQ     6$
7111  041536  000137  037704                  JMP     1$
7112  041542                          6$:
7113  041542  004737  052770                  JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
7114  041546  000517                          BR      TST40           ;;GO TO NEXT TEST
7115                                                                  ;RETURN HERE IF SW 14 IS SET OR
7116                                                                  ;SW 8 WITH SWR <7:0> APPLY
7117  041550                          8$:
7118
7119  041550  004737  050320                  JSR     PC,SUBCLR
7120  041554  104024                          ERROR   24              ;CERR AFTER SCRL
7121
7122  041556  013765  001372  000020  76$:    MOV     TOCYL,RKDC(R5)  ;CYL#
7123
7124  041564  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD TO RECONDITION DRIVE.
7125  041572  013737  001434  004400          MOV     T10,TEMP1       ;SETUP TIMEOUT
7126  041600  004737  047632                  JSR     PC,FRDY         ;FIND RDY
7127  041604  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD.
7128
7129  041606  013737  001444  004400          MOV     T5000,TEMP1
7130  041614  004737  050144                  JSR     PC,FATT2        ;FIND ATTN
```

```
7131  041620  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
7132  041622  032737  100000  004336          BIT     #CERR,HCS1
7133  041630  001401                          BEQ     77$
7134  041632  104210                          ERROR   210             ;CERR AFTER SEEK CMD.
7135
7136  041634  004737  050320          77$:    JSR     PC,SUBCLR
7137  041640  104024                          ERROR   24              ;CERR AFTER SCLR
7138
7139
7140  041642  005337  001372                  DEC     TOCYL
7141  041646  023727  001372  177777          CMP     TOCYL,#-1       ;ALL CYL DONE?
7142  041654  001340                          BNE     76$
7143
7144  041656  004737  050320                  JSR     PC,SUBCLR
7145  041662  104024                          ERROR   24              ;CERR AFTER SCLR
7146
7147  041664  005037  001176                  CLR     $ESCAPE
7148  041670  005737  001430                  TST     LPFLG
7149  041674  001402                          BEQ     78$
7150  041676  000177  137206                  JMP     @$LPERR         ;SW 9 WAS SET.
7151  041702  000177  137200          78$:    JMP     @$LPADR         ;SW 14 OR 8 WAS SET
7152
7153  041706                          10$:
7154  041706  005227  001430                  INC     LPFLG
7155  041712  032777  001000  137220          BIT     #SW9,@SWR       ;LOOP ON ERROR?
7156  041720  001313                          BNE     8$              ;YES, RECONDITION DRIVE
7157  041722  000137  040102                  JMP     2$              ;RETURN TO MAINLINE
7158  041726                          12$:
7159  041726  005237  001430                  INC     LPFLG
7160  041732  032777  001000  137200          BIT     #SW9,@SWR       ;LOOP ON ERROR?
7161  041740  001303                          BNE     8$              ;YES, RECONDITION DRIVE
7162  041742  000137  040452                  JMP     3$              ;RETURN TO MAINLINE
7163  041746                          14$:
7164  041746  005237  001430                  INC     LPFLG
7165  041752  032777  001000  137160          BIT     #SW9,@SWR       ;LOOP ON ERROR?
7166  041760  001273                          BNE     8$              ;YES, RECONDITION DRIVE
7167  041762  000137  041000                  JMP     4$              ;RETURN TO MAINLINE
7168  041766                          16$:
7169  041766  005237  001430                  INC     LPFLG
7170  041772  032777  001000  137140          BIT     #SW9,@SWR       ;LOOP ON ERROR?
7171  042000  001263                          BNE     8$              ;YES, RECONDITION DRIVE
7172  042002  000137  041350                  JMP     5$              ;RETURN TO MAINLINE
7173
7174
7175
7176                                  ;;*****************************************************************
7177                                  ;*TEST 40       SEEK TO ALL CYLINDERS FROM 0 & READ HEADERS
7178                                  ;;*****************************************************************
7179  042006  000004                  TST40:  SCOPE
7180  042010  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
7181  042016  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
7182
7183  042022  005737  001362                  TST     MODTST          ;SEE IF MODULE TESTING
7184  042026  001402                          BEQ     DOSEEK          ;BR IF NO
7185  042030  000137  046376                  JMP     CYLINV          ;ELSE BYPASS TESTS 40 & 41
7186  042034                          DOSEEK:
```

```
7187
7188  042034  012737  000000  001370          MOV     #0,FRCYL        ;SETUP FROM CYL
7189  042042  012737  000001  001372          MOV     #1,TOCYL        ;SETUP TO CYL
7190
7191  042050                          1$:
7192  042050  104415                          SCOP1
7193  042052  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
7194
7195  042056  004737  050320                  JSR     PC,SUBCLR
7196  042062  104024                          ERROR   24              ;CERR AFTER SCLR
7197
7198  042064  012737  044052  001176          MOV     #10$,$ESCAPE
7199  042072  013737  001370  004404          MOV     FRCYL,TEMP3     ;SETUP
7200  042100  013737  001372  004406          MOV     TOCYL,TEMP4     ;CYL DIFF
7201  042106  163737  004404  004406          SUB     TEMP3,TEMP4     ;FOR
7202  042114  013737  004406  001400          MOV     TEMP4,CALDIF    ;ERROR PRINTOUT
7203
7204  042122  013765  001372  000020          MOV     TOCYL,RKDC(R5)  ;GO TO CYL #
7205
7206  042130  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD
7207  042136  013737  001434  004400          MOV     T10,TEMP1       ;SETUP  TIMEOUT
7208  042144  004737  047632                  JSR     PC,FRDY         ;FIND RDY
7209  042150  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD
7210  042152  012737  030140  004374          MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,SBMR2        ;LOAD SHOULD BE DATA
7211  042160  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
7212  042164  104203                          ERROR   203             ;MSG AO ERROR DURING SEEK CMD
7213  042166  005037  004376                  CLR     SBMR3           ;MR3 SHOULD BE 0
7214  042172  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
7215  042176  104204                          ERROR   204             ;MSG BO ERROR DURING SEEK CMD
7216
7217  042200  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
7218  042206  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
7219  042214  004737  050256                  JSR     PC,GSTAT
7220  042220  012737  003720  004374          MOV     #<D.FWD!D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
7221  042226  004737  052504                  JSR     PC,CKMR2
7222  042232  104205                          ERROR   205             ;MSG A1 ERROR DURING SEEK CMD
7223  042234  005037  004376                  CLR     SBMR3
7224  042240  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
7225  042244  104206                          ERROR   206             ;MSG B1 ERROR DURING SEEK CMD
7226
7227  042246  012737  044072  001176  2$:     MOV     #12$,$ESCAPE
7228  042254  013737  001444  004400          MOV     T5000,TEMP1     ;SETUP  TIMEOUT
7229
7230  042262  004737  050144                  JSR     PC,FATT2        ;FIND ATTN
7231  042266  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
7232  042270  032737  100000  004336          BIT     #CERR,HCS1
7233  042276  001401                          BEQ     64$
7234  042300  104210                          ERROR   210             ;CERR AFTER SEEK CMD
7235  042302                          64$:
7236
7237  042302  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
7238  042310  004737  050256                  JSR     PC,GSTAT        ;GET LATEST STATUS
7239  042314  012737  050340  004374          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2 ;SHOULD BE VALUE
7240  042322  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
7241  042326  104133                          ERROR   133             ;MSG AO ERROR AFTER SEEK CMD
7242  042330  005037  004376                  CLR     SBMR3
```

```
7243  042334  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
7244  042340  104134                          ERROR   134             ;MSG B0 ERROR AFTER SEEK CMD
7245
7246  042342  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
7247  042350  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
7248  042356  004737  050256                  JSR     PC,GSTAT
7249  042362  012737  001720  004374          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
7250  042370  004737  052504                  JSR     PC,CKMR2
7251  042374  104135                          ERROR   135             ;MSG A1 ERROR AFTER SEEK CMD
7252  042376  005037  004376                  CLR     SBMR3
7253  042402  004737  052566                  JSR     PC,CKMR3
7254  042406  104136                          ERROR   136             ;MSG B1 ERROR AFTER SEEK CMD
7255
7256
7257  042410  004737  051124                  JSR     PC,RDCYLD       ;READ CYL DIFF IN MSG A2
7258  042414  005737  001402                  TST     CYLDIF
7259  042420  001401                          BEQ     65$
7260  042422  104137                          ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
7261
7262  042424                          65$:
7263
7264  042424  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
7265  042432  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
7266  042440  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
7267  042446  013737  001434  004400          MOV     T10,TEMP1
7268  042454  004737  047632                  JSR     PC,FRDY         ;FIND RDY
7269  042460  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
7270  042462  004737  050016                  JSR     PC,TSTATN       ;TEST FOR ATTN
7271  042466  000401                          BR      66$
7272  042470  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
7273  042472                          66$:
7274
7275  042472  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
7276  042500  004737  050256                  JSR     PC,GSTAT        ;GET LATEST STATUS
7277  042504  012737  010340  004374          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2     ;SHOULD BE VALUE
7278  042512  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
7279  042516  104273                          ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
7280  042520  005037  004376                  CLR     SBMR3
7281  042524  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
7282  042530  104265                          ERROR   265             ;MSG B0 ERROR AFTER DRIVE CLEAR CMD
7283
7284  042532  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
7285  042540  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
7286  042546  004737  050256                  JSR     PC,GSTAT
7287  042552  012737  001720  004374          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
7288  042560  004737  052504                  JSR     PC,CKMR2
7289  042564  104274                          ERROR   274             ;MSG A1 ERROR AFTER DRIVE CLEAR CMD
7290  042566  005037  004376                  CLR     SBMR3
7291  042572  004737  052566                  JSR     PC,CKMR3
7292  042576  104266                          ERROR   266             ;MSG B1 ERROR AFTER DRIVE CLEAR CMD
7293
7294
7295  042600  004737  051210                  JSR     PC,RDCYLA       ;READ CYL ADDR IN MSG B2
7296  042604  023737  001404  001372          CMP     CYLADD,TOCYL
7297  042612  001401                          BEQ     3$
7298  042614  104207                          ERROR   207             ;CYL ADDR IN RKMR3 NOT=RKDC
```

```
7299
7300  042616                          3$:
7301  042616  104415                        SCOP1
7302  042620  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR
7303
7304  042624  004737  050320                JSR     PC,SUBCLR
7305  042630  104024                        ERROR   24              ;CERR AFTER SCLR
7306
7307  042632  005037  001176                CLR     $ESCAPE
7308  042636  013765  001372  000020        MOV     TOCYL,RKDC(R5)  ;CYL #
7309
7310
7311  042644  012700  001704                MOV     #RHTAB,R0
7312  042650  012765  000025  000000        MOV     #<RDHEAD>,RKCS1(R5)     ;READ HEADER CMD
7313  042656  013737  001446  004400        MOV     T50000,TEMP1    ;SETUP  TIMEOUT
7314  042664  004737  047632                JSR     PC,FRDY         ;FIND RDY
7315  042670  104171                        ERROR   171             ;NO RDY AFTER READ HEADER CMD
7316  042672  032737  100000  004336        BIT     #CERR,HCS1
7317  042700  001405                        BEQ     68$
7318  042702  104174                        ERROR   174             ;CERR AFTER READ HEADER CMD
7319  042704  104401  060720                TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
7320  042710  000137  047164                JMP     $EOP            ;ABORT DRIVE
7321
7322  042714  016520  000024        68$:    MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
7323  042720  016520  000024                MOV     RKDB(R5),(R0)+  ;2'ND WORD
7324  042724  016520  000024                MOV     RKDB(R5),(R0)+  ;3'RD WORD
7325
7326
7327  042730  032765  100000  000010        BIT     #DLT,RKCS2(R5)
7328  042736  001407                        BEQ     69$
7329  042740  004737  050256                JSR     PC,GSTAT
7330  042744  104173                        ERROR   173             ;DLT AFTER READ HEADER CMD
7331  042746  104401  060720                TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
7332  042752  000137  047164                JMP     $EOP            ;ABORT DRIVE
7333  042756                          69$:
7334
7335  042756  023737  001704  001372        CMP     RHTAB,TOCYL     ;CHECK WORD 0 (CYL#) ONLY
7336  042764  001401                        BEQ     67$             ;BR IF SAME
7337  042766  104310                        ERROR   310             ;READ CYL WORD HEADER ERROR
7338  042770                          67$:
7339
7340
7341  042770  104415                        SCOP1
7342  042772  012706  001100                MOV     #STACK,SP       ;RESTORE STK PTR
7343
7344  042776  004737  050320                JSR     PC,SUBCLR
7345  043002  104024                        ERROR   24              ;CERR AFTER SCLR
7346
7347  043004  012737  044112  001176        MOV     #14$,$ESCAPE
7348  043012  013765  001370  000020        MOV     FRCYL,RKDC(R5)  ;RETURN TO CYL #
7349  043020  013737  001370  001374        MOV     FRCYL,CCYL      ;CURRENT CYL FOR TRUERR ROUTINE
7350
7351  043026  012765  000017  000000        MOV     #SEEK,RKCS1(R5) ;SEEK CMD
7352  043034  013737  001434  004400        MOV     T10,TEMP1       ;SETUP  TIMEOUT
7353  043042  004737  047632                JSR     PC,FRDY         ;FIND RDY
7354  043046  104131                        ERROR   131             ;NO RDY AFTER SEEK CMD
```

```
7355  043050  012737  030140  004374          MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,SBMR2        ;LOAD SHOULD BE DATA
7356  043056  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
7357  043062  104203                          ERROR   203             ;MSG AC ERROR DURING SEEK CMD
7358  043064  005037  004376                  CLR     SBMR3           ;MR3 SHOULD BE 0
7359  043070  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
7360  043074  104204                          ERROR   204             ;MSG BC ERROR DURING SEEK CMD
7361
7362  043076  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
7363  043104  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
7364  043112  004737  050256                  JSR     PC,GSTAT
7365  043116  012737  005720  004374          MOV     #<D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
7366  043124  004737  052504                  JSR     PC,CKMR2
7367  043130  104205                          ERROR   205             ;MSG A1 ERROR DURING SEEK CMD
7368  043132  005037  004376                  CLR     SBMR3
7369  043136  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
7370  043142  104206                          ERROR   206             ;MSG B1 ERROR DURING SEEK CMD
7371
7372
7373  043144  012737  044132  001176  4$:     MOV     #16$,$ESCAPE
7374  043152  013737  001444  004400          MOV     T5000,TEMP1     ;SETUP  TIMEOUT
7375
7376  043160  004737  050144                  JSR     PC,FATT2        ;FIND ATTN
7377  043164  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
7378  043166  032737  100000  004336          BIT     #CERR,HCS1
7379  043174  001401                          BEQ     70$
7380  043176  104210                          ERROR   210             ;CERR AFTER SEEK CMD
7381  043200                          70$:
7382
7383  043200  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
7384  043206  004737  050256                  JSR     PC,GSTAT        ;GET LATEST STATUS
7385  043212  012737  050340  004374          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2 ;SHOULD BE VALUE
7386  043220  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
7387  043224  104133                          ERROR   133             ;MSG AD ERROR AFTER SEEK CMD
7388  043226  005037  004376                  CLR     SBMR3
7389  043232  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
7390  043236  104134                          ERROR   134             ;MSG BD ERROR AFTER SEEK CMD
7391
7392  043240  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
7393  043246  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
7394  043254  004737  050256                  JSR     PC,GSTAT
7395  043260  012737  001720  004374          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
7396  043266  004737  052504                  JSR     PC,CKMR2
7397  043272  104135                          ERROR   135             ;MSG A1 ERROR AFTER SEEK CMD
7398  043274  005037  004376                  CLR     SBMR3
7399  043300  004737  052566                  JSR     PC,CKMR3
7400  043304  104136                          ERROR   136             ;MSG B1 ERROR AFTER SEEK CMD
7401
7402
7403  043306  004737  051124                  JSR     PC,RDCYLD       ;READ CYL DIFF IN MSG A2
7404  043312  005737  001402                  TST     CYLDIF
7405  043316  001401                          BEQ     71$
7406  043320  104137                          ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
7407
7408  043322                          71$:
7409
7410  043322  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
```

```
7411  043330  012765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
7412  043336  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
7413  043344  013737  001434  004400          MOV     T10,TEMP1
7414  043352  004737  047632                  JSR     PC,FRDY         ;FIND RDY
7415  043356  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
7416  043360  004737  050016                  JSR     PC,TSTATN       ;TEST FOR ATTN
7417  043364  000401                          BR      72$
7418  043366  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
7419  043370                          72$:
7420
7421  043370  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
7422  043376  004737  050256                  JSR     PC,GSTAT        ;GET LATEST STATUS
7423  043402  012737  010340  004374          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2     ;SHOULD BE VALUE
7424  043410  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
7425  043414  104273                          ERROR   273             ;MSG AO ERROR AFTER DRIVE CLEAR CMD
7426  043416  005037  004376                  CLR     SBMR3
7427  043422  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
7428  043426  104265                          ERROR   265             ;MSG BO ERROR AFTER DRIVE CLEAR CMD
7429
7430  043430  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
7431  043436  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
7432  043444  004737  050256                  JSR     PC,GSTAT
7433  043450  012737  001720  004374          MOV     #<0.SPOK!0.CART!0.DOOR!0.BRHM!0.TFOK>,SBMR2
7434  043456  004737  052504                  JSR     PC,CKMR2
7435  043462  104274                          ERROR   274             ;MSG A1 ERROR AFTER DRIVE CLEAR CMD
7436  043464  005037  004376                  CLR     SBMR3
7437  043470  004737  052566                  JSR     PC,CKMR3
7438  043474  104266                          ERROR   266             ;MSG B1 ERROR AFTER DRIVE CLEAR CMD
7439
7440
7441  043476  004737  051210                  JSR     PC,RDCYLA       ;READ CYL ADDR IN MSG B2
7442  043502  023737  001404  001370          CMP     CYLADD,FRCYL
7443  043510  001401                          BEQ     5$
7444  043512  104243                          ERROR   243             ;CYL ADDR IN RKMR3 NOT=RKDC
7445
7446  043514                          5$:
7447  043514  104415                          SCOP1
7448  043516  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
7449
7450  043522  004737  050320                  JSR     PC,SUBCLR
7451  043526  104024                          ERROR   24              ;CERR AFTER SCLR
7452
7453  043530  005037  001176                  CLR     $ESCAPE
7454  043534  013765  001370  000020          MOV     FRCYL,RKDC(R5)  ;CYL #
7455
7456
7457  043542  012700  001704                  MOV     #RHTAB,R0
7458  043546  012765  000025  000000          MOV     #<RDHEAD>,RKCS1(R5)     ;READ HEADER CMD
7459  043554  013737  001446  004400          MOV     T50000,TEMP1    ;SETUP TIMEOUT
7460  043562  004737  047632                  JSR     PC,FRDY         ;FIND RDY
7461  043566  104171                          ERROR   171             ;NO RDY AFTER READ HEADER CMD
7462  043570  032737  100000  004336          BIT     #CERR,HCS1
7463  043576  001405                          BEQ     74$
7464  043600  104174                          ERROR   174             ;CERR AFTER READ HEADER CMD
7465  043602  104401  060720                  TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
7466  043606  000137  047164                  JMP     $EOP            ;ABORT DRIVE
```

```
7467
7468  043612  016520  000024          74$:  MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
7469  043616  016520  000024                MOV     RKDB(R5),(R0)+  ;2'ND WORD
7470  043622  016520  000024                MOV     RKDB(R5),(R0)+  ;3'RD WORD
7471
7472
7473  043626  032765  100000  000010        BIT     #DLT,RKCS2(R5)
7474  043634  001407                        BEQ     75$
7475  043636  004737  050256                JSR     PC,GSTAT
7476  043642  104173                        ERROR   173             ;DLT AFTER READ HEADER CMD
7477  043644  104401  060720                TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
7478  043650  000137  047164                JMP     $EOP            ;ABORT DRIVE
7479  043654                          75$:
7480
7481  043654  023737  001704  001370        CMP     RHTAB,FRCYL     ;CHECK WORD 0 (CYL#) ONLY
7482  043662  001401                        BEQ     73$             ;BR IF SAME
7483  043664  104311                        ERROR   311             ;READ CYL WORD HEADER ERROR
7484  043666                          73$:
7485
7486
7487  043666  005237  001372                INC     TOCYL
7488  043672  023727  001372  000633        CMP     TOCYL,#411.     ;ALL CYL DONE?
7489  043700  001402                        BEQ     6$
7490  043702  000137  042050                JMP     1$
7491  043706                          6$:
7492  043706  004737  052770                JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
7493  043712  000517                        BR      TST41           ;;GO TO NEXT TEST
7494                                                                ;RETURN HERE IF SW 14 IS SET OR
7495                                                                ;SW 8 WITH SWR <7:0> APPLY
7496  043714                          8$:
7497
7498  043714  004737  050320                JSR     PC,SUBCLR
7499  043720  104024                        ERROR   24              ;CERR AFTER SCRL
7500
7501  043722  013765  001372  000020  76$:  MOV     TOCYL,RKDC(R5)  ;CYL#
7502
7503  043730  012765  000017  000000        MOV     #SEEK,RKCS1(R5) ;SEEK CMD TO RECONDITION DRIVE.
7504  043736  013737  001434  004400        MOV     T10,TEMP1       ;SETUP TIMEOUT
7505  043744  004737  047632                JSR     PC,FRDY         ;FIND RDY
7506  043750  104131                        ERROR   131             ;NO RDY AFTER SEEK CMD.
7507
7508  043752  013737  001444  004400        MOV     T5000,TEMP1
7509  043760  004737  050144                JSR     PC,FATT2        ;FIND ATTN
7510  043764  104132                        ERROR   132             ;NO ATTN AFTER SEEK CMD
7511  043766  032737  100000  004336        BIT     #CERR,HCS1
7512  043774  001401                        BEQ     77$
7513  043776  104210                        ERROR   210             ;CERR AFTER SEEK CMD.
7514
7515  044000  004737  050320          77$:  JSR     PC,SUBCLR
7516  044004  104024                        ERROR   24              ;CERR AFTER SCLR
7517
7518
7519  044006  005337  001372                DEC     TOCYL
7520  044012  023727  001372  177777        CMP     TOCYL,#-1       ;ALL CYL DONE?
7521  044020  001340                        BNE     76$
7522
```

```
7523  044022  004737  050320              JSR     PC,SUBCLR
7524  044026  104024                      ERROR   24              ;CERR AFTER SCLR
7525
7526  044030  005037  001176              CLR     $ESCAPE
7527  044034  005737  001430              TST     LPFLG
7528  044040  001402                      BEQ     78$
7529  044042  000177  135042              JMP     @$LPERR         ;SW 9 WAS SET.
7530  044046  000177  135034      78$:    JMP     @$LPADR         ;SW 14 OR 8 WAS SET
7531
7532  044052                      10$:
7533  044052  005237  001430              INC     LPFLG
7534  044056  032777  001000  135054      BIT     #SW9,@SWR       ;LOOP ON ERROR?
7535  044064  001313                      BNE     8$              ;YES, RECONDITION DRIVE
7536  044066  000137  042246              JMP     2$              ;RETURN TO MAINLINE
7537  044072                      12$:
7538  044072  005237  001430              INC     LPFLG
7539  044076  032777  001000  135034      BIT     #SW9,@SWR       ;LOOP ON ERROR?
7540  044104  001303                      BNE     8$              ;YES, RECONDITION DRIVE
7541  044106  000137  042616              JMP     3$              ;RETURN TO MAINLINE
7542  044112                      14$:
7543  044112  005237  001430              INC     LPFLG
7544  044116  032777  001000  135014      BIT     #SW9,@SWR       ;LOOP ON ERROR?
7545  044124  001273                      BNE     8$              ;YES, RECONDITION DRIVE
7546  044126  000137  043144              JMP     4$              ;RETURN TO MAINLINE
7547  044132                      16$:
7548  044132  005237  001430              INC     LPFLG
7549  044136  032777  001000  134774      BIT     #SW9,@SWR       ;LOOP ON ERROR?
7550  044144  001263                      BNE     8$              ;YES, RECONDITION DRIVE
7551  044146  000137  043514              JMP     5$              ;RETURN TO MAINLINE
7552
7553                      ;;*****************************************************************
7554                      ;*TEST 41       SEEK TO ALL CYLINDERS FROM CYL 410 & READ HEADERS
7555                      ;;*****************************************************************
7556  044152  000004      TST41:  SCOPE
7557  044154  012737  000001  001174      MOV     #1,$TIMES       ;;DO 1 ITERATION
7558  044162  012706  001100              MOV     #STACK,SP       ;RESTORE STK PTR
7559
7560
7561  044166  004737  050320              JSR     PC,SUBCLR
7562  044172  104024                      ERROR   24              ;CERR AFTER SCLR
7563
7564  044174  012765  000632  000020      MOV     #410.,RKDC(R5)  ;QUICK SEEK TO CYL 410
7565
7566  044202  012765  000017  000000      MOV     #SEEK,RKCS1(R5) ;SEEK CMD TO RECONDITION DRIVE.
7567  044210  013737  001434  004400      MOV     T10,TEMP1       ;SETUP TIMEOUT
7568  044216  004737  047632              JSR     PC,FRDY         ;FIND RDY
7569  044222  104131                      ERROR   131             ;NO RDY AFTER SEEK CMD.
7570
7571  044224  013737  001444  004400      MOV     T5000,TEMP1
7572  044232  004737  050144              JSR     PC,FATT2        ;FIND ATTN
7573  044236  104132                      ERROR   132             ;NO ATTN AFTER SEEK CMD
7574  044240  032737  100000  004336      BIT     #CERR,HCS1
7575  044246  001401                      BEQ     64$
7576  044250  104210                      ERROR   210             ;CERR AFTER SEEK CMD.
7577
7578  044252  004737  050320      64$:    JSR     PC,SUBCLR
```

```
7579  044256  104024                          ERROR   24              ;CERR AFTER SCLR
7580
7581
7582  044260  012737  000632  001370          MOV     #410.,FRCYL     ;SETUP FROM CYL
7583  044266  012737  000631  001372          MOV     #409.,TOCYL     ;SETUP TO CYL
7584
7585  044274                          1$:
7586  044274  104415                          SCOP1
7587  044276  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
7588
7589  044302  004737  050320                  JSR     PC,SUBCLR
7590  044306  104024                          ERROR   24              ;CERR AFTER SCLR
7591
7592  044310  012737  046276  001176          MOV     #10$,$ESCAPE
7593  044316  013737  001370  004404          MOV     FRCYL,TEMP3     ;SETUP
7594  044324  013737  001372  004406          MOV     TOCYL,TEMP4     ;CYL DIFF
7595  044332  163737  004406  004404          SUB     TEMP4,TEMP3     ;FOR
7596  044340  013737  004404  001400          MOV     TEMP3,CALDIF    ;ERROR PRINTOUT
7597
7598  044346  013765  001372  000020          MOV     TOCYL,RKDC(R5)  ;GO TO CYL #
7599
7600  044354  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD
7601  044362  013737  001434  004400          MOV     T10,TEMP1       ;SETUP  TIMEOUT
7602  044370  004737  047632                  JSR     PC,FRDY         ;FIND RDY
7603  044374  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD
7604  044376  012737  030140  004374          MOV     #<D.PIP!D.SPIN!D.VV!D.DRA>,SBMR2        ;LOAD SHOULD BE DATA
7605  044404  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
7606  044410  104203                          ERROR   203             ;MSG AO ERROR DURING SEEK CMD
7607  044412  005037  004376                  CLR     SBMR3           ;MR3 SHOULD BE 0
7608  044416  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
7609  044422  104204                          ERROR   204             ;MSG BO ERROR DURING SEEK CMD
7610
7611  044424  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
7612  044432  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
7613  044440  004737  050256                  JSR     PC,GSTAT
7614  044444  012737  005720  004374          MOV     #<D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
7615  044452  004737  052504                  JSR     PC,CKMR2
7616  044456  104205                          ERROR   205             ;MSG A1 ERROR DURING SEEK CMD
7617  044460  005037  004376                  CLR     SBMR3
7618  044464  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
7619  044470  104206                          ERROR   206             ;MSG B1 ERROR DURING SEEK CMD
7620
7621  044472  012737  046316  001176  2$:     MOV     #12$,$ESCAPE
7622  044500  013737  001444  004400          MOV     T5000,TEMP1     ;SETUP  TIMEOUT
7623
7624  044506  004737  050144                  JSR     PC,FATT2        ;FIND ATTN
7625  044512  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
7626  044514  032737  100000  004336          BIT     #CERR,HCS1
7627  044522  001401                          BEQ     65$
7628  044524  104210                          ERROR   210             ;CERR AFTER SEEK CMD
7629  044526                          65$:
7630
7631  044526  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
7632  044534  004737  050256                  JSR     PC,GSTAT        ;GET LATEST STATUS
7633  044540  012737  050340  004374          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2 ;SHOULD BE VALUE
7634  044546  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
```

```
7635  044552  104133                          ERROR   133             ;MSG A0 ERROR AFTER SEEK CMD
7636  044554  005037  004376                  CLR     SBMR3
7637  044560  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
7638  044564  104134                          ERROR   134             ;MSG B0 ERROR AFTER SEEK CMD
7639
7640  044566  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
7641  044574  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
7642  044602  004737  050256                  JSR     PC,GSTAT
7643  044606  012737  001720  004374          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
7644  044614  004737  052504                  JSR     PC,CKMR2
7645  044620  104135                          ERROR   135             ;MSG A1 ERROR AFTER SEEK CMD
7646  044622  005037  004376                  CLR     SBMR3
7647  044626  004737  052566                  JSR     PC,CKMR3
7648  044632  104136                          ERROR   136             ;MSG B1 ERROR AFTER SEEK CMD
7649
7650
7651  044634  004737  051124                  JSR     PC,RDCYLD       ;READ CYL DIFF IN MSG A2
7652  044640  005737  001402                  TST     CYLDIF
7653  044644  001401                          BEQ     66$
7654  044646  104137                          ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
7655
7656  044650                          66$:
7657
7658  044650  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
7659  044656  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
7660  044664  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
7661  044672  013737  001434  004400          MOV     T10,TEMP1
7662  044700  004737  047632                  JSR     PC,FRDY         ;FIND RDY
7663  044704  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
7664  044706  004737  050016                  JSR     PC,TSTATN       ;TEST FOR ATTN
7665  044712  000401                          BR      67$
7666  044714  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
7667  044716                          67$:
7668
7669  044716  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
7670  044724  004737  050256                  JSR     PC,GSTAT        ;GET LATEST STATUS
7671  044730  012737  010340  004374          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2      ;SHOULD BE VALUE
7672  044736  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
7673  044742  104273                          ERROR   273             ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
7674  044744  005037  004376                  CLR     SBMR3
7675  044750  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
7676  044754  104265                          ERROR   265             ;MSG B0 ERROR AFTER DRIVE CLEAR CMD
7677
7678  044756  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
7679  044764  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
7680  044772  004737  050256                  JSR     PC,GSTAT
7681  044776  012737  001720  004374          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
7682  045004  004737  052504                  JSR     PC,CKMR2
7683  045010  104274                          ERROR   274             ;MSG A1 ERROR AFTER DRIVE CLEAR CMD
7684  045012  005037  004376                  CLR     SBMR3
7685  045016  004737  052566                  JSR     PC,CKMR3
7686  045022  104266                          ERROR   266             ;MSG B1 ERROR AFTER DRIVE CLEAR CMD
7687
7688
7689  045024  004737  051210                  JSR     PC,RDCYLA       ;READ CYL ADDR IN MSG B2
7690  045030  023737  001404  001372          CMP     CYLADD,TOCYL
```

```
7691  045036  001401                          BEQ     3$
7692  045040  104207                          ERROR   207             ;CYL ADDR IN RKMR3 NOT=RKDC
7693
7694  045042                          3$:
7695  045042  104415                          SCOP1
7696  045044  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
7697
7698  045050  004737  050320                  JSR     PC,SUBCLR
7699  045054  104024                          ERROR   24              ;CERR AFTER SCLR
7700
7701  045056  005037  001176                  CLR     $ESCAPE
7702  045062  013765  001372  000020          MOV     TOCYL,RKDC(R5)  ;CYL #
7703
7704
7705  045070  012700  001704                  MOV     #RHTAB,R0
7706  045074  012765  000025  000000          MOV     #<RDHEAD>,RKCS1(R5)     ;READ HEADER CMD
7707  045102  013737  001446  004400          MOV     T50000,TEMP1    ;SETUP  TIMEOUT
7708  045110  004737  047632                  JSR     PC,FRDY         ;FIND RDY
7709  045114  104171                          ERROR   171             ;NO RDY AFTER READ HEADER CMD
7710  045116  032737  100000  004336          BIT     #CERR,HCS1
7711  045124  001405                          BEQ     69$
7712  045126  104174                          ERROR   174             ;CERR AFTER READ HEADER CMD
7713  045130  104401  060720                  TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
7714  045134  000137  047164                  JMP     $EOP            ;ABORT DRIVE
7715
7716  045140  016520  000024          69$:    MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
7717  045144  016520  000024                  MOV     RKDB(R5),(R0)+  ;2'ND WORD
7718  045150  016520  000024                  MOV     RKDB(R5),(R0)+  ;3'RD WORD
7719
7720
7721  045154  032765  100000  000010          BIT     #DLT,RKCS2(R5)
7722  045162  001407                          BEQ     70$
7723  045164  004737  050256                  JSR     PC,GSTAT
7724  045170  104173                          ERROR   173             ;DLT AFTER READ HEADER CMD
7725  045172  104401  060720                  TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
7726  045176  000137  047164                  JMP     $EOP            ;ABORT DRIVE
7727  045202                          70$:
7728
7729  045202  023737  001704  001372          CMP     RHTAB,TOCYL     ;CHECK WORD 0 (CYL#) ONLY
7730  045210  001401                          BEQ     68$             ;BR IF SAME
7731  045212  104310                          ERROR   310             ;READ CYL WORD HEADER ERROR
7732  045214                          68$:
7733
7734
7735  045214  104415                          SCOP1
7736  045216  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
7737
7738  045222  004737  050320                  JSR     PC,SUBCLR
7739  045226  104024                          ERROR   24              ;CERR AFTER SCLR
7740
7741  045230  012737  046336  001176          MOV     #14$,$ESCAPE
7742  045236  013765  001370  000020          MOV     FRCYL,RKDC(R5)  ;RETURN TO CYL #
7743  045244  013737  001370  001374          MOV     FRCYL,CCYL      ;CURRENT CYL FOR TRUERR ROUTINE
7744
7745  045252  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD
7746  045260  013737  001434  004400          MOV     T10,TEMP1       ;SETUP  TIMEOUT
```

```
7747  045266  004737  047632                  JSR     PC,FRDY         ;FIND RDY
7748  045272  104131                          ERROR   131             ;NO RDY AFTER SEEK CMD
7749  045274  012737  030140  004374          MOV     #<D.P!D.SPIN!D.VV!D.DRA>,SBMR2      ;LOAD SHOULD BE DATA
7750  045302  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
7751  045306  104203                          ERROR   203             ;MSG AO ERROR DURING SEEK CMD
7752  045310  005037  004376                  CLR     SBMR3           ;MR3 SHOULD BE 0
7753  045314  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
7754  045320  104204                          ERROR   204             ;MSC BO ERROR DURING SEEK CMD
7755
7756  045322  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
7757  045330  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
7758  045336  004737  050256                  JSR     PC,GSTAT
7759  045342  012737  003720  004374          MOV     #<D.FWD!D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
7760  045350  004737  052504                  JSR     PC,CKMR2
7761  045354  104205                          ERROR   205             ;MSG A1 ERROR DURING SEEK CMD
7762  045356  005037  004376                  CLR     SBMR3
7763  045362  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
7764  045366  104206                          ERROR   206             ;MSG B1 ERROR DURING SEEK CMD
7765
7766
7767  045370  012737  046356  001176  4$:     MOV     #16$,$ESCAPE
7768  045376  013737  001444  004400          MOV     T5000,TEMP1     ;SETUP  TIMEOUT
7769
7770  045404  004737  050144                  JSR     PC,FATT2        ;FIND ATTN
7771  045410  104132                          ERROR   132             ;NO ATTN AFTER SEEK CMD
7772  045412  032737  100000  004336          BIT     #CERR,HCS1
7773  045420  001401                          BEQ     71$
7774  045422  104210                          ERROR   210             ;CERR AFTER SEEK CMD
7775  045424                          71$:
7776
7777  045424  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
7778  045432  004737  050256                  JSR     PC,GSTAT        ;GET LATEST STATUS
7779  045436  012737  050340  004374          MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2 ;SHOULD BE VALUE
7780  045444  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
7781  045450  104133                          ERROR   133             ;MSG AO ERROR AFTER SEEK CMD
7782  045452  005037  004376                  CLR     SBMR3
7783  045456  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
7784  045462  104134                          ERROR   134             ;MSG BO ERROR AFTER SEEK CMD
7785
7786  045464  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
7787  045472  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
7788  045500  004737  050256                  JSR     PC,GSTAT
7789  045504  012737  001720  004374          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
7790  045512  004737  052504                  JSR     PC,CKMR2
7791  045516  104135                          ERROR   135             ;MSG A1 ERROR AFTER SEEK CMD
7792  045520  005037  004376                  CLR     SBMR3
7793  045524  004737  052566                  JSR     PC,CKMR3
7794  045530  104136                          ERROR   136             ;MSG B1 ERROR AFTER SEEK CMD
7795
7796
7797  045532  004737  051124                  JSR     PC,RDCYLD       ;READ CYL DIFF IN MSG A2
7798  045536  005737  001402                  TST     CYLDIF
7799  045542  001401                          BEQ     72$
7800  045544  104137                          ERROR   137             ;CYL DIFF NOT CLEARED AFTER SEEK CMD
7801
7802  045546                          72$:
```

```
7803
7804  045546  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
7805  045554  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
7806  045562  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
7807  045570  013737  001434  004400          MOV     T10,TEMP1
7808  045576  004737  047632                  JSR     PC,FRDY         ;FIND RDY
7809  045602  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
7810  045604  004737  050016                  JSR     PC,TSTATN       ;TEST FOR ATTN
7811  045610  000401                          BR      73$
7812  045612  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
7813  045614                          73$:
7814
7815  045614  012765  100000  000000          MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
7816  045622  004737  050256                  JSR     PC,GSTAT        ;GET LATEST STATUS
7817  045626  012737  010340  004374          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2     ;SHOULD BE VALUE
7818  045634  004737  052504                  JSR     PC,CKMR2        ;CHECK MR2
7819  045640  104273                          ERROR   273             ;MSG AO ERRCR AFTER DRIVE CLEAR CMD
7820  045642  005037  004376                  CLR     SBMR3
7821  045646  004737  052566                  JSR     PC,CKMR3        ;CHECK MR3
7822  045652  104265                          ERROR   265             ;MSG BO ERROR AFTER DRIVE CLEAR CMD
7823
7824  045654  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
7825  045662  012765  000001  000026          MOV     #1,RKMR1(R5)    ;SELECT WORD 1
7826  045670  004737  050256                  JSR     PC,GSTAT
7827  045674  012737  001720  004374          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
7828  045702  004737  052504                  JSR     PC,CKMR2
7829  045706  104274                          ERROR   274             ;MSG A1 ERROR AFTER DRIVE CLEAR CMD
7830  045710  005037  004376                  CLR     SBMR3
7831  045714  004737  052566                  JSR     PC,CKMR3
7832  045720  104266                          ERROR   266             ;MSG B1 ERROR AFTER DRIVE CLEAR CMD
7833
7834
7835  045722  004737  051210                  JSR     PC,RDCYLA       ;READ CYL ADDR IN MSG B2
7836  045726  023737  001404  001370          CMP     CYLADD,FRCYL
7837  045734  001401                          BEQ     5$
7838  045736  104243                          ERROR   243             ;CYL ADDR IN RKMR3 NOT=RKDC
7839
7840  045740                          5$:
7841  045740  104415                          SCOP1
7842  045742  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
7843
7844  045746  004737  050320                  JSR     PC,SUBCLR
7845  045752  104024                          ERROR   24              ;CERR AFTER SCLR
7846
7847  045754  005037  001176                  CLR     $ESCAPE
7848  045760  013765  001370  000020          MOV     FRCYL,RKDC(R5)  ;CYL #
7849
7850
7851  045766  012700  001704                  MOV     #RHTAB,R0
7852  045772  012765  000025  000000          MOV     #<RDHEAD>,RKCS1(R5)     ;READ HEADER CMD
7853  046000  013737  001446  004400          MOV     T50000,TEMP1    ;SETUP  TIMEOUT
7854  046006  004737  047632                  JSR     PC,FRDY         ;FIND RDY
7855  046012  104171                          ERROR   171             ;NO RDY AFTER READ HEADER CMD
7856  046014  032737  100000  004336          BIT     #CERR,HCS1
7857  046022  001405                          BEQ     75$
7858  046024  104174                          ERROR   174             ;CERR AFTER READ HEADER CMD
```

```
 7859  046026  104401  060720                TYPE    ,MSG18          ;ABORT BALANCE OF TESTS
 7860  046032  000137  047164                JMP     $EOP            ;ABORT DRIVE
 7861
 7862  046036  016520  000024        75$:    MOV     RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
 7863  046042  016520  000024                MOV     RKDB(R5),(R0)+  ;2'ND WORD
 7864  046046  016520  000024                MOV     RKDB(R5),(R0)+  ;3'RD WORD
 7865
 7866
 7867  046052  032765  100000  000010        BIT     #DLT,RKCS2(R5)
 7868  046060  001407                        BEQ     76$
 7869  046062  004737  050256                JSR     PC,GSTAT
 7870  046066  104173                        ERROR   173             ;DLT AFTER READ HEADER CMD
 7871  046070  104401  060720                TYPE    ,MSG18          ;ABORTING BALANCE OF TESTS
 7872  046074  000137  047164                JMP     $EOP            ;ABORT DRIVE
 7873  046100                        76$:
 7874
 7875  046100  023737  001704  001370        CMP     RHTAB,FRCYL     ;CHECK WORD 0 (CYL#) ONLY
 7876  046106  001401                        BEQ     74$             ;BR IF SAME
 7877  046110  104311                        ERROR   311             ;READ CYL WORD HEADER ERROR
 7878  046112                        74$:
 7879
 7880
 7881  046112  005337  001372                DEC     TOCYL
 7882  046116  023727  001372  177777        CMP     TOCYL,#-1       ;ALL CYL DONE?
 7883  046124  001402                        BEQ     6$
 7884  046126  000137  044274                JMP     1$
 7885  046132                        6$:
 7886  046132  004737  052770                JSR     PC,SWTST        ;SEE IF SW 14 OR 8 IS SET
 7887  046136  000517                        BR      TST42           ;;GO TO NEXT TEST
 7888                                                                ;RETURN HERE IF SW 14 IS SET OR
 7889                                                                ;SW 8 WITH SWR <7:0> APPLY
 7890  046140                        8$:
 7891
 7892  046140  004737  050320                JSR     PC,SUBCLR
 7893  046144  104024                        ERROR   24              ;CERR AFTER SCRL
 7894
 7895  046146  013765  001372  000020  77$:   MOV     TOCYL,RKDC(R5)  ;CYL#
 7896
 7897  046154  012765  000017  000000        MOV     #SEEK,RKCS1(R5) ;SEEK CMD TO RECONDITION DRIVE.
 7898  046162  013737  001434  004400        MOV     T10,TEMP1       ;SETUP TIMEOUT
 7899  046170  004737  047632                JSR     PC,FRDY         ;FIND RDY
 7900  046174  104131                        ERROR   131             ;NO RDY AFTER SEEK CMD.
 7901
 7902  046176  013737  001444  004400        MOV     T5000,TEMP1
 7903  046204  004737  050144                JSR     PC,FATT2        ;FIND ATTN
 7904  046210  104132                        ERROR   132             ;NO ATTN AFTER SEEK CMD
 7905  046212  032737  100000  004336        BIT     #CERR,HCS1
 7906  046220  001401                        BEQ     78$
 7907  046222  104210                        ERROR   210             ;CERR AFTER SEEK CMD.
 7908
 7909  046224  004737  050320        78$:    JSR     PC,SUBCLR
 7910  046230  104024                        ERROR   24              ;CERR AFTER SCLR
 7911
 7912
 7913  046232  005237  001372                INC     TOCYL
 7914  046236  023727  001372  000633        CMP     TOCYL,#411.     ;ALL CYL DONE?
```

```
7915  046244  001340                          BNE     77$
7916
7917  046246  004737  050320                  JSR     PC,SUBCLR
7918  046252  104024                          ERROR   24              ;CERR AFTER SCLR
7919
7920  046254  005037  001176                  CLR     $ESCAPE
7921  046260  005737  001430                  TST     LPFLG
7922  046264  001402                          BEQ     79$
7923  046266  000177  132616                  JMP     @$LPERR         ;SW 9 WAS SET.
7924  046272  000177  132610          79$:    JMP     @$LPADR         ;SW 14 OR 8 WAS SET
7925
7926  046276                          10$:
7927  046276  005237  001430                  INC     LPFLG
7928  046302  032777  001000  132630          BIT     #SW9,@SWR       ;LOOP ON ERROR?
7929  046310  001313                          BNE     8$              ;YES, RECONDITION DRIVE
7930  046312  000137  044472                  JMP     2$              ;RETURN TO MAINLINE
7931  046316                          12$:
7932  046316  005237  001430                  INC     LPFLG
7933  046322  032777  001000  132610          BIT     #SW9,@SWR       ;LOOP ON ERROR?
7934  046330  001303                          BNE     8$              ;YES, RECONDITION DRIVE
7935  046332  000137  045042                  JMP     3$              ;RETURN TO MAINLINE
7936  046336                          14$:
7937  046336  005237  001430                  INC     LPFLG
7938  046342  032777  001000  132570          BIT     #SW9,@SWR       ;LOOP ON ERROR?
7939  046350  001273                          BNE     8$              ;YES, RECONDITION DRIVE
7940  046352  000137  045370                  JMP     4$              ;RETURN TO MAINLINE
7941  046356                          16$:
7942  046356  005237  001430                  INC     LPFLG
7943  046362  032777  001000  132550          BIT     #SW9,@SWR       ;LOOP ON ERROR?
7944  046370  001263                          BNE     8$              ;YES, RECONDITION DRIVE
7945  046372  000137  045740                  JMP     5$              ;RETURN TO MAINLINE
7946
7947  046376                          CYLINV:
7948
7949
7950                                  ;;*****************************************************************
7951                                  ;*TEST 42       SEEK TO ALL KEY INVALID CYLINDERS
7952                                  ;*
7953                                  ;*      THIS TEST VERIFIES THAT 'INV ADDR' & 'SEEK INCOMPLETE' IS
7954                                  ;*      PRODUCED & THAT HEADS DO NOT MOVE OR UNLOAD IF AN ILLEGAL
7955                                  ;*      CYLINDER IS SPECIFIED IN A SEEK.
7956                                  ;*
7957                                  ;*      INVALID CYLINDERS ARE 411 THRU 511 (10)  IE. 633 THRU 777 (8)
7958                                  ;*
7959                                  ;*      THIS TEST CHECKS KEY INVALID CYLINDERS 411,412,416,448 & 480
7960                                  ;*      FOR A FULL LOGIC TEST
7961                                  ;*
7962                                  ;;*****************************************************************
7963  046376  000004                  TST42:  SCOPE
7964  046400  012737  000001  001174          MOV     #1,$TIMES       ;;DO 1 ITERATION
7965  046406  012706  001100                  MOV     #STACK,SP       ;RESTORE STK PTR
7966
7967  046412  004737  050320                  JSR     PC,SUBCLR
7968  046416  104024                          ERROR   24              ;CERR AFTER SCLR
7969
7970  046420  012765  000017  000000          MOV     #SEEK,RKCS1(R5) ;SEEK CMD TO RECONDITION DRIVE.
```

```
7971  046426 013737 001434 004400        MOV     T10,TEMP1       ;SETUP TIMEOUT
7972  046434 004737 047632               JSR     PC,FRDY         ;FIND RDY
7973  046440 104131                      ERROR   131             ;NO RDY AFTER SEEK CMD.
7974
7975  046442 013737 001444 004400        MOV     T5000,TEMP1
7976  046450 004737 050144               JSR     PC,FATT2        ;FIND ATTN
7977  046454 104132                      ERROR   132             ;NO ATTN AFTER SEEK CMD
7978  046456 032737 100000 004336        BIT     #CERR,HCS1
7979  046464 001401                      BEQ     64$
7980  046466 104210                      ERROR   210             ;CERR AFTER SEEK CMD.
7981
7982  046470 004737 050320        64$:   JSR     PC,SUBCLR
7983  046474 104024                      ERROR   24              ;CERR AFTER SCLR
7984
7985  046476 005000                      CLR     R0
7986  046500 005037 001370               CLR     FRCYL           ;FROM CYL 0
7987
7988  046504                      1$:
7989  046504 104415                      SCOP1
7990  046506 012706 001100               MOV     #STACK,SP       ;RESTORE STK PTR
7991
7992  046512 004737 050320               JSR     PC,SUBCLR
7993  046516 104024                      ERROR   24              ;CERR AFTER SCLR
7994
7995  046520 016037 004314 001372        MOV     INVCYL(R0),TOCYL        ;GET INVALID CYL ADDR
7996  046526 013737 001372 001400        MOV     TOCYL,CALDIF
7997  046534 013765 001372 000020        MOV     TOCYL,RKDC(R5)
7998  046542 012765 000017 000000        MOV     #SEEK,RKCS1(R5) ;SEEK CMD
7999  046550 012737 000005 004400        MOV     #5,TEMP1        ;SETUP 100US TIMEOUT
8000  046556 004737 047632               JSR     PC,FRDY         ;FIND RDY
8001  046562 104131                      ERROR   131             ;NO RDY AFTER SEEK CMD
8002  046564 004737 050016               JSR     PC,TSTATN
8003  046570 104245                      ERROR   245             ;NO ATTN AFTER SEEK TO INV CYL
8004
8005  046572 032737 000040 004366        BIT     #D.IDAE,HMR3
8006  046600 001001                      BNE     2$
8007  046602 104246                      ERROR   246             ;IDAE NOT SET AFTER SEEK TO INVALID ADDR
8008  046604 032737 000200 004366 2$:    BIT     #D.FLT,HMR3
8009  046612 001001                      BNE     4$
8010  046614 104247                      ERROR   247             ;FLT NOT SET AFTER SEEK TO INV ADDR
8011  046616 032737 020000 004364 4$:    BIT     #D.PIP,HMR2
8012  046624 001401                      BEQ     5$
8013  046626 104250                      ERROR   250             ;PIP SET AFTER SEEK TO INV ADDR
8014  046630 032737 040000 004364 5$:    BIT     #D.DSC,HMR2
8015  046636 001001                      BNE     6$
8016  046640 104251                      ERROR   251             ;DSC NOT SET AFTER SEEK TO INV ADDR
8017
8018  046642 012765 100000 000000 6$:    MOV     #CCLR,RKCS1(R5)
8019  046650 012737 050340 004374        MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,SBMR2
8020  046656 004737 052504               JSR     PC,CKMR2        ;CHECK REST OF MR2
8021  046662 104252                      ERROR   252             ;MSG AO ERROR AFTER SEEK TO INV ADDR
8022  046664 012737 002240 004376        MOV     #<D.SKI!D.FLT!D.IDAE>,SBMR3
8023  046672 004737 052566               JSR     PC,CKMR3        ;CHECK REST OF MR3
8024  046676 104253                      ERROR   253             ;MSG BO ERROR AFT SEEK TO INV ADDR
8025  046700 012765 100000 000000        MOV     #CCLR,RKCS1(R5) ;CONTR CLEAR
8026  046706 012765 000001 000026        MOV     #1,RKMR1(R5)    ;SELECT WORD 1
```

```
8027  046714  004737  050256                  JSR     PC,GSTAT
8028  046720  012737  001720  004374          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.TFOK>,SBMR2
8029  046726  004737  052504                  JSR     PC,CKMR2
8030  046732  104254                          ERROR   254             ;MSG A1 ERROR AFT SEEK TO INV CYL
8031  046734  005037  004376                  CLR     SBMR3
8032  046740  004737  052566                  JSR     PC,CKMR3
8033  046744  104255                          ERROR   255             ;MSG B1 ERROR AFT SEEK TO INV CYL
8034  046746  004737  051124                  JSR     PC,RDCYLD       ;READ CYL DIFF IN MSG A2
8035  046752  023737  001402  001372          CMP     CYLDIF,TOCYL
8036  046760  001401                          BEQ     7$
8037  046762  104256                          ERROR   256             ;CYL DIFF IN RKMR2 NOT=CYL DIF
8038  046764  004737  051210          7$:     JSR     PC,RDCYLA
8039  046770  023737  001404  001372          CMP     CYLADD,TOCYL
8040  046776  001401                          BEQ     8$
8041  047000  104257                          ERROR   257             ;CYL ADDR IN RKMR3 NOT=RKDC
8042
8043  047002                          8$:
8044
8045  047002  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
8046  047010  013765  001222  000010          MOV     $UNIT,RKCS2(R5)
8047  047016  012765  000013  000000          MOV     #RECAL,RKCS1(R5)        ;RECAL CMD
8048  047024  013737  001434  004400          MOV     T10,TEMP1
8049  047032  004737  047632                  JSR     PC,FRDY         ;FIND RDY
8050  047036  104124                          ERROR   124             ;RDY NOT FOUND AFTER RECAL CMD
8051
8052  047040  012765  100000  000000          MOV     #CCLR,RKCS1(R5)
8053  047046  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;DRIVE#
8054  047054  012765  000005  000000          MOV     #CLEAR,RKCS1(R5)        ;DRIVE CLEAR CMD
8055  047062  013737  001434  004400          MOV     T10,TEMP1
8056  047070  004737  047632                  JSR     PC,FRDY         ;FIND RDY
8057  047074  104151                          ERROR   151             ;NO RDY AFTER DRIVE CLEAR CMD
8058  047076  004737  050016                  JSR     PC,TSTATN       ;TEST FOR ATTN
8059  047102  000401                          BR      66$
8060  047104  104154                          ERROR   154             ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
8061  047106                          66$:
8062
8063
8064  047106  004737  050256                  JSR     PC,GSTAT
8065  047112  032737  000040  004366          BIT     #D.IDAE,HMR3    ;SEE IF IDAE IS CLEARED
8066  047120  001401                          BEQ     65$             ;BR IF YES
8067  047122  104155                          ERROR   155             ;IDAE NOT CLEARED AFTER RECAL CMD
8068
8069  047124  012765  100000  000000  65$:    MOV     #CCLR,RKCS1(R5)
8070  047132  013737  001432  004402          MOV     T1,TEMP2        ;LOOK FOR ATTN FROM RECAL
8071  047140  004737  050050                  JSR     PC,FATT1
8072  047144  104055                          ERROR   55              ;NO ATTN AFTER RECAL CMD
8073
8074
8075  047146  062700  000002                  ADD     #2,R0
8076  047152  020027  000012                  CMP     R0,#10.
8077  047156  001402                          BEQ     $EOP
8078  047160  000137  046504                  JMP     1$
8079
8080
```

```
                                .SBTTL  END OF PASS ROUTINE

                                ;;*****************************************************************
                                ;*INCREMENT THE PASS NUMBER ($PASS)
                                ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
                                ;*IF THERES A MONITOR GO TO IT
                                ;*IF THERE ISN'T JUMP TO STS

047164                          $EOP:

047164  000004                          SCOPE
047166  012706  001100                  MOV     #STACK,SP
047172  005237  001220                  INC     $DEVCT          ;INCR COUNT FOR # OF DRIVES THAT ARE CHECKED
047176  023737  004422  001220          CMP     DRIVS,$DEVCT    ;ARE ALL DRIVES PRESINT TESTED?
047204  001403                          BEQ     $EOP1+2         ;BR IF YES
047206  000137  012174                  JMP     NUDRV           ;IF NOT , TEST NEXT DRIVE PRESENT
047212  000004                  $EOP1:  SCOPE
047214  005037  001102                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
047220  005037  001174                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
047224  005237  001216                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
047230  042737  100000  001216          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
047236  005327                          DEC     (PC)+           ;;LOOP?
047240  000001                  $EOPCT: .WORD   1
047242  003022                          BGT     $DOAGN          ;;YES
047244  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
047246  000001                  $ENDCT: .WORD   1
047250  047240                          $EOPCT
047252  104401  047317                  TYPE    ,$ENDMG         ;;TYPE "END PASS #"
047256  013746  001216                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
047262  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
047264  104401  047314                  TYPE    ,$NULL          ;;TYPE A NULL CHARACTER
047270  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
047274  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
047276  000005                          RESET                   ;;CLEAR THE WORLD
047300  004710                  $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
047302  000240                          NOP                     ;;SAVE ROOM
047304  000240                          NOP                     ;;FOR
047306  000240                          NOP                     ;;ACT11
047310                          $DOAGN:
047310  000137                          JMP     @(PC)+          ;;RETURN
047312  010534                  $RTNAD: .WORD   $TS
047314     377     377     000  $NULL:  .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
047317     015  042412  042116  $ENDMG: .ASCIZ  <15><12>/END PASS #/
047324  050040  051501  020123
047332  000043
```

```
                                .SBTTL  SUBROUTINES
                                ;SUBROUTINE TO CLEAR ALL FLAGS FROM DDUMP THRU DOTIM
                                ;
047334  012700  004412          CLRFLG: MOV     #DDUMP,R0
047340  012701  177757                  MOV     #-17.,R1
047344  005020                  1$:     CLR     (R0)+
047346  005201                          INC     R1
047350  001375                          BNE     1$
047352  000207                          RTS     PC
                                ;
                                ;TYPE PROGRAM ID IF FTITLE=0
                                ;
047354  005737  001364          TITLE:  TST     FTITLE
047360  001004                          BNE     1$
047362  005237  001364                  INC     FTITLE
047366  104401  057132                  TYPE    ,MSG1           ;PROGRAM ID
047372  000207                  1$:     RTS     PC
                                ;
                                ;ROUTINE TO INPUT DRIVE NOS. TYPED IN & SET
                                ;DRIVS, DRIV0-DRIV7 REGISTERS APPROPRIATELY
                                ;
047374  104411                  GDRVS:  RDLIN
047376  012600                          MOV     (SP)+,R0        ;GET STARTING ADDR OF ASCII STRING
047400  012701  177770                  MOV     #-8.,R1         ;SET UP COUNT
047404  112002                  1$:     MOVB    (R0)+,R2        ;GET ASCII CHAR
047406  042702  177400                  BIC     #177400,R2      ;MASK HI BYTE
047412  012703  004424                  MOV     #DRIV0,R3       ;DRIVE FLAG ADDR
047416  012704  000060                  MOV     #60,R4

047422  020402                  2$:     CMP     R4,R2           ;WAS TYPED CHAR 0 THRU 7?
047424  001415                          BEQ     3$              ;BRANCH IF YES
047426  005723                          TST     (R3)+           ;NO, INCREMENT DR FLAG ADDR
047430  005204                          INC     R4
047432  020427  000070                  CMP     R4,#70
047436  001371                          BNE     2$              ;S/B 0-7 OR TERMINATOR
047440  005702                          TST     R2
047442  001022                          BNE     4$
047444  020127  177770                  CMP     R1,#-8.
047450  001426                          BEQ     6$              ;DEFAULT ALL DRIVES
047452  005037  004452          7$:     CLR     SIZFLG          ;BYPASS TEST 1 (SIZING)
047456  000207                          RTS     PC              ;FOUND TERMINATOR, EXIT

047460  005213                  3$:     INC     @R3             ;SET UP FLAG FOR THE DRIVE
047462  005237  004422                  INC     DRIVS           ;INCREMENT TOTAL # DRIVES TO BE TESTED
047466  112002                          MOVB    (R0)+,R2        ;GET NEXT ASCII CHAR.
047470  042702  177400                  BIC     #177400,R2      ;MASK
047474  022702  000054                  CMP     #54,R2          ;IS IT A COMMA?
047500  001407                          BEQ     5$              ;YES, GO TO NEXT WORD.
047502  005702                          TST     R2              ;NO, IS IT A TERMINATOR?
047504  001001                          BNE     4$              ;IF NOT, SOMETHING WRONG.
```

```
8182  047506  000761                    BR      7$              ;FOUND TERMINATOR, EXIT
8183
8184  047510  104401  061261    4$:     TYPE    ,EM1            ;ONLY 0-7 ALLOWED.
8185  047514  000137  007764            JMP     PRGSRT          ;START ALL OVER
8186
8187  047520  005201            5$:     INC     R1              ;S/B NO MORE THAN 8 DIFF
8188  047522  001330                    BNE     1$              ;DRIVES TYPED IN.
8189  047524  000771                    BR      4$              ;IF NORE, HAVE ERROR.
8190
8191  047526  005237  004452    6$:     INC     SIZFLG          ;DO TEST 1 (SIZING)
8192  047532  000207                    RTS     PC              ;EXIT.
8193
8194                            ;
8195                            ;ROUTINE TO INPUT RKBAS OR DEFAULT.
8196                            ;
8197
8198  047534  104412            GBA:    RDOCT
8199  047536  012600                    MOV     (SP)+,R0        ;GET LOW ORDER FROM STACK
8200  047540  005700                    TST     R0
8201  047542  001403                    BEQ     1$              ;BRANCH IF DEFAULT.
8202  047544  010037  001264            MOV     R0,$BASE
8203  047550  000207                    RTS     PC
8204  047552  012737  177440  001264 1$:     MOV     #177440,$BASE   ;DEFAULT VALUE
8205  047560  000207                    RTS     PC
8206
8207                            ;
8208                            ;ROUTINE TO INPUT RKVEC OR DEFAULT
8209                            ;
8210
8211  047562  104412            GINT:   RDOCT
8212  047564  012600                    MOV     (SP)+,R0        ;GET LOW ORDER FROM STACK
8213  047566  005700                    TST     R0
8214  047570  001405                    BEQ     1$              ;BRANCH IF DEFAULT
8215  047572  010037  001334            MOV     R0,RKVEC
8216  047576  004737  047614    2$:     JSR     PC,SETINT
8217  047602  000207                    RTS     PC
8218  047604  012737  000210  001334 1$:     MOV     #210,RKVEC      ;DEFAULT VALUE
8219  047612  000771                    BR      2$
8220
8221                            ;
8222                            ;ROUTINE TO SETUP INTERRUPT VECTOR & PRIORITY
8223                            ;
8224
8225  047614  013700  001334    SETINT: MOV     RKVEC,R0
8226  047620  012720  053256            MOV     #INTER,(R0)+    ;INTER ADDR TO RKVEC
8227  047624  013710  001336            MOV     RKPRI,(R0)      ;PRS TO RKVEC+2
8228  047630  000207                    RTS     PC
8229
8230
8231                            ;
8232                            ;ROUTINE TO FIND CONTROLLER READY (RDY) DURING A DELAY
8233                            ;ENTER WITH A COUNT IN TEMP1
8234                            ;RETURN IF RDY NOT PRESENT (ERROR CONDITION)
8235                            ;RETURN +2 IF RDY PRESENT (SKIP OVER ERROR)
8236                            ;STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
8237                            ;
```

```
 8238  047632  032765  000200  000000 FRDY:   BIT     #RDY,RKCS1(R5)
 8239  047640  001006                 BNE     1$
 8240  047642  005337  004400         DEC     TEMP1
 8241  047646  001371                 BNE     FRDY
 8242  047650  004737  047670         JSR     PC,HOLD         ;STORE ALL RK611 REGS IN HOLDING REGS.
 8243  047654  000207                 RTS     PC              ;NO RDY, EXIT
 8244  047656  062716  000002 1$:     ADD     #2,(SP)         ;SKIP OVER ERROR
 8245  047662  004737  047670         JSR     PC,HOLD
 8246  047666  000207                 RTS     PC
 8247
 8248                                 ;
 8249                                 ;STORE ALL RK611 REGISTERS IN HOLDING REGS
 8250                                 ;
 8251
 8252  047670  016 37  000000  004336 HOLD:   MOV     RKCS1(R5),HCS1
 8253  047676  016537  000010  004340         MOV     RKCS2(R5),HCS2
 8254  047704  016537  000002  004342         MOV     RKWC(R5),HWC
 8255  047712  016537  000004  004344         MOV     RKBA(R5),HBA
 8256  047720  016537  000006  004346         MOV     RKDA(R5),HDA
 8257  047726  016537  000012  004350         MOV     RKDS(R5),HDS
 8258  047734  016537  000014  004352         MOV     RKER(R5),HER
 8259  047742  016537  000016  004354         MOV     RKASOF(R5),HASOF
 8260  047750  016537  000020  004356         MOV     RKDC(R5),HDC
 8261  047756  016537  000026  004362         MOV     RKMR1(R5),HMR1
 8262  047764  016537  000034  004364         MOV     RKMR2(R5),HMR2
 8263  047772  016537  000036  004366         MOV     RKMR3(R5),HMR3
 8264  050000  016537  000030  004370         MOV     RKECPS(R5),HPOS
 8265  050006  016537  000032  004372         MOV     RKECPT(R5),HPAT
 8266  050014  000207                 RTS     PC
 8267
 8268                                 ;
 8269                                 ;ROUTINE TO CHECK FOR CORRECT ATTN
 8270                                 ;RETURN IF ATTN NOT PRESENT (ERROR CONDITION)
 8271                                 ;RETURN +2 IF ATTN PRESENT (SKIP OVER ERROR)
 8272                                 ;
 8273  050016  010446                 ;STATN: MOV     R4,-(SP)        ;SAV R4
 8274  050020  013704  001222         MOV     $UNIT,R4
 8275  050024  136437  004326  004355 BITB    ATTN(R4),HASOF+1
 8276  050032  001404                 BEQ     1$              ;BRANCH IF ATTN NOT PRESENT
 8277  050034  012604                 MOV     (SP)+,R4        ;RESTOR R4
 8278  050036  062716  000002         ADD     #2,(SP)         ;INCR RET ADDR TO JUMP OVER ERROR.
 8279  050042  000207                 RTS     PC
 8280  050044  012604         1$:     MOV     (SP)+,R4        ;RESTOR R4
 8281  050046  000207                 RTS     PC
 8282
 8283                                 ;
 8284                                 ;ROUTINE TO FIND ATTN WITHIN TIMES GREATER THAN 1 SEC
 8285                                 ;ENTER WITH TIME IN SECONDS IN TEMP2
 8286                                 ;RETURN IF NO ATTN (ERROR CONDITION)
 8287                                 ;RETURN +2 IF ATTN FOUND
 8288                                 ;STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
 8289                                 ;
 8290
 8291  050050  010446                 FATT1:  MOV     R4,-(SP)        ;SAV R4
 8292  050052  012737  177777  004400 3$:     MOV     #-1,TEMP1
 8293  050060  013704  001222         MOV     $UNIT,R4
```

```
8294  050064  136465  004326  000017  1$:     BITB    ATTN(R4),RKASOF+1(R5) ;FIND CORRECT ATTN
8295  050072  001014                          BNE     2$
8296  050074  005337  004400                  DEC     TEMP1
8297  050100  001371                          BNE     1$
8298  050102  005337  004402                  DEC     TEMP2
8299  050106  001361                          BNE     3$
8300  050110  005065  000026                  CLR     RKMR1(R5)       ;SELECT WORD 0
8301  050114  004737  050256                  JSR     PC,GSTAT        ;GET LATEST STATUS
8302  050120  012604                          MOV     (SP)+,R4        ;RESTOR R4
8303  050122  000207                          RTS     PC
8304  050124  005065  000026          2$:     CLR     RKMR1(R5)
8305  050130  004737  050256                  JSR     PC,GSTAT        ;GET STATUS AFTER ATTN SEEN
8306  050134  012604                          MOV     (SP)+,R4        ;RESTOR R4
8307  050136  062716  000002                  ADD     #2,(SP)         ;SKIP OVER ERROR
8308  050142  000207                          RTS     PC
8309
8310                                  ;
8311                                  ;ROUTINE TO FIND ATTN WITHIN 1 SEC
8312                                  ;ENTER WITH COUNT IN TEMP1
8313                                  ;RETURN IF NO ATTN (ERROR)
8314                                  ;RETURN +2 IF ATTN FOUND
8315                                  ;STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
8316                                  ;
8317
8318  050144  010446                  FATT2:  MOV     R4,-(SP)        ;SAV R4
8319  050146  013704  001222          2$:     MOV     $UNIT,R4
8320  050152  136465  004326  000017          BITB    ATTN(R4),RKASOF+1(R5)  ;FIND CORRECT ATTN
8321  050160  001011                          BNE     1$
8322  050162  005337  004400                  DEC     TEMP1
8323  050166  001367                          BNE     2$
8324  050170  005065  000026                  CLR     RKMR1(R5)       ;SELECT WORD 0
8325  050174  004737  050256                  JSR     PC,GSTAT        ;GET LATEST STATUS.
8326  050200  012604                          MOV     (SP)+,R4        ;RESTOR R4
8327  050202  000207                          RTS     PC
8328  050204  005065  000026          1$:     CLR     RKMR1(R5)
8329  050210  004737  050256                  JSR     PC,GSTAT
8330  050214  012604                          MOV     (SP)+,R4        ;RESTOR R4
8331  050216  062716  000002                  ADD     #2,(SP)         ;SKIP OVER ERROR
8332  050222  000207                          RTS     PC
8333                                  ;
8334                                  ;ENTER WITH A COUNT IN TEMP1
8335                                  ;THE DELAY IS APPROX 17 US/ITERATION + 12 US TO EXIT
8336                                  ;WHEN COUNT IS 0...BASED ON AN 11/05.
8337                                  ;
8338  050224  005737  004400          DLY:    TST     TEMP1           ;5.6 US
8339  050230  001403                          BEQ     1$              ;2.5 US
8340  050232  005337  004400                  DEC     TEMP1           ;6.8 US
8341  050236  000772                          BR      DLY             ;2.5 US
8342  050240  000207                  1$:     RTS     PC              ;3.8 US
8343                                  ;
8344                                  ;THIS ROUTINE TYPES BYPASSED DRIVE#.  ENTER WITH DRIVE# IN R0
8345                                  ;
8346
8347  050242  104401  060607          BYP:    TYPE    ,MSG14          ;BYPASS DRIVE
8348  050246  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
8349                                                                  ;;TYPE DR#
```

```
8350  050250  104403                        TYPOS                   ;;GO TYPE--OCTAL ASCII
8351  050252  001                           .BYTE   1               ;;TYPE 1 DIGIT(S)
8352  050253  000                           .BYTE   0               ;;SUPPRESS LEADING ZEROS
8353  050254  000207                        RTS     PC
8354
8355                                  ;
8356                                  ;THIS ROUTINE DOES THE SELECT DRIVE COMMAND TO GET STATUS
8357                                  ;IT THEN WAITS FOR CONTROLLER  READY.
8358                                  ;IF RDY NOT RECEIVED BY A TIMEOUT, AN ERROR IS FLAGGED
8359                                  ;
8360  050256  013746  004400          GSTAT:  MOV     TEMP1,-(SP)     ;SAVE TEMP1
8361  050262  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;CURRENT DRIVE #
8362  050270  012765  000001  000000          MOV     #SELDRV,RKCS1(R5)  ;GET STATUS WITH SELECT DRIVE CMD
8363  050276  013737  001434  004400          MOV     T10,TEMP1
8364  050304  004737  047632                  JSR     PC,FRDY         ;FIND RDY
8365  050310  104117                          ERROR   117             ;RDY NOT SET BY END OF SELECT DRIVE CMD
8366  050312  012637  004400                  MOV     (SP)+,TEMP1     ;RESTOR TEMP1
8367  050316  000207                          RTS     PC
8368
8369
8370                                  ;
8371                                  ;THIS ROUTINE DOES A SUBSYSTEM CLEAR & WAITS FOR CONTROLLER READY
8372                                  ;IF RDY IS NOT RECEIVED BY THE END OF THE TIMEOUT, AN ERROR IS FLAGGED.
8373                                  ;THE ROUTINE THEN GETS CURRENT STATUS & CHECKS FOR CONTROLLER ERROR (CERR)
8374                                  ;RETURN IF CERR SET
8375                                  ;RETURN +2 IF CERR CLEAR
8376                                  ;
8377  050320  012765  000040  000010  SUBCLR: MOV     #SCLR,RKCS2(R5) ;SUBSYS CLEAR
8378  050326  013737  001434  004400          MOV     T10,TEMP1
8379  050334  004737  047632                  JSR     PC,FRDY         ;FIND RDY
8380  050340  104120                          ERROR   120             ;RDY NOT SET BY END OF SCLR
8381  050342  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;CURRENT DRIVE #
8382  050350  005065  000026                  CLR     RKMR1(R5)       ;SELECT WORD 0
8383  050354  004737  050256                  JSR     PC,GSTAT        ;GET STATUS
8384  050360  032737  100000  004336          BIT     #CERR,HCS1      ;CHECK FOR CONT ERROR
8385  050366  001401                          BEQ     1$
8386  050370  000207                          RTS     PC
8387  050372  062716  000002          1$:     ADD     #2,(SP)         ;SKIP OVER ERROR
8388  050376  000207                          RTS     PC
8389
8390                                  ;
8391                                  ;READ THE SECTOR COUNT IN RKMR3, RIGHT JUSTIFY IT & STORE IT IN 'SECTOR'
8392                                  ;
8393  050400  012765  000003  000026  RDSEC:  MOV     #3,RKMR1(R5)    ;WORD 3
8394  050406  004737  050256                  JSR     PC,GSTAT
8395  050412  013737  004366  001426          MOV     HMR3,SECTOR
8396  050420  042737  177017  001426          BIC     #^C<M.SECT>,SECTOR
8397  050426  006237  001426                  ASR     SECTOR          ;RIGHT JUSTIFY
8398  050432  006237  001426                  ASR     SECTOR          ;SECTOR
8399  050436  006237  001426                  ASR     SECTOR          ;INFO
8400  050442  006237  001426                  ASR     SECTOR
8401  050446  000207                          RTS     PC
8402                                  ;
8403                                  ;READ THE SECTOR COUNT TO RKMR3 IN 20 SECTOR FORMAT
8404                                  ;RIGHT JUSTIFY IT & STORE IT IN 'SECTOR'
8405                                  ;
```

```
8406  050450  013746  004400          R20SEC: MOV     TEMP1,-(SP)      ;SAVE TEMP 1
8407  050454  012765  000003  000026          MOV     #3,RKMR1(R5)     ;SELECT WORD 3
8408  050462  013765  001222  000010          MOV     $UNIT,RKCS2(R5) ;CURRENT DRIVE #
8409  050470  012765  010001  000000          MOV     #<CFMT!SELDRV>,RKCS1(R5)        ;SETUP 20 SECTOR STATUS
8410  050476  013737  001434  004400          MOV     T10,TEMP1
8411  050504  032765  000200  000000  1$:     BIT     #RDY,RKCS1(R5)
8412  050512  001007                          BNE     2$
8413  050514  005337  004400                  DEC     TEMP1            ;TRY AGAIN IF TIME NOT UP
8414  050520  001371                          BNE     1$
8415  050522  104117                          ERROR   117              ;RDY NOT SET AFTER SELECT DRIVE C.1D
8416  050524  012637  004400                  MOV     (SP)+,TEMP1      ;RESTORE TEMP 1
8417  050530  000207                          RTS     PC
8418
8419  050532  013737  004366  001426  2$:     MOV     HMR3,SECTOR      ;GET SECTOR INFO
8420  050540  042737  177017  001426          BIC     #^C<M.SECT>,SECTOR
8421  050546  006237  001426                  ASR     SECTOR
8422  050552  006237  001426                  ASR     SECTOR
8423  050556  006237  001426                  ASR     SECTOR
8424  050562  006237  001426                  ASR     SECTOR
8425
8426  050566  012637  004400                  MOV     (SP)+,TEMP1      ;RESTORE TEMP 1
8427  050572  000207                          RTS     PC
8428
8429
8430                                  ;
8431                                  ;FIND SECTOR 0 IN 22 SECTOR FORMAT.
8432                                  ;ERROR FLAGGED IF NOT FOUND BY TIMEOUT
8433                                  ;
8434  050574  013746  004400          FS022:  MOV     TEMP1,-(SP)      ;SAVE TEMP1
8435  050600  013737  001444  004400          MOV     T5000,TEMP1      ;SETUP TIMEOUT
8436  050606  004737  050400  1$:     JSR     PC,RDSEC                 ;READ SECTOR
8437  050612  005737  001426                  TST     SECTOR           ;LOOK FOR SECTOR 0
8438  050616  001005                          BNE     2$
8439  050620  004737  050400                  JSR     PC,RDSEC
8440  050624  005737  001426                  TST     SECTOR
8441  050630  001406                          BEQ     3$               ;BR IF SAME TWICE
8442  050632  005337  004400          2$:     DEC     TEMP1
8443  050636  001363                          BNE     1$               ;TRY AGAIN IF TIMEOUT NOT UP
8444  050640  012637  004400                  MOV     (SP)+,TEMP1      ;ELSE RESTORE TEMP1
8445  050644  000207                          RTS     PC               ;EXIT
8446  050646  012637  004400          3$:     MOV     (SP)+,TEMP1
8447  050652  062716  000002                  ADD     #2,(SP)          ;SKIP OVER ERROR
8448  050656  000207                          RTS     PC
8449
8450                                  ;
8451                                  ;FIND SECTOR 0 IN 20 SECTOR FORMAT.
8452                                  ;ERROR FLAGGED IF NOT FOUND BY TIMEOUT
8453                                  ;
8454  050660  013746  004400          FS020:  MOV     TEMP1,-(SP)      ;SAVE TEMP1
8455  050664  013737  001444  004400          MOV     T5000,TEMP1      ;SETUP TIMEOUT
8456  050672  004737  050450  1$:     JSR     PC,R20SEC                ;READ SECTOR
8457  050676  005737  001426                  TST     SECTOR           ;LOOK FOR SECTOR 0
8458  050702  001005                          BNE     2$
8459  050704  004737  050450                  JSR     PC,R20SEC
8460  050710  005737  001426                  TST     SECTOR
8461  050714  001406                          BEQ     3$               ;BR IF SAME TWICE
```

```
8462  050716  005337  004400          2$:     DEC     TEMP1
8463  050722  001363                          BNE     1$                      ;TRY AGAIN IF TIMEOUT NOT UP
8464  050724  012637  004400                  MOV     (SP)+,TEMP1             ;ELSE RESTORE TEMP1
8465  050730  000207                          RTS     PC                      ;EXIT
8466  050732  012637  004400          3$:     MOV     (SP)+,TEMP1
8467  050736  062716  000002                  ADD     #2,(SP)                 ;SKIP OVER ERROR
8468  050742  000207                          RTS     PC
8469
8470                                          ;
8471                                          ;FIND NEXT SECTOR IN 22 SECTOR FORMAT
8472                                          ;ERROR FLAGGED IF NOT FOUND BY TIMEOUT
8473                                          ;
8474  050744  013746  004400          FNS22:  MOV     TEMP1,-(SP)             ;SAVE TEMP 1
8475  050750  013737  001440  004400          MOV     T500,TEMP1              ;SETUP TIMEOUT
8476  050756  004737  050400          1$:     JSR     PC,RDSEC                        ;READ SECTOR
8477  050762  023737  001422  001426          CMP     PSEC,SECTOR
8478  050770  001406                          BEQ     3$                      ;BR IF SAME
8479  050772  004737  050400                  JSR     PC,RDSEC                        ;ELSE TRY READ DIFFERENT TWICE
8480  050776  023737  001422  001426          CMP     PSEC,SECTOR
8481  051004  001006                          BNE     2$                      ;BR IF DIFFERENT TWICE
8482  051006  005337  004400          3$:     DEC     TEMP1                   ;ELSE TRY AGAIN IF TIME LEFT
8483  051012  001361                          BNE     1$
8484  051014  012637  004400                  MOV     (SP)+,TEMP1             ;RESTORE TEMP 1
8485  051020  000207                          RTS     PC
8486  051022  012637  004400          2$:     MOV     (SP)+,TEMP1             ;RESTORE TEMP 1
8487  051026  062716  000002                  ADD     #2,(SP)                 ;SKIP OVER ERROR
8488  051032  000207                          RTS     PC
3489
8490                                          ;
8491                                          ;FIND NEXT SECTOR IN 20 SECTOR FORMAT
8492                                          ;ERROR FLAGGED IF NOT FOUND BY TIMEOUT
8493                                          ;
8494  051034  013746  004400          FNS20:  MOV     TEMP1,-(SP)             ;SAVE TEMP 1
8495  051040  013737  001440  004400          MOV     T500,TEMP1              ;SETUP TIMEOUT
8496  051046  004737  050450          1$:     JSR     PC,R20SEC                       ;READ SECTOR
8497  051052  023737  001422  001426          CMP     PSEC,SECTOR
8498  051060  001406                          BEQ     3$                      ;BR IF SAME
8499  051062  004737  050450                  JSR     PC,R20SEC                       ;ELSE TRY READ DIFFERENT TWICE
8500  051066  023737  001422  001426          CMP     PSEC,SECTOR
8501  051074  001006                          BNE     2$                      ;BR IF DIFFERENT TWICE
8502  051076  005337  004400          3$:     DEC     TEMP1                   ;ELSE TRY AGAIN IF TIME LEFT
8503  051102  001361                          BNE     1$
8504  051104  012637  004400                  MOV     (SP)+,TEMP1             ;RESTORE TEMP 1
8505  051110  000207                          RTS     PC
8506  051112  012637  004400          2$:     MOV     (SP)+,TEMP1             ;RESTORE TEMP 1
8507  051116  062716  000002                  ADD     #2,(SP)                 ;SKIP OVER ERROR
8508  051122  000207                          RTS     PC
8509                                          ;
8510                                          ;READ THE CYL DIFF/OFFSET IN RKMR2, RIGHT JUSTIFY IT & STORE IT IN 'CYLDIF'
8511                                          ;
8512  051124  012765  000002  000026  RDCYLD: MOV     #2,RKMR1(R5)            ;WORD 2
8513  051132  004737  050256                  JSR     PC,GSTAT
8514  051136  013737  004364  001402          MOV     HMR2,CYLDIF
8515  051144  042737  160017  001402          BIC     #^C<M.CDIF>,CYLDIF
8516  051152  006237  001402                  ASR     CYLDIF                  ;RIGHT JUSTIFY
8517  051156  006237  001402                  ASR     CYLDIF                  ;CYL DIFF/OFFSET
```

```
 8518  051162  006237  001402          ASR     CYLDIF          ;INFO
 8519  051166  006237  001402          ASR     CYLDIF
 8520  051172  023727  001402  000777  CMP     CYLDIF,#777     ;CHK TO SEE IF RET IN COMPL. FORM
 8521  051200  001002                  BNE     1$              ;BR IF NOT
 8522  051202  005037  001402          CLR     CYLDIF          ;CLR IF YES
 8523  051206  000207              1$: RTS     PC
 8524                                ;
 8525                                ;READ THE CYL ADDR IN RKMR3, RIGHT JUSTIFY IT & STORE IT IN  'CYLADD'
 8526                                ;
 8527  051210  012765  000002  000026 RDCYLA: MOV     #2,RKMR1(R5)    ;WORD 2
 8528  051216  004737  050256          JSR     PC,GSTAT
 8529  051222  013737  004366  001404  MOV     HMR3,CYLADD
 8530  051230  042737  160017  001404  BIC     #^C<M.CADD>,CYLADD
 8531  051236  006237  001404          ASR     CYLADD          ;RIGHT JUSTIFY
 8532  051242  006237  001404          ASR     CYLADD          ;CYL ADDR
 8533  051246  006237  001404          ASR     CYLADD          ;INFO
 8534  051252  006237  001404          ASR     CYLADD
 8535  051256  000207                  RTS     PC
 8536                                ;
 8537                                ;FIND LIMIT DETECT ON SEEK IN RKMR3 BEFORE TIMEOUT
 8538                                ;RETURN IF NOT FOUND: ERROR
 8539                                ;RETURN+2 IF FOUND: SKIP OVER ERROR
 8540                                ;
 8541  051260  005037  001476      FLIM:   CLR     LIMERR          ;LIMIT DETECT ERROR FLAG
 8542  051264  012737  000764  004400      MOV     #500.,TEMP1     ;SETUP TIMEOUT
 8543  051272  012765  000001  000026      MOV     #1,RKMR1(R5)    ;WORD 1
 8544  051300  004737  050256          1$: JSR     PC,GSTAT
 8545  051304  032737  020000  004366      BIT     #D.LIMD,HMR3
 8546  051312  001006                      BNE     2$              ;EXIT IF SET
 8547  051314  005337  004400              DEC     TEMP1
 8548  051320  001367                      BNE     1$
 8549  051322  005237  001476              INC     LIMERR          ;SET LIMIT DETECT FLAG
 8550  051326  000207                      RTS     PC
 8551  051330  062716  000002          2$: ADD     #2,(SP)         ;SKIP OVER ERROR
 8552  051334  000207                      RTS     PC
 8553                                ;
 8554                                ;ROUTINE TO FIND HEADS HOME IN RKMR2 WORD 1 BEFORE TIMEOUT
 8555                                ;ENTER WITH TIME IN SECONDS IN TEMP2
 8556                                ;RETURN IF NOT FOUND
 8557                                ;RETURN+2 IF FOUND - SKIP OVER ERROR
 8558                                ;
 8559  051336  012737  177777  004400 FHDHM:  MOV     #-1,TEMP1       ;ALL 1'S
 8560  051344  012765  000001  000026      MOV     #1,RKMR1(R5)    ;WORD 1
```

```
 8561  051352  004737  050256          1$:     JSR     PC,GSTAT
 8562  051356  032737  000040  004364          BIT     #D.HDHM,HMR2
 8563  051364  001007                          BNE     2$
 8564  051366  005337  004400                  DEC     TEMP1
 8565  051372  001367                          BNE     1$
 8566  051374  005337  004402                  DEC     TEMP2
 8567  051400  001356                          BNE     FHDHM
 8568  051402  000207                          RTS     PC
 8569  051404  062716  000002          2$:     ADD     #2,(SP)         ;SKIP OVER ERROR
 8570  051410  000207                          RTS     PC
 8571                                  ;
 8572                                  ;ROUTINE TO FIND LOAD HEADS IN RKMR2 WORD 1 BEFORE TIMEOUT
 8573                                  ;RETURN IF NOT FOUND
 8574                                  ;RETURN+2 IF FOUND: SKIP OVER ERROR
 8575                                  ;
 8576  051412  012737  000372  004400  FLOAD:  MOV     #250.,TEMP1     ;SETUP TIMEOUT
```

```
8577  051420  012765  000001  000026          MOV     #1,RKMR1(R5)    ;WORD 1
8578  051426  004737  050256          1$:     JSR     PC,GSTAT
8579  051432  032737  010000  004364          BIT     #D.LOAD,HMR2
8580  051440  001004                          BNE     2$
8581  051442  005337  004400                  DEC     TEMP1
8582  051446  001367                          BNE     1$
8583  051450  000207                          RTS     PC
8584  051452  062716  000002          2$:     ADD     #2,(SP)         ;SKIP OVER ERROR
8585  051456  000207                          RTS     PC
8586                                          ;
8587                                          ;FILL HEADER TABLE WITH 66 WORDS OF VALID HEADERS
8588                                          ;ENTER WITH CYL # IN 'CALADD'
8589                                          ;ENTER WITH HEAD # IN 'HEAD'
8590                                          ;ENTER WITH FORMAT IN 'FORMAT'
8591                                          ;
8592  051460  010046                  FHDTAB: MOV     R0,-(SP)        ;SAV R0
8593  051462  010146                          MOV     R1,-(SP)        ;SAV R1
8594  051464  012700  001500                  MOV     #HDTAB,R0       ;HEADER WORD TABLE ADDR
8595  051470  005001                          CLR     R1              ;SECTOR COUNTER
8596  051472  013737  001450  001452          MOV     HEAD,HD1
8597  051500  006337  001452                  ASL     HD1
8598  051504  006337  001452                  ASL     HD1
8599  051510  006337  001452                  ASL     HD1
8600  051514  006337  001452                  ASL     HD1
8601  051520  006337  001452                  ASL     HD1             ;SETUP HEAD # FOR WORD 2 OF HEADER
8602  051524  013737  001454  001456          MOV     FORMAT,FMT1
8603  051532  000337  001456                  SWAB    FMT1
8604  051536  006337  001456                  ASL     FMT1            ;SETUP FORMAT FOR WORD 2 OF HEADER
8605
8606  051542  013720  001406          1$:     MOV     CALADD,(R0)+    ;HEADER WORD 1-CYL ADDR
8607  051546  010110                          MOV     R1,(R0)         ;HEADER WORD 2-SECTOR NO
```

```
 8608  051550  053710  001452          BIS     HD1,(R0)        ;               -HEAD NO
 8609  051554  053710  001456          BIS     FMT1,(R0)       ;               -FORMAT
 8610  051560  052710  140000          BIS     #(BIT14!BIT15),(R0)     ;       -GOOD SECTOR FLAGS
 8611
 8612  051564  013737  001406  004400          MOV     CALADD,TEMP1
 8613  051572  011037  004402                  MOV     (R0),TEMP2
 8614  051576  043737  001406  004402          BIC     CALADD,TEMP2
 8615  051604  042037  004400                  BIC     (R0)+,TEMP1
 8616  051610  053737  004400  004402          BIS     TEMP1,TEMP2
 8617  051616  013720  004402                  MOV     TEMP2,(R0)+     ;HEADER WORD 3-HEADER CHECK
 8618
 8619  051622  005201                  INC     R1              ;SECTOR CTR
 8620  051624  020127  000026          CMP     R1,#22.         ;ALL 22 SECTORS DONE? (66 WORDS)
 8621  051630  001344                  BNE     1$              ;BR IF NO
 8622
 8623  051632  012601                  MOV     (SP)+,R1        ;RESTOR R1
 8624  051634  012600                  MOV     (SP)+,R0        ;RESTOR R0
 8625  051636  000207                  RTS     PC
 8626
 8627
 8628                          ;
 8629                          ;THIS ROUTINE SORTS THE RHTAB TABLE FROM WHATEVER SECTOR IT BEGINS
 8630                          ;WITH AND RE-WRITES THE INFO IN SRTTAB TABLE TO BEGIN WITH SECTOR 0
 8631                          ;
 8632  051640  010046          SORT:   MOV     R0,-(SP)        ;SAVE R0
 8633  051642  010146                  MOV     R1,-(SP)        ;SAVE R1
 8634  051644  004737  050400          JSR     PC,RDSEC
 8635  051650  062737  000001  001426  ADD     #1,SECTOR
 8636  051656  004737  051746          JSR     PC,MULT6        ;MULT SECTOR BY 6
 8637  051662  012700  000204          MOV     #132.,R0
 8638  051666  163700  001426          SUB     SECTOR,R0       ;R0-SECTOR TO R0 = INDEX
 8639  051672  010037  001426          MOV     R0,SECTOR
 8640  051676  062737  001704  001426  ADD     #RHTAB,SECTOR   ;SAVE INDEX
 8641
 8642  051704  062700  001704          ADD     #RHTAB,R0       ;INDEX TO BOT HALF OF RHTAB
 8643  051710  012701  002110          MOV     #SRTTAB,R1      ;INDEX TO TOP HALF OF SRTTAB
 8644
 8645  051714  012021          1$:     MOV     (R0)+,(R1)+     ;PUT BOTTOM OF RHTAB TO TOP OF SRTTAB
 8646  051716  020027  002110          CMP     R0,#RHTAB+132.
 8647  051722  001374                  BNE     1$
 8648
 8649  051724  012700  001704          MOV     #RHTAB,R0       ;PUT TOP OF RHTAB TO BOT OF SRTTAB
 8650  051730  012021          2$:     MOV     (R0)+,(R1)+
 8651  051732  020037  001426          CMP     R0,SECTOR
 8652  051736  001374                  BNE     2$
 8653
 8654  051740  012601                  MOV     (SP)+,R1        ;RESTOR R1
 8655  051742  012600                  MOV     (SP)+,R0        ;RESTOR R0
 8656  051744  000207                  RTS     PC
 8657
 8658
 8659                          ;MULT BY 6. ENTER WITH DESIRED # IN 'SECTOR'
 8660                          ;
 8661  051746  006337  001426  MULT6:  ASL     SECTOR          ;2 X SECTOR
 8662  051752  013746  001426          MOV     SECTOR,-(SP)
 8663  051756  006337  001426          ASL     SECTOR          ;4 X SECTOR
```

```
8664  051762  062637  001426              ADD     (SP)+,SECTOR    ;(4 X S)+(2 X S) = 6 X SECTOR
8665  051766  000207                      RTS     PC
8666
8667                                  ;
8668                                  ;THIS ROUTINE IS ENTERED IF THERE IS A READ HEADER MISMATCH.
8669                                  ;IT COMPARES THE CYLINDER, SECTOR AND HEAD NUMBERS OF WHERE THE
8670                                  ;HEADS WERE AT THE TIME OF READING AGAINST THE BSE INFO
8671                                  ;OBTAINED IN A PREVIOUS TEST.
8672                                  ;RETURN IF NO COMPARE: ERROR CONDITION
8673                                  ;RETURN+2 IF COMPARE FOUND: SKIP OVER ERROR
8674                                  ;
8675  051770  010446                  TRUERR: MOV     R4,-(SP)        ;SAVE R4
8676  051772  013746  001160                  MOV     $TMP0,-(SP)     ;SAVE
8677  051776  013746  001162                  MOV     $TMP1,-(SP)     ;SAVE
8678
8679  052002  013737  004346  001160          MOV     HDA,$TMP0       ;READ TRK SECTOR INFO
8680  052010  013737  004346  001162          MOV     HDA,$TMP1
8681
8682  052016  042737  177740  001160          BIC     #^C<37>,$TMP0   ;TMP0 HAS SECTOR INFO
8683  052024  042737  174377  001162          BIC     #^C<3400>,$TMP1
8684  052032  000337  001162                  SWAB    $TMP1           ;TMP1 HAS HEAD INFO
8685
8686  052036  005737  001160                  TST     $TMP0           ;SEE IF SECTOR 0
8687  052042  001414                          BEQ     6$
8688  052044  005337  001160                  DEC     $TMP0           ;GET ACTUAL WLE SECTOR
8689  052050  013737  001160  001426          MOV     $TMP0,SECTOR    ;STORE SECTOR
8690  052056  013737  001162  001450          MOV     $TMP1,HEAD      ;STORE HEAD
8691  052064  013737  004356  001374          MOV     HDC,CCYL        ;STORE CYLINDER
8692  052072  000440                          BR      9$
8693
8694  052074  005737  001162          6$:     TST     $TMP1           ;SEE IF HEAD 0
8695  052100  001414                          BEQ     7$              ;BR IF YES
8696  052102  005337  001162                  DEC     $TMP1           ;GET ACTUAL WLE HEAD
8697  052106  013737  001162  001450          MOV     $TMP1,HEAD
8698  052114  012737  000025  001426          MOV     #21.,SECTOR
8699  052122  013737  004356  001374          MOV     HDC,CC -
8700  052130  000421                          BR 9$
8701
8702  052132  005737  004356          7$:     TST     HDC             ;SEE IF CYL 0
8703  052136  001414                          BEQ     8$
8704  052140  005337  004356                  DEC     HDC             ;GET ACTUAL WLE CYL
8705  052144  013737  004356  001374          MOV     HDC,CCYL
8706  052152  012737  000002  001450          MOV     #2,HEAD
8707  052160  012737  000025  001426          MOV     #21.,SECTOR
8708  052166  000402                          BR      9$
8709
8710  052170  104314                  8$:     ERROR   314             ;RKDC & RKDA INDICATES WCE
8711  052172  000453                          BR      5$              ;OCCURRED AT CYL 411,HD 2 SECTOR 21
8712
8713  052174  012704  003324          9$:     MOV     #BSE22+8.,R4
8714  052200  012437  001470          1$:     MOV     (R4)+,WORD      ;GET CYL# OFF BSE TABLE
8715  052204  023727  001470  177777          CMP     WORD,#-1        ;SEE IF ALL 1'S
8716  052212  001406                          BEQ     2$              ;EXIT IF YES
8717  052214  023737  001470  001374          CMP     WORD,CCYL       ;COMPARE CYL #
8718  052222  001410                          BEQ     3$              ;BR IF CYL MATCH
8719  052224  005724                          TST     (R4)+           ;ADV TO NEXT CYL WORD
```

```
 8720  052226  000764                          BR      1$
 8721
 8722  052230  012637  001162          2$:     MOV     (SP)+,$TMP1     ;RESTOR
 8723  052234  012637  001160                  MOV     (SP)+,$TMP0     ;RESTOR
 8724  052240  012604                          MOV     (SP)+,R4        ;RESTOR R4
 8725  052242  000207                          RTS     PC
 8726
 8727  052244  011437  001470          3$:     MOV     (R4),WORD       ;GET HEAD & SECTOR FROM BSE TABLE
 8728  052250  042737  177400  001470          BIC     #177400,WORD    ;KEEP SECTOR# ONLY
 8729  052256  023737  001470  001426          CMP     WORD,SECTOR     ;SECTOR COMPARE?
 8730  052264  001402                          BEQ     4$              ;BR IF YES
 8731  052266  005724                          TST     (R4)+           ;ELSE GET NEXT CYL # OFF TABLE
 8732  052270  000743                          BR      1$
 8733
 8734  052272  012437  001470          4$:     MOV     (R4)+,WORD      ;GET HEAD & SECTOR FROM BSE TABLE
 8735  052276  000337  001470                  SWAB    WORD
 8736  052302  042737  177400  001470          BIC     #177400,WORD    ;KEEP HEAD# ONLY
 8737  052310  023737  001470  001450          CMP     WORD,HEAD       ;HEAD COMPARE?
 8738  052316  001401                          BEQ     5$              ;BR IF YES
 8739  052320  000727                          BR      1$
 8740
 8741  052322  012637  001162          5$:     MOV     (SP)+,$TMP1     ;RESTOR
 8742  052326  012637  001160                  MOV     (SP)+,$TMP0     ;RESTOR
 8743  052332  012604                          MOV     (SP)+,R4        ;RESTOR R4
 8744  052334  062716  000002                  ADD     #2,(SP)         ;SKIP OVER ERR ON RETURN
 8745  052340  000207                          RTS     PC
 8746                                  ;
 8747                                  ;ROUTINE TO TURN L OR P CLOCK INTERRUPT ON
 8748                                  ;
 8749  052342  012746  000000          CLKON:  MOV     #PR0,-(SP)      ;PSW LOADED TO BE
 8750  052346  012746  052354                  MOV     #2$,-(SP)       ;LSI-11 COMPATABLE
 8751  052352  000002                          RTI                     ;ENABLE INTERRUPTS FROM CLOCK
 8752  052354  005037  001416          2$:     CLR     TIMUP
 8753  052360  005737  004446                  TST     PCLKF
 8754  052364  001004                          BNE     1$              ;BRANCH IF P-CLOCK PRESENT
 8755  052366  012777  000100  126752          MOV     #100,@LKS       ;L-CLOCK, ENABLE INT
 8756  052374  000207                          RTS     PC
 8757  052376  012777  177777  126736  1$:     MOV     #-1,@PKSB       ;P-CLOCK, ALL 1'S
 8758  052404  012777  000135  126726          MOV     #135,@PKS       ;ENABLE INT, CT UP, REP INT
 8759  052412  000207                          RTS     PC              ;LINE FREQ & RUN
 8760
 8761                                  ;KW11-L & KW11-P INTERRUPT HANDLER
 8762                                  ;
 8763  052414  005037  001416          CLOCK:  CLR     TIMUP
 8764  052420  005337  001412                  DEC     COUNT
 8765  052424  001010                          BNE     1$
 8766  052426  013737  001410  001412          MOV     HZ,COUNT
 8767  052434  005337  001414                  DEC     SEC
 8768  052440  001002                          BNE     1$
 8769  052442  005237  001416                  INC     TIMUP           ;SORRY, TIME IS UP
 8770  052446  000002                  1$:     RTI
 8771                                  ;
 8772                                  ;ROUTINE TO TURN L OR P CLOCK INTERRUPT OFF
 8773                                  ;
 8774  052450  012746  000340          CLKOF:  MOV     #PR7,-(SP)      ;PSW LOADED TO BE
 8775  052454  012746  052462                  MOV     #2$,-(SP)       ;LSI-11 COMPATABLE
```

```
8776  052460  000002                          RTI                     ;DISABLE ALL INTERRUPTS
8777  052462  005737  004446          2$:     TST     PCLKF
8778  052466  001003                          BNE     1$              ;BRACH IF P-CLOCK PRESENT
8779  052470  005077  126652                  CLR     @LKS            ;L-CLOCK, CLEAR INTERRUPT
8780  052474  000207                          RTS     PC
8781  052476  005077  126636          1$:     CLR     @PKS            ;P-CLOCK, CLEAR INTERRUPT
8782  052502  000207                          RTS     PC
8783                                  ;
8784                                  ;THIS ROUTINE CHECKS RKMR2 (MSGA)-WORDS 0 & 1
8785                                  ;ENTER WITH SHOULD BE VALUE IN SBMR2.
8786                                  ;RETURN IF NO COMPARE, RETURN +2 IF COMPARE
8787                                  ;
8788  052504  013746  004400          CKMR2:  MOV     TEMP1,-(SP)     ;SAV TEMP1
8789  052510  053737  001222  004374          BIS     $UNIT,SBMR2     ;INSERT DRIVE #
8790  052516  013737  004374  004400          MOV     SBMR2,TEMP1
8791  052524  004737  052664                  JSR     PC,SBPAR        ;GET PARITY FOR SBMR2
8792  052530  013737  004400  004374          MOV     TEMP1,SBMR2     ;NOW HAS PARITY
8793  052536  023737  004374  004364          CMP     SBMR2,HMR2      ;SHOULD BE SAME
8794  052544  001005                          BNE     1$
8795  052546  012637  004400                  MOV     (SP)+,TEMP1     ;RESTOR TEMP1
8796  052552  062716  000002                  ADD     #2,(SP)         ;COMPARE OK, SKIP OVER ERROR.
8797  052556  000207                          RTS     PC
8798  052560  012637  004400          1$:     MOV     (SP)+,TEMP1     ;RESTOR TEMP1
8799  052564  000207                          RTS     PC
8800
8801
8802                                  ;
8803                                  ;THIS ROUTINE CHECKS RKMR3 (MSGB)-WORDS 0 & 1
8804                                  ;ENTER WITH SHOULD BE VALUE IN SBMR3
8805                                  ;RETURN IF NO COMPARE, RETURN +2 IF COMPARE
8806                                  ;
8807  052566  013746  004400          CKMR3:  MOV     TEMP1,-(SP)     ;SAV TEMP1
8808  052572  013737  004362  004400          MOV     HMR1,TEMP1
8809  052600  042737  177774  004400          BIC     #^C<M.ID>,TEMP1
8810  052606  053737  004400  004376          BIS     TEMP1,SBMR3     ;INSERT WORD #
8811  052614  013737  004376  004400          MOV     SBMR3,TEMP1
8812  052622  004737  052664                  JSR     PC,SBPAR        ;GET PARITY FOR SBMR3
8813  052626  013737  004400  004376          MOV     TEMP1,SBMR3     ;NOW HAS PARITY
8814  052634  023737  004376  004366          CMP     SBMR3,HMR3      ;SHOULD BE SAME
8815  052642  001005                          BNE     1$
8816  052644  012637  004400                  MOV     (SP)+,TEMP1     ;RESTOR TEMP1
8817  052650  062716  000002                  ADD     #2,(SP)         ;COMPARE OK, SKIP OVER ERROR.
8818  052654  000207                          RTS     PC
8819  052656  012637  004400          1$:     MOV     (SP)+,TEMP1     ;RESTOR TEMP1
```

```
 8820  052662  000207                        RTS     PC
 8821
 8822                                  ;
 8823                                  ;THIS ROUTINE GENERATES PARITY FOR SBMR2 AND SBMR3
 8824                                  ;ENTER WITH SBMR2 / SBMR3 IN TEMP1
 8825                                  ; TEMP1 IS ROTATED LEFT 17 TIMES. EACH TIME THE CARRY BIT IS SET,
 8826                                  ;R1 IS INCREMENTED. AT THE END OF 17 ROTATES ( TEMP1 BACK TO ORIG),
 8827                                  ;R1 BIT 0 IS EXAMINED. IF IT IS SET, INDICATING AN ODD # OF 1'S,
 8828                                  ; THE PARITY BIT IS NOT SET IN B .
 8829                                  ;IF IT IS NOT SET, INDICATING AN EVEN # OF 1'S ,THE PARITY BIT IS
 8830                                  ;SET IN TEMP1
 8831                                  ;
 8832  052664  010046                  $BPAR:  MOV     R0,-(SP)        ;SAVE R0
 8833  052666  010146                          MOV     R1,-(SP)        ;SAVE R1
 8834  052670  012700  000021                  MOV     #17.,R0         ;SHIFT COUNTER
 8835  052674  005001                          CLR     R1              ;COUNT # OF 1'S IN TEMP1
 8836  052676  000241                          CLC                     ;CLEAR CARRY
 8837
 8838  052700  006137  004400          1$:     ROL     TEMP1
 8839  052704  103001                          BCC     2$              ;BR IF CARRY CLEAR
 8840  052706  005201                          INC     R1              ;COUNT # OF 1'S
 8841  052710  005300                  2$:     DEC     R0              ;SHIFT COUNTER
 8842  052712  001372                          BNE     1$
 8843
 8844  052714  032701  000001                  BIT     #BIT0,R1
 8845  052720  001003                          BNE     3$              ;BR IF ODD # IN R0
 8846  052722  052737  100000  004400          BIS     #M.PAR,TEMP1    ;SET PARITY BIT
 8847  052730  012601                  3$:     MOV     (SP)+,R1        ;RESTORE R1
 8848  052732  012600                          MOV     (SP)+,R0        ;RESTORE R0
 8849  052734  000207                          RTS     PC
 8850
 8851                                  ;
 8852                                  ;ROUTINE TO ENABLE LOOPING ON INTERMITTANT ERRORS
 8853                                  ;WHEN $LPERR SET BY OTHER THAN SCOPE ROUTINE
 8854                                  ; IE: MY LOOP MACRO
 8855                                  ;
 8856  052736  032777  001000  126174  $COP1$: BIT     #SW9,@SWR       ;LOOP ON ERROR?
 8857  052744  001406                          BEQ     1$              ;BR IF NO
 8858  052746  105737  001103                  TSTB    $ERFLG          ;HAD ERROR?
 8859  052752  001403                          BEQ     1$              ;BR IF NO
 8860  052754  013716  001110                  MOV     $LPERR,(SP)
 8861  052760  000002                          RTI
 8862
 8863  052762  011637  001110          1$:     MOV     (SP),$LPERR     ;SET LOOP ADDR FOR TIGHT SCOPE LOOP
 8864  052766  000002                          RTI
 8865
 8866                                  ;
 8867                                  ;CHECK FOR SW14 (LOOP ON TEST) OR SW8 (LOOP ON SPECIFIC TEST)
 8868                                  ;
 8869                                  ;RETURN IF NEITHER SET
 8870                                  ;RETURN +2 IF EITHER SET
 8871                                  ;
 8872                                  ;THIS SUBROUTINE IS USED AT THE END OF ANY TEST THAT REQUIRES
 8873                                  ;RECONDITIONING OF THE DRIVE BEFORE LOOPING ON AN ERROR OR TEST
 8874                                  ;
 8875  052770  005037  001176          $WTST:  CLR     $ESCAPE
```

```
8876  052774  005037  001430                    CLR     LPFLG
8877  053000  032777  040000  126132            BIT     #SW14,@SWR      ;LOOP ON TEST?
8878  053006  001403                            BEQ     3$              ;BR IF NO
8879  053010  062716  000002            1$:     ADD     #2,(SP)
8880  053014  000207                    2$:     RTS     PC
8881
8882  053016  032777  000400  126114    3$:     BIT     #SW8,@SWR       ;LOOP ON SPECIFIC TEST?
8883  053024  001773                            BEQ     2$              ;BR IF NO
8884  053026  127737  126106  001102            CMPB    @SWR,$TSTNM     ;RIGHT TEST? SWR <7:0>
8885  053034  001765                            BEQ     1$              ;BR IF YES
8886  053036  000207                            RTS     PC
8887
8888
8889                                    .SBTTL  UNEXPECTED TIMEOUT HANDLER
8890
8891                                    ;
8892                                    ;THIS ROUTINE IS ENTERED IF THERE IS
8893                                    ;       A.  NON EXISTANT MEMORY (NO SSYN)
8894                                    ;       B.  BOUNDRY ERROR
8895                                    ;       C.  STACK OVERFLOW
8896                                    ;
8897
8898  053040  011600                    BADTMO: MOV     (SP),R0         ;SAVE PC WHERE TIMEOUT OCCURRED.
8899  053042  005740                            TST     -(R0)           ;GET PC BEFORE UPDATE
8900  053044  032777  020000  126066            BIT     #SW13,@SWR      ;INHIBIT ERR TYPOUT?
8901  053052  001005                            BNE     1$              ;YES, DON'T TYPE
8902  053054  104401  061436                    TYPE    ,EM3            ;ABORT TESTS,UNEXP T.O. @ PC=
8903  053060  010046                            MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
8904                                                                    ;;TYPE PC
8905  053062  104403                            TYPOS                   ;;GO TYPE--OCTAL ASCII
8906  053064     006                            .BYTE   6               ;;TYPE 6 DIGIT(S)
8907  053065     000                            .BYTE   0               ;;SUPPRESS LEADING ZEROS
8908  053066  032777  001000  126044    1$:     BIT     #SW9,@SWR       ;LOOP ON ERROR?
8909  053074  001403                            BEQ     2$              ;NO, BRANCH
8910  053076  022626                            CMP     (SP)+,(SP)+     ;YES, RESTORE STACK
8911  053100  000177  126002                    JMP     @$LPADR         ;GO TO STARTING ADDR OF TEST
8912                                                                    ;THAT GAVE BAD TIMEOUT
8913  053104  032777  040000  126026    2$:     BIT     #SW14,@SWR      ;LOOP ON TEST?
8914  053112  001401                            BEQ     3$              ;NO BRANCH
8915  053114  000002                            RTI                     ;YES
8916
8917  053116  000000                    3$:     HALT                    ;UNEXPECTED TIME OUT OCCURRED
8918                                                                    ;AS INDICATED. YOU CAN LOOP ON
8919                                                                    ;ERROR, LOOP ON TEST OR INHIBIT
8920                                                                    ;ERROR TYPEOUT BY SETTING THOSE
8921                                                                    ;SWITCHES.
8922                                                                    ;
8923  053120  022626                            CMP     (SP)+,(SP)+     ;RESTORE STACK
8924  053122  000137  047212                    JMP     $EOP1           ;ABORT TESTS
8925
8926                                    .SBTTL  UNEXPECTED INTERRUPT HANDLER
8927
8928                                    ;
8929                                    ;THIS ROUTINE CHECKS SW13 (INH ERR TYPOUT), SW9 (LOOP ON ERR)
8930                                    ;& SW14 (LOOP ON TEST).
8931                                    ;
```

```
8932
8933  053126  011600                  BADINT: MOV     (SP),R0         ;SAVE PC WHERE INT OCCURRED
8934  053130  005740                          TST     -(R0)           ;GET PC BEFORE UPDATE
8935  053132  032777  020000  126000          BIT     #SW13,@SWR      ;INHIBIT ERR TYPEOUT?
8936  053140  001005                          BNE     1$              ;YES, DONT TYPE
8937  053142  104401  061511                  TYPE    ,EM4            ;ABORT TESTS, UNEXP INT @ PC=
8938  053146  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
8939                                                                  ;;TYPE PC
8940  053150  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
8941  053152     006                          .BYTE   6               ;;TYPE 6 DIGIT(S)
8942  053153     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
8943
8944  053154  032777  001000  125756  1$:     BIT     #SW9,@SWR       ;LOOP ON ERROR?
8945  053162  001403                          BEQ     2$              ;NO, BRANCH
8946  053164  022626                          CMP     (SP)+,(SP)+     ;YES, RESTORE STACK
8947  053166  000177  125714                  JMP     @$LPADR         ;GO TO THE STARTING ADDR OF
8948                                                                  ;TEST THAT GAVE UNEXP. INT.
8949  053172  032777  040000  125740  2$:     BIT     #SW14,@SWR      ;LOOP ON TEST?
8950  053200  001401                          BEQ     3$              ;NO, BRANCH
8951  053202  000002                          RTI                     ;YES.
8952
8953  053204  000000                  3$:     HALT                    ;UNEXPECTED INTERRUPT OCCURRED AS
8954                                                                  ;INDICATED. YOU CAN LOOP ON ERROR,
8955                                                                  ;LOOP ON TEST OR INHIBIT
8956                                                                  ;ERROR TYPEOUT BY SETTING THOSE
8957                                                                  ;SWITCHES
8958                                                                  ;
8959  053206  022626                          CMP     (SP)+,(SP)+     ;RESTORE STACK
8960  053210  000137  047212                  JMP     $EOP1           ;ABORT TESTS
8961
8962                          .SBTTL  MEMORY CHECK ENABLE TRAP
8963
8964  053214  012737  053230  001176  MEMERR: MOV     #1$,$ESCAPE     ;LOAD ESCAPE
8965  053222  011637  001354                  MOV     (SP),TRAPPC     ;STORE PC
8966  053226  104236                          ERROR   236             ;UNEXP MEM PARITY TRAP
8967
8968  053230  005037  001176          1$:     CLR     $ESCAPE
8969  053234  032777  001000  125676          BIT     #SW9,@SWR       ;CHECK IF LOOP ON ERROR
8970  053242  001001                          BNE     2$              ;YES, FORCE STACK AND TRY AGAIN
8971  053244  000002                          RTI                     ;ELSE RETURN
8972
8973  053246  012706  001100          2$:     MOV     #STACK,SP       ;INIT STACK
8974  053252  000177  125632                  JMP     @$LPERR         ;LOOP ON ERROR
8975
8976                          .SBTTL  RK06 INTERRUPT HANDLER
8977
8978  053256  000240                  INTER:  NOP
8979  053260  000240                          NOP
8980  053262  000240                          NOP
8981  053264  011600                          MOV     (SP),R0         ;SAVE PC WHERE INT OCCURRED.
8982  053266  005740                          TST     -(R0)           ;GET PC BEFORE UPDATE.
8983  053270  104401  06023C                  TYPE    ,MSG6           ;INT AT PC=
8984  053274  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
8985                                                                  ;;TYPE PC
8986  053276  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
8987  053300     006                          .BYTE   6               ;;TYPE 6 DIGIT(S)
```

```
 8988  053301     000                        .BYTE   0               ;;SUPPRESS LEADING ZEROS
 8989  053302  000000                        HALT
 8990  053304  000240                        NOP
 8991  053306  000240                        NOP
 8992  053310  000002                        RTI
 8993
 8994                                .SBTTL  POWER DOWN AND UP ROUTINES
 8995
 8996                                ;POWER DOWN ROUTINE
 8997
 8998  053312  012737  053324  000024  $PWRDN: MOV     #$PWRUP,PWRVEC  ;SET UP VECTOR
 8999  053320  000000                        HALT
 9000  053322  000776                        BR      .-2             ;HANG UP.
 9001
 9002                                ;POWER UP ROUTINE
 9003
 9004  053324  005037  053376        $PWRUP: CLR     $PWRCT          ;WAIT LOOP FOR TTY
 9005  053330  005237  053376        1$:     INC     $PWRCT          ;WAIT FOR THE INCR
 9006  053334  001375                        BNE     1$              ;OF WORD
 9007  053336  012737  053312  000024        MOV     #$PWRDN,PWRVEC  ;SET POWER DOWN VECTOR
 9008  053344  012737  000340  000026        MOV     #PR7,PWRVEC+2   ;PRIORITY 7
 9009  053352  012737  000340  000036        MOV     #PR7,TRAPVEC+2  ;LOCKOUT ALL INTERRUPTS FOR TRAPS
 9010  053360  012706  001100                MOV     #STACK,SP       ;INITIALIZE STACK
 9011  053364  104401  060452                TYPE    ,MSG11          ;REPORT POWER FAIL
 9012  053370  000005                        RESET
 9013  053372  000137  012326                JMP     PFSRT
 9014
 9015  053376  000000        $PWRCT: 0                               ;WAIT COUNT FOR TTY
 9016
```

```
                                        .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
                                        ;*SW08=1         LOOP ON TEST IN SWR<7:0>
                                        ;*CALL
                                        ;*      SCOPE           ;;SCOPE=IOT

053400                                  $SCOPE:
053400  104407                                 CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
053402  032777  040000  125530          1$:    BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
053410  001114                                 BNE     $OVER           ;;YES IF SW14=1
                                        ;*****START OF CODE FOR THE XOR TESTER*****
053412  000416                          $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                        ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
053414  013746  000004                         MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
053420  012737  053440  000004                 MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
053426  005737  177060                        TST     @#177060        ;;TIME OUT ON XOR?
053432  012637  000004                         MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
053436  000463                                 BR      $SVLAD          ;;GO TO THE NEXT TEST
053440  022626                          5$:    CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
053442  012637  000004                         MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
053446  000423                                 BR      7$              ;;LOOP ON THE PRESENT TEST
053450                                  6$:;*****END OF CODE FOR THE XOR TESTER*****
053450  032777  000400  125462                 BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
053456  001404                                 BEQ     2$              ;;BR IF NO
053460  127737  125454  001102                 CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
053466  001465                                 BEQ     $OVER           ;;BR IF YES
053470  105737  001103          2$:    TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
053474  001421                                 BEQ     3$              ;;BR IF NO
053476  123737  001115  001103                 CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
053504  101015                                 BHI     3$              ;;BR IF NO
053506  032777  001000  125424                 BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
053514  001404                                 BEQ     4$              ;;BR IF NO
053516  013737  001110  001106          7$:    MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
053524  000446                                 BR      $OVER
053526  105037  001103          4$:    CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
053532  005037  001174                        CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
053536  000415                                 BR      1$              ;;ESCAPE TO THE NEXT TEST
053540  032777  004000  125372          3$:    BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
053546  001011                                 BNE     1$              ;;BR IF YES
053550  005737  001216                        TST     $PASS           ;;IF FIRST PASS OF PROGRAM
053554  001406                                 BEQ     1$              ;;      INHIBIT ITERATIONS
053556  005237  001104                        INC     $ICNT           ;;INCREMENT ITERATION COUNT
053562  023737  001174  001104                 CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
053570  002024                                 BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
053572  012737  000001  001104          1$:    MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
053600  013737  053656  001174                 MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
053606  105237  001102                  $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
053612  113737  001102  001214                 MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
```

```
 9073  053620  011637  001106                 MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
 9074  053624  011637  001110                 MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
 9075  053630  005037  001176                 CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
 9076  053634  112737  000001  001115         MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
 9077  053642  013777  001102  125272 $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
 9078  053650  013716  001106                 MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
 9079  053654  000002                         RTI                     ;;FIXES PS
 9080  053656  003720                 $MXCNT: 2000.                   ;;MAX. NUMBER OF ITERATIONS
 9081                                 .SBTTL  ERROR HANDLER ROUTINE
 9082
 9083                                 ;;**************************************************************
 9084                                 ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
 9085                                 ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
 9086                                 ;*AND GO TO TYPERR ON ERROR
 9087                                 ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
 9088                                 ;*SW15=1         HALT ON ERROR
 9089                                 ;*SW13=1         INHIBIT ERROR TYPEOUTS
 9090                                 ;*SW10=1         BELL ON ERROR
 9091                                 ;*SW09=1         LOOP ON ERROR
 9092                                 ;*CALL
 9093                                 ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
 9094
 9095  053660                         $ERROR:
 9096  053660  104407                         CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
 9097  053662  105237  001103         7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
 9098  053666  001775                         BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
 9099  053670  013777  001102  125244         MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
 9100  053676  032777  002000  125234         BIT     #BIT10,@SWR     ;;BELL ON ERROR?
 9101  053704  001402                         BEQ     1$              ;;NO - SKIP
 9102  053706  104401  001200                 TYPE    ,$BELL          ;;RING BELL
 9103  053712  005237  001112         1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
 9104  053716  011637  001116                 MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
 9105  053722  162737  000002  001116         SUB     #2,$ERRPC
 9106  053730  117737  125162  001114         MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
 9107  053736  032777  020000  125174         BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
 9108  053744  001004                         BNE     20$             ;;SKIP TYPEOUTS
 9109  053746  004737  072126                 JSR     PC,TYPERR       ;;GO TO USER ERROR ROUTINE
 9110  053752  104401  001205                 TYPE    ,$CRLF
 9111  053755                         20$:
 9112  053756  122737  000001  001230         CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
 9113  053764  001007                         BNE     2$              ;;NO,SKIP APT ERROR REPORT
 9114  053766  113737  001114  054000         MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
 9115  053774  004737  054604                 JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
 9116  054000     000                 21$:    .BYTE   0
 9117  054001     000                         .BYTE   0
 9118  054002  000777                 22$:    BR      22$             ;;APT ERROR LOOP
 9119  054004  005777  125130         2$:     TST     @SWR            ;;HALT ON ERROR
 9120  054010  100002                         BPL     3$              ;;SKIP IF CONTINUE
 9121  054012  000000                         HALT                    ;;HALT ON ERROR!
 9122  054014  104407                         CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
 9123  054016  032777  001000  125114 3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
 9124  054024  001402                         BEQ     4$              ;;BR IF NO
 9125  054026  013716  001110                 MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
 9126  054032  005737  001176         4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
 9127  054036  001402                         BEQ     5$              ;;BR IF NONE
 9128  054040  013716  001176                 MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
```

```
                                5$:
054044                          
054044  022737  047300  000042          CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
054052  001001                          BNE     6$              ;;BRANCH IF NO
054054  000000                          HALT                    ;;YES
054056                          6$:
054056  000002                          RTI                     ;;RETURN
                                .SBTTL  TYPE ROUTINE

                                ;;**************************************************************
                                ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
                                ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
                                ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
                                ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
                                ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
                                ;*
                                ;*CALL:
                                ;*1) USING A TRAP INSTRUCTION
                                ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                                ;*OR
                                ;*      TYPE
                                ;*      MESADR
                                ;*

054060  105737  001157          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
054064  100002                          BPL     1$              ;;BR IF YES
054066  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
054070  000430                          BR      3$              ;;LEAVE
054072  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
054074  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
054100  122737  000001  001230          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
054106  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
054110  132737  000100  001231          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
054116  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
054120  010037  054130                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
054124  004737  054574                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
054130  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
054132  132737  000040  001231  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
054140  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
054142  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
054144  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
054146  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
054150  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
054152  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
054156  000002                          RTI                     ;;RETURN
054160  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
054164  001430                          BEQ     8$
054166  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
054172  001006                          BNE     5$
054174  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
054176  104401                          TYPE                    ;;TYPE A CR AND LF
054200  001205                          $CRLF
054202  105037  054336                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
054206  000755                          BR      2$              ;;GET NEXT CHARACTER
054210  004737  054272          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
054214  123726  001156          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
054220  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
```

```
 9185  054222  013746  001154                MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
 9186                                                                  ;;AND THE NULL CHAR.
 9187  054226  105366  000001        7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
 9188  054232  002770                        BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
 9189  054234  004737  054272                JSR     PC,$TYPEC       ;;GO TYPE A NULL
 9190  054240  105337  054336                DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
 9191  054244  000770                        BR      7$              ;;LOOP
 9192
 9193                                  ;HORIZONTAL TAB PROCESSOR
 9194
 9195  054246  112716  000040        8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
 9196  054252  004737  054272        9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
 9197  054256  132737  000007  054336        BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
 9198  054264  001372                        BNE     9$              ;;TAB STOP
 9199  054266  005726                        TST     (SP)+           ;;POP SPACE OFF STACK
 9200  054270  000724                        BR      2$              ;;GET NEXT CHARACTER
 9201  054272  105777  124652        $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
 9202  054276  100375                        BPL     $TYPEC
 9203  054300  116677  000002  124644        MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
 9204  054306  122766  000015  000002        CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
 9205  054314  001003                        BNE     1$              ;;BRANCH IF NO
 9206  054316  105037  054336                CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
 9207  054322  000406                        BR      $TYPEX          ;;EXIT
 9208  054324  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
 9209  054332  001402                        BEQ     $TYPEX          ;;BRANCH IF YES
 9210  054334  105227                        INCB    (PC)+           ;;COUNT THE CHARACTER
 9211  054336  000000        $CHARCNT:.WORD  0                       ;;CHARACTER COUNT STORAGE
 9212  054340  000207        $TYPEX: RTS     PC
 9213
 9214                                  .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
 9215
 9216                                  ;;*****************************************************************
 9217                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
 9218                                  ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
 9219                                  ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
 9220                                  ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
 9221                                  ;*REPLACED WITH SPACES.
 9222                                  ;*CALL:
 9223                                  ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
 9224                                  ;*      TYPDS                   ;;GO TO THE ROUTINE
 9225
 9226  054342                        $TYPDS:
 9227  054342  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 9228  054344  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 9229  054346  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 9230  054350  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 9231  054352  010546                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 9232  054354  012746  020200                MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
 9233  054360  016605  000020                MOV     20(SP),R5       ;;GET THE INPUT NUMBER
 9234  054364  100004                        BPL     1$              ;;BR IF INPUT IS POS.
 9235  054366  005405                        NEG     R5              ;;MAKE THE BINARY NUMBER POS.
 9236  054370  112766  000055  000001        MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
 9237  054376  005000                1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
 9238  054400  012703  054556                MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
 9239  054404  112723  000040                MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
 9240  054410  005002                2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
```

```
9241  054412  016001  054546                  MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
9242  054416  160105                  3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
9243  054420  002402                          BLT     4$              ;;BR IF DONE
9244  054422  005202                          INC     R2              ;;INCREASE THE BCD DIGIT BY 1
9245  054424  000774                          BR      3$
9246  054426  060105                  4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
9247  054430  005702                          TST     R2              ;;CHECK IF BCD DIGIT=0
9248  054432  001002                          BNE     5$              ;;FALL THROUGH IF 0
9249  054434  105716                          TSTB    (SP)            ;;STILL DOING LEADING 0'S?
9250  054436  100407                          BMI     7$              ;;BR IF YES
9251  054440  106316                  5$:     ASLB    (SP)            ;;MSD?
9252  054442  103003                          BCC     6$              ;;BR IF NO
9253  054444  116663  000001  177777          MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
9254  054452  052702  000060          6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
9255  054456  052702  000040          7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
9256  054462  110223                          MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
9257  054464  005720                          TST     (R0)+           ;;JUST INCREMENTING
9258  054466  020027  000010                  CMP     R0,#10          ;;CHECK THE TABLE INDEX
9259  054472  002746                          BLT     2$              ;;GO DO THE NEXT DIGIT
9260  054474  003002                          BGT     8$              ;;GO TO EXIT
9261  054476  010502                          MOV     R5,R2           ;;GET THE LSD
9262  054500  000764                          BR      6$              ;;GO CHANGE TO ASCII
9263  054502  105726                  8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
9264  054504  100003                          BPL     9$              ;;BR IF NO
9265  054506  116663  177777  177776          MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
9266  054514  105013                  9$:     CLRB    (R3)            ;;SET THE TERMINATOR
9267  054516  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
9268  054520  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
9269  054522  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
9270  054524  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
9271  054526  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
9272  054530  104401  054556                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
9273  054534  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
9274  054542  012616                          MOV     (SP)+,(SP)
9275  054544  000002                          RTI                     ;;RETURN TO USER
9276  054546  023420                  $DTBL:  10000.
9277  054550  001750                          1000.
9278  054552  000144                          100.
9279  054554  000012                          10.
9280  054556  000004                  $DBLK:  .BLKW   4
9281                                  .SBTTL  APT COMMUNICATIONS ROUTINE
9282
9283                                  ;;**************************************************************
9284  054566  112737  000001  055032  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
9285  054574  112737  000001  055030  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
9286  054602  000403                          BR      $ATYC
9287  054604  112737  000001  055032  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
9288  054612                          $ATYC:
9289  054612  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
9290  054614  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
9291  054616  105737  055030                  TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
9292  054622  001450                          BEQ     5$              ;;IF NOT:  BR
9293  054624  122737  000001  001230          CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
9294  054632  001031                          BNE     3$              ;;IF NOT:  BR
9295  054634  132737  000100  001231          BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
9296  054642  001425                          BEQ     3$              ;;IF NOT:  BR
```

```
 9297  054644  017600  000004              MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
 9298  054650  062766  000002  000004      ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
 9299  054656  005737  001210          1$: TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
 9300  054662  001375                      BNE     1$              ;;IF NOT:  WAIT
 9301  054664  010037  001224              MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
 9302  054670  105720                  2$: TSTB    (R0)+           ;;FIND END OF MESSAGE
 9303  054672  001376                      BNE     2$
 9304  054674  163700  001224              SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
 9305  054700  006200                      ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
 9306  054702  010037  001226              MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
 9307  054706  012737  000004  001210      MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
 9308  054714  000413                      BR      5$
 9309  054716  017637  000004  054742  3$: MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
 9310  054724  062766  000002  000004      ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
 9311  054732  013746  177776              MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
 9312  054736  004737  054060              JSR     PC,$TYPE        ;;CALL TYPE MACRO
 9313  054742  000000                  4$: .WORD   0
 9314  054744                          5$:
 9315  054744  105737  055032          10$: TSTB   $FFLG           ;;SHOULD REPORT FATAL ERROR?
 9316  054750  001416                      BEQ     12$             ;;IF NOT:  BR
 9317  054752  005737  001230              TST     $ENV            ;;RUNNING UNDER APT?
 9318  054756  001413                      BEQ     12$             ;;IF NOT:  BR
 9319  054760  005737  001210          11$: TST    $MSGTYPE        ;;FINISHED LAST MESSAGE?
 9320  054764  001375                      BNE     11$             ;;IF NOT:  WAIT
 9321  054766  017637  000004  001212      MOV     @4(SP),$FATAL   ;;GET ERROR #
 9322  054774  062766  000002  000004      ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
 9323  055002  005237  001210              INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
 9324  055006  105037  055032          12$: CLRB   $FFLG           ;;CLEAR FATAL FLAG
 9325  055012  105037  055031              CLRB    $LFLG           ;;CLEAR LOG FLAG
 9326  055016  105037  055030              CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
 9327  055022  012601                      MOV     (SP)+,R1        ;;POP STACK INTO R1
 9328  055024  012600                      MOV     (SP)+,R0        ;;POP STACK INTO R0
 9329  055026  000207                      RTS     PC              ;;RETURN
 9330  055030     000              $MFLG: .BYTE   0               ;;MESSG. FLAG
 9331  055031     000              $LFLG: .BYTE   0               ;;LOG FLAG
 9332  055032     000              $FFLG: .BYTE   0               ;;FATAL FLAG
 9333        055034                      .EVEN
 9334        000200              APTSIZE=200
 9335        000001              APTENV=001
 9336        000100              APTSPOOL=100
 9337        000040              APTCSUP=040
 9338                            .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
 9339
 9340                            ;;**************************************************************
 9341                            ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
 9342                            ;*OCTAL (ASCII) NUMBER AND TYPE IT.
 9343                            ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
 9344                            ;*CALL:
 9345                            ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 9346                            ;*      TYPOS                   ;;CALL FOR TYPEOUT
 9347                            ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
 9348                            ;*      .BYTE   M               ;;M=1 OR 0
 9349                            ;*                                      ;;1=TYPE LEADING ZEROS
 9350                            ;*                                      ;;0=SUPPRESS LEADING ZEROS
 9351                            ;*
 9352                            ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
```

```
                                        ;*$TYPOS OR $TYPOC
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                        ;*      TYPON                   ;;CALL FOR TYPEOUT
                                        ;*
                                        ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                        ;*      TYPOC                   ;;CALL FOR TYPEOUT

055034  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
055040  116637  000001  055257          MOVB    1(SP),$CFILL    ;;LOAD ZERO FILL SWITCH
055046  112637  055261                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
055052  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
055056  000406                          BR      $TYPON
055060  112737  000001  055257  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
055066  112737  000006  055261          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
055074  112737  000005  055256  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
055102  010346                          MOV     R3,-(SP)        ;;SAVE R3
055104  010446                          MOV     R4,-(SP)        ;;SAVE R4
055106  010546                          MOV     R5,-(SP)        ;;SAVE R5
055110  113704  055261                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
055114  005404                          NEG     R4
055116  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
055122  110437  055260                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
055126  113704  055257                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
055132  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
055136  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
055140  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
055142  000404                          BR      3$              ;;GO DO MSB
055144  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
055146  006105                          ROL     R5
055150  006105                          ROL     R5
055152  010503                          MOV     R5,R3
055154  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
055156  105337  055260                  DECB    $OMODE          ;;TYPE THIS DIGIT?
055162  100016                          BPL     7$              ;;BR IF NO
055164  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
055170  001002                          BNE     4$              ;;TEST FOR 0
055172  005704                          TST     R4              ;;SUPPRESS THIS 0?
055174  001403                          BEQ     5$              ;;BR IF YES
055176  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
055200  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
055204  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
055210  110337  055254                  MOVB    R3,8$           ;;SAVE FOR TYPING
055214  104401  055254                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
055220  105337  055256          7$:     DECB    $OCNT           ;;COUNT BY 1
055224  003347                          BGT     2$              ;;BR IF MORE TO DO
055226  002402                          BLT     6$              ;;BR IF DONE
055230  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
055232  000744                          BR      2$              ;;GO DO THE LAST DIGIT
055234  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
055236  012604                          MOV     (SP)+,R4        ;;RESTORE R4
055240  012603                          MOV     (SP)+,R3        ;;RESTORE R3
055242  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
055250  012616                          MOV     (SP)+,(SP)
```

```
055252  000002                          RTI                             ;;RETURN
055254     000                  8$:     .BYTE   0                       ;;STORAGE FOR ASCII DIGIT
055255     000                          .BYTE   0                       ;;TERMINATOR FOR TYPE ROUTINE
055256     000                  $OCNT:  .BYTE   0                       ;;OCTAL DIGIT COUNTER
055257     000                  $OFILL: .BYTE   0                       ;;ZERO FILL SWITCH
055260  000000                  $OMODE: .WORD   0                       ;;NUMBER OF DIGITS TO TYPE
                                .SBTTL  TTY INPUT ROUTINE

                                ;;*****************************************************************
                                .ENABL  LSB

                                ;;*****************************************************************
                                ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
                                ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
                                ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
                                ;*WHEN OPERATING IN TTY FLAG MODE.
055262  022737  000176  001140  $CKSWR: CMP     #SWREG,SWR              ;;IS THE SOFT-SWR SELECTED?
055270  001074                          BNE     15$                     ;;BRANCH IF NO
055272  105777  123646                  TSTB    @$TKS                   ;;CHAR THERE?
055276  100071                          BPL     15$                     ;;IF NO, DON'T WAIT AROUND
055300  117746  123642                  MOVB    @$TKB,-(SP)             ;;SAVE THE CHAR
055304  042716  177600                  BIC     #^C177,(SP)             ;;STRIP-OFF THE ASCII
055310  022726  000007                  CMP     #7,(SP)+                ;;IS IT A CONTROL G?
055314  001062                          BNE     15$                     ;;NO, RETURN TO USER
055316  123727  001134  000001          CMPB    $AUTOB,#1               ;;ARE WE RUNNING IN AUTO-MODE?
055324  001456                          BEQ     15$                     ;;BRANCH IF YES

055326  104401  056147                  TYPE    ,$CNTLG                 ;;ECHO THE CONTROL-G (^G)
055332  104401  056154          $GTSWR: TYPE    ,$MSWR                  ;;TYPE CURRENT CONTENTS
055336  013746  000176                  MOV     SWREG,-(SP)             ;;SAVE SWREG FOR TYPEOUT
055342  104402                          TYPOC                           ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
055344  104401  056165                  TYPE    ,$MNEW                  ;;PROMPT FOR NEW SWR
055350  005046                  19$:    CLR     -(SP)                   ;;CLEAR COUNTER
055352  005046                          CLR     -(SP)                   ;;THE NEW SWR
055354  105777  123564          7$:     TSTB    @$TKS                   ;;CHAR THERE?
055360  100375                          BPL     7$                      ;;IF NOT TRY AGAIN

055362  117746  123560                  MOVB    @$TKB,-(SP)             ;;PICK UP CHAR
055366  042716  177600                  BIC     #^C177,(SP)             ;;MAKE IT 7-BIT ASCII



055372  021627  000025          9$:     CMP     (SP),#25                ;;IS IT A CONTROL-U?
055376  001005                          BNE     10$                     ;;BRANCH IF NOT
055400  104401  056142                  TYPE    ,$CNTLU                 ;;YES, ECHO CONTROL-U (^U)
055404  062706  000006          20$:    ADD     #6,SP                   ;;IGNORE PREVIOUS INPUT
055410  000757                          BR      19$                     ;;LET'S TRY IT AGAIN


055412  021627  000015          10$:    CMP     (SP),#15                ;;IS IT A <CR>?
055416  001022                          BNE     16$                     ;;BRANCH IF NO
055420  005766  000004                  TST     4(SP)                   ;;YES, IS IT THE FIRST CHAR?
055424  001403                          BEQ     11$                     ;;BRANCH IF YES
055426  016677  000002  123504          MOV     2(SP),@SWR              ;;SAVE NEW SWR
055434  062706  000006          11$:    ADD     #6,SP                   ;;CLEAR UP STACK
055440  104401  001205          14$:    TYPE    ,$CRLF                  ;;ECHO <CR> AND <LF>
```

```
 9465  055444  123727  001135  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
 9466  055452  001003                          BNE     15$             ;;BRANCH IF NOT
 9467  055454  012777  000100  123462          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
 9468  055462  000002                  15$:    RTI                     ;;RETURN
 9469  055464  004737  054272          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
 9470  055470  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
 9471  055474  002420                          BLT     18$             ;;BRANCH IF YES
 9472  055476  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
 9473  055502  003015                          BGT     18$             ;;BRANCH IF YES
 9474  055504  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
 9475  055510  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
 9476  055514  001403                          BEQ     17$             ;;BRANCH IF YES
 9477  055516  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
 9478  055520  006316                          ASL     (SP)            ;;   CHAR OVER TO MAKE
 9479  055522  006316                          ASL     (SP)            ;;   ROOM FOR NEW ONE.
 9480  055524  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
 9481  055530  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
 9482  055534  000707                          BR      7$              ;;GET THE NEXT ONE
 9483  055536  104401  001204          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
 9484  055542  000720                          BR      20$             ;;SIMULATE CONTROL-U
 9485                                  .DSABL  LSB
 9486
 9487
 9488                                  ;;**************************************************************
 9489                                  ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
 9490                                  ;*CALL:
 9491                                  ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
 9492                                  ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
 9493                                  ;*                              ;;WITH PARITY BIT STRIPPED OFF
 9494                                  ;
 9495
 9496  055544  011646          $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
 9497  055546  016666  000004  000002          MOV     4(SP),2(SP)     ;;SAVE THE PS
 9498  055554  105777  123364          1$:     TSTB    @$TKS           ;;WAIT FOR
 9499  055560  100375                          BPL     1$              ;;A CHARACTER
 9500  055562  117766  123360  000004          MOVB    @$TKB,4(SP)     ;;READ THE TTY
 9501  055570  042766  177600  000004          BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
 9502  055576  026627  000004  000023          CMP     4(SP),#23       ;;IS IT A CONTROL-S?
 9503  055604  001013                          BNE     3$              ;;BRANCH IF NO
 9504  055606  105777  123332          2$:     TSTB    @$TKS           ;;WAIT FOR A CHARACTER
 9505  055612  100375                          BPL     2$              ;;LOOP UNTIL ITS THERE
 9506  055614  117746  123326                  MOVB    @$TKB,-(SP)     ;;GET CHARACTER
 9507  055620  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
 9508  055624  022627  000021                  CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
 9509  055630  001366                          BNE     2$              ;;IF NOT DISCARD IT
 9510  055632  000750                          BR      1$              ;;YES, RESUME
 9511  055634  026627  000004  000140  3$:     CMP     4(SP),#140      ;;IS IT UPPER CASE?
 9512  055642  002407                          BLT     4$              ;;BRANCH IF YES
 9513  055644  026627  000004  000175          CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
 9514  055652  003003                          BGT     4$              ;;BRANCH IF YES
 9515  055654  042766  000040  000004          BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
 9516  055662  000002                  4$:     RTI                     ;;GO BACK TO USER
 9517                                  ;;**************************************************************
 9518                                  ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
 9519                                  ;*CALL:
 9520                                  ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
```

```
                                        ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
                                        ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
                                        $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
055664  010346                                CLR     -(SP)           ;;CLEAR THE RUBOUT KEY
055666  005046                        1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
055670  012703  056120                2$:     CMP     #$TTYIN+22,R3   ;;BUFFER FULL?
055674  022703  056142                        BLOS    4$              ;;BR IF YES
055700  101456                              RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
055702  104410                              MOVB    (SP)+,(R3)      ;;GET CHARACTER
055704  112613                      10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
055706  122713  000177                      BNE     5$              ;;BR IF NO
055712  001022                              TST     (SP)            ;;IS THIS THE FIRST RUBOUT?
055714  005716                              BNE     6$              ;;BR IF NO
055716  001007                              MOVB    #'\,9$          ;;TYPE A BACK SLASH
055720  112737  000134  056116              TYPE    ,9$
055726  104401  056116                      MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
055732  012716  177777              6$:     DEC     R3              ;;BACKUP BY ONE
055736  005303                              CMP     R3,#$TTYIN      ;;STACK EMPTY?
055740  020327  056120                      BLO     4$              ;;BR IF YES
055744  103434                              MOVB    (R3),9$         ;;SETUP TO TYPEOUT THE DELETED CHAR.
055746  111337  056116                      TYPE    ,9$             ;;GO TYPE
055752  104401  056116                      BR      2$              ;;GO READ ANOTHER CHAR.
055756  000746                      5$:     TST     (SP)            ;;RUBOUT KEY SET?
055760  005716                              BEQ     7$              ;;BR IF NO
055762  001406                              MOVB    #'\,9$          ;;TYPE A BACK SLASH
055764  112737  000134  056116              TYPE    ,9$
055772  104401  056116                      CLR     (SP)            ;;CLEAR THE RUBOUT KEY
055776  005016                      7$:     CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
056000  122713  000025                      BNE     8$              ;;BR IF NO
056004  001003                              TYPE    ,$CNTLU         ;;TYPE A CONTROL "U"
056006  104401  056142                      BR      1$              ;;GO START OVER
056012  000726                      8$:     CMPB    #22,(R3)        ;;IS CHARACTER A "↑R"?
056014  122713  000022                      BNE     3$              ;;BRANCH IF NO
056020  001011                              CLRB    (R3)            ;;CLEAR THE CHARACTER
056022  105013                              TYPE    ,$CRLF          ;;TYPE A "CR" & "LF"
056024  104401  001205                      TYPE    ,$TTYIN         ;;TYPE THE INPUT STRING
056030  104401  056120                      BR      2$              ;;GO PICKUP ANOTHER CHACTER
056034  000717                      4$:     TYPE    ,$QUES          ;;TYPE A '?'
056036  104401  001204                      BR      1$              ;;CLEAR THE BUFFER AND LOOP
056042  000712                      3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
056044  111337  056116                      TYPE    ,9$
056050  104401  056116                      CMPB    #15,(R3)+       ;;CHECK FOR RETURN
056054  122723  000015                      BNE     2$              ;;LOOP IF NOT RETURN
056060  001305                              CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
056062  105063  177777                      TYPE    ,$LF            ;;TYPE A LINE FEED
056066  104401  001206                      TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
056072  005726                              MOV     (SP)+,R3        ;;RESTORE R3
056074  012603                              MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
056076  011646                              MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
056100  016666  000004  000002              MOV     #$TTYIN,4(SP)
056106  012766  056120  000004              RTI                     ;;RETURN
056114  000002                      9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
056116     000                              .BYTE   0               ;;TERMINATOR
056117     000                      $TTYIN: .BLKB   22              ;;RESERVE 22 BYTES FOR TTY INPUT
056120  000022                      $CNTLU: .ASCIZ  /↑U/<15><12>    ;;CONTROL "U"
056142  052536  005015     000
```

```
                                .SBTTL  READ AN OCTAL NUMBER FROM THE TTY

                                ;;*****************************************************************
                                ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
                                ;*CHANGE IT TO BINARY.
                                ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
                                ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
                                ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
                                ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
                                ;*CALL:
                                ;*      RDOCT                   ;;READ AN OCTAL NUMBER
                                ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
                                ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT

056176  011646                  $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
056200  016666  000004  000002          MOV     4(SP),2(SP)     ;;INPUT NUMBER
056206  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
056210  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
056212  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
056214  104411                  1$:     RDLIN                   ;;READ AN ASCIZ LINE
056216  012600                          MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
056220  010037  056324                  MOV     R0,5$           ;;AND SAVE IT
056224  005001                          CLR     R1              ;;CLEAR DATA WORD
056226  005002                          CLR     R2
056230  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
056232  001420                          BEQ     3$              ;;IF ZERO GET OUT
056234  122716  000060                  CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
056240  003026                          BGT     4$              ;;IS AN OCTAL DIGIT
056242  122716  000067                  CMPB    #'7,(SP)
056246  002423                          BLT     4$
056250  006301                          ASL     R1              ;;*2
056252  006102                          ROL     R2
056254  006301                          ASL     R1              ;;*4
056256  006102                          ROL     R2
056260  006301                          ASL     R1              ;;*8
056262  006102                          ROL     R2
056264  042716  177770                  BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
056270  062601                          ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
056272  000756                          BR      2$              ;;LOOP
056274  005726                  3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
056276  010166  000012                  MOV     R1,12(SP)       ;;SAVE THE RESULT
056302  010237  056334                  MOV     R2,$HIOCT
056306  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
056310  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
056312  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
056314  000002                          RTI                     ;;RETURN
056316  005726                  4$:     TST     (SP)+           ;;CLEAN PARTIAL FROM STACK
056320  105010                          CLRB    (R0)            ;;SET A TERMINATOR
056322  104401                          TYPE                    ;;TYPE UP THRU THE BAD CHAR.
056324  000000                  5$:     .WORD   0
056326  104401  001204                  TYPE    ,$QUES          ;;"?" "CR" & "LF"
```

```
 9633  056332  000730                          BR      1$              ;;TRY AGAIN
 9634  056334  000000                  $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
 9635                                  .SBTTL  DOUBLE LENGTH BINARY TO OCTAL ASCII CONVERT ROUTINE
 9636
 9637                                  ;;*****************************************************************
 9638                                  ;*THIS ROUTINE WILL CONVERT A 32-BIT UNSIGNED BINARY NUMBER TO AN
 9639                                  ;*UNSIGNED OCTAL ASCIZ NUMBER.
 9640                                  ;*CALL
 9641                                  ;*      MOV     #PNTR,-(SP)     ;;POINTER TO LOW WORD OF BINARY NUMBER
 9642                                  ;*      JSR     PC,@#$DB2O      ;;CALL THE ROUTINE
 9643                                  ;*      RETURN                  ;;THE ADDRESS OF THE FIRST ASCIZ CHAR. IS ON THE STACK
 9644
 9645
 9646  056336  104413                  $DB2O:  SAVREG                  ;;SAVE ALL REGISTERS
 9647  056340  016601  000002                  MOV     2(SP),R1        ;;PICKUP THE POINTER TO LOW WORD
 9648  056344  012705  056455                  MOV     #$OCTVL+13.,R5  ;;POINTER TO DATA TABLE
 9649  056350  012704  000014                  MOV     #12.,R4         ;;DO ELEVEN CHARACTERS
 9650  056354  012703  177770                  MOV     #^C7,R3         ;;MASK
 9651  056360  012100                          MOV     (R1)+,R0        ;;LOWER WORD
 9652  056362  012101                          MOV     (R1)+,R1        ;;HIGH WORD
 9653  056364  005002                          CLR     R2              ;;TERMINATOR
 9654  056366  110245                  1$:     MOVB    R2,-(R5)        ;;PUT CHARACTER IN DATA TABLE
 9655  056370  010002                          MOV     R0,R2           ;;GET THIS DIGIT
 9656  056372  005304                          DEC     R4              ;;COUNT THIS CHARACTER
 9657  056374  003007                          BGT     3$              ;;BR IF NOT THE LAST DIGIT
 9658  056376  001405                          BEQ     2$              ;;BR IF IT IS THE LAST DIGIT
 9659  056400  005205                          INC     R5              ;;ALL DIGITS DONE-ADJUST POINTER FOR FIRST
 9660  056402  010566  000002                  MOV     R5,2(SP)        ;;ASCIZ CHAR. & PUT IT ON THE STACK
 9661  056406  104414                          RESREG                  ;;RESTORE ALL REGISTERS
 9662  056410  000207                          RTS     PC              ;;RETURN TO USER
 9663  056412  006203                  2$:     ASR     R3              ;;POSITION THE MASK FOR THE LAST DIGIT
 9664  056414  006001                  3$:     ROR     R1              ;;POSITION THE BINARY NUMBER FOR
 9665  056416  006000                          ROR     R0              ;;      THE NEXT OCTAL DIGIT
 9666  056420  006001                          ROR     R1
 9667  056422  006000                          ROR     R0
 9668  056424  006001                          ROR     R1
 9669  056426  006000                          ROR     R0
 9670  056430  040302                          BIC     R3,R2           ;;MASK OUT ALL JUNK
 9671  056432  062702  000060                  ADD     #'0,R2          ;;MAKE THIS CHAR. ASCII
 9672  056436  000753                          BR      1$              ;;GO PUT IT IN THE DATA TABLE
 9673  056440  000016                  $OCTVL: .BLKB   14.             ;;RESERVE DATA TABLE
 9674                                  .SBTTL  SINGLE LENGTH BINARY TO DECIMAL ASCIZ ROUTINE
 9675
 9676                                  ;;*****************************************************************
 9677                                  ;*THIS ROUTINE WILL CONVERT A 16-BIT UNSIGNED BINARY NUMBER TO AN
 9678                                  ;*UNSIGNED DECIMAL ASCIZ NUMBER.
 9679                                  ;*CALL
 9680                                  ;*      MOV     NUMBER,-(SP)    ;;PUT BINARY NUMBER ON THE STACK
 9681                                  ;*      JSR     PC,@#$SB2D      ;;CALL
 9682                                  ;*      RETURN                  ;;ADDRESS OF THE 1ST ASCIZ CHAR.IS ON THE STACK
 9683
 9684
 9685  056456  016637  000002  056506  $SB2D:  MOV     2(SP),1$        ;;SAVE BINARY NUMBER
 9686  056464  012746  056506                  MOV     #1$,-(SP)       ;;SET POINTER
 9687  056470  004737  056512                  JSR     PC,@#$DB2D      ;;CALL DOUBLE LENGTH CONVERT
 9688  056474  062716  000005                  ADD     #5,(SP)         ;;ONLY ALLOW FIVE CHARACTERS
```

```
 9689  056500  012666  000002                  MOV     (SP)+,2(SP)     ;;PICKUP POINTER
 9690  056504  000207                          RTS     PC              ;;RETURN
 9691  056506  000000  000000          1$:     .WORD   0,0
 9692                                  .SBTTL  DOUBLE LENGTH BINARY TO DECIMAL ASCII CONVERT ROUTINE
 9693
 9694                                  ;;*****************************************************************
 9695                                  ;*THIS ROUTINE WILL CONVERT A 32-BIT BINARY NUMBER TO AN UNSIGNED
 9696                                  ;*DECIMAL (ASCII) NUMBER. THE SIGN OF THE BINARY NUMBER MUST BE
 9697                                  ;*POSITIVE.
 9698                                  ;*CALL
 9699                                  ;*      MOV     #PNTR,-(SP)     ;;POINTER TO LOW WORD OF BINARY NUMBER
 9700                                  ;*      JSR     PC,@#$DB2D
 9701                                  ;*      RETURN                  ;;THE FIRST ADDRESS OF ASCIZ
 9702                                                                  ;;IS ON THE STACK
 9703
 9704
 9705  056512  104413                  $DB2D:  SAVREG                  ;;SAVE REGISTERS
 9706  056514  016602  000002                  MOV     2(SP),R2        ;;PICKUP THE DATA POINTER
 9707  056520  012700  056672                  MOV     #$DECVL,R0      ;;GET ADDRESS OF "$DECVL" STRING
 9708  056524  010066  000002                  MOV     R0,2(SP)        ;;PUT ACDRESS OF ASCIZ STRING ON STACK
 9709  056530  012201                          MOV     (R2)+,R1        ;;PICKUP THE BINARY NUMBER
 9710  056532  012202                          MOV     (R2)+,R2
 9711  056534  012737  000012  056610          MOV     #10.,4$         ;;SET UP TO DO 10 CONVERSIONS
 9712  056542  012704  056622                  MOV     #$TNPWR,R4      ;;ADDRESS OF TEN POWER
 9713  056546  012705  056624                  MOV     #$TNPWR+2,R5
 9714  056552  005003                  1$:     CLR     R3              ;;CLEAR PARTIAL
 9715  056554  161401                  2$:     SUB     (R4),R1         ;;SUBTRACT TEN POWER
 9716  056556  005602                          SBC     R2
 9717  056560  161502                          SUB     (R5),R2
 9718  056562  002402                          BLT     3$              ;;BR IF TEN POWER TO LARGE
 9719  056564  005203                          INC     R3              ;;ADD 1 TO PARTIAL
 9720  056566  000772                          BR      2$              ;;LOOP
 9721  056570  062401                  3$:     ADD     (R4)+,R1        ;;RESTORE SUBTRACTED VALUE
 9722  056572  005502                          ADC     R2
 9723  056574  062402                          ADD     (R4)+,R2
 9724  056576  022525                          CMP     (R5)+,(R5)+     ;;MOVE TO NEXT TEN POWER
 9725  056600  052703  000060                  BIS     #'0,R3          ;;CHANGE PARTIAL TO ASCII
 9726  056604  110320                          MOVB    R3,(R0)+        ;;SAVE IT
 9727  056606  005327                          DEC     (PC)+           ;;DONE?
 9728  056610  000000                  4$:     .WORD   0
 9729  056612  001357                          BNE     1$              ;;BR IF NO
 9730  056614  105020                          CLRB    (R0)+           ;;TERMINATOR
 9731  056616  104414                          RESREG                  ;;RESTORE REGISTERS
 9732  056620  000207                          RTS     PC              ;;RETURN
 9733  056622  145000                  $TNPWR: 145000                  ;;1.0E09
 9734  056624  035632                          35632
 9735  056626  160400                          160400                  ;;1.0E08
 9736  056630  002765                          2765
 9737  056632  113200                          113200                  ;;1.0E07
 9738  056634  000230                          230
 9739  056636  041100                          041100                  ;;1.0E06
 9740  056640  000017                          17
 9741  056642  103240                          103240                  ;;1.0E05
 9742  056644  000001                          1
 9743  056646  023420                          23420                   ;;1.0E04
 9744  056650  000000                          0
```

```
 9745  056652  001750                        1750                    ;;1.0E03
 9746  056654  000000                        0
 9747  056656  000144                        144                     ;;1.0E02
 9748  056660  000000                        0
 9749  056662  000012                        12                      ;;1.0E01
 9750  056664  000000                        0
 9751  056666  000001                        1                       ;;1.0E00
 9752  056670  000000                        0
 9753  056672  000014              $DECVL: .BLKB   12.               ;;RESERVE STORAGE FOR ASCIZ STRING
 9754                              .SBTTL  TYPE NUMERICAL ASCIZ STRING SUPPRESS LEADING ZEROS
 9755
 9756                              ;;**************************************************************
 9757                              ;*THIS ROUTINE IS USED TO TYPE AN ASCIZ NUMBER SUPPRESSING THE
 9758                              ;*LEADING NUMBERS.
 9759                              ;*CALL
 9760                              ;*      MOV     #NUMADR,-(SP)   ;;FIRST ADDRESS OF ASCIZ STRING
 9761                              ;*      JSR     PC,@#$SUPRS
 9762
 9763
 9764  056706  010046              $SUPRS: MOV     R0,-(SP)        ;;SAVE R0
 9765  056710  016600  000004              MOV     4(SP),R0        ;;PICKUP THE POINTER
 9766  056714  105710              1$:     TSTB    (R0)            ;;TERMINATEOR?
 9767  056716  001403                      BEQ     2$              ;;BR IF YES
 9768  056720  122720  000060              CMPB    #'0,(R0)+       ;;IS THIS AN ASCII "0" ?
 9769  056724  001773                      BEQ     1$              ;;BR IF YES
 9770  056726  005300              2$:     DEC     R0              ;;BACKUP BY "1"
 9771  056730  010037  056736              MOV     R0,3$           ;;SAVE FOR TYPING
 9772  056734  104401                      TYPE                    ;;GO TYPE
 9773  056736  000000              3$:     .WORD   0               ;;ASCIZ POINTER GOES HERE
 9774  056740  012600                      MOV     (SP)+,R0        ;;RESTORE R0
 9775  056742  012616                      MOV     (SP)+,(SP)      ;;RESTORE THE STACK
 9776  056744  000207                      RTS     PC              ;;RETURN
 9777                              .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES
 9778
 9779                              ;;**************************************************************
 9780                              ;*SAVE R0-R5
 9781                              ;*CALL:
 9782                              ;*      SAVREG
 9783                              ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
 9784                              ;*
 9785                              ;*TOP---(+16)
 9786                              ;* +2---(+18)
 9787                              ;* +4---R5
 9788                              ;* +6---R4
 9789                              ;* +8---R3
 9790                              ;*+10---R2
 9791                              ;*+12---R1
 9792                              ;*+14---R0
 9793
 9794  056746              $SAVREG:
 9795  056746  010046                      MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 9796  056750  010146                      MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 9797  056752  010246                      MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 9798  056754  010346                      MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 9799  056756  010446                      MOV     R4,-(SP)        ;;PUSH R4 ON STACK
 9800  056760  010546                      MOV     R5,-(SP)        ;;PUSH R5 ON STACK
```

```
 9801  056762  016646  000022                MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
 9802  056766  016646  000022                MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
 9803  056772  016646  000022                MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
 9804  056776  016646  000022                MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
 9805  057002  000002                        RTI
 9806
 9807                                  ;*RESTORE R0-R5
 9808                                  ;*CALL:
 9809                                  ;*      RESREG
 9810  057004                          $RESREG:
 9811  057004  012666  000022                MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
 9812  057010  012666  000022                MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
 9813  057014  012666  000022                MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
 9814  057020  012666  000022                MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
 9815  057024  012605                        MOV     (SP)+,R5        ;;POP STACK INTO R5
 9816  057026  012604                        MOV     (SP)+,R4        ;;POP STACK INTO R4
 9817  057030  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
 9818  057032  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
 9819  057034  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
 9820  057036  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
 9821  057040  000002                        RTI
 9822                                  .SBTTL  TRAP DECODER
 9823
 9824                                  ;;**************************************************************
 9825                                  ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
 9826                                  ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
 9827                                  ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
 9828                                  ;*GO TO THAT ROUTINE.
 9829
 9830  057042  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
 9831  057044  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
 9832  057050  005740                          TST     -(R0)           ;;BACKUP BY 2
 9833  057052  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
 9834  057054  006300                          ASL     R0              ;;POSITION FOR INDEXING
 9835  057056  016000  057076                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
 9836  057062  000200                          RTS     R0              ;;GO TO ROUTINE
 9837
 9838
 9839                                  ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
 9840
 9841  057064  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
 9842  057066  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
 9843  057074  000002                          RTI                     ;;RESTORE THE PSW
 9844
 9845                                  .SBTTL  TRAP TABLE
 9846
 9847                                  ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
 9848                                  ;*BY THE "TRAP" INSTRUCTION.
 9849
 9850                                  ;       ROUTINE
 9851                                  ;       -------
 9852  057076  057064                  $TRPAD: .WORD   $TRAP2
 9853  057100  054060                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
 9854  057102  055060                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
 9855  057104  055034                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
 9856  057106  055074                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
```

```
 9857  057110  054342                $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
 9858
 9859  057112  055332                $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING
 9860
 9861  057114  055262                $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
 9862  057116  055544                $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
 9863  057120  055664                $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
 9864  057122  056176                $RDOCT  ;;CALL=RDOCT    TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
 9865  057124  056746                $SAVREG ;;CALL=SAVREG   TRAP+13(104413) SAVE R0-R5 ROUTINE
 9866  057126  057004                $RESREG ;;CALL=RESREG   TRAP+14(104414) RESTORE R0-R5 ROUTINE
 9867  057130  052736                SCOP1$  ;;CALL=SCOP1    TRAP+15(104415) INTERNAL LOOP ON ERROR
 9868
```

```
 9869
 9870                                    .SBTTL  SERVICE MESSAGES
 9871
 9872
 9873  057132  005015  047125  041111  MSG1:   .ASCII  <CR><LF>/UNIBUS RK06 DRIVE DIAGNOSTIC/
 9874  057140  051525  051040  030113
 9875  057146  020066  051104  053111
 9876  057154  020105  044504  043501
 9877  057162  047516  052123  041511
 9878  057170  005015  050011  051101          .ASCII  <CR><LF>/       PART 1  014A/<CR><LF>
 9879  057176  020124  004461  030460
 9880  057204  040464  005015
 9881  057210  005015  025011  025052          .ASCII  <CR><LF>/       ***** CAUTION *****/<CR><LF>
 9882  057216  025052  041440  052501
 9883  057224  044524  047117  025040
 9884  057232  025052  025052  005015
 9885  057240  005015  044124  051511          .ASCII  <CR><LF>/THIS PROGRAM SHOULD  BE HALTED ONLY AT THE END/
 9886  057246  050040  047522  051107
 9887  057254  046501  051440  047510
 9888  057262  046125  020104  041040
 9889  057270  020105  040510  052114          -
 9890  057276  042105  047440  046116
 9891  057304  020131  052101  052040
 9892  057312  042510  042440  042116
 9893  057320  005015  043117  040440          .ASCII  <CR><LF>/OF A PASS, OTHERWISE HEADERS WRITTEN ON THE/
 9894  057326  050040  051501  026123
 9895  057334  047440  044124  051105
 9896  057342  044527  042523  044040
 9897  057350  040505  042504  051522
 9898  057356  053440  044522  052124
 9899  057364  047105  047440  020116
 9900  057372  044124     105
 9901  057375     015  042012  051511          .ASCII  <CR><LF>/DISK CARTRIDGE MAY BE LEFT IN AN UNFORMATTED STATE/<CR><LF>
 9902  057402  020113  040503  052122
 9903  057410  044522  043504  020105
 9904  057416  040515  020131  042502
 9905  057424  046040  043105  020124
 9906  057432  047111  040440  020116
 9907  057440  047125  047506  046522
 9908  057446  052101  042524  020104
 9909  057454  052123  052101  006505
 9910  057462     012
 9911  057463     015  040412  051514          .ASCII  <CR><LF>/ALSO, DRIVES TO BE TESTED SHOULD HAVE:/<CR><LF>
 9912  057470  026117  042040  044522
 9913  057476  042526  020123  047524
 9914  057504  041040  020105  042524
 9915  057512  052123  042105  051440
 9916  057520  047510  046125  020104
 9917  057526  040510  042526  006472
 9918  057534     012
 9919  057535     015  040412  020056          .ASCII  <CR><LF>/A. HEADS MANUALLY LOADED/
 9920  057542  042510  042101  020123
 9921  057550  040515  052516  046101
 9922  057556  054514  046040  040517
 9923  057564  042504     104
 9924  057567     015  041012  020056          .ASCII  <CR><LF>/B. CORRECT PORT SELECTED/
```

```
 9925  057574  047503  051122  041505
 9926  057602  020124  047520  052122
 9927  057610  051440  046105  041505
 9928  057616  042524     104
 9929  057621     015  041412  020056          .ASCII  <CR><LF>/C. WRITE LOCK DISABLED/
 9930  057626  051127  052111  020105
 9931  057634  047514  045503  042040
 9932  057642  051511  041101  042514
 9933  057650     104
 9934  057651     015  042012  020056          .ASCII  <CR><LF>/D. DRIVE READY INDICATOR ON/<CR><LF>
 9935  057656  051104  053111  020105
 9936  057664  042522  042101  020131
 9937  057672  047111  044504  040503
 9938  057700  047524  020122  047117
 9939  057706  005015
 9940  057710  005015  051104  053111          .ASCII  <CR><LF>/DRIVES NOT TO BE TESTED MUST HAVE/
 9941  057716  051505  047040  052117
 9942  057724  052040  020117  042502
 9943  057732  052040  051505  042524
 9944  057740  020104  052515  052123
 9945  057746  044040   53101     105
 9946  057753     015  041012  052117          .ASCIZ  <CR><LF>/BOTH PORTS DESELECTED/
 9947  057760  020110  047520  052122
 9948  057766  020123  042504  042523
 9949  057774  042514  052103  042105
 9950  060002     000
 9951
 9952
 9953
 9954
 9955  060003     015  041012  020105  MSG2:   .ASCIZ  <CR><LF>/BE SURE TO PUT SCRATCH PACK IN DRIVE 0/
 9956  060010  052523  042522  052040
 9957  060016  020117  052520  020124
 9958  060024  041523  040522  041524
 9959  060032  020110  040520  045503
 9960  060040  044440  020116  051104
 9961  060046  053111  020105  000060
 9962  060054  005015  051104  053111  MSG3:   .ASCIZ  <CR><LF>/DRIVE(S) TO BE TESTED: /
 9963  060062  024105  024523  052040
 9964  060070  020117  042502  052040
 9965  060076  051505  042524  035104
 9966  060104  000040
 9967  060106  005015  054524  042520  MSG4:   .ASCIZ  <CR><LF>/TYPE BUSS ADDRESS IF NOT 177440 /
 9968  060114  041040  051525  020123
 9969  060122  042101  051104  051505
 9970  060130  020123  043111  047040
 9971  060136  052117  030440  033467
 9972  060144  032064  020060     000
 9973  060151     015  052012  050131  MSG5:   .ASCIZ  <CR><LF>/TYPE CONTROLLER INTERRUPT VECTOR IF NOT 210 /
 9974  060156  020105  047503  052116
 9975  060164  047522  046114  051105
 9976  060172  044440  052116  051105
 9977  060200  052522  052120  053040
 9978  060206  041505  047524  020122
 9979  060214  043111  047040  052117
 9980  060222  031040  030061  000040
```

```
 9981  060230  005015  047111  042524  MSG6:   .ASCIZ  <CR><LF>/INTERRUPT OCCURRED AT PC=/
 9982  060236  051122  050125  020124
 9983  060244  041517  052503  051122
 9984  060252  042105  040440  020124
 9985  060260  041520  000075
 9986  060264  005015  051104  053111  MSG7:   .ASCIZ  <CR><LF>/DRIVE 0 WILL NOT BE TESTED/
 9987  060272  020105  020060  044527
 9988  060300  046114  047040  052117
 9989  060306  041040  020105  042524
 9990  060314  052123  042105     000
 9991  060321     015  050012  042514  MSG8:   .ASCIZ  <CR><LF>/PLEASE WAIT, TEST 16 TAKES 2 TO 4 MINUTES/<CR><LF>
 9992  060326  051501  020105  040527
 9993  060334  052111  020054  042524
 9994  060342  052123  030440  020066
 9995  060350  040524  042513  020123
 9996  060356  020062  047524  032040
 9997  060364  046440  047111  052125
 9998  060372  051505  005015     000
 9999  060377     015  041012  050131  MSG9:   .ASCIZ  <CR><LF>/BYPASSING TEST 16/<CR><LF>
10000  060404  051501  044523  043516
10001  060412  052040  051505  020124
10002  060420  033061  005015     000
10003  060425     015  005012  044527  MSG10:  .ASCIZ  <CR><LF><LF>/WILL TEST DRIVES:/
10004  060432  046114  052040  051505
10005  060440  020124  051104  053111
10006  060446  051505  000072
10007  060452  005015  050012  053517  MSG11:  .ASCIZ  <CR><LF><LF>/POWER UP RESTART TO TEST 1/<CR><LF>
10008  060460  051105  052440  020120
10009  060466  042522  052123  051101
10010  060474  020124  047524  052040
10011  060502  051505  020124  006461
10012  060510  000012
10013  060512  047516  042516  005015  MSG12:  .ASCIZ  /NONE/<CR><LF>
10014  060520     000
10015  060521     015  047012  020117  MSG13:  .ASCII  <CR><LF>/NO L OR P CLOCKS PRESENT/
10016  060535  020114  051117  050040
10017  060543  041440  047514  045503
10018  060542  020123  051120  051505
10019  060550  047105     124
10020  060553     015  040412  046114          .ASCIZ  <CR><LF>/ALL TIMING TESTS BYPASSED/
10021  060560  052040  046511  047111
10022  060566  020107  042524  052123
10023  060574  020123  054502  040520
10024  060602  051523  042105     000
10025  060607     015  041012  050131  MSG14:  .ASCIZ  <CR><LF>/BYPASSING DRIVE /
10026  060614  051501  044523  043516
10027  060622  042040  044522  042526
10028  060630  000040
10029  060632  005015  042012  044522  MSG15:  .ASCIZ  <CR><LF><LF>/DRIVE /
10030  060640  042526  000040
10031  060644  005015  051104  053111  MSG16:  .ASCIZ  <CR><LF>/DRIVE SERIAL NO. /
10032  060652  020105  042523  044522
10033  060660  046101  047040  027117
10034  060666  000040
10035  060670  005015  040503  052122  MSG17:  .ASCIZ  <CR><LF>/CARTRIDGE SERIAL NO. /
10036  060676  044522  043504  020105
```

```
 10037  060704  042523  044522  046101
 10038  060712  047040  027117  000040
 10039  060720  005015  040412  047502  MSG18:  .ASCIZ  <CR><LF><LF>/ABORTING BALANCE OF TESTS/<CR><LF><LF>
 10040  060726  052122  047111  020107
 10041  060734  040502  040514  041516
 10042  060742  020105  043117  052040
 10043  060750  051505  051524  005015
 10044  060756  000012
 10045  060760  005015  040412  046114  MSG19:  .ASCIZ  <CR><LF><LF>/ALL DRIVES TESTED/<CR><LF><LF>
 10046  060766  042040  044522  042526
 10047  060774  020123  042524  052123
 10048  061002  042105  005015  000012
 10049  061010  005015  052522  047116  MSG20:  .ASCII  <CR><LF>/RUNNING MODIFIED VERSION OF FORMAT PACK TEST/
 10050  061016  047111  020107  047515
 10051  061024  044504  044506  042105
 10052  061032  053040  051105  044523
 10053  061040  047117  047440  020106
 10054  061046  047506  046522  052101
 10055  061054  050040  041501  020113
 10056  061062  042524  052123
 10057  061066  005015  047524  051040          .ASCII  <CR><LF>/TO RESTORE HEADERS ON CYLINDER 0 & 1, ALL TRACKS/
 10058  061074  051505  047524  042522
 10059  061102  044040  040505  042504
 10060  061110  051522  047440  020116
 10061  061116  054503  044514  042116
 10062  061124  051105  030040  023040
 10063  061132  030440  020054  046101
 10064  061140  020114  051124  041501
 10065  061146  051513
 10066  061150  005015  047506  020122          .ASCIZ  <CR><LF>/FOR MODULE TESTING/<CR><LF>
 10067  061156  047515  052504  042514
 10068  061164  052040  051505  044524
 10069  061172  043516  005015     000
 10070  061177     015  041012  050131  MSG21:  .ASCIZ  <CR><LF>/BYPASSING TESTS 36,40 & 41 FOR MODULE TESTING/<CR><LF>
 10071  061204  051501  044523  043516
 10072  061212  052040  051505  051524
 10073  061220  031440  026066  030064
 10074  061226  023040  032040  020061
 10075  061234  047506  020122  047515
 10076  061242  052504  042514  052040
 10077  061250  051505  044524  043516
 10078  061256  005015     000
 10079
 10080
 10081                                  .SBTTL  ERROR MESSAGES
 10082
 10083  061261     015  042412  051122  EM1:    .ASCIZ  <CR><LF>/ERROR, ONLY 0 THRU 7 ALLOWED, TRY AGAIN/<CR><LF>
 10084  061266  051117  020054  047117
 10085  061274  054514  030040  052040
 10086  061302  051110  020125  020067
 10087  061310  046101  047514  042527
 10088  061316  026104  052040  054522
 10089  061324  040440  040507  047111
 10090  061332  005015     000
 10091  061335     123  046105  041505  EM2:    .ASCIZ  /SELECTED DRIVE # IN RKCS2 CANNOT BE READ BACK CORRECTLY IN RKMR2/
 10092  061342  042524  020104  051104
```

```
 10093  061350  053111  020105  020043
 10094  061356  047111  051040  041513
 10095  061364  031123  041440  047101
 10096  061372  047516  020124  042502
 10097  061400  051040  040505  020104
 10098  061406  040502  045503  041440
 10099  061414  051117  042522  052103
 10100  061422  054514  044440  020116
 10101  061430  045522  051115  000062
 10102  061436  005015  041101  051117  EM3:    .ASCIZ  <CR><LF>/ABORT TESTS...UNEXPECTED TIME OUT AT PC=/
 10103  061444  020124  042524  052123.
 10104  061452  027123  027056  047125
 10105  061460  054105  042520  052103
 10106  061466  042105  052040  046511
 10107  061474  020105  052517  020124
 10108  061502  052101  050040  036503
 10109  061510     000
 10110  061511     015  040412  047502  EM4:    .ASCIZ  <CR><LF>/ABORT TESTS...UNEXPECTED INTERRUPT AT PC=/
 10111  061516  052122  020040  051505
 10112  061524  051524  027056  052456
 10113  061532  042516  050130  041505
 10114  061540  042524  020104  047111
 10115  061546  042524  051122  050125
 10116  061554  020124  052101  050040
 10117  061562  036503     000
 10118  061565     115  051504  051440  EM5:    .ASCIZ  /MDS SET IN RKCS2/
 10119  061572  052105  044440  020116
 10120  061600  045522  051503  000062
 10121  061606  043125  020105  042523  EM6:    .ASCIZ  /UFE SET IN RKCS2/
 10122  061614  020124  047111  051040
 10123  061622  041513  031123     000
 10124  061627     104  040522  044440  EM7:    .ASCIZ  /DRA IN RKDS & NED IN RKCS2 RESET; WRONG PORT SELECTED?/
 10125  061634  020116  045522  051504
 10126  061642  023040  047040  042105
 10127  061650  044440  020116  045522
 10128  061656  051503  020062  042522
 10129  061664  042523  035524  053440
 10130  061672  047522  043516  050040
 10131  061700  051117  020124  042523
 10132  061706  042514  052103  042105
 10133  061714  000077
 10134  061716  051104  053111  020105  EM8:    .ASCIZ  /DRIVE PRESENT BUT NOT SPECIFIED BY OPERATOR/
 10135  061724  051120  051505  047105
 10136  061732  020124  052502  020124
 10137  061740  047516  020124  050123
 10138  061746  041505  043111  042511
 10139  061754  020104  054502  047440
 10140  061762  042520  040522  047524
 10141  061770  000122
 10142  061772  051104  053111  020105  EM9:    .ASCIZ  /DRIVE NOT PRESENT BUT SPECIFIED BY OPERATOR/
 10143  062000  047516  020124  051120
 10144  062006  051505  047105  020124
 10145  062014  052502  020124  050123
 10146  062022  041505  043111  042511
 10147  062030  020104  054502  047440
 10148  062036  042520  040522  047524
```

```
10149  062044  000122
10150  062046  041101  051117  020124  EM10:   .ASCIZ  /ABORT TESTS...CANNOT REFERENCE CONTROLLER REGISTER/
10151  062054  042524  052123  027123
10152  062062  027056  040503  047116
10153  062070  052117  051040  043105
10154  062076  051105  047105  042503
10155  062104  041440  047117  051124
10156  062112  046117  042514  020122
10157  062120  042522  044507  052123
10158  062126  051105     000
10159  062131     104  040522  044440  EM11:   .ASCIZ  /DRA IN RKDS & NED IN RKCS2 BOTH SET/
10160  062136  020116  045522  051504
10161  062144  023040  047040  042105
10162  062152  044440  020116  045522
10163  062160  051503  020062  047502
10164  062166  044124  051440  052105
10165  062174     000
10166  062175     103  047117  051124  EM12:   .ASCIZ  /CONTROLLER NOT READY IN RKCS1/
10167  062202  046117  042514  020122
10168  062210  047516  020124  042522
10169  062216  042101  020131  047111
10170  062224  051040  041513  030523
10171  062232     000
10172  062233     116  020117  052101  EM13:   .ASCIZ  /NO ATTN IN RKASOF/
10173  062240  047124  044440  020116
10174  062246  045522  051501  043117
10175  062254     000
10176  062255     127  047522  043516  EM14:   .ASCIZ  /WRONG ATTN IN RKASOF/
10177  062262  040440  052124  020116
10178  062270  047111  051040  040513
10179  062276  047523  000106
10180  062302  051104  054504  047040  EM15:   .ASCIZ  /DRDY NOT CLEARED IN RKMR2/
1018.  062310  052117  041440  042514
10182  062316  051101  042105  044440
10183  062324  020116  045522  051115
10184  062332  000062
10185  062334  051504  020103  047516  EM16:   .ASCIZ  /DSC NOT SET IN RKMR2/
10186  062342  020124  042523  020124
10187  062350  047111  051040  046513
10188  062356  031122     000
10189  062361     115  051505  040523  EM17:   .ASCIZ  /MESSAGE A0 ERROR/
10190  062366  042507  040440  020060
10191  062374  051105  047522  000122
10192  062402  042515  051523  043501  EM18:   .ASCIZ  /MESSAGE B0 ERROR/
10193  062410  020105  030102  042440
10194  062416  051122  051117     000
10195  062423     115  051505  040523  EM19:   .ASCIZ  /MESSAGE A1 ERROR/
10196  062430  042507  040440  020061
10197  062436  051105  047522  000122
10198  062444  042515  051523  043501  EM20:   .ASCIZ  /MESSAGE B1 ERROR/
10199  062452  020105  030502  042440
10200  062460  051122  051117     000
10201  062465     103  051105  020122  EM21:   .ASCIZ  /CERR SET IN RKCS1/
10202  062472  042523  020124  047111
10203  062500  051040  041513  030523
10204  062506     000
```

```
10205  062507     122  051514  044440  EM22:   .ASCIZ  /RLS IN RKCS2 SET CERR IN RKCS1/
10206  062514  020116  045522  051503
10207  062522  020062  042523  020124
10208  062530  042503  051122  044440
10209  062536  020116  045522  051503
10210  062544  000061
10211  062546  043125  020105  047111  EM23:   .ASCIZ  /UFE IN RKCS2 SET (IE:SACK) AFTER RLS IN RKCS2 SENT/
10212  062554  051040  041513  031123
10213  062562  051440  052105  024040
10214  062570  042511  051472  041501
10215  062576  024513  040440  052106
10216  062604  051105  051040  051514
10217  062612  044440  020116  045522
10218  062620  051503  020062  042523
10219  062626  052116     000
10220  062631     126  046117  053040  EM24:   .ASCIZ  /VOL VALID NOT SET IN RKMR2/
10221  062636  046101  042111  047040
10222  062644  052117  051440  052105
10223  062652  044440  020116  045522
10224  062660  051115  000062
10225  062664  051104  020126  054524  EM25:   .ASCIZ  /DRV TYPE SET IN RKMR2/
10226  062672  042520  051440  052105
10227  062700  044440  020116  045522
10228  062706  051115  000062
10229  062712  042104  020124  042523  EM26:   .ASCIZ  /DDT SET IN RKDS/
10230  062720  020124  047111  051040
10231  062726  042113  000123
10232  062732  052104  042531  051440  EM27:   .ASCIZ  /DTYE SET IN RKER/
10233  062740  052105  044440  020116
10234  062746  045522  051105     000
10235  062753     104  054524  020105  EM28:   .ASCIZ  /DTYE NOT SET IN RKER/
10236  062760  047516  020124  042523
10237  062766  020124  047111  051040
10238  062774  042513  000122
10239  063000  052104  042531  044440  EM29:   .ASCIZ  /DTYE IN RKER DID NOT SET CERR IN RKCS1/
10240  063006  020116  045522  051105
10241  063014  042040  042111  047040
10242  063022  052117  051440  052105
10243  063030  041440  051105  020122
10244  063036  047111  051040  041513
10245  063044  030523     000
10246  063047     103  042055  050040  EM30:   .ASCIZ  /C-D PARITY ERR SET IN RKMR3/
10247  063054  051101  052111  020131
10248  063062  051105  020122  042523
10249  063070  020124  047111  051040
10250  063076  046513  031522     000
10251  063103     123  040520  020122  EM31:   .ASCIZ  /SPAR SET IN RKCS1/
10252  063110  042523  020124  047111
10253  063116  051040  041513  030523
10254  063124     000
10255  063125     106  052501  052114  EM32:   .ASCIZ  /FAULT NOT SET IN RKMR3/
10256  063132  047040  052117  051440
10257  063140  052105  044440  020116
10258  063146  045522  051115  000063
10259  063154  026503  020104  040520  EM33:   .ASCIZ  /C-D PARITY ERR NOT SET IN RKMR3/
10260  063162  044522  054524  042440
```

```
10261  063170  051122  047040  052117
10262  063176  051440  052105  044440
10263  063204  020116  045522  051115
10264  063212  000063
10265  063214  050123  051101  047040  EM34:   .ASCIZ  /SPAR NOT SET IN RKCS1/
10266  063222  052117  051440  052105
10267  063230  044440  020116  045522
10268  063236  051503  000061
10269  063242  050123  051101  044440  EM35:   .ASCIZ  /SPAR IN RKCS1 DID NOT SET CERR IN RKCS1/
10270  063250  020116  045522  051503
10271  063256  020061  044504  020104
10272  063264  047516  020124  042523
10273  063272  020124  042503  051122
10274  063300  044440  020116  045522
10275  063306  051503  000061
10276  063312  054503  020114  042101  EM36:   .ASCIZ  /CYL ADDR IN RKMR3 NOT SAME AS RKDC/
10277  063320  051104  044440  020116
10278  063326  045522  051115  020063
10279  063334  047516  020124  040523
10280  063342  042515  040440  020123
10281  063350  045522  041504     000
10282  063355     103  046131  042040  EM37:   .ASCIZ  /CYL DIFF IN RKMR2 NOT SAME AS RKDC/
10283  063362  043111  020106  047111
10284  063370  051040  046513  031122
10285  063376  047040  052117  051440
10286  063404  046501  020105  051501
10287  063412  051040  042113  000103
10288  063420  054503  020114  044504  EM38:   .ASCIZ  /CYL DIFF IN RKMR2 NOT SAME AS 'CYL DIFF'/
10289  063426  043106  044440  020116
10290  063434  045522  051115  020062
10291  063442  047516  020124  040523
10292  063450  042515  040440  020123
10293  063456  041447  046131  042040
10294  063464  043111  023506     000
10295  063471     103  046131  042040  EM39:   .ASCIZ  /CYL DIFF & OFFSET IN RKMR2 NOT CLEARED/
10296  063476  043111  020106  020046
10297  063504  043117  051506  052105
10298  063512  044440  020116  045522
10299  063520  051115  020062  047516
10300  063526  020124  046103  040505
10301  063534  042522  000104
10302  063540  054503  020114  042101  EM40:   .ASCIZ  /CYL ADDR IN RKMR3 NOT CLEARED/
10303  063546  051104  044440  020116
10304  063554  045522  051115  020063
10305  063562  047516  020124  046103
10306  063570  040505  042522  000104
10307  063576  054503  020114  042101  EM41:   .ASCIZ  /CYL ADDR IN RKMR3 DID NOT REMAIN CLEARED/
10308  063604  051104  044440  020116
10309  063612  045522  051115  020063
10310  063620  044504  020104  047516
10311  063626  020124  042522  040515
10312  063634  047111  041440  042514
10313  063642  051101  042105     000
10314  063647     117  043106  042523  EM42:   .ASCIZ  /OFFSET REGISTER IN RKMR2 NOT SAME AS RKASOF/
10315  063654  020124  042522  044507
10316  063662  052123  051105  044440
```

```
10317  063670  020116  045522  051115
10318  063676  020062  047516  020124
10319  063704  040523  042515  040440
10320  063712  020123  045522  051501
10321  063720  043117     000
10322  063723     110  040505  020104  EM43:   .ASCIZ  /HEAD ADDR IN RKMR3 NOT CLEARED/
10323  063730  042101  051104  044440
10324  063736  020116  045522  051115
10325  063744  020063  047516  020124
10326  063752  046103  040505  042522
10327  063760  000104
10328  063762  042510  042101  042040  EM44:   .ASCIZ  /HEAD DECODE IN RKMR3 DOES NOT CORRELATE WITH RKDA/
10329  063770  041505  042117  020105
10330  063776  047111  051040  046513
10331  064004  031522  042040  042517
10332  064012  020123  047516  020124
10333  064020  047503  051122  046105
10334  064026  052101  020105  044527
10335  064034  044124  051040  042113
10336  064042  000101
10337  064044  051104  053111  020105  EM45:   .ASCII  /DRIVE READY IN RKMR2 NOT SET WITHIN APPROX 1 SEC FROM FWD/
10338  064052  042522  042101  020131
10339  064060  047111  051040  046513
10340  064066  031122  047040  052117
10341  064074  051440  052105  053440
10342  064102  052111  044510  020116
10343  064110  050101  051120  054117
10344  064116  030440  051440  041505
10345  064124  043040  047522  020115
10346  064132  053506     104
10347  064135     015  044412  020116          .ASCIZ  <CR><LF>/IN RTZ PORTION OF START SPINDLE CMD/
10348  064142  052122  020132  047520
10349  064150  052122  047511  020116
10350  064156  043117  051440  040524
10351  064164  052122  051440  044520
10352  064172  042116  042514  041440
10353  064200  042115     000
10354  064203     115  051505  040523  EM46:   .ASCIZ  /MESSAGE A2 ERROR/
10355  064210  042507  040440  020062
10356  064216  051105  047522  000122
10357  064224  042515  051523  043501  EM47:   .ASCIZ  /MESSAGE B2 ERROR/
10358  064232  020105  031102  042440
10359  064240  051122  051117     000
10360  064245     115  051505  040523  EM48:   .ASCIZ  /MESSAGE B3 ERROR/
10361  064252  042507  041040  020063
10362  064260  051105  047522  000122
10363  064266  053506  020104  047516  EM49:   .ASCIZ  /FWD NOT SET IN RKMR2 WITHIN APPROX 4 SEC IN RTZ PORTION OF START SPINDL
10364  064274  020124  042523  020124
10365  064302  047111  051040  046513
10366  064310  031122  053440  052111
10367  064316  044510  020116  050101
10368  064324  051120  054117  032040
10369  064332  051440  041505  044440
10370  064340  020116  052122  020132
10371  064346  047520  052122  047511
10372  064354  020116  043117  051440
```

```
10373  064362  040524  052122  051440
10374  064370  044520  042116  042514
10375  064376  041440  042115     000
10376  064403     106  042127  047040  EM50:   .ASCIZ  /FWD NOT SET IN RKMR2 WITHIN APPROX 60 SEC FROM START SPINDLE CMD/
10377  064410  052117  051440  052105
10378  064416  044440  020116  045522
10379  064424  051115  020062  044527
10380  064432  044124  047111  040440
10381  064440  050120  047522  020130
10382  064446  030066  051440  041505
10383  064454  043040  047522  020115
10384  064462  052123  051101  020124
10385  064470  050123  047111  046104
10386  064476  020105  046503  000104
10387  064504  053506  020104  047516  EM51:   .ASCIZ  /FWD NOT CLEARED IN RKMR2 WITHIN APPROX 5 SEC OF MOTION FROM START SPIND
10388  064512  020124  046103  040505
10389  064520  042522  020104  047111
10390  064526  051040  046513  031122
10391  064534  053440  052111  044510
10392  064542  020116  050101  051120
10393  064550  054117  032440  051440
10394  064556  041505  047440  020106
10395  064564  047515  044524  047117
10396  064572  043040  047522  020115
10397  064600  052123  051101  020124
10398  064606  050123  047111  046104
10399  064614  020105  046503  000104
10400  064622  030062  051440  041505  EM52:   .ASCIZ  /20 SECTOR FORMAT NOT SET IN RKMR2/
10401  064630  047524  020122  047506
10402  064636  046522  052101  047040
10403  064644  052117  051440  052105
10404  064652  044440  020116  045522
10405  064660  051115  000062
10406  064664  042523  052103  051117  EM53:   .ASCIZ  /SECTOR 0 NOT FOUND WITHIN 50 MS/
10407  064672  030040  047040  052117
10408  064700  043040  052517  042116
10409  064706  053440  052111  044510
10410  064714  020116  030065  046440
10411  064722  000123
10412  064724  044504  043106  051440  EM54:   .ASCIZ  /DIFF SECTOR NOT FOUND WITHIN 3 MS/
10413  064732  041505  047524  020122
10414  064740  047516  020124  047506
10415  064746  047125  020104  044527
10416  064754  044124  047111  031440
10417  064762  046440  000123
10418  064766  052101  047124  047040  EM55:   .ASCIZ  /ATTN NOT CLEARED IN RKASOF/
10419  064774  052117  041440  042514
10420  065002  051101  042105  044440
10421  065010  020116  045522  051501
10422  065016  043117     000
10423  065021     125  042516  050130  EM56:   .ASCIZ  /UNEXPECTED MEMORY PARITY TRAP/
10424  065026  041505  042524  020104
10425  065034  042515  047515  054522
10426  065042  050040  051101  052111
10427  065050  020131  051124  050101
10428  065056     000
```

```
10429  065057     122  042113  020103  EM57:   .ASCII  /RKDC &RKDA INDICATE THAT WCE OCCURRED AT/
10430  065064  051046  042113  020101
10431  065072  047111  044504  040503
10432  065100  042524  052040  040510
10433  065106  020124  041527  020105
10434  065114  041517  052503  051122
10435  065122  042105  040440     124
10436  065127     015  041412  046131          .ASCIZ  <CR><LF>/CYL 411, TRACK 2, SECTOR 21/
10437  065134  032040  030461  020054
10438  065142  051124  041501  020113
10439  065150  026062  051440  041505
10440  065156  047524  020122  030462
10441  065164     000
10442  065165     015  051412  042520  EM58:   .ASCIZ  <CR><LF>/SPEED OK IN RKMR2 NOT CLEARED BY TIMEOUT/
10443  065172  042105  047440  020113
10444  065200  047111  051040  046513
10445  065206  031122  047040  052117
10446  065214  041440  042514  051101
10447  065222  042105  041040  020131
10448  065230  044524  042515  052517
10449  065236  000124
10450  065240  044514  044515  020124  EM59:   .ASCIZ  /LIMIT DETECT NOT FOUND IN RKMR3/
10451  065246  042504  042524  052103
10452  065254  047040  052117  043040
10453  065262  052517  042116  044440
10454  065270  020116  045522  051115
10455  065276  000063
10456  065300  042510  042101  020123  EM60:   .ASCIZ  /HEADS HOME NOT FOUND IN RKMR2/
10457  065306  047510  042515  047040
10458  065314  052117  043040  052517
10459  065322  042116  044440  020116
10460  065330  045522  051115  000062
10461  065336  047514  042101  044040  EM61:   .ASCIZ  /LOAD HEADS NOT FOUND IN RKMR2/
10462  065344  040505  051504  047040
10463  065352  052117  043040  052517
10464  065360  042116  044440  020116
10465  065366  045522  051115  000062
10466  065374  052104  020105  042523  EM62:   .ASCIZ  /DTE SET IN RKER/
10467  065402  020124  047111  051040
10468  065410  042513  000122
10469  065414  046104  020124  042523  EM63:   .ASCIZ  /DLT SET IN RKCS2/
10470  065422  020124  047111  051040
10471  065430  041513  031123     000
10472  065435     115  043523  041040  EM64:   .ASCIZ  /MSG B3 HEAD REG NOT CLEARED IN UNLOAD/
10473  065442  020063  042510  042101
10474  065450  051040  043505  047040
10475  065456  052117  041440  042514
10476  065464  051101  042105  044440
10477  065472  020116  047125  047514
10478  065500  042101     000
10479  065503     122  040505  020104  EM65:   .ASCIZ  /READ HEADER ERROR/
10480  065510  042510  042101  051105
10481  065516  042440  051122  051117
10482  065524     000
10483  065525     103  046131  040440  EM66:   .ASCIZ  /CYL ADDR IN RKMR3 INCORRECT/
10484  065532  042104  020122  047111
```

```
10485  065540  051040  046513  031522
10486  065546  044440  041516  051117
10487  065554  042522  052103     000
10488  065561     122  040505  044504  EM67:   .ASCIZ  /READING CYL 0 HEADERS ON CYL 1/
10489  065566  043516  041440  046131
10490  065574  030040  044040  040505
10491  065602  042504  051522  047440
10492  065610  020116  054503  020114
10493  065616  000061
10494  065620  042522  042101  047111  EM68:   .ASCIZ  /READING CYL 1 HEADERS ON CYL 0/
10495  065626  020107  054503  020114
10496  065634  020061  042510  042101
10497  065642  051105  020123  047117
10498  065650  041440  046131  030040
10499  065656     000
10500  065657     101  044514  047107  EM69:   .ASCIZ  /ALIGNMENT CARTRIDGE USED/
10501  065664  042515  052116  041440
10502  065672  051101  051124  042111
10503  065700  042507  052440  042523
10504  065706  000104
10505  065710  051504  020103  042523  EM71:   .ASCIZ  /DSC SET IN RKMR2/
10506  065716  020124  047111  051040
10507  065724  046513  031122     000
10508  065731     126  051117  040515  EM72:   .ASCIZ  /FORMAT TEST BYPASSED/
10509  065736  020124  042524  052123
10510  065744  041040  050131  051501
10511  065752  042523  000104
10512  065756  052122  020132  047516  EM74:   .ASCIZ  /RTZ NOT SET IN RKMR2/
10513  065764  020124  042523  020124
10514  065772  047111  051040  046513
10515  066000  031122     000
10516  066003     111  040504  020105  EM75:   .ASCIZ  /IDAE NOT SET IN RKMR3/
10517  066010  047516  020124  042523
10518  066016  020124  047111  051040
10519  066024  046513  031522     000
10520  066031     120  050111  051440  EM76:   .ASCIZ  /PIP SET IN RKMR2/
10521  066036  052105  044440  020116
10522  066044  045522  051115  000062
10523  066052  040506  046125  020124  EM77:   .ASCIZ  /FAULT NOT CLEARED IN RKMR3/
10524  066060  047516  020124  046103
10525  066066  040505  042522  020104
10526  066074  047111  051040  046513
10527  066102  031522     000
10528  066105     103  046131  042040  EM78:   .ASCIZ  /CYL DIFF IN RKMR2 DID NOT REMAIN = 1 IN SEEK TO SELF/
10529  066112  043111  020106  047111
10530  066120  051040  046513  031122
10531  066126  042040  042111  047040
10532  066134  052117  051040  046505
10533  066142  044501  020116  020075
10534  066150  020061  047111  051440
10535  066156  042505  020113  047524
10536  066164  051440  046105  000106
10537  066172  047125  047514  042101  EM80:   .ASCIZ  /UNLOAD NOT SET IN RKMR2/
10538  066200  047040  052117  051440
10539  066206  052105  044440  020116
10540  066214  045522  051115  000062
```

```
10541  066222  050123  047111  047040  EM81:   .ASCIZ  /SPIN NOT SET IN RKMR2/
10542  066230  052117  051440  052105
10543  066236  044440  020116  045522
10544  066244  051115  000062
10545  066250  052122  020132  047516  EM82:   .ASCIZ  'RTZ NOT SET IN RKMR2/
10546  066256  020124  042523  020124
10547  066264  047111  051040  046513
10548  066272  031122     000
10549  066275     122  040505  020104  EM83:   .ASCIZ  /READ HEADER ERROR   WORD 0 (CYLINDER#)/
10550  066302  042510  042101  051105
10551  066310  042440  051122  051117
10552  066316  020040  053440  051117
10553  066324  020104  020060  041450
10554  066332  046131  047111  042504
10555  066340  021522  000051
10556  066344  047506  046522  052101  EM84:   .ASCIZ  /FORMAT IN RKMR3 NOT SET/
10557  066352  044440  020116  045522
10558  066360  051115  020063  047516
10559  066366  020124  042523  000124
10560  066374  046111  042514  040507  EM85:   .ASCIZ  /ILLEGAL ADDRESS IN RKMR3 NOT CLEARED/
10561  066402  020114  042101  051104
10562  066410  051505  020123  047111
10563  066416  051040  046513  031522
10564  066424  047040  052117  041440
10565  066432  042514  051101  042105
10566  066440     000
10567  066441     103  047101  047516  EM87:   .ASCIZ  /CANNOT READ BAD SECTOR INFORMATION/
10568  066446  020124  042522  042101
10569  066454  041040  042101  051440
10570  066462  041505  047524  020122
10571  066470  047111  047506  046522
10572  066476  052101  047511  000116
10573
10574                                  .SBTTL  DATA HEADERS
10575
10576  066504  042524  052123  047040  DH1:    .ASCIZ  /TEST NO.  PC/
10577  066512  027117  020040  041520
10578  066520     000
10579  066521     122  046513  031122  DH2:    .ASCIZ  /RKMR2  RKMR3   RKER    RKDS    RKCS1   RKCS2/
10580  066526  051011  046513  031522
10581  066534  051011  042513  004522
10582  066542  045522  051504  051011
10583  066550  041513  030523  051011
10584  066556  041513  031123     000
10585  066563     122  040513  047523  DH3:    .ASCIZ  /RKASOF/
10586  066570  000106
10587  066572  044123  052517  042114  DH4:    .ASCIZ  /SHOULD BE/
10588  066600  041040  000105
10589  066604  045522  051115  004462  DH5:    .ASCIZ  /RKMR2  RKMR3   RKER    RKDS    RKDC    RKCS1   RKCS2/
10590  066612  045522  051115  004463
10591  066620  045522  051105  051011
10592  066626  042113  004523  045522
10593  066634  041504  051011  041513
10594  066642  030523  051011  041513
10595  066650  031123     000
10596  066653     122  046513  031122  DH6:    .ASCIZ  /RKMR2   RKMR3   RKDC   FROM CYL  TO CYL  CYL DIFF/
```

```
10597  066660  020040  051040  046513
10598  066666  031522  020040  051040
10599  066674  042113  020103  020040
10600  066702  051106  046517  041440
10601  066710  046131  020040  047524
10602  066716  041440  046131  020040
10603  066724  054503  020114  044504
10604  066732  043106     000
10605  066735     122  046513  031122  DH7:    .ASCIZ  /RKMR2  RKMR3   RKER    RKDS    RKDA    RKCS1   RKCS2/
10606  066742  051011  046513  031522
10607  066750  051011  042513  004522
10608  066756  045522  051504  051011
10609  066764  042113  004501  045522
10610  066772  051503  004461  045522
10611  067000  051503  000062
10612  067004  044127  047105  042040  DH8:    .ASCIZ  /WHEN DRIVE UNLOADED/
10613  067012  044522  042526  052440
10614  067020  046116  040517  042504
10615  067026  000104
10616  067030  043101  042524  020122  DH9:    .ASCIZ  /AFTER START SPINDLE COMMAND REC'D BY DRIVE/
10617  067036  052123  051101  020124
10618  067044  050123  047111  046104
10619  067052  020105  047503  046515
10620  067060  047101  020104  042522
10621  067066  023503  020104  054502
10622  067074  042040  044522  042526
10623  067102     000
10624  067103     101  020124  047105  DH10:   .ASCIZ  /AT END OF HEAD LOADING/
10625  067110  020104  043117  044040
10626  067116  040505  020104  047514
10627  067124  042101  047111  000107
10628  067132  043101  042524  020122  DH11:   .ASCIZ  /AFTER START SPINDLE CMD & FWD SET/
10629  067140  052123  051101  020124
10630  067146  050123  047111  046104
10631  067154  020105  046503  020104
10632  067162  020046  053506  020104
10633  067170  042523  000124
10634  067174  052101  044440  047116  DH12:   .ASCIZ  /AT INNER LIMIT FROM START SPINDLE CMD/
10635  067202  051105  046040  046511
10636  067210  052111  043040  047522
10637  067216  020115  052123  051101
10638  067224  020124  050123  047111
10639  067232  046104  020105  046503
10640  067240  000104
10641  067242  051106  046517  047440  DH13:   .ASCIZ  /FROM OUTER LIMIT TO CYL 0 DURING LOADING/
10642  067250  052125  051105  046040
10643  067256  046511  052111  052040
10644  067264  020117  054503  020114
10645  067272  020060  052504  044522
10646  067300  043516  046040  040517
10647  067306  044504  043516     000
10648  067313     101  052106  051105  DH14:   .ASCIZ  /AFTER SEEK WITH BAD PARITY/
10649  067320  051440  042505  020113
10650  067326  044527  044124  041040
10651  067334  042101  050040  051101
10652  067342  052111  000131
```

```
 10653  067346  043101  042524  020122  DH15:   .ASCIZ  /AFTER OFFSET COMMAND WITH BAD PARITY/
 10654  067354  043117  051506  052105
 10655  067362  041440  046517  040515
 10656  067370  042116  053440  052111
 10657  067376  020110  040502  020104
 10658  067404  040520  044522  054524
 10659  067412     000
 10660  067413     101  052106  051105  DH16:   .ASCIZ  /AFTER LOADING HEAD REGISTER & SEEK COMMAND/
 10661  067420  046040  040517  044504
 10662  067426  043516  044040  040505
 10663  067434  020104  042522  044507
 10664  067442  052123  051105  023040
 10665  067450  051440  042505  020113
 10666  067456  047503  046515  047101
 10667  067464  000104
 10668  067466  043101  042524  020122  DH17:   .ASCIZ  /AFTER RECAL COMMAND/
 10669  067474  042522  040503  020114
 10670  067502  047503  046515  047101
 10671  067510  000104
 10672  067512  043101  042524  020122  DH18:   .ASCIZ  /AFTER UNLOAD COMMAND/
 10673  067520  047125  047514  042101
 10674  067526  041440  046517  040515
 10675  067534  042116     000
 10676  067537     101  052106  051105  DH19:   .ASCIZ  /AFTER PACK COMMAND/
 10677  067544  050040  041501  020113
 10678  067552  047503  046515  047101
 10679  067560  000104
 10680  067562  043101  042524  020122  DH20:   .ASCIZ  /AFTER SELECT DRIVE COMMAND/
 10681  067570  042523  042514  052103
 10682  067576  042040  044522  042526
 10683  067604  041440  046517  040515
 10684  067612  042116     000
 10685  067615     101  052106  051105  DH21:   .ASCIZ  /AFTER SUBSYSTEM CLEAR/
 10686  067622  051440  041125  054523
 10687  067630  052123  046505  041440
 10688  067636  042514  051101     000
 10689  067643     101  052106  051105  DH22:   .ASCIZ  /AFTER DRIVE CLEAR COMMAND/
 10690  067650  042040  044522  042526
 10691  067656  041440  042514  051101
 10692  067664  041440  046517  040515
 10693  067672  042116     000
 10694  067675     124  051505  020124  DH23:   .ASCIZ  /TEST NO.       TRAP PC/
 10695  067702  047516  004456  051124
 10696  067710  050101  050040  000103
 10697  067716  043101  042524  020122  DH24:   .ASCIZ  /AFTER OFFSET COMMAND/
 10698  067724  043117  051506  052105
 10699  067732  041440  046517  040515
 10700  067740  042116     000
 10701  067743     101  052106  051105  DH25:   .ASCIZ  /AFTER SEEK COMMAND/
 10702  067750  051440  042505  020113
 10703  067756  047503  046515  047101
 10704  067764  000104
 10705  067766  043101  042524  020122  DH26:   .ASCIZ  /AFTER READ DATA COMMAND/
 10706  067774  042522  042101  042040
 10707  070002  052101  020101  047503
 10708  070010  046515  047101  000104
```

```
 10709  070016  043101  042524  020122  DH27:   .ASCIZ  /AFTER WRITE DATA COMMAND/
 10710  070024  051127  052111  020105
 10711  070032  040504  040524  041440
 10712  070040  046517  040515  042116
 10713  070046     000
 10714  070047     101  052106  051105  DH30:   .ASCIZ  /AFTER READ HEADER COMMAND/
 10715  070054  051040  040505  020104
 10716  070062  042510  042101  051105
 10717  070070  041440  046517  040515
 10718  070076  042116     000
 10719  070101     122  046513  031122  DH31:   .ASCIZ  /RKMR2  RKMR3   RKCS1   RKCS2   RKDC    RKDA/
 10720  070106  051011  046513  031522
 10721  070114  051011  041513  030523
 10722  070122  051011  041513  031123
 10723  070130  051011  042113  004503
 10724  070136  045522  040504     000
 10725  070143     104  051125  047111  DH33:   .ASCIZ  /DURING SEEK COMMAND/
 10726  070150  020107  042523  045505
 10727  070156  041440  046517  040515
 10728  070164  042116     000
 10729  070167     123  041505  047524  DH34:   .ASCIZ  /SECTOR REG UNSTABLE/
 10730  070174  020122  042522  020107
 10731  070202  047125  052123  041101
 10732  070210  042514     000
 10733  070213     102  052105  042527  DH35:   .ASCIZ  /BETWEEN SECTOR COUNTS/
 10734  070220  047105  051440  041505
 10735  070226  047524  020122  047503
 10736  070234  047125  051524     000
 10737  070241     122  046513  031122  DH36:   .ASCIZ  /RKMR2   RKMR3   RKCS1   RKCS2  FROM SECT  TO SECT/
 10738  070246  020040  051040  046513
 10739  070254  031522  020040  051040
 10740  070262  041513  030523  020040
 10741  070270  051040  041513  031123
 10742  070276  020040  051106  046517
 10743  070304  051440  041505  020124
 10744  070312  052040  020117  042523
 10745  070320  052103     000
 10746  070323     123  042520  042105  DH37:   .ASCIZ  /SPEED OK NOT CLEARED 10 SEC AFTER UNLOAD/
 10747  070330  047440  020113  047516
 10748  070336  020124  046103  040505
 10749  070344  042522  020104  030061
 10750  070352  051440  041505  040440
 10751  070360  052106  051105  052440
 10752  070366  046116  040517  000104
 10753  070374  043101  042524  020122  DH38:   .ASCIZ  /AFTER LIMIT DETECT/
 10754  070402  044514  044515  020124
 10755  070410  042504  042524  052103
 10756  070416     000
 10757  070417     101  052106  051105  DH39:   .ASCIZ  /AFTER WRITE HEADER COMMAND/
 10758  070424  053440  044522  042524
 10759  070432  044040  040505  042504
 10760  070440  020122  047503  046515
 10761  070446  047101  000104
 10762  070452  045522  051115  004462  DH40:   .ASCIZ  /RKMR2  RKMR3   RKDA    WORD#   HEADER WAS  SHOULD BE/
 10763  070460  045522  051115  004463
 10764  070466  045522  040504  053411
```

```
 10765  070474  051117  021504  044011
 10766  070502  040505  042504  020122
 10767  070510  040527  020123  051440
 10768  070516  047510  046125  020104
 10769  070524  042502     000
 10770  070527     104  051125  047111  DH41:   .ASCIZ  /DURING RECAL COMMAND/
 10771  070534  020107  042522  040503
 10772  070542  020114  047503  046515
 10773  070550  047101  000104
 10774  070554  047117  051440  041505  DH42:   .ASCIZ  /ON SECTORS 0,2,4,6 OR 8  CYL 410 TRACK 2/
 10775  070562  047524  051522  030040
 10776  070570  031054  032054  033054
 10777  070576  047440  020122  020070
 10778  070604  041440  046131  032040
 10779  070612  030061  052040  040522
 10780  070620  045503  031040     000
 10781  070625     117  020116  042523  DH43:   .ASCIZ  /ON SECTORS 1,3,5,7 OR 9  CYL 410 TRACK2/
 10782  070632  052103  051117  020123
 10783  070640  026061  026063  026065
 10784  070646  020067  051117  034440
 10785  070654  020040  054503  020114
 10786  070662  030464  020060  051124
 10787  070670  041501  031113     000
 10788  070675     106  051117  040515  DH44:   .ASCIZ  /FORMAT S ALL READ-WRITE TESTS WILL BE BYPASSED/
 10789  070702  020124  020046  046101
 10790  070710  020114  042522  042101
 10791  070716  053455  044522  042524
 10792  070724  052040  051505  051524
 10793  070732  053440  046111  020114
 10794  070740  042502  041040  050131
 10795  070746  051501  042523  000104
 10796  070754  042502  040503  051525  DH45:   .ASCIZ  /BECAUSE OF LIMIT DETECT ERROR ON PREVIOUS TEST/
 10797  070762  020105  043117  046040
 10798  070770  046511  052111  042040
 10799  070776  052105  041505  020124
 10800  071004  051105  047522  020122
 10801  071012  047117  050040  042522
 10802  071020  044526  052517  020123
 10803  071026  042524  052123     000
 10804  071033     103  052517  042114  DH46:   .ASCIZ  /COULD NOT READ BSE INFO ON PREVIOUS TEST/
 10805  071040  047040  052117  051040
 10806  071046  040505  020104  051502
 10807  071054  020105  047111  047506
 10808  071062  047440  020116  051120
 10809  071070  053105  047511  051525
 10810  071076  052040  051505  000124
 10811  071104  043101  042524  020122  DH48:   .ASCIZ  /AFTER SEEK COMMAND TO INVALID CYLINDER/
 10812  071112  042523  045505  041440
 10813  071120  046517  040515  042116
 10814  071126  052040  020117  047111
 10815  071134  040526  044514  020104
 10816  071142  054503  044514  042116
 10817  071150  051105     000
 10818  071153     101  052106  051105  DH51:   .ASCIZ  /AFTER SEEK TO SELF COMMAND/
 10819  071160  051440  042505  020113
 10820  071166  047524  051440  046105
```

```
10821  071174  020106  047503  046515
10822  071202  047101  000104
10823  071206  045522  051115  004462  DH52:   .ASCIZ  /RKMR2  RKMR3   RKDA    EXP CYL#        CYL HEADER WAS/
10824  071214  045522  051115  004463
10825  071222  045522  040504  042411
10826  071230  050130  041440  046131
10827  071236  004443  054503  020114
10828  071244  042510  042101  051105
10829  071252  053440  051501     000
10830                                  .SBTTL  ERROR OUTPUT DATA
10831
10832          071260                  .EVEN
10833  071260  001214  001116  004364  DT1:    $TESTN,$ERRPC,HMR2,HMR3,HER,HDS,HCS1,HCS2,SBMR2,SBMR3
10834  071266  004366  004352  004350
10835  071274  004336  004340  004374
10836  071302  004376
10837  071304  001214  001116  004364  DT2:    $TESTN,$ERRPC,HMR2,HMR3,HER,HDS,HCS1,HCS2,HASOF
10838  071312  004366  004352  004350
10839  071320  004336  004340  004354
10840  071326  001214  001116  004364  DT3:    $TESTN,$ERRPC,HMR2,HMR3,HER,HDS,HDC,HCS1,HCS2
10841  071334  004366  004352  004350
10842  071342  004356  004336  004340
10843  071350  001214  001116  004364  DT4:    $TESTN,$ERRPC,HMR2,HMR3,HDC,FRCYL,TOCYL,CALDIF
10844  071356  004366  004356  001370
10845  071364  001372  001400
10846  071370  001214  001116  004364  DT5:    $TESTN,$ERRPC,HMR2,HMR3,HER,HDS,HDA,HCS1,HCS2,SBMR2,SBMR3
10847  071376  004366  004352  004350
10848  071404  004346  004336  004340
10849  071412  004374  004376
10850  071416  001214  001116  004364  DT6:    $TESTN,$ERRPC,HMR2,HMR3,HCS1,HCS2,PSEC,ESEC
10851  071424  004366  004336  004340
10852  071432  001422  001424
10853  071436  001214  001116  004364  DT7:    $TESTN,$ERRPC,HMR2,HMR3,HDA,WDCNT,HDWD,TEMP1
10854  071444  004366  004346  001460
10855  071452  001472  004400
10856  071456  001214  001116  004364  DT8:    $TESTN,$ERRPC,HMR2,HMR3,HDC,TOCYL,FRCYL,CALDIF
10857  071464  004366  004356  001372
10858  071472  001370  001400
10859  071476  001214  001116  004364  DT9:    $TESTN,$ERRPC,HMR2,HMR3,HDA,TOCYL,RHTAB
10860  071504  004366  004346  001372
10861  071512  001704
10862  071514  001214  001116  004364  DT10:   $TESTN,$ERRPC,HMR2,HMR3,HDA,FRCYL,RHTAB
10863  071522  004366  004346  001370
10864  071530  001704
10865  071532  001214  001354          DT11:   $TESTN,TRAPPC
10866  071536  001214  001116  004364  DT12:   $TESTN,$ERRPC,HMR2,HMR3,HCS1,HCS2,HDC,HDA
10867  071544  004366  004336  004340
10868  071552  004356  004346
10869
10870                                  .SBTTL  ERROR DATA FORMATS
10871
10872  071556  000002                  DF1:    2
10873  071560     002     000                  .BYTE 2,0
10874  071562  066521                         DH2
10875  071564     006     000                  .BYTE 6,0
10876
```

```
10877  071566  000003                DF2:    3
10878  071570     002      000               .BYTE 2,0
10879  071572  066521                        DH2
10880  071574     006      000               .BYTE 6,0
10881  071576  066563                        DH3
10882  071600     001      000               .BYTE 1,0
10883
10884  071602  000001                DF3:    1
10885  071604     002      000               .BYTE   2,0
10886  071606  000002                DF4:    2
10887  071610     002      000               .BYTE   2,0
10888  071612  070101                        DH31
10889  071614     006      000               .BYTE   6,0
10890
10891  071616  000003                DF5:    3
10892  071620     000      000               .BYTE 0,0
10893  071622  066504                        DH1
10894  071624     002      000               .BYTE 2,0
10895  071626  066604                        DH5
10896  071630     007      000               .BYTE 7,0
10897
10898  071632  000003                DF6:    3
10899  071634     000      000               .BYTE 0,0
10900  071636  066504                        DH1
10901  071640     002      000               .BYTE 2,0
10902  071642  066653                        DH6
10903  071644     006      000               .BYTE 6,0
10904
10905  071646  000002                DF7:    2
10906  071650     002      000               .BYTE 2,0
10907  071652  066735                        DH7
10908  071654     007      000               .BYTE 7,0
10909
10910  071656  000004                DF8:    4
10911  071660     000      000               .BYTE 0,0
10912  071662  066504                        DH1
10913  071664     002      000               .BYTE 2,0
```

```
10914  071666  066521                  DH2
10915  071670     006      000         .BYTE 6,0
10916  071672  066572                  DH4
10917  071674     001      000         .BYTE 1,0
10918
10919  071676  000004            DF9:  4
10920  071700     000      000         .BYTE 0,0
10921  071702  066504                  DH1
10922  071704     002      000         .BYTE 2,0
10923  071706  066521                  DH2
10924  071710     006      000         .BYTE 6,0
10925  071712  066572                  DH4
10926  071714     002      000         .BYTE 2,0
10927
10928  071716  000003            DF10: 3
10929  071720     000      000         .BYTE 0,0
10930  071722  066504                  DH1
10931  071724     002      000         .BYTE 2,0
10932  071726  066521                  DH2
10933  071730     006      000         .BYTE 6,0
10934
10935  071732  000004            DF11: 4
10936  071734     000      000         .BYTE 0,0
10937  071736  066504                  DH1
10938  071740     002      000         .BYTE 2,0
10939  071742  066521                  DH2
10940  071744     006      000         .BYTE 6,0
10941  071746  066563                  DH3
10942  071750     001      000         .BYTE 1,0
10943  071752  000003            DF12: 3
10944  071754     000      000         .BYTE 0,0
10945  071756  066504                  DH1
10946  071760     002      000         .BYTE 2,0
10947  071762  070241                  DH36
10948  071764     006      000         .BYTE 6,0
10949  071766  000003            DF13: 3
10950  071770     000      000         .BYTE   0,0
10951  071772  066504                  DH1
10952  071774     002      000         .BYTE   2,0
10953  071776  066735                  DH7
10954  072000     007      000         .BYTE   7,0
10955
10956  072002  000002            DF14: 2
10957  072004     002      000         .BYTE   2,0
10958  072006  070452                  DH40
10959  072010     006      000         .BYTE   6,0
10960
10961
10962  072012  000003            DF15: 3
10963  072014     000      000         .BYTE   0,0
10964  072016  066504                  DH1
10965  072020     002      000         .BYTE   2,0
10966  072022  066735                  DH7
10967  072024     007      000         .BYTE   7,0
10968
10969  072026  000002            DF16: 2
```

```
10970  072030     000     000                .BYTE 0,0
10971  072032  066504                        DH1
10972  072034     002     000                .BYTE 2,0
10973
10974  072036  000004              DF17:     4
10975  072040     000     000                .BYTE   0,0
10976  072042  070675                        DH44
10977  072044     000     000                .BYTE   0,0
10978  072046  066504                        DH1
10979  072050     002     000                .BYTE   2,0
10980  072052  066521                        DH2
10981  072054     006     000                .BYTE   6,0
10982  072056  000004              DF18:     4
10983  072060     000     000                .BYTE   0,0
10984  072062  066504                        DH1
10985  072064     002     000                .BYTE   2,0
10986  072066  066735                        DH7
10987  072070     007     000                .BYTE   7,0
10988  072072  066572                        DH4
10989  072074     001     000                .BYTE   1,0
10990  072076  000004              DF19:     4
10991  072100     000     000                .BYTE   0,0
10992  072102  066504                        DH1
10993  072104     002     000                .BYTE   2,0
10994  072106  066735                        DH7
10995  072110     007     000                .BYTE   7,0
10996  072112  066572                        DH4
10997  072114     002     000                .BYTE   2,0
10998  072116  000002              DF21:     2
10999  072120     002     000                .BYTE   2,0
11000  072122  071206                        DH52
11001  072124     005     000                .BYTE   5,0
```

```
11002                                   ;;*****************************************************************
11003                                   .SBTTL  TYPE ERROR ROUTINE
11004                                   ;*ENTRY JSR PC,TYP ERR
11005                                   ;*RETURN RTS PC
11006                                   ;*
11007                                   ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
11008                                   ;*ERROR IS TO BE REPORTED. IT THEN USES THE "ERROR TABLE" ($ERRTB)
11009                                   ;*ENTRY TO DEFINE WHAT INFORMATION IS TO BE REPORTED CONCERNING
11010                                   ;*THE ERROR.
11011                                   ;;*****************************************************************
11012  072126  104413                  TYPERR: SAVREG
11013  072130  113700  001114                  MOVB    $ITEMB,R0       ;ENTER ERROR NUMBER
11014  072134  042700  177400                  BIC     #177400,R0      ;CLEAR SIGN EXTENSION
11015  072140  005300                          DEC     R0              ;FORM INDEX FOR ERROR TABLE
11016  072142  006300                          ASL     R0
11017  072144  006300                          ASL     R0
11018  072146  006300                          ASL     R0
11019  072150  062700  004454          1$:     ADD     #$ERRTB,R0      ;FORM ADDRESS OF ERROR ENTRY
11020  072154  012037  072170                  MOV     (R0)+,2$        ;GET EM POINTER
11021  072160  001404                          BEQ     3$              ;BRANCH IF THERE ISN'T ONE
11022  072162  104401  001205                  TYPE    ,$CRLF          ;TYPE CARRIAGE RETURN LINE FEED
11023  072166  104401                          TYPE                    ;TYPE ERROR MESSAGE (EM)
11024  072170  000000                  2$:     .WORD   0               ;EM POINTER GOES HERE
11025  072172  012037  072206          3$:     MOV     (R0)+,4$        ;GET DH POINTER
11026  072176  001404                          BEQ     5$              ;BRANCH IF THERE ISN'T ONE
11027  072200  104401  001205                  TYPE    ,$CRLF          ;TYPE CR-LF
11028  072204  104401                          TYPE                    ;TYPE DATA HEADER
11029  072206  000000                  4$:     .WORD   0               ;DH POINTER GOES HERE
11030  072210  012001                  5$:     MOV     (R0)+,R1        ;GET DT POINTER
11031  072212  001455                          BEQ     20$             ;BRANCH IF THERE ARE NONE
11032  072214  005004                          CLR     R4              ;SET INDENT SWITCH
11033  072216  012000                          MOV     (R0)+,R0        ;GET DF POINTER
11034  072220  012002                          MOV     (R0)+,R2        ;STORE NUMBER OF DH'S
11035  072222  001446                          BEQ     17$             ;DH NUM IS 0-BRANCH
11036  072224  005104                          COM     R4              ;NO INDENT
```

```
11037  072226  104401  001205            TYPE    $CRLF
11038  072232  112003                10$:  MOVB    (R0)+,R3        ;GET & STORE NUMBER OF DATA WORDS
11039  072234  105720                      TSTB    (R0)+           ;BUMP PAST FORMAT WORD
11040  072236  005703                      TST     R3              ;TEST IF ANY DATA FOR THIS HEADER
11041  072240  001407                      BEQ     14$             ;NO - SKIP DATA PRINT
11042  072242  013146                11$:  MOV     @(R1)+,-(SP)    ;PUT FIRST DATA WORD ON STACK
11043  072244  104402                      TYPOC                   ;TYPE IT
11044  072246  005303                      DEC     R3              ;MORE DATA WORDS
11045  072250  001403                      BEQ     14$             ;NO-BRANCH
11046  072252  104401  072402              TYPE    $SPACE2         ;TYPE SEPARATORS
11047  072256  000771                      BR      11$             ;LOOP
11048  072260  005302                14$:  DEC     R2              ;MORE DH'S?
11049  072262  003431                      BLE     20$             ;NO-BRANCH
11050  072264  104401  001205              TYPE    $CRLF
11051  072270  005760  000002              TST     2(R0)           ;ONLY A DH IN THIS REQUEST?
11052  072274  001404                      BEQ     15$             ;YES-BRANCH BYPASS INDENT
11053  072276  005104                      COM     R4              ;INDENT?
11054  072300  001002                      BNE     15$             ;NO-BRANCH
11055  072302  104401  072402              TYPE    $SPACE2         ;YES-TYPE SPACES
11056  072306  012037  072314        15$:  MOV     (R0)+,16$       ;GET NEXT DH POINTER
11057  072312  104401                      TYPE                    ;TYPE DH
11058  072314  000000                16$:  .WORD   0               ;DH POINTER GOES HERE
11059  072316  105710                      TSTB    (R0)            ;TYPE A DT?
11060  072320  001003                      BNE     21$             ;YES-BRANCH
11061  072322  062700  000002              ADD     #2,R0           ;INCREMENT DF POINTER
11062  072326  000754                      BR      14$             ;SEE IF END OF DF BLOCK
11063  072330  104401  001205        21$:  TYPE    $CRLF
11064  072334  005704                      TST     R4              ;INDENT?
11065  072336  001335                      BNE     10$             ;NO-BRANCH
11066  072340  104401  072402        17$:  TYPE    $SPACE2         ;YES-TYPE SPACES
11067  072344  000732                      BR      10$             ;LOOP
11068  072346  104414                20$:  RESREG
11069
11070  072350  032777  010000  106562      BIT     #SW12,@SWR      ;ABORT DRV AFTER 20 ERRORS?
11071  072356  001410                      BEQ     25$             ;BR IF NO
11072  072360  023727  001103  000024      CMP     $ERFLG,#20.     ;ELSE SEE IF 20 ERRORS
11073  072366  001004                      BNE     25$             ;BR IF NO
11074  072370  012706  001100              MOV     #STACK,SP       ;ELSE RESTORE STK
11075  072374  000137  047164              JMP     $EOP            ;AND DROP DRIVE
11076  072400  000207                25$:  RTS     PC
11077  072402  020040     000        SPACE2: .ASCIZ/ /             ;2 SPACES
```

```
                                ;  ODT-11 -- V005A
                                ;
                                ; DEC-11-UODPA-A-LA
                                ;
                                ; COPYRIGHT 1969,1970,1972
                                ; DIGITAL EQUIPMENT CORPORATION
                                ; MAYNARD, MASSACHUSETTS 01754
                                        .ENABL  ABS,AMA
072406                                  .EVEN
072466                                  .=.+60
000000                          R0      =       %0      ; REGISTER
000001                          R1      =       %1      ;  NAMING
000002                          R2      =       %2      ;   CONVENTIONS
000003                          R3      =       %3
000004                          R4      =       %4
000005                          R5      =       %5
000006                          SP      =       %6
000007                          PC      =       %7
177776                          ST      =       177776          ;STATUS REGISTER
                                ;
000014                          O.TVEC  =       14              ;TRT VECTOR LOCATION
000340                          O.STM   =       340             ;PRIORITY MASK - STATUS REGISTER
000020                          O.TBT   =       20              ;T-BIT MASK - STATUS REGISTER
000003                          TRT     =       000003          ;TRT INSTRUCTION
000006                          RTT     =       000006          ;RTT INSTRUCTION
                                ;
                                ; R5 IS USUALLY CONSIDERED SAFE, THE CURRENT ADDRESS WORD
                                ; RESIDES IN IT.  AFTER A BREAKPOINT, IT IS SET TO ZERO, AND SEARCH
                                ; OPERATIONS LEAVE IT RANDOMLY FILLED.  OTHERWISE, IT SHOULD NOT
                                ; BE USED EXCEPT FOR JSR'S AND THE CURRENT ADDRESS POINTER (CAD).
                                ;
177562                          O.RDB   =       177562  ;R DATA BUFFER
177560                          O.RCSR  =       177560  ;R C/SR
177566                          O.TDB   =       177566  ;T DATA BUFFER
177564                          O.TCSR  =       177564  ;T C/SR
                                ;
                                ;
                                ; INITIALIZE ODT
                                ;  USE O.ODT FOR A NORMAL ENTRY
                                ;  USE O.ODT+2 TO RESTART ODT - WIPING OUT ALL BREAKPOINTS
                                ;  USE O.ODT+4 TO RE-ENTER (I.E. - FAKE A BREAKPOINT)
                                ;
072466  000413                  O.ODT:  BR      O.STRT          ;NORMAL ENTRY
072470  000417                          BR      O.RST           ;RESTART
072472  013737  177776  072446  O.ENTR: MOV     ST,O.UST        ;RE-ENTER  --  SAVE STATUS
072500  013737  000016  177776          MOV     O.TVEC+2,ST     ;SET UP LOCAL STATUS
Z 072506  010737  072444                  MOV     PC,O.UPC        ;FAKE THE PC
072512  000137  073644                  JMP     O.BK1
                                ;
072516  012706  072426          O.STRT: MOV     #O.URO,SP       ;SET UP STACK
072522  010637  072442                  MOV     SP,O.USP        ;FAKE THE SAVED STACK
072526  000414                          BR      O.RST1          ;CLEAR BREAKPOINT TABLES
072530  004037  074052          O.RST:  JSR     0,O.SVR         ;SAVE REGISTERS
072534  013777  072464  177716          MOV     O.UIN,@O.ADR1   ;REMOVE THE BREAKPOINT
072542  113704  072450                  MOVB    O.PRI,R4        ;GET ODT PRIORITY
072546  106004                          RORB    R4              ;SHIFT
```

```
 11134  072550  106004                          RORB    R4              ; INTO
 11135  072552  106004                          RORB    R4              ;  POSITION
 11136  072554  110437  177776                  MOVB    R4,ST           ;STORE IN STATUS
Z 11137 072560  000127                  0.RST1: JMP     (PC)+
 11138  072562  000403                          BR      0.45
 11139  072564  012737  000002  073554          MOV     #RTI,0.RTIT     ;SET TO RTI IF 11/20 OR /05
 11140  072572  105037  074473          0.45:   CLRB    0.P             ;DISALLOW PROCEED
 11141  072576  012737  000340  000016          MOV     #0.STM,0.TVEC+2 ;STATUS WORD TO TRT VECTOR + 2
 11142  072604  012737  073634  000014          MOV     #0.BRK,0.TVEC   ;PC TO TRT VECTOR
 11143  072612  000447                          BR      0.RALL          ;CLEAR BREAKPOINT TABLES
 11144                                  ;
 11145                                  ; SPECIAL NAME HANDLER
 11146                                  ;  DEPENDS UPON THE EXPLICIT ORDER OF THE TWO TABLES 0.TL AND 0.URO
 11147                                  ;
 11148  072614  004537  074274          0.REGT: JSR     5,0.GET         ;SPECIAL NAME, GET ONE MORE CHARACTER
 11149  072620  012704  074517                  MOV     #0.TL,R4        ;TABLE START ADDRESS
 11150  072624  120024                  0.RSP:  CMPB    R0,(R4)+        ;IS THIS THE CORRECT CHARACTER?
 11151  072626  001413                          BEQ     0.SP            ;JUMP IF YES
 11152  072630  022704  074525                  CMP     #0.TL+0.LG,R4   ;IS THE SEARCH DONE?
 11153  072634  101373                          BHI     0.RSP           ;BRANCH IF NOT
 11154  072636  042700  177770                  BIC     #177770,R0      ;MASK OFF OCTAL
 11155  072642  010004                          MOV     R0,R4
 11156  072644  006304                  0.SP1:  ASL     R4
 11157  072646  062704  072426                  ADD     #0.URO,R4       ;GENERATE ADDRESS
 11158  072652  005202                          INC     R2              ;SET FOUND FLAG
 11159  072654  000444                          BR      0.SCAN          ;GO FIND NEXT CHARACTER
 11160  072656  162704  074510          0.SP:   SUB     #0.TL-7,R4      ;CORRECT CONSTANT
 11161  072662  000770                          BR      0.SP1
 11162                                  ;
 11163                                  ; ← HANDLER - OPEN INDEXED ON THE PC
 11164                                  ;
 11165  072664  004737  074420          0.ORPC: JSR     PC,0.TCLS
 11166  072670  010502                          MOV     R5,R2           ;CURRENT ADDRESS IN R2
 11167  072672  061202                          ADD     @R2,R2          ;COMPUTE
 11168  072674  006202                          ASR     R2              ;MOVE ONE BIT TO CARRY
 11169  072676  103421                          BCS     0.ERR           ;ERROR IF ODD NUMBER
 11170  072700  006302                          ASL     R2              ;RESTORE WORD
 11171  072702  005722                          TST     (R2)+           ; AND INCREMENT BY TWO
 11172  072704  010205                          MOV     R2,R5           ;UPDATE CAD
 11173  072706  000137  073160                  JMP     0.OP2           ;GO FINISH UP
 11174                                  ;
 11175                                  ; B HANDLER - SET AND REMOVE BREAKPOINTS
 11176                                  ;
 11177  072712  005702                  0.BKPT: TST     R2              ;IF NO NUMBER TYPED
 11178  072714  001406                          BEQ     0.RALL          ;  REMOVE BREAKPOINT
 11179  072716  006204                          ASR     R4              ;CHECK IF ODD
 11180  072720  103410                          BCS     0.ERR           ;JUMP IF ODD
 11181  072722  006304                          ASL     R4              ;RESTORE ONE BIT
 11182  072724  010437  072460                  MOV     R4,0.ADR1       ;SET A BREAKPOINT
 11183  072730  000412                          BR      0.DCD
 11184  072732  012737  074534  072460  0.RALL: MOV     #0.TRTC,0.ADR1  ;CLEAR BREAKPOINT
 11185  072740  000406                          BR      0.DCD
 11186                                  ;
 11187                                  ; COMMAND DECODER - ODT11
 11188                                  ;
 11189                                  ;   REGISTERS R0-R4 MAY BE USED,
```

```
11190                                   ;    REGISTER R5 WILL BE CONSIDERED SAFE
11191                                   ;
11192  072742  052705  000001          O.ERR:  BIS     #1,R5           ;CLOSE EVERYTHING
11193  072746  012700  000077                  MOV     #'?,R0          ;  ? TO BE TYPED
11194  072752  004537  074352                  JSR     5,O.FTYP        ; OUTPUT ?
11195  072756  004537  074452          O.DCD:  JSR     5,O.CRLS        ;TYPE  <CR><LF>*
11196  072762  005004                  O.DCD1: CLR     R4              ; R4 CONTAINS THE CONVERTED OCTAL
11197  072764  005002                          CLR     R2              ; R2 IS THE NUMBER FOUND FLAG
11198  072766  004537  074274          O.SCAN: JSR     5,O.GET         ;GET A CHAR, RETURN IN R0
11199  072772  022700  000060                  CMP     #'0,R0          ;COMPARE WITH ASCII 0
11200  072776  101013                          BHI     O.CLGL          ;CHECK LEGALITY IF NON-NUMERIC
11201  073000  022700  000067                  CMP     #'7,R0          ;COMPARE WITH ASCII 7
11202  073004  103410                          BLO     O.CLGL          ;CHECK LEGALITY IF NOT OCTAL
11203  073006  042700  177770                  BIC     #177770,R0      ;CONVERT TO BCD
11204  073012  006304                          ASL     R4              ; MAKE ROOM
11205  073014  006304                          ASL     R4              ;  IN
11206  073016  006304                          ASL     R4              ;    R4
11207  073020  060004                          ADD     R0,R4           ;PACK THREE BITS IN R4
11208  073022  005202                          INC     R2              ;R2 HAS NUMERIC FLAG
11209  073024  000760                          BR      O.SCAN          ;   AND TRY AGAIN
11210  073026  005001          O.CLGL: CLR     R1              ;CLEAR INDEX
11211  073030  120061  074503  O.LGL1: CMPB    R0,O.LGCH(R1)   ;DO THE CODES MATCH?
11212  073034  001405                  BEQ     O.LGL2          ;JUMP IF YES
11213  073036  005201                  INC     R1              ; SET INDEX FOR NEXT SEARCH
11214  073040  020127  000014          CMP     R1,#O.CLGT      ;IS THE SEARCH DONE?
11215  073044  103336                  BHIS    O.ERR           ;    OOPS!
11216  073046  000770                  BR      O.LGL1          ;RE-LOOP
11217  073050  006301          O.LGL2: ASL     R1              ;MULTIPLY BY TWO
11218  073052  000171  073056          JMP     @O.LGDR(R1)     ;GO TO PROPER ROUTINE
11219                                   ;
11220  073056  073106          O.LGDR: O.WRD   ;  /    OPEN WORD
11221  073060  073140                  O.CRET  ;  CARRIAGE RETURN    CLOSE
11222  073062  072614                  O.REGT  ;  $  REGISTER OPS
11223  073064  073450                  O.GO    ;  G  GO TO ADDRESS K
11224  073066  073152                  O.OP1   ;  <LF>  MODIFY, CLOSE, OPEN NEXT
11225  073070  072664                  O.ORPC  ;  ←  OPEN RELATED, INDEX - PC
11226  073072  073204                  O.BACK  ;  ↑  OPEN PREVIOUS
11227  073074  073214                  O.OFST  ;  O  OFFSET
11228  073076  073272                  O.WSCH  ;  W  SEARCH WORD
11229  073100  073266                  O.EFF   ;  E  SEARCH EFFECTIVE ADDRESS
11230  073102  072712                  O.BKPT  ;  B  BREAKPOINTS
11231  073104  073556                  O.PROC  ;  P  PROCEED
11232           000030          O.LGL   =       .-O.LGDR        ;LGL MUST EQUAL 2X CHLGT ALWAYS
11233                                   ;
11234                                   ; PROCESS / - OPEN WORD
11235                                   ;
11236  073106  005702          O.WRD:  TST     R2              ;GET VALUE IF R2 IS NON-ZERO
11237  073110  001410                  BEQ     O.WRDA          ;SKIP OTHERWISE
11238  073112  010405                  MOV     R4,R5           ;   PUT VALUE IN CAD
11239  073114  006205          O.WRD1: ASR     R5              ;MOVE ONE BIT TO CARRY
11240  073116  103711          O.ERR2: BCS     O.ERR           ;JUMP IF ODD ADDRESS
11241  073120  006305                  ASL     R5              ;RESTORE THE CARRY BIT
11242  073122  011500                  MOV     @R5,R0          ;GET CONTENTS OF WORD
11243  073124  004537  074210          JSR     5,O.CADV        ;GO GET AND TYPE OUT @CAD
11244  073130  000714                  BR      O.DCD1          ;GO BACK TO DECODER
11245  073132  042705  000001  O.WRDA: BIC     #1,R5           ;CLEAR CLOSED BIT
```

```
11246  073136  000766                  BR      0.WRD1          ;GO BACK TO MAIN-LINE
11247                                  ;
11248                                  ; PROCESS CARRIAGE RETURN
11249                                  ;
11250  073140  004737  074420          0.CRET: JSR     PC,0.TCLS       ;CLOSE LOCATION
11251  073144  052705  000001                  BIS     #1,R5           ;CLOSE EVERYTHING
11252  073150  000702                          BR      0.DCD           ;RETURN TO DECODER
11253                                  ;
11254                                  ; PROCESS <LF>, OPEN NEXT WORD
11255                                  ;
11256  073152  004737  074420          0.OP1:  JSR     PC,0.TCLS       ;CLOSE PRESENT CELL
11257  073156  005725                          TST     (R5)+           ;GENERATE NEW ADDRESS
11258  073160  004537  074444          0.OP2:  JSR     5,0.CRLF        ;<CR><LF>
11259  073164  010500                          MOV     R5,R0           ;NUMBER TO TYPE
11260  073166  004537  074210                  JSR     5,0.CADV        ; TYPE OUT ADDRESS
11261  073172  012700  000057                  MOV     #'/,R0          ;TYPE A /
11262  073176  004537  074352                  JSR     5,0.FTYP
11263  073202  000744                          BR      0.WRD1          ;GO PROCESS IT
11264                                  ;
11265                                  ; PROCESS ↑, OPEN PREVIOUS WORD
11266                                  ;
11267  073204  004737  074420          0.BACK: JSR     PC,0.TCLS
11268  073210  005745                          TST     -(R5)           ;GENERATE NEW ADDRESS
11269  073212  000762                          BR      0.OP2           ;GO DO THE REST
11270                                  ;
11271                                  ; PROCESS 0, COMPUTE OFFSET
11272                                  ;
11273  073214  006205                  0.OFST: ASR     R5              ;GET LOW ORDER BIT
11274  073216  103737                          BCS     0.ERR2          ;ERROR IF CLOSED
11275  073220  006305                          ASL     R5              ;RESTORE WORD
11276  073222  012700  000040                  MOV     #' ,R0          ;TYPE ONE BLANK
11277  073226  004537  074352                  JSR     5,0.FTYP        ; AS A SEPARATOR
11278  073232  160504                          SUB     R5,R4           ;COMPUTE
11279  073234  005304                          DEC     R4
11280  073236  005304                          DEC     R4              ;     16 BIT OFFSET
11281  073240  010400                          MOV     R4,R0           ;TYPE A
11282  073242  010402                          MOV     R4,R2           ;SAVE R4
11283  073244  004537  074210                  JSR     5,0.CADV        ;NUMBER IN R0 - WORD MODE
11284  073250  010200                          MOV     R2,R0
11285  073252  006200                          ASR     R0              ;DIVIDE BY TWO
11286  073254  103402                          BCS     0.OF1           ;BRANCH IF ODD
11287  073256  004537  074210                  JSR     5,0.CADV        ;NUMBER IN R0 - BYTE MODE
11288  073262  000137  072762          0.OF1:  JMP     0.DCD1          ;ALL DONE
11289                                  ;
11290                                  ; SEARCHES - $MSK   HAS THE MASK
11291                                  ;            $MSK+2 HAS THE FWA
11292                                  ;            $MSK+4 HAS THE LWA
11293                                  ;
11294  073266  005201                  0.EFF:  INC     R1              ;SET EFFECTIVE SEARCH
11295  073270  000401                          BR      0.WDS
11296  073272  005001                  0.WSCH: CLR     R1              ;SET WORD SEARCH
11297  073274  005702                  0.WDS:  TST     R2              ;CHECK FOR OBJECT FOUND
11298  073276  001621                  0.ERR1: BEQ     0.ERR           ;ERROR IF NO OBJECT
11299  073300  013702  072454                  MOV     0.MSK+2,R2      ;SET ORIGIN
11300  073304  013705  072452                  MOV     0.MSK,R5        ;SET MASK
11301  073310  005105                          COM     R5              ;AND COMPLEMENT IT
```

```
11302  073312  020237  072456          0.WDS2: CMP     R2,0.MSK+4      ; IS THE SEARCH ALL DONE?
11303  073316  101217                          BHI     0.DCD           ;  YES
11304  073320  011200                          MOV     @R2,R0          ; GET OBJECT
11305  073322  005701                          TST     R1              ;NO
11306  073324  001027                          BNE     0.EFF1          ;BRANCH IF EFFECTIVE SEARCH
11307  073326  010046                          MOV     R0,-(SP)
11308  073330  010403                          MOV     R4,R3           ;EXCLUSIVE OR
11309  073332  040400                          BIC     R4,R0           ; IS DONE
11310  073334  042603                          BIC     (SP)+,R3                ;  IN A VERY
11311  073336  050003                          BIS     R0,R3           ;     FANCY MANNER HERE
11312  073340  040503                          BIC     R5,R3           ;AND RESULT WITH MASK
11313  073342  00..16          0.WDS3: BNE     0.WDS4          ;RE-LOOP IF NO MATCH
11314  073344  010446                          MOV     R4,-(SP)        ;REGISTERS R2,R4, AND R5 ARE SAFE
11315  073346  004537  074444                 JSR     5,0.CRLF        ;TYPE <CR,LF>
11316  073352  010200                          MOV     R2,R0           ;GET READY TO TYPE
11317  073354  004537  074210                 JSR     5,0.CADV        ;  TYPE ADDRESS
11318  073360  012700  000057                 MOV     #'/,R0          ;SLASH TO R0
11319  073364  004537  074352                 JSR     5,0.FTYP        ;TYPE IT
11320  073370  011200                          MOV     @R2,R0          ;GET CONTENTS
11321  073372  004537  074210                 JSR     5,0.CADV        ;TYPE CONTENTS
11322  073376  012604                          MOV     (SP)+,R4        ; RESTORE R4
11323  073400  005722          0.WDS4: TST     (R2)+           ;INCREMENT TO NEXT CELL AND
11324  073402  000743                          BR      0.WDS2          ;     RETURN
11325  073404  020004          0.EFF1: CMP     R0,R4           ; IS (X)=K?
11326  073406  001755                          BEQ     0.WDS3          ;TYPE IF EQUAL
11327  073410  010003                          MOV     R0,R3           ;(X) TO R3
11328  073412  060203                          ADD     R2,R3           ;(X)+X
11329  073414  005203                          INC     R3
11330  073416  005203                          INC     R3              ;(X)+X+2
11331  073420  020304                          CMP     R3,R4           ;IS (X)+X+2=K?
11332  073422  001747                          BEQ     0.WDS3          ;BRANCH IF EQUAL
11333  073424  042700  177400                 BIC     #177400,R0      ;WIPE OUT EXTRANEOUS BITS
11334  073430  110000                          MOVB    R0,R0           ;EXTEND SIGN
11335  073432  000257                          CCC
11336  073434  006300                          ASL     R0              ;MULTIPLY BY TWO
11337  073436  005200                          INC     R0              ;ADD TWO
11338  073440  005200                          INC     R0
11339  073442  060200                          ADD     R2,R0           ;ADD PC
11340  073444  020004                          CMP     R0,R4           ;IS THE RESULT A PROPER REL. BRANCH?
11341  073446  000735                          BR      0.WDS3
11342
11343                          ;
                               ; PROCESS G - GO
11344                          ;
11345  073450  105037  074473          0.GO:   CLRB    0.P             ;DISALLOW PROCEED
11346  073454  006204                          ASR     R4              ;CHECK LOW ORDER BIT
11347  073456  103617                          BCS     0.ERR2          ;ERROR IF ODD NUMBER
11348  073460  006304                          ASL     R4              ;RESTORE WORD
11349  073462  010437  072444                 MOV     R4,0.UPC        ;SET UP NEW PC
11350  073466  112737  000340  177776         MOVB    #0.STM,ST       ;SET HIGH PRIORITY
11351  073474  004537  074142                 JSR     5,0.RSTT        ;RESTORE TELETYPE
11352  073500  105037  074472          0.TBIT: CLRB    0.T             ;CLEAR BOTH
11353  073504  042737  000020  072446         BIC     #0.TBT,0.UST    ;  T-BIT FLAGS
11354  073512  017737  176742  072464         MOV     @0.ADR1,0.UIN   ;SAVE INSTRUCTION
11355  073520  013777  074534  176732         MOV     0.TRTC,@0.ADR1  ;REPLACE WITH TRAP
11356  073526  012600          0.GO2:  MOV     (SP)+,R0        ;RESTORE
11357  073530  012601                          MOV     (SP)+,R1        ; R0
```

```
11358  073532  012602                        MOV     (SP)+,R2        ;   THRU
11359  073534  012603                        MOV     (SP)+,R3
11360  073536  012604                        MOV     (SP)+,R4        ;
11361  073540  012605                        MOV     (SP)+,R5        ;    R5
11362  073542  012606                        MOV     (SP)+,SP        ; AND SP
11363  073544  013746  072446                MOV     0.UST,-(SP)     ;    AND STATUS
11364  073550  013746  072444                MOV     0.UPC,-(SP)     ;    AND PC
11365  073554  000006              0.RTIT:   RTT                     ;CHANGED TO RTI FOR 11/20 AND /05
11366                                ;
11367                                ; PROCESS P - PROCEED
11368                                ;   ONLY ALLOWED AFTER A BREAKPOINT
11369                                ;
11370  073556  105737  074473       0.PROC:   TSTB    0.P             ;CHECK LEGALITY OF PROCEED
11371  073562  001645                        BEQ     0.ERR1          ;NOT LEGAL
11372  073564  105037  074473                CLRB    0.P             ;CLEAR PROCEED FLAG
11373  073570  005702                        TST     R2              ;WAS COUNT SPECIFIED?
11374  073572  001402                        BEQ     0.PR1           ;NO
11375  073574  010437  072462                MOV     R4,0.CT         ;YES, PUT AWAY COUNT
11376  073600  112737  000340  177776 0.PR1:   MOVB    #0.STM,ST       ;FORCE HIGH PRIORITY
11377  073606  004537  074142                JSR     5,0.RSTT        ;RESTORE TTY
11378  073612  112737  000340  177776 0.C1:    MOVB    #0.STM,ST       ;SET HIGH PRIORITY
11379  073620  105237  074472                INCB    0.T             ;SET T-BIT FLAG
11380  073624  052737  000020  072446        BIS     #0.TBT,0.UST    ;SET T-BIT
11381  073632  000735                        BR      0.G02
11382                                ;
11383                                ; BREAKPOINT HANDLER
11384                                ;  A TRT BREAKPOINT CAUSES 0.BRK TO BE ENTERED, WHICH SAVES
11385                                ;  VARIOUS ODDS AND ENDS, FINDS OUT IF THE BREAKPOINT WAS LEGAL,
11386                                ;  AND GIVES CONTROL TO THE COMMAND DECODER
11387                                ;
11388  073634  012637  072444       0.BRK:    MOV     (SP)+,0.UPC     ;PRIORITY IS 7 UPON ENTRY
11389  073640  012637  072446                MOV     (SP)+,0.UST     ;SAVE STATUS AND PC
11390  073644  004037  074052       0.BK1:    JSR     0,0.SVR         ;SAVE VARIOUS REGISTERS
11391  073650  105737  074472                TSTB    0.T             ;CHECK FOR T-BIT SET
11392  073654  001311                        BNE     0.TBIT          ;JUMP IF SET
11393  073656  013777  072464  176574        MOV     0.UIN,@0.ADR1   ;REMOVE BREAKPOINTS
11394  073664  105737  072450                TSTB    0.PRI           ;CHECK IF PRIORITY
11395  073670  100003                        BPL     0.BK2           ;  IS AS SAME AS USER PGM
11396  073672  113705  072446                MOVB    0.UST,R5        ;PICK UP USER UST IF SO
11397  073676  000407                        BR      0.BK3           ;AND DON'T COMPUTE THE PRIORITY
11398  073700  113705  072450       0.BK2:    MOVB    0.PRI,R5        ;OTHERWISE PICK UP ACTUAL PRIORITY
11399  073704  000257                        CCC                     ;CLEAR CARRY
11400  073706  106005                        RORB    R5              ;SHIFT LOW ORDER BITS
11401  073710  106005                        RORB    R5              ;  INTO
11402  073712  106005                        RORB    R5              ;    HIGH ORDER
11403  073714  106005                        RORB    R5              ;      POSITION
11404  073716  110537  177776       0.BK3:    MOVB    R5,ST           ;PUT THE STATUS AWAY WHERE IT BELONGS
11405  073722  013705  072444                MOV     0.UPC,R5        ;GET PC, IT POINTS TO THE TRT
11406  073726  005745                        TST     -(R5)           ;SUBTRACT TWO
11407  073730  010537  072444                MOV     R5,0.UPC        ;FROM THE USER'S PC
11408  073734  020537  072460                CMP     R5,0.ADR1       ;COMPARE WITH LIST
11409  073740  001417                        BEQ     0.B2            ;JUMP IF FOUND
11410  073742  004537  074110                JSR     5,0.SVTT        ;SAVE TELETYPE STATUS
11411  073746  004537  074444                JSR     5,0.CRLF
11412  073752  012704  074476                MOV     #0.BD,R4        ;ERROR, NOTHING FOUND
11413  073756  012703  074477                MOV     #0.BD+1,R3
```

```
11414  073762  004537  074336          JSR     5,0.TYPE        ;OUTPUT "BE" FOR BAD ENTRY
11415  073766  010500                  MOV     R5,R0
11416  073770  042737  000020  072446  BIC     #0.TBT,0.UST    ;CLEAR OUT ANY POSSIBLE FAKE T-BIT
11417  073776  000420                  BR      0.B3            ; AND CONTINUE
11418  074000  005337  072462  0.B2:   DEC     0.CT
11419  074004  003302                  BGT     0.C1            ;JUMP IF REPEAT
11420  074006  012737  000001  072462  MOV     #1,0.CT         ;RESET COUNT TO 1
11421  074014  105237  074473          INCB    0.P             ;ALLOW PROCEED
11422  074020  004537  074110          JSR     5,0.SVTT        ;SAVE TELETYPE STATUS, R4 IS SAFE
11423  074024  012700  000102          MOV     #'B,R0
11424  074030  004537  074352          JSR     5,0.FTYP        ;TYPE "B"
11425  074034  013700  072460          MOV     0.ADR1,R0       ;GET ADDRESS OF BREAK
11426  074040  004537  074210  0.B3:   JSR     5,0.CADV        ;TYPE ADDRESS
11427  074044  005005                  CLR     R5              ;CLEAR CAD
11428  074046  000137  072756          JMP     0.DCD           ;GO TO DECODER
11429                                  ;
11430                                  ; SAVE REGISTERS R0-R6 IN INTERNAL STACK
11431                                  ;
11432  074052  012637  074470  0.SVR:  MOV     (SP)+,0.XXX     ;PICK REGISTER FROM STACK AND SAVE
11433  074056  010637  072442          MOV     SP,0.USP        ;SAVE USER STACK ADDRESS
11434  074062  012706  072442          MOV     #0.USP,SP       ;SET TO INTERNAL STACK
11435  074066  010546                  MOV     R5,-(SP)        ;SAVE
11436  074070  010446                  MOV     R4,-(SP)        ; REGISTERS
11437  074072  010346                  MOV     R3,-(SP)        ;1
11438  074074  010246                  MOV     R2,-(SP)        ; THRU
11439  074076  010146                  MOV     R1,-(SP)        ;    5
11440  074100  013746  074470          MOV     0.XXX,-(SP)     ;PUT SAVED REGISTER ON STACK
11441  074104  005746                  TST     -(SP)
11442  074106  000200                  RTS     R0
11443                                  ;
11444                                  ; SAVE TELETYPE STATUS
11445                                  ;
11446  074110  113737  177560  074474  0.SVTT: MOVB   0.RCSR,0.CSR1  ;SAVE R C/SR
11447  074116  113737  177564  074475          MOVB   0.TCSR,0.CSR2  ;SAVE T C/SR
11448  074124  105037  177560                  CLRB   0.RCSR         ;CLEAR ENABLE AND MAINTENANCE
11449  074130  105037  177564                  CLRB   0.TCSR         ;  BITS IN BOTH C/SR
11450  074134  004537  074444                  JSR    5,0.CRLF       ;TYPE <CR,LF>
11451  074140  000205                          RTS    R5
11452                                  ;
11453                                  ; RESTORE TELETYPE STATUS
11454                                  ;
11455  074142  004537  074444  0.RSTT: JSR     5,0.CRLF                ;<CR,LF> BEFORE RESTORING
11456  074146  105737  177564          TSTB    0.TCSR          ;WAIT READY ON PRINTER
11457  074152  100375                  BPL     .-4
11458  074154  032737  004000  177560  BIT     #4000,0.RCSR    ;CHECK BUSY FLAG ON READER
11459  074162  001403                  BEQ     0.RSE1          ;SKIP READY LOOP IF NOT BUSY
11460  074164  105737  177560          TSTB    0.RCSR          ;WAIT READY
11461  074170  100375                  BPL     .-4             ;    ON READER
11462  074172  113737  074474  177560  0.RSE1: MOVB   0.CSR1,0.RCSR  ;RESTORE
11463  074200  113737  074475  177564          MOVB   0.CSR2,0.TCSR  ;  THE STATUS REGISTERS
11464  074206  000205                          RTS    R5
11465                                  ;
11466                                  ; TYPE OUT CONTENTS OF WORD OR BYTE WITH ONE TRAILING SPACE
11467                                  ; WORD IS IN R0
11468                                  ;
11469  074210  010246          0.CADV: MOV     R2,-(SP)        ;SAVE R2
```

```
11470  074212  012704  074533          MOV     #0.BUF+6,R4     ;BUFFER START ADDRESS
11471  074216  012746  000060          MOV     #'0,-(SP)               ;CONSTANT ASCII 0
11472  074222  010002          0.SPC:  MOV     R0,R2           ; GET
11473  074224  042702  177770          BIC     #177770,R2      ;    OCTAL CHARACTER
11474  074230  061602                  ADD     @SP,R2          ;CONVERT TO ASCII
11475  074232  110244                  MOVB    R2,-(R4)        ;STORE IN BUFFER
11476  074234  006200                  ASR     R0              ;SHIFT THIS MESS
11477  074236  006200                  ASR     R0              ; RIGHT
11478  074240  006200                  ASR     R0              ;   THREE WHOLE PLACES
11479  074242  020427  074526          CMP     R4,#0.BUF+1     ;DONE?
11480  074246  101365                  BHI     0.SPC           ;   NO
11481  074250  042700  177776          BIC     #177776,R0      ;GET LAST BIT
11482  074254  062600                  ADD     (SP)+,R0                ;CONVERT TO ASCII
11483  074256  110044                  MOVB    R0,-(R4)        ;AND PUT IT AWAY
11484  074260  012703  074533          MOV     #0.BUF+6,R3     ;LWA
11485  074264  004537  074336          JSR     5,0.TYPE        ;TYPE WHOLE STRING OF CHARACTERS
11486  074270  012602                  MOV     (SP)+,R2        ;RESTORE R2
11487  074272  000205                  RTS     R5
11488                          ;
11489                          ; GENERAL CHARACTER INPUT ROUTINE
11490                          ; CHARACTER INPUT GOES TO R0
11491                          ;
11492  074274  105737  177560  0.GET:  TSTB    0.RCSR          ;WAIT FOR
11493  074300  100375                  BPL     .-4             ;   INPUT FROM KEYBOARD
11494  074302  113700  177562          MOVB    0.RDB,R0        ;GET A CHARACTER
11495  074306  004537  074352          JSR     5,0.FTYP        ;ECHO CHARACTER
11496  074312  042700  177600          BIC     #177600,R0      ;STRIP OFF PARITY FROM CHARACTER
11497  074316  001766                  BEQ     0.GET           ;IGNORE NULLS
11498  074320  122700  000040          CMPB    #40,R0          ;CHECK FOR SPACES
11499  074324  001763                  BEQ     0.GET           ;IGNORE NULLS
11500  074326  122700  000073          CMPB    #';,R0          ;CHECK FOR SEMI-COLON
11501  074332  001760                  BEQ     0.GET           ;IGNORE THEM IF FOUND
11502  074334  000205                  RTS     R5
11503                          ;
11504                          ; GENERAL CHARACTER OUTPUT ROUTINE
11505                          ;  ADDRESS OF FIRST BYTE IN R4,
11506                          ;  ADDRESS OF LAST BYTE IN R3, (R3)>(R4)
11507                          ;
11508  074336  020304          0.TYPE: CMP     R3,R4           ;CHECK FOR COMPLETION
11509  074340  103426                  BLO     0.TYP1          ;  EXIT WHEN DONE
11510  074342  112400                  MOVB    (R4)+,R0        ;GET A CHARACTER
11511  074344  004537  074352          JSR     5,0.FTYP        ;TYPE ONE CHARACTER
11512  074350  000772                  BR      0.TYPE          ;LOOP UNTIL DONE
11513                          ;
11514                          ; TYPE ONLY ONE CHARACTER (CONTAINED IN R0)
11515                          ;
11516  074352  105737  177564  0.FTYP: TSTB    0.TCSR          ;CHECK STATUS
11517  074356  100375                  BPL     .-4             ;WAIT UNTIL READY
11518  074360  110037  177566          MOVB    R0,0.TDB        ;TYPE ONE CHARACTER
11519  074364  120037  000045          CMPB    R0,@#45         ;IS CHAR TO BE FILLED?
11520  074370  001012                  BNE     0.TYP1          ;NO
11521  074372  113746  000044          MOVB    @#44,-(SP)      ;YES, INIT THE COUNT
11522  074376  105737  177564  0.TYP2: TSTB    0.TCSR
11523  074402  100375                  BPL     0.TYP2
11524  074404  105037  177566          CLRB    0.TDB           ;GENERATE NULL FILLER
11525  074410  105316                  DECB    @SP
```

```
11526  074412  003371                  BGT     0.TYP2
11527  074414  005726                  TST     (SP)+           ;POP STACK
11528  074416  000205          0.TYP1: RTS     R5
11529                          ;
11530                          ; CLOSE WORD OR BYTE AND EXIT,
11531                          ; UPON ENTERING, R2 HAS NUMERIC FLAG, R4 HAS CONTENTS
11532                          ;
11533  074420  006205          0.TCLS: ASR     R5              ;GET LOW ORDER BIT
11534  074422  103405                  BCS     0.TC            ;JUMP IF ALREADY CLOSED
11535  074424  006305                  ASL     R5
11536  074426  005702                  TST     R2              ;IF NO NUMBER WAS TYPED THERE IS
11537  074430  001401                  BEQ     0.CLS1          ;NO CHANGE TO THE OPEN CELL
11538  074432  010415                  MOV     R4,@R5          ;STORE WORD
11539  074434  000207          0.CLS1: RTS     PC
11540  074436  005746          0.TC:   TST     -(SP)           ;POP EXTRA CELL FROM STACK
11541  074440  000137  072742          JMP     0.ERR           ;AND SCREAM BLOODY MURDER
11542                          ;
11543                          ; 0.CRLF - TYPE <CR,LF>
11544                          ; 0.CRLS - TYPE <CR,LF>*
11545                          ;
11546  074444  012703  074501  0.CRLF: MOV     #0.CR+1,R3      ;LWA <CR,LF>
11547  074450  000402                  BR      0.CRS
11548  074452  012703  074502  0.CRLS: MOV     #0.CR+2,R3      ;LWA <CR,LF>*
11549  074456  012704  074500  0.CRS:  MOV     #0.CR,R4        ;FWA
11550  074462  004537  074336          JSR     5,0.TYPE        ;TYPE SOMETHING
11551  074466  000205                  RTS     R5
11552                          ;
11553  074470  000000          0.XXX:  .WORD   0               ;TEMPORARY STORAGE
11554  074472     000          0.T:    .BYTE   0               ;  T-BIT FLAG
11555  074473     000          0.P:    .BYTE   0               ;PROCEED FLAG = 0 IF PROCEED NOT ALLOWED
11556                                                          ;               = 1 IF PROCEED ALLOWED
11557  074474     000          0.CSR1: .BYTE   0               ;SAVE CELL - R C/SR
11558  074475     000          0.CSR2: .BYTE   0               ;SAVE CELL - T C/SR
11559                          ;
11560                                  .EVEN
11561  074476  042502          0.BD:   .WORD   "BE
11562                          ;
11563  074500     015          0.CR:   .BYTE   015     ;  <CR>
11564  074501     012                  .BYTE   012     ;  <LF>
11565  074502     052                  .BYTE   '*      ;  *
11566                          ;
11567  074503     057          0.LGCH: .BYTE   '/      ;  /
11568  074504     015                  .BYTE   015     ;  CARRIAGE RETURN
11569  074505     044                  .BYTE   '$      ;  $
11570  074506     107                  .BYTE   'G      ;  G
11571  074507     012                  .BYTE   012     ;  <LF>
11572  074510     137                  .BYTE   '<-     ;  <-
11573  074511     136                  .BYTE   '^      ;  ^
11574  074512     117                  .BYTE   'O      ;  O
11575  074513     127                  .BYTE   'W      ;  W
11576  074514     105                  .BYTE   'E      ;  E
11577  074515     102                  .BYTE   'B      ;  B
11578  074516     120                  .BYTE   'P      ;  P
11579             000014  0.CLGT  =       .-0.LGCH                ;TABLE LENGTH
11580                          ;
11581  074517     123          0.TL:   .BYTE   'S      ;DO       1
```

```
11582  074520  000120                  .BYTE   'P      ;NOT      2
11583  074521  000115                  .BYTE   'M      ;CHANGE   3
11584  074522  000000                  .BYTE   0       ;THE      4
11585  074523  000000                  .BYTE   0       ;ORDER    5
11586  074524  000102                  .BYTE   'B      ;HERE     6
11587          000006          O.LG    =       .-O.TL
11588                          ;
11589  074525                  O.BUF:                          ;6 CHAR. BUFFER WITH
11590          074533          .       =       .+6
11591  074533  000040                  .BYTE                   ;TRAILING BLANK
11592                                  .EVEN
11593                          ;
11594  074534  000003          O.TRTC: TRT                     ;TRACE TRAP PROTOTYPE
11595                          ;
11596                          ;THE ORDER OF THE FOLLOWING ENTRIES IS CRITICAL
11597                          ;
11598          072426          .       =       O.ODT-40
11599  072426  000000          O.URO:  0       ;USER R0
11600  072430  000000                  0       ;       R1
11601  072432  000000                  0       ;       R2
11602  072434  000000                  0       ;       R3
11603  072436  000000                  0       ;       R4
11604  072440  000000                  0       ;       R5
11605  072442  000000          O.USP:  0       ;USER SP
11606  072444  000000          O.UPC:  0       ;USER PC
11607  072446  000000          O.UST:  0       ;USER ST
11608  072450  000007          O.PRI:  7       ;ODT PRIORITY
11609  072452  000000          O.MSK:  0       ;MASK
11610  072454  000000                  0       ;LOW LIMIT
11611  072456  000000                  0       ;HIGH LIMIT
11612                          ;
11613                          ; BREAK POINT LISTS, ADR1 = ADDRESS OF BREAKPOINT,CT = COUNT,
11614                          ;   UIN = CONTENTS
11615                          ;
11616  072460  000000          O.ADR1: 0
11617  072462  000000          O.CT:   0
11618  072464  000000          O.UIN:  0
```

```
 11619          000001                          .END
```

```
ABASE = 177440            468     509     533#
ACDW1 = 000000            468     511
ACDW2 = 000000            468     512
ACLO  = 000010            259#
ACPUOP= 000C00            468     483
ACT11   004416           1392#   2609*
ADDW0 = 000000            468     513
ADDW1 = 000000            468     514
ADDW10= 000000            468     523
ADDW11= 000000            468     524
ADDW12= 000000            468     525
ADDW13= 000000            468     526
ADDW14= 000000            468     527
ADDW15= 000000            468     528
ADDW2 = 000000            468     515
ADDW3 = 000000            468     516
ADDW4 = 000000            468     517
ADDW5 = 000000            468     518
ADDW6 = 000000            468     519
ADDW7 = 000000            468     520
ADDW8 = 000000            468     521
ADDW9 = 000000            468     522
ADEVCT= 000000            468     474
ADEVM = 000000            468     510
AENV  = 000000            468     479
AENVM = 000000            468     480
AFATAL= 000000            468     471
AMADR1= 000000            468     496
AMADR2= 000000            468     500
AMADR3= 000000            468     503
AMADR4= 000000            468     506
AMAMS1= 000000            468     490
AMAMS2= 000000            468     498
AMAMS3= 000000            468     501
AMAMS4= 000000            468     504
AMSGAD= 000000            468     476
AMSGLG= 000000            468     477
AMSGTY= 000000            468     470
AMTYP1= 000000            468     491
AMTYP2= 000000            468     499
AMTYP3= 000000            468     502
AMTYP4= 000000            468     505
APASS = 000000            468     473
APRIOR= 000000            468
APTCSU= 000040           9165    9337#
APTENV= 000001           9112    9158    9293    9335#
APTSIZ= 000200           2542    9334#
APTSPO= 000100           9160    9295    9336#
ASWREG= 000000            468     481
ATESTN= 000000            468     472
ATTN    004326            617#   2925    4143    8275    8294    8320
AUNIT = 000000            468     475
AUSWR = 000000            468     482
AVECT1= 000000            468     507
AVECT2= 000000            468     508
BADINT  053126           8933#
```

```
BADTMO  053040          2685   8898#
BAI    = 000020          222#  4673   4840
BA16   = 000400          208#
BA17   = 001000          209#
BIT0   = 000001          151#   205    237    256   8844
BIT00  = 000001          141#   151
BIT01  = 000002          140#   150
BIT02  = 000004          139#   149
BIT03  = 000010          138#   148
BIT04  = 000020          137#   147
BIT05  = 000040          136#   146
BIT06  = 000100          135#   145
BIT07  = 000200          134#   144
BIT08  = 000400          133#   143   9047
BIT09  = 001000          132#   142   9055   9123
BIT1   = 000002          150#   238
BIT10  = 002000          131#   210    228    247    279    293    307    320    334   9100
BIT11  = 004000          130#   211    229    248    265    280    294    308    321    335   9062
BIT12  = 010000          129#   212    230    249    281    295    309    322    336
BIT13  = 020000          128#   213    231    250    266    282    296    310    323    337   9107
BIT14  = 040000          127#   214    232    251    267    283    297    311    324    338    351   8610   9033
BIT15  = 100000          126#   215    216    233    252    268    284    354   3612   8610
BIT2   = 000004          149#   239    258
BIT3   = 000010          148#   221    240    259
BIT4   = 000020          147#   222    241    260    273    301    328
BIT5   = 000040          146#   223    242    261    274    288    302    315    329
BIT6   = 000100          145#   206    224    243    262    275    289    303    316    330
BIT7   = 000200          144#   207    225    244    263    276    290    304    317    331
BIT8   = 000400          143#   208    226    245    264    277    291    305    318    332
BIT9   = 001000          142#   209    227    246    278    292    306    319    333   3091   4108
BPTVEC= 000014           158#
BSE    = 000200          244#
BSERR   001474           600#  5843*  5858*  5867*  5868   5885*  6179
BSE20   002314           607#  5893
BSE22   003314           608#  5845   5882   5900   8713
BYP     050242          2754   2776   2780   2795   2847   2855   2873   2877   2881   2885   8347#
BYPT16  001360           549#  2470*  2475*  2480*  2485*  2490*  2494*  3767
BYT16   007644           368   2474#
BYT16A  007664           372   2479#
CALADD  001406           562#  4989*  5317*  6188*  6219*  6240   6345   6348   6366   6409   6416*  8606   8612
                        8614
CALDIF  001400           559#  3885*  3901   3933*  3964   5104*  5431*  5642*  6223*  6823*  7202*  7596*  7996*
                       10843  10856
CCLR   = 100000          216#  3322   3345   3351   3360   3371   3383   3394   3403   3416   3429   3464   3497
                        3536   3564   3622   3640   3649   3660   3672   3683   3692   3718   3729   3738   3788
                        3806   3815   3826   3838   3849   3858   3893   3907   3941   3970   4022   4040   4049
                        4060   4072   4083   4092   4129   4173   4180   4197   4242   4251   4269   4280   4289
                        4371   4382   4393   4402   4413   4427   4438   4447   4468   4477   4488   4500   4511
                        4520   4611   4620   4632   4696   4705   4717   4770   4779   4791   4867   4876   4930
                        4939   4950   5008   5017   5119   5146   5155   5173   5184   5193   5204   5336   5345
                        5437   5478   5487   5498   5510   5521   5530   5665   5690   5699   5717   5728   5737
                        5934   5945   5956   5965   5976   5990   6001   6010   6031   6040   6051   6063   6074
                        6083   6256   6286   6295   6313   6324   6333   6344   6381   6390   6592   6617   6626
                        6644   6655   6664   6675   6838   6858   6867   6885   6896   6905   6983   7004   7013
                        7031   7042   7051   7217   7237   7246   7264   7275   7284   7362   7383   7392   7410
                        7421   7430   7611   7631   7640   7658   7669   7678   7756   7777   7786   7804   7815
```

```
                    7824    8018    8025    8045    8052    8069
CCYL    001374      557*    3778*   3879*   3880    3886    3927*   3928    3934    6970*   7349*   7743*   8691*   8699*
                    8705*   8717
CDT   = 002000      210*    3229
CERR  = 100000      215*    2734    2833    3192    3242    3289    4237    4592    4689    4751    4850    4901    5001
                    5043    5141    5228    5271    5329    5371    5473    5555    5598    5695    5782    5855    6115
                    6205    6281    6374    6438    6495    6509    6612    6699    6750    6853    6937    6999    7083
                    7132    7232    7316    7378    7462    7511    7574    7626    7710    7772    7856    7905    7978
                    8384
CFMT  = 010000      212*    4845    4857    4897    4920    8409
CKMR2   052504      3032    3037    3053    3069    3086    3324    3335    3354    3364    3397    3407    3435    3445
                    3468    3478    3501    3511    3643    3653    3686    3696    3733    3742    3809    3819    3852
                    3862    4043    4053    4086    4096    4153    4245    4255    4283    4293    4374    4396    4406
                    4416    4442    4451    4471    4481    4514    4524    4614    4624    4699    4709    4773    4783
                    4870    4880    4933    4943    5011    5021    5113    5123    5149    5159    5187    5197    5339
                    5349    5448    5458    5481    5491    5524    5534    5659    5669    5693    5703    5731    5741
                    5937    5959    5969    5979    6005    6014    6034    6044    6077    6087    6250    6260    6289
                    6299    6327    6337    6384    6394    6586    6596    6620    6630    6658    6668    6832    6842
                    6861    6871    6899    6909    6977    6987    7007    7017    7045    7055    7211    7221    7240
                    7250    7278    7288    7356    7366    7386    7396    7424    7434    7605    7615    7634    7644
                    7672    7682    7750    7760    7780    7790    7818    7828    8020    8029    8788*
CKMR3   052566      3041    3057    3073    3329    3339    3357    3367    3400    3410    3439    3449    3472    3482
                    3505    3515    3646    3656    3689    3699    3736    3745    3812    3822    3855    3865    4046
                    4056    4089    4099    4162    4248    4258    4286    4296    4377    4399    4409    4419    4445
                    4474    4484    4517    4527    4617    4627    4702    4712    4776    4786    4873    4883    4936
                    4946    5014    5024    5116    5126    5152    5162    5190    5200    5342    5352    5452    5461
                    5484    5494    5527    5537    5662    5672    5696    5706    5734    5744    5940    5962    5972
                    5982    6008    6037    6047    6080    6090    6253    6263    6292    6302    6330    6340    6387
                    6397    6589    6599    6623    6633    6661    6671    6835    6845    6864    6874    6902    6912
                    6980    6990    7010    7020    7048    7058    7214    7224    7243    7253    7281    7291    7359
                    7369    7389    7399    7427    7437    7608    7618    7637    7647    7675    7685    7753    7763
                    7783    7793    7821    7831    8023    8032    8807#
CKSWR = 104407      9032    9096    9122    9861#
CLEAR = 000005      193*    3385    3674    3940    3309    3972    4074    4182    4271    4384    4502    5175    5512
                    5719    5947    6065    6315    6646    6887    7033    7266    7412    7660    7806    8054
CLKOF   052450      3428    3463    3496    3529    3584    3589    8774#
CLKON   052342      3419    3454    3487    3520    8749#
CLOCK   052414      2639    8763#
CLRFLG  047334      2559    8131#
COE   = 001000      246#
COUNT   001412      566*    3417*   3452*   3485*   3518*   8764*   8766*
CR    = 000015      66#     9204    9214    9873    9878    9881    9885    9893    9901    9911    9919    9924    9929
                    9934    9940    9946    9955    9962    9967    9973    9981    9986    9991    9999    10003   10007
                    10013   10015   10020   10025   10029   10031   10035   10039   10045   10049   10057   10066   10070
                    10083   10102   10110   10347   10436   10442
CRLF  = 000200      67#     9175    9214
CTO   = 004000      211#
CYLADD  001404      561*    3377    3666    3721    3832    3896    3944    4066    4219    4301    4494    4638    4723
                    4797    4956    5207    5208    5279    5504    5607    5749    6057    6348    6678    6917    7063
                    7296    7442    7690    7836    8039    8529*   8530*   8531*   8532*   8533*   8534*
CYLDIF  001402      560*    3373    3662    3725    3828    3901    3953    3959    3964    4062    4263    4490    4634
                    4719    4793    4952    5130    5167    5465    5500    5676    5711    6053    6267    6307    6603
                    6638    6879    7025    7258    7404    7652    7798    8035    8514*   8515*   8516*   8517*   8518*
                    8519*   8520    8522*
CYLINV  046376      7185    7947#
DATAO   001462      593*    4682    4811
```

```
DATAD1  001464            594#
DATA1   001466            595#  4841   4964
DCK    = 100000           252#
DCLO   = 000020           260#
DDISP  = 177570            73#   443   2530
DDPCH   004414           1391#  2605*  2742  2975
DDT    = 000400           264#  3215   3236
DDUMP   004412           1390#  2572*  8131
DF1     071556           1432   1438   1444  1450  1455  1461  1467  1473  1541  1547  1558  1563  1568
                         1573   1578   1583  1588  1593  1598  1603  1608  1668  1718  1743  1768  1793
                         1923   1928   1933  1998  10872#
DF10    071716           1479   1495   1500  1536  1553  1633  1638  1653  1663  1713  1828  1833  1838
                         1843   1848   1858  1873  1878  1883  1888  1913  1918  1938  1958  1963  1983
                         1988   2028   2033  2043  2048  2063  2068  2118  2123  2158  2173  2178  2183
                         2188   2223   2238  2259  2329  2339  2414  2419  2425  2430  2435  2450  2455
                         2465   10928#
DF11    071732           1485   1490   1853  1863  1868  1978  1993  2023  2233  2264  10935#
DF12    071752           1943   10943#
DF13    071766           2078   2083   10949#
DF14    072002           2088   2445   10956#
DF15    072012           2053   2058   2193  2198  2203  10962#
DF16    072026           2038   2073   10969#
DF17    072036           2213   2218   10974#
DF18    072056           1818   10982#
DF19    072076           1823   10990#
DF2     071566           1648   10877#
DF21    072116           2440   10998#
DF3     071602           2228   10884#
DF4     071606           2460   10886#
DF5     071616           1613   1618   1643  2208  10891#
DF6     071632           1623   1628   2113  2128  2153  2243  2248  2254  2269  2274  2279  2284  2309
                         2314   2334   10898#
DF7     071646           1658   10905#
DF8     071656           1505   1515   1525  1673  1683  1693  1703  1723  1733  1748  1758  1773  1783
                         1798   1808   1893  1903  1948  1968  2003  2013  2093  2103  2133  2143  2163
                         2168   2289   2299  2319  2374  2379  2394  2399  2404  2409  10910#
DF9     071676           1510   1520   1530  1678  1688  1698  1708  1728  1738  1753  1763  1778  1788
                         1803   1813   1898  1908  1953  1973  2008  2018  2098  2108  2138  2148  2294
                         2304   2324   2344  2349  2354  2359  2364  2369  2384  2389  10919#
DH1     066504           1430   1436   1442  1448  1453  1459  1465  1471  1539  1545  1556  1561  1566
                         1571   1576   1581  1586  1591  1596  1601  1606  1646  1656  1666  1716  1741
                         1766   1791   1921  1926  1931  1996  2086  2438  2443  2458  10576#  10893  10900
                         10912  10921  10930  10937  10945  10951  10964  10971  10978  10984  10992
DH10    067103           1691   1696   1701  1706  1711  2061  2066  10624#
DH11    067132           1721   1726   1731  1736  10628#
DH12    067174           1746   1751   1756  1761  10634#
DH13    067242           1771   1776   1781  1786  10641#
DH14    067313           1611   1616   1621  1626  1796  1801  1846  1861  1946  1951  2241  2246  2332
                         10648#
DH15    067346           1641   1806   1811  1851  1866  10653#
DH16    067413           1816   1821   10660#
DH17    067466           1631   1636   1651  1661  1856  1981  2156  2161  2166  2171  2176  2181  2236
                         2382   2387   10668#
DH18    067512           1477   1483   1488  1493  1498  2463  10672#
DH19    067537           1551   1826   10676#
DH2     066521           10579#  10874  10879  10914  10923  10932  10939  10980
```

```
DH20    067562          1831   10680#
DH21    067615          1534    1836    1871    1876   10685#
DH22    067643          1961    1966    1971    1976    2231    2327    2342    2347    2372    2377   10689#
DH23    067675          2226   10694#
DH24    067716          2317    2322   10697#
DH25    067743          1881    1886    1891    1896    1901    1906    1911    1916    2111    2116    2121    2206    2252
                       10701#
DH26    067766          2186    2191    2196    2201   10705#
DH27    070016         10709#
DH3     066563         10585#  10881   10941
DH30    070047          2041    2046    2051    2056    2337    2362    2367    2402    2407    2453   10714#
DH31    070101         10719#  10888
DH33    070143          2091    2096    2101    2106    2126   10725#
DH34    070167          1936   10729#
DH35    070213          1941   10733#
DH36    070241         10737#  10947
DH37    070323          1956   10746#
DH38    070374          2001    2006    2011    2016    2021    2026    2031    2423   10753#
DH39    070417          2076    2081    2352    2357    2392    2397    2412    2417    2448   10757#
DH4     066572         10587#  10916   10925   10988   10996
DH40    070452         10762#  10958
DH41    070527          2131    2136    2141    2146    2151    2257    2433   10770#
DH42    070554          2211   10774#
DH43    070625          2216   10781#
DH44    070675          2221   10788#  10976
DH45    070754          2036   10796#
DH46    071033          2071   10804#
DH48    071104          2262    2267    2272    2277    2282    2287    2292    2297    2302    2307    2312   10811#
DH5     066604         10589#  10895
DH51    071153          1986    1991   10818#
DH52    071206         10823#  11000
DH6     066653         10596#  10902
DH7     066735         10605#  10907   10953   10966   10986   10994
DH8     067004          1503    1508    1513    1518    1523    1528   10612#
DH9     067030          1671    1676    1681    1686    1841    2428   10616#
DI    = 040000           214#
DISPLA  001142           443#   2530*   2538*   9077*   9099*
DISPRE  000174           363#   2538
DLT   = 100000           233#   4603    4762    4912    5053    5239    5381    5566    5862    6520    6710    6948    7094
                        7327    7473    7721    7867
DLY     050224          3431    8338#   8341
DMD   = 000040           274#
DOSEEK  042034          7184    7186#
DOTIM   004450          1410#   2638*   2647*   3342
DRA   = 000001           256#   2770    2863
DRDY  = 000200           263#
DRIVS   004422          1394#   2718*   2744*   2761    2851*   2964    8094    8175*
DRIV0   004424          1399#   2620    2720    2819    2905    8158
DRIV1   004426          1400#
DRIV2   004430          1401#
DRIV3   004432          1402#
DRIV4   004434          1403#
DRIV5   004436          1404#
DRIV6   004440          1405#
DRIV7   004442          1406#
DROT  = 000040           261#
```

```
CRPAR = 000010            240*
DRVMSK= 000007            220*  2737  2836
DRVPTR  001366            553*  2620* 2969  2979*
DSC   = 040000            267*
DSWR  = 177570             72*   442  2529
DTE   = 010000            249*  5859
DTYE  = 000040            242*  3218  3239
DT1     071260           1431  1437  1443  1449  1454  1460  1466  1472  1478  1494  1499  1504  1509
                         1514  1519  1524  1529  1535  1540  1546  1552  1557  1562  1567  1572  1577
                         1582  1587  1592  1597  1602  1607  1632  1637  1652  1662  1667  1672  1677
                         1682  1687  1692  1697  1702  1707  1712  1717  1722  1727  1732  1737  1742
                         1747  1752  1757  1762  1767  1772  1777  1782  1787  1792  1797  1802  1807
                         1812  1827  1832  1837  1842  1847  1857  1872  1877  1882  1887  1892  1897
                         1902  1907  1912  1917  1922  1927  1932  1937  1947  1952  1957  1962  1967
                         1972  1982  1987  1997  2002  2007  2012  2017  2027  2032  2037  2042  2047
                         2062  2067  2072  2092  2097  2102  2107  2117  2122  2132  2137  2142  2147
                         2157  2162  2167  2172  2177  2182  2187  2212  2217  2222  2237  2258  2288
                         2293  2298  2303  2318  2323  2328  2338  2343  2348  2353  2358  2363  2368
                         2373  2378  2383  2388  2393  2398  2403  2408  2413  2418  2424  2429  2434
                         2449  2454  2464 10833*
DT10    071514           2444 10862*
DT11    071532           2227 10865*
DT12    071536           2459 10866*
DT2     071304           1484  1489  1647  1852  1862  1867  1977  1992  2022  2232  2263 10837*
DT3     071326           1612  1617  1642  2207 10840*
DT4     071350           1622  1627  2112  2127  2152  2268  2273  2278  2283  2308  2313 10843*
DT5     071370           1657  1817  1822  2052  2057  2077  2082  2192  2197  2202 10846*
DT6     071416           1942 10850*
DT7     071436           2087 10853*
DT8     071456           2242  2247  2253  2333 10856*
DT9     071476           2439 10859*
D.ACLO= 000100            316*
D.BRHM= 000100            303*  3052  3334  3363  3406  3444  3477  3510  3652  3695  3741  3818  3861
                         4052  4095  4254  4292  4405  4450  4480  4523  4623  4708  4782  4879  4942
                         5020  5122  5158  5196  5348  5457  5490  5533  5668  5702  5740  5968  6013
                         6043  6086  6259  6298  6336  6393  6595  6629  6667  6841  6870  6908  6986
                         7016  7054  7220  7249  7287  7365  7395  7433  7614  7643  7681  7759  7789
                         7827  8028
D.CART= 000400            305*  3052  3334  3363  3406  3444  3477  3510  3652  3695  3741  3818  3861
                         4052  4095  4254  4292  4405  4450  4480  4523  4623  4708  4782  4879  4942
                         5020  5122  5158  5196  5348  5457  5490  5533  5668  5702  5740  5968  6013
                         6043  6086  6259  6298  6336  6393  6595  6629  6667  6841  6870  6908  6986
                         7016  7054  7220  7249  7287  7365  7395  7433  7614  7643  7681  7759  7789
                         7827  8028
D.DDT = 000400            291*  3212  3233
D.DOOR= 000200            304*  3052  3334  3363  3406  3444  3477  3510  3652  3695  3741  3818  3861
                         4052  4095  4254  4292  4405  4450  4480  4523  4623  4708  4782  4879  4942
                         5020  5122  5158  5196  5348  5457  5490  5533  5668  5702  5740  5968  6013
                         6043  6086  6259  6298  6336  6393  6595  6629  6667  6841  6870  6908  6986
                         7016  7054  7220  7249  7287  7365  7395  7433  7614  7643  7681  7759  7789
                         7827  8028
D.DRA = 000040            288*  3031  3036  3323  3353  3396  3434  3467  3500  3642  3685  3732  3808
                         3851  4042  4085  4152  4244  4282  4373  4395  4415  4441  4470  4513  4613
                         4698  4772  4869  4932  5010  5112  5148  5186  5338  5447  5480  5523  5658
                         5692  5730  5936  5958  5978  6004  6033  6076  6249  6288  6326  6383  6585
                         6619  6657  6831  6860  6898  6976  7006  7044  7210  7239  7277  7355  7385
```

```
                    7423    7604    7633    7671    7749    7779    7817    8019
D.DRDY= 000200       290#   2928    3353    3396    3523    3642    3685    3732    3808    3851    4042    4085    4152
                    4244    4282    4373    4395    4415    4470    4513    4613    4698    4772    4869    4932    5010
                    5148    5186    5338    5480    5523    5692    5730    5936    5958    5978    6033    6076    6288
                    6326    6383    6619    6657    6860    6898    7006    7044    7239    7277    7385    7423    7633
                    7671    7779    7817    8019
D.DROT= 020000       323#
D.DSC = 040000       297#   2931    3353    3642    3732    3808    4042    4152    4244    4373    4441    4470    5148
                    5480    5692    5936    6004    6033    6288    6619    6860    7006    7239    7385    7633    7779
                    8014    8019
D.FLT = 000200       317#   3280    3735    4161    4376    4444    5939    6007    8008    8022
D.FORM= 001000       292#   4861    4924
D.FWD = 002000       307#   3422    3444    3457    3490    3510    5122    5668    6259    6841    7220    7759
D.HOFL= 000200       331#
D.HOHM= 000040       302#   3052    3334    8562
D.IDAE= 000040       315#   4161    4193    8005    6022    8065
D.ILF = 000400       318#
D.LIMO= 020000       337#   8545
D.LORO= 010000       309#   3444    8579
D.MHO = 000400       332#
D.NMOV= 010000       336#
D.OFF = 002000       293#
D.PAR = 001000       319#   3264    3283    3735    4376    5939
D.PIP = 020000       296#   3434    3467    3500    4441    5112    5447    5658    6004    6249    6585    6831    6976
                    7210    7355    7604    7749    8011
D.PLO = 004000       335#
D.REV = 004000       308#   3377    4450    5457    6013    6595    6986    7365    7614
D.RTZ = 020000       310#   3477    3510    3633    3799    4033    5440    5457
D.SECT= 000020       328#
D.SKI = 002000       320#   4161    4444    6007    8022
D.SPIN= 010000       295#   3316    3323    3353    3396    3434    3467    3500    3539    3642    3685    3732    3808
                    3851    4042    4085    4152    4244    4282    4373    4395    4415    4441    4470    4513    4613
                    4698    4772    4869    4932    5010    5112    5148    5186    5338    5447    5480    5523    5658
                    5692    5730    5936    5958    5978    6004    6033    6076    6249    6288    6326    6383    6585
                    6619    6657    6831    6860    6898    6976    7006    7044    7210    7239    7277    7355    7385
                    7423    7604    7633    7671    7749    7779    7817    8019
D.SPLS= 010000       322#
D.SPOK= 001000       306#   3013    3363    3406    3444    3477    3510    3567    3652    3695    3741    3818    3861
                    4052    4095    4254    4292    4405    4450    4480    4523    4623    4708    4782    4879    4942
                    5020    5122    5158    5196    5348    5457    5490    5533    5668    5702    5740    5968    6013
                    6043    6086    6259    6298    6336    6393    6595    6629    6667    6841    6870    6908    6986
                    7016    7054    7220    7249    7287    7365    7395    7433    7614    7643    7681    7759    7789
                    7827    8028
D.SUNS= 040000       338#
D.TFOK= 000020       301#   3363    3406    3477    3510    3652    3695    3741    3818    3861    4052    4095    4254
                    4292    4405    4450    4480    4523    4623    4708    4782    4879    4942    5020    5122    5158
                    5196    5348    5457    5490    5533    5668    5702    5740    5968    6013    6043    6086    6259
                    6298    6336    6393    6595    6629    6667    6841    6870    6908    6986    7016    7054    7220
                    7249    7287    7365    7395    7433    7614    7643    7681    7759    7789    7827    8028
D.TIB = 002000       334#
D.UNLD= 040000       311#   4432    4450    5995    6013
D.UNS = 040000       324#
D.VV  = 000100       289#   3029    3036    3148    3162    3323    3353    3396    3434    3467    3500    3642    3685
                    3732    3808    3851    4042    4085    4152    4244    4282    4373    4395    4415    4441    4470
                    4513    4613    4698    4772    4869    4932    5010    5112    5148    5186    5338    5447    5480
                    5523    5658    5692    5730    5936    5958    5978    6004    6033    6076    6249    6288    6326
```

```
                    6383   6585   6619   6657   6831   6880   6898   6976   7006   7044   7210   7239   7277
                    7355   7385   7423   7604   7633   7671   7749   7779   7817   8019
D.WCUR= 000040       329#
D.WGAT= 000100       330#
D.WLE = 004000       321#
D.WRL = 004000       294#
D.XERR= 001000       333#
ECCW  = 020000       282#
ECH   = 000100       243#
EMTVEC= 000030       161#  2513#  2514#
EM1     061261      8184  10083#
EM10    062046      1464  10150#
EM11    062131      1470  10159#
EM12    062175      1476   1825   1830   1835   1840   1845   1850   1855   1880   1960   1985   2040   2075
                    2185  10166#
EM13    062233      1482   1660   1710   1860   1865   1885   1990   2020   2261  10172#
EM14    062255      1487  10176#
EM15    062302      1492  10180#
EM16    062334      1497   2281  10185#
EM17    062361      1502   1670   1690   1720   1745   1770   1795   1805   1815   1890   1965   2000   2090
                    2130   2160   2286   2316   2371   2391   2401  10189#
EM18    062402      1507   1675   1695   1725   1750   1775   1800   1810   1820   1895   1970   2005   2095
                    2135   2291   2321   2341   2351   2361   2381  10192#
EM19    062423      1512   1680   1700   1730   1755   1780   1900   1945   1955   2010   2100   2140   2165
                    2296   2376   2396   2406  10195#
EM2     061335      1429  10091#
EM20    062444      1517   1685   1705     35   1760   1785   1905   1950   2015   2105   2145   2301   2346
                    2356   2366   2386  10198#
EM21    062465      1533   2055   2080   2115   2155   2190  10201#
EM22    062507      1538  10205#
EM23    062546      1544  10211#
EM24    062631      1550  10220#
EM25    062664      1555  10225#
EM26    062712      1560  10229#
EM27    062732      1565  10232#
EM28    062753      1570  10235#
EM29    063000      1575  10239#
EM3     061436      8902  10102#
EM30    063047      1580  10246#
EM31    063103      1585  10251#
EM32    063125      1590   2271  10255#
EM33    063154      1595  10259#
EM34    063214      1600  10265#
EM35    063242      1605  10269#
EM36    063312      1610   1620   2110   2205   2240   2251   2311  10276#
EM37    063355      1615  10282#
EM38    063420      1625   2125   2150   2245   2306  10288#
EM39    063471      1630   1870   1910   2045   2060   2170   2411  10295#
EM4     061511      8937  10110#
EM40    063540      1635   1875   1915   2065  10302#
EM41    063576      1640   2336   2416  10307#
EM42    063647      1645  10314#
EM43    063723      1650  10322#
EM44    063762      1655  10328#
EM45    064044      1790  10337#
EM46    064203      1522  10354#
```

```
EM47    064224          1527   10357#
EM48    064245          1935    1940   10360#
EM49    064266          1765   10363#
EM5     061565          1435   10118#
EM50    064403          1715   10376#
EM51    064504          1740   10387#
EM52    064622          1920   10400#
EM53    064664          1925   10406#
EM54    064724          1930   10412#
EM55    064766          1975   10418#
EM56    065021          2225   10423#
EM57    065057          2457   10429#
EM58    065165          2452   10442#
EM59    065240          1995   10450#
EM6     061606          1441   10121#
EM60    065300          2025   10456#
EM61    065336          2030   10461#
EM62    065374          2195   10466#
EM63    065414          2050    2200   10469#
EM64    065435          1665   10472#
EM65    065503          2085   10479#
EM66    065525          2175   10483#
EM67    065561          2120   10488#
EM68    065620          2180    2235   10494#
EM69    065657          2220   10500#
EM7     061627          1447   10124#
EM71    065710          2230   10505#
EM72    065731          2035    2070   10508#
EM74    065756          2256   10512#
EM75    066003          2266   10516#
EM76    066031          2276   10520#
EM77    066052          2326   10523#
EM78    066105          2331   10528#
EM8     061716          1452   10134#
EM80    066172          2422   10537#
EM81    066222          2427   10541#
EM82    066250          2432   10545#
EM83    066275          2437    2442   10549#
EM84    066344          2447    2452   10556#
EM85    066374          1980   10560#
EM87    066441          2210    2215   10567#
EM9     061772          1458   10142#
ERRVEC= 000004           154#   2527    2528#   2539#   2546#   2547#   2556#   2557#   2623#   2627#   2634#   2667#   2685#
                        9038   9039#   9041#   9044#
ESEC    001424           572#   4324#   4325#   4335   10850
FATT1   050050          3347   3637    3803    4037    4199    4464    4550    6027    6147    8071    8291#
FATT2   050144          4235   5139    5269    5471    5596    5683    5780    6113    6279    6436    6493    6610    6748
                        6851   6997    7130    7230    7376    7509    7572    7624    7770    7903    7976    8318#
FHDMM   051336          4459   6022    8559#   8567
FHDTAB  051460          4992   5320    6191    6216    6360    8592#
FLIM    051260          4429   5992    8541#
FLOAD   051412          4461   6024    8576#
FMTE  = 000020           241#
FMT1    001456           590#   8602#   8603#   8604#   8609
FNS20   051034          8494#
FNS22   050744          4322    8474#
```

```
FORMAT  001454            589#  4991*  5319*  6190*  6221*  8602
FRCYL   001370            555#  3776*  3926   3927   3942   3944   3948   3987*  3988   3990   5102*  5429*  5640*
                         5753*  6224*  6417*  6568*  6723*  6809*  6820   6969   6970   7063   7075   7102   7188*
                         7199   7348   7349   7442   7454   7481   7582*  7593   7742   7743   7836   7848   7875
                         7986*  10843  10856  10862
FRDY    047632           2713   2727   2732   2826   2831   2913   2920   3159   3231   3312   3387   3555   3628
                         3676   3714   3794   3842   3889   3911   3937   3974   4028   4076   4125   4177   4184
                         4231   4273   4367   4386   4425   4504   4546   4590   4686   4749   4847   4859   4899
                         4922   4998   5041   5110   5177   5226   5265   5326   5369   5435   5514   5553   5592
                         5656   5721   5776   5852   5930   5949   5988   6067   6109   6143   6202   6247   6317
                         6371   6432   6489   6507   6583   6648   6697   6744   6829   6889   6935   6974   7035
                         7081   7126   7208   7268   7314   7353   7414   7460   7505   7568   7602   7662   7708
                         7747   7808   7854   7899   7972   8000   8049   8056   8238#  8241   8364   8379
FSO20   050660           8454#
FSO22   050574           4318   8434#
FTITLE  001364            551#  8142   8144*
GBA     047534           2591   8198#
GDRVS   047374           2589   8153#
GINT    047562           2593   8211#
GNS   = ******  U         362   9853   9854   9855   9856   9857   9859   9861   9862   9863   9864   9865   9866
                         9867
GO    = 000001            205#
GSTAT   050256           3012   3028   3051   3067   3083   3114   3147   3190   3279   3315   3333   3352   3362
                         3395   3405   3421   3433   3443   3456   3466   3476   3489   3499   3509   3522   3538
                         3566   3632   3641   3651   3684   3694   3731   3740   3798   3807   3817   3850   3860
                         4032   4041   4051   4084   4094   4105   4137   4151   4192   4243   4253   4281   4291
                         4372   4394   4404   4414   4428   4440   4449   4469   4479   4512   4522   4605   4612
                         4622   4688   4697   4707   4764   4771   4781   4849   4868   4878   4914   4931   4941
                         5000   5009   5019   5055   5121   5147   5157   5185   5195   5241   5328   5337   5347
                         5383   5439   5445   5456   5479   5489   5522   5532   5568   5667   5691   5701   5729
                         5739   5854   5935   5957   5967   5977   5991   6003   6012   6032   6042   6075   6085
                         6204   6258   6287   6297   6325   6335   6373   6382   6392   6522   6594   6618   6628
                         6656   6666   6712   6840   6859   6869   6897   6907   6950   6985   7005   7015   7043
                         7053   7096   7219   7238   7248   7276   7286   7329   7364   7384   7394   7422   7432
                         7475   7613   7632   7642   7670   7680   7723   7758   7778   7788   7816   7826   7869
                         8027   8064   8301   8305   8325   8329   8360#  8383   8394   8513   8528   8544   8561
                         8578
GTSWR = 104406           9859#
HASOF   004354           1365#  2922   2925   8259*  8275   10837
HBA     004344           1361#  8255*
HCS1    004336           1358#  2734   2833   3192   3242   3267   3286   3289   4237   4592   4689   4751   4850
                         4901   5001   5043   5141   5228   5271   5329   5371   5473   5555   5598   5685   5782
                         5855   6115   6205   6281   6374   6438   6495   6509   6612   6699   6750   6853   6937
                         6999   7083   7132   7232   7316   7378   7462   7511   7574   7626   7710   7772   7856
                         7905   7978   8252*  8384   10833  10837  10840  10846  10850  10866
HCS2    004340           1359#  2766   2768   2772   2784   2859   2861   2865   2868   3178   3195   5862   8253*
                        10833  10837  10840  10846  10850  10866
HDA     004346           1362#  8256*  8679   8680   10846  10853  10859  10862  10866
HDB     004360           1367#
HDC     004356           1366#  8260*  8691   8699   8702   8704*  8705   10840  10843  10856  10866
HDS     004350           1363#  2770   2863   3215   3236   8257*  10833  10837  10840  10846
HDTAB   001500            603#  4674   4682*  4985   5067   5313   5395   6194   6364   8594
HDWD    001472            598#  4812*  4813   4965*  4966   5069*  5071   5397*  5399   10853
HDI     001452            588#  8596*  8597*  8598*  8599*  8600*  8601*  8608
HERD    001450            587#  4574*  4583*  4584   4585*  4643*  4644   4665*  4676*  4677   4678*  4680   4729*
                         4730   4733*  4742*  4743   4744*  4802*  4803   4804*  4809   4820*  4821   4990*  5318*
```

```
                    6189*  6196*  6197   6198*  6212   6215*  6220*  6241*  6242   6243*  6361*  6362   6363*
                    6406*  6407   6413   6415*  6479*  6480   8596   8690*  8697*  8706*  8737
HER      004352     1364*  3218   3239   5859   8258*  10833  10837  10840  10846
HMR1     004362     1368*  8261*  8808
HMR2     004264     1369*  2736   2835   2928   2931   3013   3029   3084   3115   3148   3162   3212   3233
                    3316   3422   3457   3490   3523   3539   3567   3633   3799   4033   4432   4861   4924
                    5440   5995   8011   8014   8262*  8514   8562   8579   8793   10833  10837  10840  10843
                    10846  10850  10853  10856  10859  10862  10866
HMR3     004366     1370*  3089   3264   3280   3283   4106   4138   4193   8005   8008   8065   8263*  8335
                    8419   8529   8545   8814   10833  10837  10840  10843  10846  10850  10853  10856  10859
                    10862  10866
HOLD     047670     8242   8245   8252*
HPAT     004372     1372*  8265*
HPOS     004370     1371*  8264*
HT     = 000011       64*  9173   9214
HVRC   = 000400     245*
HWC      004342     1360*  8254*
HZ       001410     564*   3417   3452   3485   3518   8766
IDAE   = 002000     247*
IE     = 000100     206*
ILF    = 000001     237*
INTER    053256     8226   8978*
INVCYL   004314     610*   7995
IOTVEC= 000020      159*   2511*  2512*
IR     = 000100     224*
LCLKF    004444     1408*  2625*
LCVEC    001350     541*   2626
LF     = 000012       65*  9208   9214   9873   9878   9881   9885   9893   9901   9911   9919   9924   9929
                    9934   9940   9946   9955   9962   9967   9973   9981   9986   9991   9999   10003  10007
                    10013  10015  10020  10025  10029  10031  10035  10039  10045  10049  10057  10066  10070
                    10083  10102  10110  10347  10436  10442
LIMERR   001476     601*   6175   8541*  8549*
LKS      001346     539*   2624   8755*  8779*
LPFLG    001430     575*   3577   3585*  3590*  4361*  4553*  5285   5291*  5296*  5612   5617*  5622*  5798
                    5806*  5812*  5924*  6131   6150*  6454   6467*  6473*  6538*  6543*  6766   6774*  6780*
                    7148   7154*  7159*  7164*  7169*  7527   7533*  7538*  7543*  7548*  7921   7927*  7932*
                    7937*  7942*  8876*
MCLK   = 000400     277*
MDS    = 001000     227*   2766   2859
MDTST    007706     377    2484*
MDTSTA   007726     379    2489*
MEMBAS= 172100      545*   2549*
MEMERR   053214     2551   8964*
MEMPAR   010254     2546*
MEMVEC= 000114      544*   2551*  2552*
MERD   = 001000     278*
MESMSK= 000017      272*
MEWD   = 002000     279*
MIND   = 000200     276*
MODTST   001362     550*   2471*  2476*  2481*  2486*  2491*  2495*  6171   6235   6271   6402   6564   7183
MSG1     057132     8145   9873*
MSG10    060425     2717   10003*
MSG11    060452     9011   10007*
MSG12    060512     2763   10013*
MSG13    060521     2648   10015*
MSG14    060607     8347   10025*
```

```
MSG15   060632          2786   2980   10029#
MSG16   060644          3112  10031#
MSG17   060670          5899  10035#
MSG18   060720          4595   4607   4692   4754   4766   4853   4904   4916   5004   5046   5057   5231   5243
                        5332   5374   5385   5558   5570   6208   6377   6512   6524   6702   6714   6940   6952
                        7086   7098   7319   7331   7465   7477   7713   7725   7859   7671  10039#
MSG19   060760          2966  10045#
MSG2    060003          2573   9955#
MSG20   061010          6173  10049#
MSG21   061177          6566  10070#
MSG3    060054          2588   9962#
MSG4    060106          2590   9967#
MSG5    060151          2592   9973#
MSG6    060230          8983   9981#
MSG7    060264          2607   9986#
MSG8    060321          3774   9991#
MSG9    060377          3769   9999#
MSP   = 000100           275#
MULT6   051746          8635   8661#
M.ALGN= 040000           351#
M.CADD= 017760           350#  8530
M.CDIF= 017760           343#  8515
M.DRV = 000007           342#
M.HEAD= 007000           353#  3090   4107   4139
M.ID  = 000003           349#  8809
M.OFST= 017760           344#
M.PAR = 100000           354#  3085   8846
M.SECT= 000760           352#  8396   8420
M.SER = 077770           345#
NED   = 010000           230#  2772   2784   2865   2868
NEM   = 004000           229#
NUDRV   012174          2947#  8096
NXF   = 000004           239#
OFFSET= 000015           197#
OFST  = 000004           258#
OPI   = 020000           250#
OR    = 000200           225#
O.ADR1  072460         11131#  11182#  11184#  11354   11355#  11393#  11408   11425   11616#
O.BACK  073204         11226   11267#
O.BD    074476         11412   11413   11561#
O.BKPT  072712         11177#  11230
O.BK1   073644         11125   11390#
O.BK2   073700         11395   11398#
O.BK3   073716         11397   11404#
O.BRK   073634         11142   11388#
O.BUF   074525         11470   11479   11484   11589#
O.B2    074000         11409   11418#
O.B3    074040         11417   11426#
O.CADV  074210         11243   11260   11283   11287   11317   11321   11426   11469#
O.CLGL  073026         11200   11202   11210#
O.CLGT= 000014         11214   11579#
O.CLS1  074434         11537   11539#
O.CR    074500         11546   11548   11549   11563#
O.CRET  073140         11221   11250#
O.CRLF  074444         11258   11315   11411   11450   11455   11546#
O.CRLS  074452         11195   11548#
```

```
0.CRS   074456          11547   11549#
0.CSR1  074474          11446*  11462   11557#
0.CSR2  074475          11447*  11463   11558#
0.CT    072462          11375*  11418*  11420*  11617#
0.C1    073612          11378#  11419
0.DCD   072756          11183   11185   11195*  11252   11303   11428
0.DCD1  072762          11196#  11244   11288
0.EFF   073266          11229   11294#
0.EFF1  073404          11306   11325#
0.ENTR  072472          11122#
0.ERR   072742          11169   11180   11192*  11215   11240   11298   11541
0.ERR1  073276          11298#  11371
0.ERR2  073116          11240#  11274   11347
0.FTYP  074352          11194   11262   11277   11319   11424   11495   11511   11516#
0.GET   074274          11148   11198   11492*  11497   11499   11501
0.GO    073450          11223   11345#
0.GO2   073526          11356#  11381
0.LG  = 000006          11152   11587#
0.LGCH  074503          11211   11567#  11579
0.LGDR  073056          11218   11220#  11232
0.LGL = 000030          11232#
0.LGL1  073030          11211#  11216
0.LGL2  073050          11212   11217#
0.MSK   072452          11299   11300   11302   11609#
0.ODT   072466            375   11120#  11598
0.OFST  073214          11227   11273#
0.OF1   073262          11286   11288#
0.OP1   073152          11224   11256#
0.OP2   073160          11173   11258#  11269
0.ORPC  072664          11165#  11225
0.P     074473          11140*  11345*  11370   11372*  11421*  11555#
0.PRI   072450          11132   11394   11398   11608#
0.PROC  073556          11231   11370#
0.PR1   073600          11374   11376#
0.RALL  072732          11143   11178   11184#
0.RCSR= 177560          11110#  11446   11448*  11458   11460   11462*  11492
0.RDB = 177562          11109#  11494
0.REGT  072614          11148#  11222
0.RSE1  074172          11459   11462#
0.RSP   072624          11150#  11153
0.RST   072530          11121   11130#
0.RSTT  074142          11351   11377   11455#
0.RST1  072560          11129   11137#
0.RTIT  073554          11139*  11365#
0.SCAN  072766          11159   11198#  11209
0.SP    072656          11151   11160#
0.SPC   074222          11472#  11480
0.SP1   072644          11156#  11161
0.STM = 000340          11099#  11141   11350   11376   11378
0.STRT  072516          11120   11127#
0.SVR   074052          11130   11390   11432#
0.SVTT  074110          11410   11422   11446#
0.T     074472          11352*  11379*  11391   11554#
0.TBIT  073500          11352#  11392
0.TBT = 000020          11100#  11353   11380   11416
0.TC    074436          11534   11540#
```

```
0.TCLS  074420          11165   11250   11256   11267   11533#
0.TCSR= 177564          11112#  11447   11449#  11456   11463#  11516   11522
0.TDB = 177566          11111#  11518#  11524#
0.TL    074517          11149   11152   11160   11581#  11587
0.TRTC  074534          11184   11355   11594#
0.TVEC= 000014          11098#  11123   11141#  11142#
0.TYPE  074336          11414   11485   11508#  11512   11550
0.TYPI  074416          11509   11520   11528#
0.TYP2  074376          11522#  11523   11526
0.UIN   072464          11131   11354#  11393   11618#
0.UPC   072444          11124#  11349#  11364   11388#  11405   11407#  11606#
0.URO   072426          11127   11157   11599#
0.USP   072442          11128#  11433#  11434   11605#
0.UST   072446          11122#  11353#  11363   11380#  11389#  11396   11416#  11607#
0.WDS   073274          11295   11297#
0.WDS2  073312          11302#  11324
0.WDS3  073342          11313#  11326   11332   11341
0.WDS4  073400          11313   11323#
0.WRD   073106          11220   11236#
0.WRDA  073132          11237   11245#
0.WRD1  073114          11239#  11246   11263
0.WSCH  073272          11228   11296#
0.XXX   074470          11432#  11440   11553#
0.45    072572          11138   11140#
PACK  = 000003            192#   3157
PARAM   001356            548#   2469#  2474#  2479#  2484#  2489#  2493#  2585
PARSRT  007624            370    2469#
PAT   = 000020            273#   3278   3710   3877   3925   4363   5926
PC    =%000007             85#   2559#  2569#  2589#  2591#  2593#  2606#  2613#  2713#  2727#  2732#  2754#  2776#
                        2780#  2795#  2826#  2831#  2847#  2855#  2873#  2877#  2881#  2885#  2913#  2920#  3006#
                        3012#  3024#  3028#  3032#  3037#  3041#  3047#  3051#  3053#  3057#  3063#  3067#  3069#
                        3073#  3079#  3083#  3086#  3109#  3114#  3132#  3147#  3154#  3159#  3176#  3186#  3190#
                        3210#  3226#  3231#  3262#  3275#  3279#  3307#  3312#  3315#  3324#  3329#  3333#  3335#
                        3339#  3347#  3352#  3354#  3357#  3362#  3364#  3367#  3372#  3376#  3387#  3389#  3395#
                        3397#  3400#  3405#  3407#  3410#  3419#  3421#  3428#  3431#  3433#  3435#  3439#  3443#
                        3445#  3449#  3454#  3456#  3463#  3466#  3468#  3472#  3476#  3478#  3482#  3487#  3489#
                        3496#  3499#  3501#  3505#  3509#  3511#  3515#  3520#  3522#  3529#  3530#  3538#  3544#
                        3550#  3555#  3557#  3560#  3566#  3573#  3584#  3589#  3618#  3628#  3632#  3637#  3641#
                        3643#  3646#  3651#  3653#  3656#  3661#  3665#  3676#  3678#  3684#  3686#  3689#  3694#
                        3696#  3699#  3707#  3714#  3716#  3720#  3724#  3731#  3733#  3736#  3740#  3742#  3745#
                        3784#  3794#  3798#  3803#  3807#  3809#  3812#  3817#  3819#  3822#  3827#  3831#  3842#
                        3844#  3850#  3852#  3855#  3860#  3862#  3865#  3873#  3889#  3891#  3895#  3900#  3911#
                        3913#  3922#  3937#  3939#  3943#  3952#  3958#  3963#  3974#  3976#  4017#  4028#  4032#
                        4037#  4041#  4043#  4046#  4051#  4053#  4056#  4061#  4065#  4076#  4078#  4084#  4086#
                        4089#  4094#  4096#  4099#  4105#  4116#  4125#  4127#  4137#  4151#  4153#  4162#  4177#
                        4184#  4186#  4192#  4199#  4216#  4218#  4226#  4231#  4235#  4243#  4245#  4248#  4253#
                        4255#  4258#  4262#  4273#  4275#  4281#  4283#  4286#  4291#  4293#  4296#  4300#  4312#
                        4318#  4322#  4327#  4330#  4358#  4367#  4369#  4372#  4374#  4377#  4386#  4388#  4394#
                        4396#  4399#  4404#  4406#  4409#  4414#  4416#  4419#  4425#  4428#  4429#  4440#  4442#
                        4445#  4449#  4451#  4453#  4456#  4459#  4461#  4464#  4469#  4471#  4474#  4479#  4481#
                        4484#  4489#  4493#  4504#  4506#  4512#  4514#  4517#  4522#  4524#  4527#  4531#  4541#
                        4546#  4550#  4580#  4590#  4605#  4612#  4614#  4617#  4622#  4624#  4627#  4633#  4637#
                        4670#  4686#  4688#  4697#  4699#  4702#  4707#  4709#  4712#  4718#  4722#  4737#  4749#
                        4764#  4771#  4773#  4776#  4781#  4783#  4786#  4792#  4796#  4838#  4847#  4849#  4859#
                        4868#  4870#  4873#  4878#  4880#  4883#  4892#  4899#  4914#  4922#  4931#  4933#  4936#
                        4941#  4943#  4946#  4951#  4955#  4983#  4992#  4998#  5000#  5009#  5011#  5014#  5019#
```

```
                    5021*  5024*  5032*  5041*  5055*  5063*  5100*  5110*  5113*  5116*  5121*  5123*  5126*
                    5129*  5139*  5147*  5149*  5152*  5157*  5159*  5162*  5166*  5177*  5179*  5185*  5187*
                    5190*  5195*  5197*  5200*  5206*  5217*  5226*  5241*  5256*  5260*  5265*  5269*  5275*
                    5278*  5311*  5320*  5326*  5328*  5337*  5339*  5342*  5347*  5349*  5352*  5360*  5369*
                    5383*  5391*  5427*  5435*  5439*  5445*  5448*  5452*  5456*  5458*  5461*  5464*  5471*
                    5479*  5481*  5484*  5489*  5491*  5494*  5499*  5503*  5514*  5516*  5522*  5524*  5527*
                    5532*  5534*  5537*  5545*  5553*  5568*  5582*  5586*  5592*  5596*  5602*  5606*  5638*
                    5648*  5656*  5659*  5662*  5667*  5669*  5672*  5675*  5683*  5691*  5693*  5696*  5701*
                    5703*  5706*  5710*  5721*  5723*  5729*  5731*  5734*  5739*  5741*  5744*  5748*  5760*
                    5769*  5776*  5780*  5786*  5794*  5839*  5852*  5854*  5901*  5902*  5921*  5930*  5932*
                    5935*  5937*  5940*  5949*  5951*  5957*  5959*  5962*  5967*  5969*  5972*  5977*  5979*
                    5982*  5988*  5991*  5992*  6003*  6005*  6008*  6012*  6014*  6016*  6019*  6022*  6024*
                    6027*  6032*  6034*  6037*  6042*  6044*  6047*  6052*  6056*  6067*  6069*  6075*  6077*
                    6080*  6085*  6087*  6090*  6094*  6102*  6109*  6113*  6119*  6127*  6138*  6143*  6147*
                    6183*  6191*  6202*  6204*  6216*  6232*  6247*  6250*  6253*  6258*  6260*  6263*  6266*
                    6279*  6287*  6289*  6292*  6297*  6299*  6302*  6306*  6317*  6319*  6325*  6327*  6330*
                    6335*  6337*  6340*  6347*  6356*  6360*  6371*  6373*  6382*  6384*  6387*  6392*  6394*
                    6397*  6425*  6432*  6436*  6442*  6450*  6461*  6489*  6493*  6499*  6507*  6522*  6533*
                    6575*  6583*  6586*  6589*  6594*  6596*  6599*  6602*  6610*  6618*  6620*  6623*  6628*
                    6630*  6633*  6637*  6648*  6650*  6656*  6658*  6661*  6666*  6668*  6671*  6677*  6686*
                    6697*  6712*  6729*  6737*  6744*  6748*  6754*  6762*  6816*  6829*  6832*  6835*  6840*
                    6842*  6845*  6851*  6859*  6861*  6864*  6869*  6871*  6874*  6878*  6889*  6891*  6897*
                    6899*  6902*  6907*  6909*  6912*  6916*  6925*  6935*  6950*  6965*  6974*  6977*  6980*
                    6985*  6987*  6990*  6997*  7005*  7007*  7010*  7015*  7017*  7020*  7024*  7035*  7037*
                    7043*  7045*  7048*  7053*  7055*  7058*  7062*  7071*  7081*  7096*  7113*  7119*  7126*
                    7130*  7136*  7144*  7195*  7208*  7211*  7214*  7219*  7221*  7224*  7230*  7238*  7240*
                    7243*  7248*  7250*  7253*  7257*  7268*  7270*  7276*  7278*  7281*  7286*  7288*  7291*
                    7295*  7304*  7314*  7329*  7344*  7353*  7356*  7359*  7364*  7366*  7369*  7376*  7384*
                    7386*  7389*  7394*  7396*  7399*  7403*  7414*  7416*  7422*  7424*  7427*  7432*  7434*
                    7437*  7441*  7450*  7460*  7475*  7492*  7498*  7505*  7509*  7515*  7523*  7561*  7568*
                    7572*  7578*  7589*  7602*  7605*  7608*  7613*  7615*  7618*  7624*  7632*  7634*  7637*
                    7642*  7644*  7647*  7651*  7662*  7664*  7670*  7672*  7675*  7680*  7682*  7685*  7689*
                    7698*  7708*  7723*  7738*  7747*  7750*  7753*  7758*  7760*  7763*  7770*  7778*  7780*
                    7783*  7788*  7790*  7793*  7797*  7808*  7810*  7816*  7818*  7821*  7826*  7828*  7831*
                    7835*  7844*  7854*  7869*  7886*  7892*  7899*  7903*  7909*  7917*  7967*  7972*  7976*
                    7982*  7992*  8000*  8002*  8020*  8023*  8027*  8029*  8032*  8034*  8038*  8049*  8056*
                    8058*  8064*  8071*  8102*  8105*  8115*  8120    8136*  8146*  8172*  8192*  8203*  8205*
                    8216*  8217*  8228*  8242*  8243*  8245*  8246*  8266*  8279*  8281*  8301*  8303*  8305*
                    8308*  8325*  8327*  8329*  8332*  8342*  8353*  8364*  8367*  8379*  8383*  8386*  8389*
                    8394*  8401*  8417*  8427*  8436*  8439*  8445*  8448*  8456*  8459*  8465*  8468*  8476*
                    8479*  8485*  8488*  8496*  8499*  8505*  8508*  8513*  8523*  8528*  8535*  8544*  8550*
                    8552*  8561*  8568*  8570*  8578*  8583*  8585*  8625*  8633*  8635*  8656*  8665*  8725*
                    8745*  8756*  8759*  8780*  8782*  8791*  8797*  8799*  8812*  8818*  8820*  8849*  8880*
                    8886*  9109*  9115*  9163*  9182*  9189*  9196*  9210*  9212*  9312*  9329*  9469*  9662*
                    9687*  9690*  9727*  9732*  9776*  11076*  11095*  11124  11137  11165*  11250*  11256*  11267*
                   11539*
PCA    = 004000     280#
PCD    = 010000     281#
PCLKF    004446    1409#  2629*  2636*  8753   8777
PCVEC    001352     542#  2630   2637
PCYL     001376     558#  3779*  3881   3886*  3929   3934*
PFSRT    012326    2990#  9013
PGE    = 002000     228#
PIP    = 020000     266#
PIRQ   = 177772      71#
PIRQVE= 000240      165#
```

```
PKRB    001344          538#
PKS     001340          536#  2628  2635  8758*  8781*
PKSB    001342          537#  8757*
PPTP    004420         1393#  2576*
PRGSRT  007764         2472   2477  2482  2487   2492   2496#  8185
PR0   = 000000           88#  8749
PR1   = 000040           89#
PR2   = 000100           90#
PR3   = 000140           91#
PR4   = 000200           92#
PR5   = 000240           93#   535
PR6   = 000300           94#
PR7   = 000340           95#  2498  2547  2552  2617  2640  8774  9008  9009
PS    = 177776           68#    69
PSEC    001422          571#  4321* 4324  4326*  4328   4331   8477  8480  8497  8500  10850
PSW   = 177776           69#
PWRVEC= 000024          160#  2517* 2518* 8998*  9007*  9008*
RDCHR = 104410         9529   9862#
RDCYLA  051210         3376   3665  3720  3831   3895   3943   4065  4218  4300  4493  4637  4722  4796
                       4955   5206  5278  5503   5606   5748   6056  6347  6677  6916  7062  7295  7441
                       7689   7835  8038  8527#
RD  LD  001124         3372   3661  3724  3827   3900   3952   3958  3963  4061  4262  4489  4633  4718
                       4792   4951  5129  5166   5464   5499   5675  5710  6052  6266  6306  6602  6637
                       6878   7024  7257  7403   7651   7797   8034  8512#
RDDA A= 000021          199#  5850
RDGA E= 100000          284#
RDHEAD= 000025          201#  4588  4747  4897   5039   5224   5367  5551  6505  6695  6933  7079  7312
                       7458   7706  7852
RDLIN = 104411         8153   9601  9863#
RDOCT = 104412         8198   8211  9864#
RDSEC   050400         4327   4330  8393#  8436   8439   8476  8479  8633
RDY   = 000200          207#  8238  8411
RECAL = 000013          196#  3624  3790   4024   4175   5433  8047
RESREG= 104414         9661   9731  9866#  11068
RESVEC= 000010          155#
RHTAB   001704          604#  4587  4746  4806   4896   4961  5037  5060  5223  5248  5252  5365  5388
                       5550   5574  5578  6504   6528   6694  6718  6932  6956  7078  7102  7311  7335
                       7457   7481  7705  7729   7851   7875  8640  8642  8646  8649  10859 10862
RKASOF= 000016          178#  2676  8259  8294   8320
RKBA  = 000004          173#  2672  4674* 4841*  4985*  5313* 5845* 5893* 6194* 6364* 8255
RKCS1 = 000000          171#  2669  2730* 2829*  2918*  3157* 3229* 3310* 3322* 3345* 3351* 3360* 3371*
                       3383*  3385* 3394* 3403*  3416*  3429* 3464* 3497* 3536* 3553* 3564* 3622* 3624*
                       3640*  3649* 3660* 3672*  3674*  3683* 3692* 3712* 3718* 3729* 3738* 3788* 3790*
                       3806*  3815* 3826* 3838*  3840*  3849* 3858* 3887* 3893* 3907* 3909* 3935* 3941*
                       3970*  3972* 4022* 4024*  4040*  4049* 4060* 4072* 4074* 4083* 4092* 4123* 4129*
                       4173*  4175* 4180* 4182*  4197*  4229* 4242* 4251* 4269* 4271* 4280* 4289* 4365*
                       4371*  4382* 4384* 4393*  4402*  4413* 4423* 4427* 4438* 4447* 4468* 4477* 4488*
                       4500*  4502* 4511* 4520*  4544*  4588* 4611* 4620* 4632* 4684* 4696* 4705* 4717*
                       4747*  4770* 4779* 4791*  4845*  4857* 4867* 4876* 4897* 4920* 4930* 4939* 4950*
                       4996*  5008* 5017* 5037*  5108*  5119* 5146* 5155* 5173* 5175* 5184* 5193* 5204*
                       5224*  5263* 5324* 5336*  5345*  5367* 5433* 5437* 5478* 5487* 5498* 5510* 5512*
                       5521*  5530* 5551* 5590*  5654*  5665* 5690* 5699* 5717* 5719* 5728* 5737* 5774*
                       5850*  5928* 5934* 5945*  5947*  5956* 5965* 5976* 5986* 5990* 6001* 6010* 6031*
                       6040*  6051* 6063* 6065*  6074*  6083* 6107* 6141* 6200* 6245* 6256* 6286* 6295*
                       6313*  6315* 6324* 6333*  6344*  6369* 6381* 6390* 6430* 6487* 6505* 6581* 6592*
                       6617*  6626* 6644* 6646*  6655*  6664* 6675* 6695* 6742* 6827* 6838* 6858* 6867*
```

```
                    6885*  6887*  6896*  6905*  6933*  6972*  6983*  7004*  7013*  7031*  7033*  7042*  7051*
                    7079*  7124*  7206*  7217*  7237*  7246*  7264*  7266*  7275*  7284*  7312*  7351*  7362*
                    7383*  7392*  7410*  7412*  7421*  7430*  7458*  7503*  7566*  7600*  7611*  7631*  7640*
                    7658*  7660*  7669*  7678*  7706*  7745*  7756*  7777*  7786*  7804*  7806*  7815*  7824*
                    7852*  7897*  7970*  7998*  8018*  8025*  8045*  8047*  8052*  8054*  8069*  8238   8252
                    8362*  8409*  8411
RKCS2  = 000010      175#  2670   2711*  2725*  2729*  2824*  2828*  2911*  2917*  3189*  3384*  3623*  3673*
                    3789*  3839*  3908*  3971*  4023*  4073*  4174*  4181*  4270*  4383*  4501*  4603   4673*
                    4762   4840*  4912   5053   5174*  5205*  5239   5381   5511*  5566   5718*  5946*  6064*
                    6314*  6520   6645*  6676*  6710   6886*  6948   7032*  7094   7265*  7327   7411*  7473
                    7659*  7721   7805*  7867   8046*  8053*  8253   8361*  8377*  8381*  8408*
RKCA   = 000006      174#  2673   4120*  4131*  4584*  4677*  4743*  4803*  5846*  5870*  5894*  6197*  6242*
                    6362*  6484*  8256
RKDB   = 000024      180#  2678   4598   4599   4600   4757   4758   4759   4907   4908   4909   5049   5050
                    5051   5234   5235   5236   5377   5378   5379   5561   5562   5563   6515   6516   6517
                    6705   6706   6707   6943   6944   6945   7089   7090   7091   7322   7323   7324   7468
                    7469   7470   7716   7717   7718   7862   7863   7864
RKDC   = 000020      179#  2677   3711*  3719*  3878*  3894*  3926*  3942*  4364*  4422*  4994*  5035*  5105*
                    5221*  5322*  5363*  5588*  5652*  5772*  5847*  5927*  5985*  6105*  6240*  6345*  6366*
                    6428*  6579*  6691*  6740*  6825*  6929*  6969*  7075*  7122*  7204*  7308*  7348*  7454*
                    7501*  7564*  7598*  7702*  7742*  7848*  7895*  7997*  8260
RKDS   = 000012      176#  2674   8257
RKECPS= 000030       184#  2682   8264
RKECPT= 000032       185#  2683   8265
RKER   = 000014      177#  2675   8258
RKMR1  = 000026      181#  2679   3011*  3050*  3066*  3082*  3113*  3146*  3278*  3332*  3361*  3404*  3420*
                    3432*  3442*  3455*  3465*  3475*  3488*  3498*  3508*  3521*  3537*  3565*  3631*  3650*
                    3693*  3710*  3730*  3739*  3797*  3816*  3859*  3877*  3925*  4031*  4050*  4093*  4104*
                    4136*  4150*  4252*  4290*  4363*  4380*  4403*  4439*  4448*  4478*  4521*  4621*  4706*
                    4780*  4877*  4940*  5018*  5120*  5156*  5194*  5346*  5438*  5444*  5455*  5488*  5531*
                    5666*  5700*  5738*  5926*  5943*  5966*  6002*  6011*  6041*  6084*  6257*  6296*  6334*
                    6391*  6593*  6627*  6665*  6839*  6868*  6906*  6984*  7014*  7052*  7218*  7247*  7285*
                    7363*  7393*  7431*  7612*  7641*  7679*  7757*  7787*  7825*  8026*  8261   8300*  8304*
                    8324*  8328*  8382*  8393*  8407*  8512*  8527*  8543*  8560*  8577*
RKMR2  = 000034      182#  2680   8262
RKMR3  = 000036      183#  2681   8263
RKPRI    001336      535#  8227
RKVEC    001334      534#  2612*  8215*  8218*  8225
RKWC   = 000002      172#  2671   4675*  4842*  4986*  5314*  5849*  6195*  6365*  8254
RLS    = 000010      221#  3189
RTT    = 000006    11102#
R0     =%000000       76#  2626*  2630*  2637*  2639*  2640*  2719*  2729   2738   2740   2747   2758*  2759
                    2787   2818*  2828   2837   2842*  2843   2904*  2917   2925   2935*  2936   2981*  2982
                    3125*  3126*  3127*  3128   3611*  3711   3719   3721   3725   3748*  3749   4012*  4119*
                    4120   4121*  4130*  4131   4132*  4133   4143   4156   4166*  4167   4587*  4598*  4599*
                    4600*  4680*  4681*  4682   4746*  4757*  4758*  4759*  4809*  4810*  4811   4896*  4907*
                    4908*  4909*  5037*  5049*  5050*  5051*  5060   5066*  5069   5223*  5234*  5235*  5236*
                    5365*  5377*  5378*  5379*  5388   5394*  5397   5550*  5561*  5562*  5563*  5881*  5882
                    6504*  6515*  6516*  6517*  6694*  6705*  6706*  6707*  6932*  6943*  6944*  6945*  7078*
                    7039*  7090*  7091*  7311*  7322*  7323*  7324*  7457*  7468*  7469*  7470*  7705*  7716*
                    7717*  7718*  7851*  7862*  7863*  7864*  7985*  7995   8075*  8076   8112*  8115   8131*
                    8133*  8154*  8156   8176   8199*  8200   8202   8212*  8213   8215   8225*  8226*  8227*
                    8348   8592   8594*  8606*  8607*  8608*  8609*  8610*  8613   8615   8617*  8624*  8631
                    8637*  8638*  8639   8642*  8645   8646   8649*  8650   8651   8655*  8832   8834*  8841*
                    8848*  8898*  8899   8903   8933*  8934   8938   8981*  8982   8984   9156   9157*  9162
                    9167   9170*  9227   9237*  9241   9257   9258   9271*  9289   9297*  9301   9302   9304*
```

```
                    9305*  9306   9328*  9598   9602*  9603   9606   9626*  9629*  9651*  9655   9665*  9667*
                    9669*  9707*  9708   9726*  9730*  9764   9765*  9766   9768   9770*  9771   9774*  9795
                    9820*  9830   9831*  9832   9833*  9834*  9835*  9836*  11013* 11014* 11015* 11016* 11017*
                   11018*  11019* 11020  11025  11030  11033* 11034  11038  11039  11051  11056  11059  11061*
                   11088*  11150  11154* 11155  11193* 11199  11201  11203* 11207  11211  11242* 11259* 11261*
                   11276*  11281* 11284* 11285* 11304* 11307  11309* 11311  11316* 11318* 11320* 11325  11327
                   11333*  11334* 11336* 11337* 11338* 11339* 11340  11356* 11415* 11423* 11425* 11442* 11472
                   11476*  11477* 11478* 11481* 11482* 11483  11494* 11496* 11498  11500  11510* 11518  11519
R1      =%000001     77#   2720*  2745*  2757   2819*  2839   2841   2849   2852*  2870   2905*  2915   2969*
                    2973   2979   3115*  3119*  3120*  3121*  3122*  3123*  3124*  3125   4806*  4812   4961*
                    4965   5067*  5070   5395*  5398   8132*  8134*  8155*  8169   8187*  8593   8595*  8607
                    8619*  8620   8623*  8632   8643*  8645*  8650*  8654*  8833   8835*  8840*  8844   8847*
                    9228   9241*  9242   9246   9270*  9290   9327*  9599   9604*  9612*  9614*  9616*  9619*
                    9622   9625*  9647*  9651   9652*  9664*  9666*  9668*  9709*  9715*  9721*  9796   9819*
                   11030*  11042  11089#  11210*  11211  11213*  11214  11217*  11218  11294*  11296*  11305  11357*
                   11439
R2      =%000002     78#   8156*  8157*  8161   8167   8176*  8177*  8178   8180   9229   9240*  9244*  9247
                    9254*  9255*  9256   9261*  9269*  9600   9605*  9613*  9615*  9617*  9623   9624*  9653*
                    9654   9655*  9670*  9671*  9706*  9709   9710*  9716*  9717*  9722*  9723*  9797   9818*
                   11034*  11048*  11090#  11158*  11166*  11167*  11168*  11170*  11171  11172  11177  11197*  11208*
                   11236  11282*  11284  11297  11299*  11302  11304  11316  11320  11323  11328  11339  11358*
                   11373  11438  11469  11472*  11473*  11474*  11475  11486*  11536
R2OSEC  050450      8406#  8456   8459   8496   8499
R3      =%000003     79#   3118*  3129*  8158*  8163   8174*  9230   9238*  9239*  9253*  9256*  9265*  9266*
                    9268*  9371   9380*  9386*  9387*  9390*  9395*  9396*  9397   9406*  9524   9526*  9527
                    9530*  9531   9538*  9539   9541   9549   9553   9555*  9561   9563   9565*  9568*  9650*
                    9663*  9670   9714*  9719*  9725*  9726   9798   9817*  11038*  11040  11044*  11091#  11308*
                   11310*  11311*  11312*  11327*  11328*  11329*  11330*  11331  11359*  11413*  11437  11484*  11508
                   11546*  11548*
R4      =%000004     80#   3116*  3117   3128*  3131*  8159*  8161   8164*  8165   8273   8274*  8275   8277*
                    8280*  8291   8293*  8294   8302*  8306*  8318   8319*  8320   8326*  8330*  8675   8713*
                    8714   8719   8724*  8727   8731   8734   8743*  9372   9374*  9375*  9376*  9377   9378*
                    9392   9394*  9402*  9405*  9649*  9656*  9712*  9715   9721   9723   9799   9816*  11032*
                   11036*  11053*  11064  11092#  11132*  11133*  11134*  11135*  11136  11149*  11150  11152  11155*
                   11156*  11157*  11160*  11179*  11181*  11182  11196*  11204*  11205*  11206*  11207*  11238  11278*
                   11279*  11280*  11281  11282  11308  11309  11314  11322*  11325  11331  11340  11346*  11348*
                   11349  11360*  11375  11412*  11436  11470*  11475*  11479  11483*  11508  11510  11538  11549*
R5      =%000005     81#   2668*  2669   2670   2671   2672   2673   2674   2675   2676   2677   2678   2679
                    2680   2681   2682   2683   2711*  2725*  2729*  2730*  2824*  2828*  2829*  2911*  2917*
                    2918*  3011*  3050*  3066*  3082*  3113*  3146*  3157*  3189*  3229*  3278*  3310*  3322*
                    3332*  3345*  3351*  3360*  3361*  3371*  3383*  3384*  3385*  3394*  3403*  3404*  3416*
                    3420*  3429*  3432*  3442*  3455*  3464*  3465*  3475*  3488*  3497*  3498*  3508*  3521*
                    3536*  3537*  3553*  3564*  3565*  3622*  3623*  3624*  3631*  3640*  3649*  3650*  3660*
                    3672*  3673*  3674*  3683*  3692*  3693*  3710*  3711*  3712*  3718*  3719*  3729*  3730*
                    3738*  3739*  3788*  3789*  3790*  3797*  3806*  3815*  3816*  3826*  3838*  3839*  3840*
                    3849*  3858*  3859*  3877*  3878*  3887*  3893*  3894*  3907*  3908*  3909*  3925*  3926*
                    3935*  3941*  3942*  3970*  3971*  3972*  4022*  4023*  4024*  4031*  4040*  4049*  4050*
                    4060*  4072*  4073*  4074*  4083*  4092*  4093*  4104*  4120*  4123*  4129*  4131*  4136*
                    4150*  4173*  4174*  4175*  4180*  4181*  4182*  4197*  4229*  4242*  4251*  4252*  4269*
                    4270*  4271*  4280*  4289*  4290*  4363*  4364*  4365*  4371*  4380*  4382*  4383*  4384*
                    4393*  4402*  4403*  4413*  4422*  4423*  4427*  4438*  4439*  4447*  4448*  4468*  4477*
                    4478*  4488*  4500*  4501*  4502*  4511*  4520*  4521*  4544*  4584*  4588*  4598   4599
                    4600   4603   4611*  4620*  4621*  4632*  4673*  4674*  4675*  4677*  4684*  4696*  4705*
                    4706*  4717*  4743*  4747*  4757   4758   4759   4762   4770*  4779*  4780*  4791*  4803*
                    4840*  4841*  4842*  4845*  4857*  4867*  4876*  4877*  4897*  4907   4908   4909   4912
                    4920*  4930*  4939*  4940*  4950*  4985*  4986*  4994*  4996*  5008*  5017*  5018*  5035*
```

```
                    5039*  5049   5050   5051   5053   5105*  5108*  5119*  5120*  5146*  5155*  5156*  5173*
                    5174*  5175*  5184*  5193*  5194*  5204*  5205*  5221*  5224*  5234   5235   5236   5239
                    5263*  5313*  5314*  5322*  5324*  5336*  5345*  5346*  5363*  5367*  5377   5378   5379
                    5381   5433*  5437*  5438*  5444*  5455*  5478*  5487*  5488*  5498*  5510*  5511*  5512*
                    5521*  5530*  5531*  5551*  5561   5562   5563   5566   5588*  5590*  5652*  5654*  5665*
                    5666*  5690*  5699*  5700*  5717*  5718*  5719*  5728*  5737*  5738*  5772*  5774*  5845*
                    5846*  5847*  5849*  5850*  5870*  5893*  5894*  5926*  5927*  5928*  5934*  5943*  5945*
                    5946*  5947*  5956*  5965*  5966*  5976*  5985*  5986*  5990*  6001*  6002*  6010*  6011*
                    6031*  6040*  6041*  6051*  6063*  6064*  6065*  6074*  6083*  6084*  6105*  6107*  6141*
                    6194*  6195*  6197*  6200*  6240*  6242*  6245*  6256*  6257*  6286*  6295*  6296*  6313*
                    6314*  6315*  6324*  6333*  6334*  6344*  6345*  6362*  6364*  6365*  6366*  6369*  6381*
                    6390*  6391*  6428*  6430*  6484*  6487*  6505*  6515   6516   6517   6520   6579*  6581*
                    6592*  6593*  6617*  6626*  6627*  6644*  6645*  6646*  6655*  6664*  6665*  6675*  6676*
                    6691*  6695*  6705   6706   6707   6710   6740*  6742*  6825*  6827*  6838*  6839*  6858*
                    6867*  6868*  6885*  6886*  6887*  6896*  6905*  6906*  6929*  6933*  6943   6944   6945
                    6948   6969*  6972*  6983*  6984*  7004*  7013*  7014*  7031*  7032*  7033*  7042*  7051*
                    7052*  7075*  7079*  7089   7090   7091   7094   7122*  7124*  7204*  7206*  7217*  7218*
                    7237*  7246*  7247*  7264*  7265*  7266*  7275*  7284*  7285*  7308*  7312*  7322   7323
                    7324   7327   7348*  7351*  7362*  7363*  7383*  7392*  7393*  7410*  7411*  7412*  7421*
                    7430*  7431*  7454*  7458*  7468   7469   7470   7473   7501*  7503*  7564*  7566*  7598*
                    7600*  7611*  7612*  7631*  7640*  7641*  7658*  7659*  7660*  7669*  7678*  7679*  7702*
                    7706*  7716   7717   7718   7721   7742*  7745*  7756*  7757*  7777*  7786*  7787*  7804*
                    7805*  7806*  7815*  7824*  7825*  7848*  7852*  7862   7863   7864   7867   7895*  7897*
                    7970*  7997*  7998*  8018*  8025*  8026*  8045*  8046*  8047*  8052*  8053*  8054*  8069*
                    8238   8252   8253   8254   8255   8256   8257   8258   8259   8260   8261   8262   8263
                    8264   8265   8294   8300*  8304*  8320   8324*  8328*  8361*  8362*  8377*  8381*  8382*
                    8393*  8407*  8408*  8409*  8411   8512*  8527*  8543*  8560*  8577*  9231   9233*  9235*
                    9242*  9246*  9261   9267*  9373   9379*  9381*  9383*  9384*  9385*  9386   9404*  9648*
                    9654*  9659*  9660   9713*  9717   9724   9800   9815*  11093*  11166  11172*  11192*  11238*
                   11239*  11241*  11242  11245*  11251*  11257  11259  11268  11273*  11275*  11278  11300*  11301*
                   11312   11361*  11396*  11398*  11400*  11401*  11402*  11403*  11404  11405*  11406  11407  11408
                   11415   11427*  11435  11451*  11464*  11487*  11502*  11528*  11533*  11535*  11538*  11551*

R6      =%000006    82#    84     2505*  2506*  2507
R7      =%000007    83#    85
SAVREG= 104413     9646   9705   9865*  11012
SBMR2   004374     1377#  3031*  3036*  3052*  3068*  3084*  3085*  3323*  3334*  3353*  3363*  3396*  3406*
                   3434*  3444*  3467*  3477*  3500*  3510*  3642*  3652*  3685*  3695*  3732*  3741*  3808*
                   3818*  3851*  3861*  4042*  4052*  4085*  4095*  4152*  4244*  4254*  4282*  4292*  4373*
                   4395*  4405*  4415*  4441*  4450*  4470*  4480*  4513*  4523*  4613*  4623*  4698*  4708*
                   4772*  4782*  4869*  4879*  4932*  4942*  5010*  5020*  5112*  5122*  5148*  5158*  5186*
                   5196*  5338*  5348*  5447*  5457*  5480*  5490*  5523*  5533*  5658*  5668*  5692*  5702*
                   5730*  5740*  5936*  5958*  5968*  5978*  6004*  6013*  6033*  6043*  6076*  6086*  6249*
                   6259*  6288*  6298*  6326*  6336*  6383*  6393*  6585*  6595*  6619*  6629*  6657*  6667*
                   6831*  6841*  6860*  6870*  6898*  6908*  6976*  6986*  7006*  7016*  7044*  7054*  7210*
                   7220*  7239*  7249*  7277*  7287*  7355*  7365*  7385*  7395*  7423*  7433*  7604*  7614*
                   7633*  7643*  7671*  7681*  7749*  7759*  7779*  7789*  7817*  7827*  8019*  8028*  8789*
                   8790   8792*  8793   10833  10846
SBMR3   004376     1378#  3040*  3056*  3072*  3328*  3338*  3356*  3366*  3399*  3409*  3438*  3448*  3471*
                   3481*  3504*  3514*  3645*  3655*  3688*  3698*  3735*  3744*  3811*  3821*  3854*  3864*
                   4045*  4055*  4088*  4098*  4159*  4161*  4247*  4257*  4285*  4295*  4376*  4398*  4408*
                   4418*  4444*  4473*  4483*  4516*  4526*  4616*  4626*  4701*  4711*  4775*  4785*  4872*
                   4882*  4935*  4945*  5013*  5023*  5115*  5125*  5151*  5161*  5189*  5199*  5341*  5351*
                   5451*  5460*  5483*  5493*  5526*  5536*  5661*  5671*  5695*  5705*  5733*  5743*  5939*
                   5961*  5971*  5981*  6007*  6036*  6046*  6079*  6089*  6252*  6262*  6291*  6301*  6329*
                   6339*  6386*  6396*  6588*  6598*  6622*  6632*  6660*  6670*  6834*  6844*  6863*  6873*
                   6901*  6911*  6979*  6989*  7009*  7019*  7047*  7057*  7213*  7223*  7242*  7252*  7280*
```

```
                    7290*  7358*  7368*  7388*  7398*  7426*  7436*  7607*  7617*  7636*  7646*  7674*  7684*
                    7752*  7762*  7782*  7792*  7820*  7830*  8022*  8031*  8810*  8811   8813*  8814   10833
                   10846
SBPAR   052664      8791   8812   8832*
SCLR  = 000040       223*  2711   2725   2824   2911   8377
SCOP1 = 104415      2722   2821   2908   3021   3044   3060   3076   3151   3183   3223   3272   3615   3704
                    3781   3870   3919   4014   4113   4223   4577   4667   4734   4889   5029   5214   5357
                    5542   5645   6229   6353   6572   6683   6813   6922   6962   7068   7192   7301   7341
                    7447   7586   7695   7735   7841   7989   9867*
SCOP1$  052736      8856*  9867
SDC   = ****** U    9872
SEC     001414       568*  3418*  3453*  3486*  3519*  9767*
SECNT   001420       570*  4316*  4338*  4887*  5028*  5356*
SECTOR  001426       573*  4326   4328   4331   4335   8395*  8396*  8397*  8398*  8399*  8400*  8419*  8420*
                    8421*  8422*  8423*  8424*  8437   8440   8457   8460   8477   8480   8497   8500   8634*
                    8638   8639*  8640*  8651   8661*  8662   8663*  8664*  8689*  8698*  8707*  8729
SEEK  = 000017       198*  3712   3887   3935   4123   4229   4365   4423   5108   5263   5590   5654   5774
                    5928   5986   6107   6245   6430   6487   6581   6742   6827   6972   7124   7206   7351
                    7503   7566   7600   7745   7897   7970   7998
SELDRV= 000001       191*  2730   2829   3229   4857   4920   8362   8409
SETINT  047614      2613   8216   8225*
SIZFLG  004452      1411*  2614*  2715   8171*  8191*
SKI   = 002002       238*
SORT    051640      5063   5391   8631*
SP    =%000006        84*  2497*  2498*  2499*  2509*  2527*  2535*  2539   2555   2616*  2617*  2618*  2633
                    2643   2646   2665*  2688   2709*  2723*  2747*  2787*  2817*  2822*  2902*  2909*  2960*
                    2982*  3004*  3022*  3045*  3061*  3077*  3105*  3117*  3144*  3152*  3174*  3184*  3208*
                    3224*  3260*  3273*  3305*  3609*  3616*  3705*  3765*  3782*  3871*  3920*  4010*  4015*
                    4114*  4214*  4224*  4310*  4357*  4572*  4578*  4662*  4668*  4735*  4836*  4890*  4981*
                    5030*  5099*  5215*  5309*  5358*  5425*  5543*  5636*  5646*  5837*  5900*  5920*  6170*
                    6230*  6354*  6562*  6573*  6684*  6807*  6814*  6923*  6963*  7069*  7181*  7193*  7302*
                    7342*  7448*  7558*  7587*  7696*  7736*  7842*  7965*  7990*  8092*  8109*  8154   8199
                    8212   8244*  8273*  8277   8278*  8280   8291*  8302   8306   8307*  8318*  8326   8330
                    8331*  8348*  8360*  8366   8387*  8406*  8416   8426   8434*  8444   8446   8447*  8454*
                    8464   8466   8467*  8474*  8484   8486   8487*  8494*  8504   8506   8507*  8551*  8569*
                    8584*  8592*  8593*  8623   8624   8631*  8632*  8654   8655   8662*  8664   8675*  8676*
                    8677*  8722   8723   8724   8741   8742   8743   8744*  8749*  8750*  8774*  8775*  8788*
                    8795   8796*  8798   8807*  8816   8817*  8819   8832*  8833*  8847   8848   8860*  8863
                    8879*  8898   8903*  8910   8923   8933   8938*  8946   8959   8965   8973*  8981   8984*
                    9010*  9038*  9041   9043   9044   9073   9074   9078*  9104   9125*  9128*  9156*  9157
                    9167*  9169   9170   9171*  9173   9175   9177   9183   9185*  9187*  9195*  9199   9203
                    9204   9208   9227*  9228*  9229*  9230*  9231*  9232*  9233   9236*  9249   9251*  9253
                    9263   9265   9267   9268   9269   9270   9271   9273*  9274*  9289*  9290*  9297   9298*
                    9309   9310*  9311*  9321   9322*  9327   9328   9363*  9364   9365   9366*  9371*  9372*
                    9373*  9379   9404   9405   9406   9407*  9408*  9429*  9430*  9431   9438*  9441*  9442*
                    9446*  9447*  9451   9454*  9458   9460   9462   9463*  9470   9472   9474*  9475   9477*
                    9478*  9479*  9480*  9481*  9496*  9497*  9500*  9501*  9502   9506*  9507*  9508   9511
                    9513   9515*  9524*  9525*  9530   9533   9537*  9544   9548*  9567   9568   9569*  9570*
                    9571*  9596*  9597*  9598*  9599*  9600*  9602   9606*  9608   9610   9618*  9619   9621
                    9622*  9624   9625   9626   9628   9647   9660*  9685   9686*  9688*  9689*  9706   9708*
                    9764*  9765   9774   9775*  9795*  9796*  9797*  9798*  9799*  9800*  9801*  9802*  9803*
                    9804*  9811*  9812*  9813*  9814*  9815   9816   9817   9818   9819   9820   9830*  9831
                    9841*  9842*  11042*  11074*  11094*  11127*  11128  11307*  11310  11314*  11322  11356  11357
                   11358  11359  11360  11361  11362*  11363*  11364*  11388  11389  11432  11433  11434*  11435*
                   11436*  11437*  11438*  11439*  11440*  11441  11463*  11471*  11474  11482  11486  11521*  11525*
                   11527  11540
```

```
SPACE2  072402          11046  11055  11066  11077#
SPAR  = 020000            213#  3267   3286
SRTSPL= 000011            195#  3310   4544   6141
SRTTAB  002110            605#  5066   5394   8643
ST    = 177776          11096# 11122  11123* 11136* 11350* 11376* 11378* 11404*
STACK = 001100             59#  2497   2509   2616   2665   2709   2723   2817   2822   2502   2909   2960   3004
                         3022   3045   3061   3077   3105   3144   3152   3174   3184   3208   3224   3260   3273
                         3305   3609   3616   3705   3765   3782   3871   3920   4010   4015   4114   4214   4224
                         4310   4357   4572   4578   4662   4668   4735   4836   4890   4981   5030   5099   5215
                         5309   5358   5425   5543   5636   5646   5837   5920   6170   6230   6354   6562   6573
                         6684   6807   6814   6923   6963   7069   7181   7193   7302   7342   7448   7558   7587
                         7696   7736   7842   7965   7990   8092   8973   9010  11074
START   007750            366  2493#
START1  010344           2567#
STKLMT= 177774             70#
ST2     010412           2574   2585#
ST3     010456           2575   2603#
ST4     010510           2587   2608   2611#
ST5     010534           2594   2616#  8121
SUBCLR  050320           3006   3024   3047   3063   3079   3109   3154   3176   3186   3210   3226   3262   3275
                         3307   3550   3560   3573   3618   3707   3784   3873   3922   4017   4116   4216   4226
                         4312   4358   4456   4541   4580   4670   4737   4838   4892   4983   5032   5100   5217
                         5260   5275   5311   5360   5427   5545   5586   5602   5638   5648   5769   5786   5794
                         5839   5921   6019   6102   6119   6127   6138   6183   6232   6356   6425   6442   6450
                         6499   6575   6686   6737   6754   6762   6816   6925   6965   7071   7119   7136   7144
                         7195   7304   7344   7450   7498   7515   7523   7561   7578   7589   7698   7738   7844
                         7892   7909   7917   7967   7982   7992   8377#
SVAL  = 100000            268#
SWR     001140            442#  2507   2529*  2531   2537*  2544*  3586   3591   4554   5292   5297   5618   5623
                         5807   5813   6151   6468   6474   6539   6544   6775   6781   7155   7160   7165   7170
                         7534   7539   7544   7549   7928   7933   7938   7943   8856   8877   8882   8884   8900
                         8908   8913   8935   8944   8949   8969   9033   9047   9049   9055   9062   9100   9107
                         9119   9123   9425   9462* 11070
SWREG   000176            364#  2537   9425   9438
SWTST   052770           3544   4531   5256   5582   5760   6094   6461   6533   6729   7113   7492   7886   8875#
SW0   = 000001            123#
SW00  = 000001            113#   123
SW01  = 000002            112#   122
SW02  = 000004            111#   121
SW03  = 000010            110#   120
SW04  = 000020            109#   119
SW05  = 000040            108#   118
SW06  = 000100            107#   117
SW07  = 000200            106#   116
SW08  = 000400            105#   115
SW09  = 001000            104#   114
SW1   = 000002            122#
SW10  = 002000            103#
SW11  = 004000            102#
SW12  = 010000            101# 11070
SW13  = 020000            100#  8900   8935
SW14  = 040000             99#  8877   8913   8949
SW15  = 100000             98#
SW2   = 000004            121#
SW3   = 000010            120#
SW4   = 000020            119#
```

```
SW5     = 000040            118#
SW6     = 000100            117#
SW7     = 000200            116#
SW8     = 000400            115#  8882
SW9     = 001000            114#  3586  3591  4554  5292  5297  5618  5623  5807  5813  6151  6468  6474
                            6539  6544  6775  6781  7155  7160  7165  7170  7534  7539  7544  7549  7928
                            7933  7938  7943  8856  8908  8944  8969
TBITVE= 000014              156#
TEMP1   004400             1380#  2712*  2726*  2731*  2736*  2737*  2738   2825*  2830*  2835*  2836*  2837   2912*
                            2919*  3010*  3015*  3089*  3090*  3091   3158*  3230*  3311*  3386*  3430*  3554*  3627*
                            3675*  3713*  3793*  3841*  3888*  3910*  3936*  3973*  4027*  4075*  4106*  4107*  4108
                            4124*  4142*  4143*  4144   4176*  4183*  4230*  4233*  4272*  4366*  4385*  4424*  4503*
                            4545*  4589*  4685*  4748*  4811*  4813   4846*  4858*  4898*  4921*  4964*  4966   4997*
                            5040*  5070*  5071   5109*  5136*  5176*  5225*  5264*  5268*  5325*  5368*  5398*  5399
                            5434*  5470*  5513*  5552*  5591*  5595*  5655*  5681*  5720*  5775*  5779*  5851*  5929*
                            5948*  5987*  6066*  6108*  6112*  6142*  6201*  6246*  6277*  6316*  6370*  6431*  6435*
                            6488*  6492*  6506*  6582*  6608*  6647*  6696*  6743*  6747*  6828*  6849*  6888*  6934*
                            6973*  6995*  7034*  7080*  7125*  7129*  7207*  7228*  7267*  7313*  7352*  7374*  7413*
                            7459*  7504*  7508*  7567*  7571*  7601*  7622*  7661*  7707*  7746*  7768*  7807*  7853*
                            7898*  7902*  7971*  7975*  7999*  8048*  8055*  8240*  8292*  8296*  8322*  8338   8340*
                            8360   8363*  8366*  8378*  8406   8410*  8413*  8416*  8426*  8434   8435*  8442*  8444*
                            8446*  8454   8455*  8462*  8464*  8466*  8474   8475*  8482*  8484*  8486*  8494   8495*
                            8502*  8504*  8506*  8542*  8547*  8559*  8564*  8576*  8581*  8612*  8615*  8616   8788
                            8790*  8792   8795*  8798*  8807   8808*  8809*  8810   8811*  8813   8816*  8819*  8838*
                            8846*  10853
TEMP2   004402             1381#  3009*  3017*  3346*  3636*  3802*  4036*  4138*  4139*  4140*  4141*  4144   4198*
                            4458*  4463*  4549*  5841*  5871*  5872   5892*  6021*  6026*  6146*  8070*  8298*  8566*
                            8613*  8614*  8616*  8617
TEMP3   004404             1382#  3880*  3882   3928*  3930   5842*  5874   5888*  5889   6820*  6822   7199*  7201
                            7593*  7595*  7596
TEMP4   004406             1383#  3881*  3882*  3884*  3885   3929*  3930*  3932*  3933   5882*  6821*  6822*  6823
                            7200*  7201*  7202   7594*  7595
TEMP5   004410             1384#  3612*  3747*
TIMUP   001416              569#  3424   3459   3492   3525   8752*  8763*  8769*
TITLE   047354             2569   2606   8142#
TKVEC = 000060              163#
TOCYL   001372              556#  3777*  3878   3879   3894   3896   3948   3950   3982*  3983   3990*  4987*  4989
                            5103*  5315*  5317   5430*  5641*  5652   5749   5754*  5755   5772   5790*  5791   6105
                            6123*  6124   6186*  6188   6225*  6418*  6428   6446*  6447   6485*  6528   6569*  6579
                            6678   6691   6718   6725*  6740   6758*  6759   6810*  6821   6825   6917   6929   6956
                            7108*  7109   7122   7140*  7141   7189*  7200   7204   7296   7308   7335   7487*  7488
                            7501   7519*  7520   7583*  7594   7598   7690   7702   7729   7881*  7882   7895   7913*
                            7914   7995*  7996   7997   8035   8039   10843  10856  10859
TPVEC = 000064              164#
TRAPPC  001354              546#  8965*  10865
TRAPVE= 000034              162#  2515*  2516*  9009*
TRT     = 000003           11101#  11594
TRTVEC= 000014              157#
TRUERR  051770             8675#
TSTATN  050016             3389   3530   3557   3678   3716   3844   3891   3913   3939   3976   4078   4127   4186
                            4275   4369   4388   4453   4506   5179   5516   5723   5932   5951   6016   6069   6319
                            6650   6891   7037   7270   7416   7664   7810   8002   8058   8273#
TST1    010716             2641   2663#
TST10   013102             3108   3142#
TST11   013206             3149   3163   3172#
TST12   013314             3196   3206#
```

```
TST13   013502          3243    3258#
TST14   013646          3290    3303#
TST15   015542          3545    3607#
TST16   016476          3751    3763#
TST17   020056          3989    4008#
TST2    011060          2686    2707#
TST20   021204          4212#
TST21   021640          4302    4308#
TST22   022016          4355#
TST23   023270          4532    4570#
TST24   023634          4660#
TST25   024624          4822    4834#
TST26   025504          4979#
TST27   026206          5097#
TST3    011470          2716    2815#
TST30   027362          5257    5307#
TST31   030064          5423#
TST32   031252          5583    5634#
TST33   032300          5761    5835#
TST34   032652          5877    5879    5898    5918#
TST35   034252          6095    6166#
TST36   036442          6462    6534    6560#
TST37   037654          6730    6805#
TST4    012012          2845    2900#
TST40   042006          7114    7179#
TST41   044152          7493    7556#
TST42   046376          7887    7963#
TST5    012174          2958#
TST6    012326          3002#
TST7    012744          3092    3103#
TYPOS = 104405          8110    9857#
TYPE  = 104401          2573    2588    2590    2592    2607    2648    2717    2746    2763    2786    2966    2980    2988
                        3112    3133    3134    3769    3774    4595    4607    4692    4754    4766    4853    4904    4916
                        5004    5046    5057    5231    5243    5332    5374    5385    5558    5570    5899    5903    5904
                        6173    6208    6377    6512    6524    6566    6702    6714    6940    6952    7086    7098    7319
                        7331    7465    7477    7713    7725    7859    7871    8108    8111    8145    8184    8347    8902
                        8937    8983    9011    9102    9110    9178    9272    9398    9436    9437    9440    9453    9464
                        9483    9536    9542    9547    9551    9556    9557    9559    9562    9566    9630    9632    9772
                        9853#   11022   11023   11027   11028   11037   11046   11050   11055   11057   11063   11066
TYPERR  072126          9109    11012#
TYPOC = 104402          9439    9854#   11043
TYPON = 104404          9856#
TYPOS = 104403          2749    2789    2984    8350    8905    8940    8986    9855#
T1      001432           579#   3636    3802    4036    4198    8070
T10     001434           580#   2712    2728    2731    2825    2830    2912    2919    3158    3230    3311    3386    3554
                        3627    3675    3713    3793    3841    3888    3910    3936    3973    4027    4075    4124    4176
                        4183    4272    4366    4385    4424    4458    4463    4503    4545    4858    4921    5109    5176
                        5264    5434    5513    5591    5555    5720    5775    5929    5948    5987    6021    6026    6066
                        6108    6142    6246    6316    6431    6488    6582    6647    6743    6828    6888    6973    7034
                        7125    7207    7267    7352    7413    7504    7567    7601    7661    7746    7807    7898    7971
                        9048    8055    8363    8378    8410
T100    001436           581#   3346    4549    6146
T2500   001442           583#   5136    5681    5851
T500    001440           582#   5040    5368    8475    8495
T5000   001444           594#   5268    5595    5779    6112    6435    6492    6747    6849    6995    7129    7228    7374
                        7508    7571    7622    7768    7902    7975    8435    8455
```

```
TSOOOO  001446          585*  3563  4589  4685  4748  4846  4898  4997  5225  5325  5552  6201  6370
                       6506  6696  6934  7080  7313  7459  7707  7853
UFE   = 000400          226*  2768  2861  3178  3195
UNLDRV  012012         2798  2889*
UNLOAD= 000007           194*  2918  3553
UNS   = 040000           251*
UPE   = 020000           231*
VV    = 000100           262*
WCE   = 040000           232*
WDCNT   001460           591*  4808* 4816* 4817  4963* 4969* 4970  5065* 5075* 5076  5393* 5403* 5404
                       10853
WLE   = 004000           248*
WORD    001470           597*  8714* 8715  8717  8727* 8728* 8729  8734* 8735* 8736* 8737
WPDATA= 000023           200*
WRHEAD= 000027           202*  4684  4845  4996  5324  6200  6369
WRL   = 004000           265*
WRTCHK= 000031           203*
WRTGAT= 040000           283*
$APTHD  001000           401   407*
$ASTAT= ****** U        9315  9330
$ATYC   054612          9286  9288*
$ATY1   054566          9284*
$ATY3   054574          9163  9285*
$ATY4   054604          9115  9287*
$AUTOB  001134           439*  9433  9582
$BASE   001264           509*  2611* 2668  8202* 8204*
$BDADR  001122           434*
$BDDAT  001126           436*
$BELL   001200           460*  9102  9135
$CDW1   001270           511*
$CDW2   001272           512*
$CHARC  054336          9180* 9190* 9197  9206* 9211*
$CKSUM  055262          9425* 9861
$CMTAG  001100           422*  2504  2505  2513  2519  2520  2521
$CM3  = 000000           452*
$CM4  = 000006           452*   453*  454*  455*  456*  457*  458*
$CNTLG  056147          9436  9577*
$CNTLU  056142          9453  9551  9576*
$CPUOP  001236           483*
$CRLF   001205           462*  2746  2988  3133  3134  5903  5904  9110  9135  9179  9214  9464  9556
                       9576  9635 11022 11027 11037 11050 11063
$DBLK   054556          9238  9272  9280*
$DB2D   056512          9687  9705*
$DB2O   056336          5901  9646*
$DDW0   001274           513*
$DDW1   001276           514*
$DDW10  001320           523*
$DDW11  001322           524*
$DDW12  001324           525*
$DDW13  001326           526*
$DDW14  001330           527*
$DDW15  001332           528*
$DDW2   001300           515*
$DDW3   001302           516*
$DDW4   001304           517*
$DDW5   001306           518*
```

```
$DDW6   001310            519#
$DDW7   001312            520#
$DDW8   001314            521#
$DDW9   001316            522#
$DECVL  056672           9707    9753#
$DEVCT  001220            474#   2560*   2621*   2970    8093*   8094
$DEVM   001266            510#
$DOAGN  047310           8104    8113    8119#
$DTBL   054546           9241    9276#
$ENDAD  047300            387    8115#   9130
$ENDCT  047246           2519    8106#
$ENDMG  047317           8108    8123#
$ENULL  047314           8111    8122#
$ENV    001230            479#   9112    9158    9293    9317
$ENVM   001231            480#   2542    9160    9165    9295
$EOP    047164           3319    3542    4596    4608    4693    4755    4767    4854    4905    4917    5005    5047    5058
                         5232    5244    5333    5375    5386    5559    5571    6209    6378    6513    6525    6703    6715
                         6941    6953    7087    7099    7320    7332    7466    7478    7714    7726    7860    7872    8077
                         8089#  11075
$EOPCT  047240           2519*   8103#   8107
$EOPI   047212           2690    2764    2967    8095    8097#   8924    8960
$ERFLG  001103            425#   8858    9022    9051    9053    9059*   9081    9097*   9135   11072
$ERMAX  001115            431#   2522*   9053    9076*   9081
$ERROR  053660           2513    9095#
$ERRPC  001116            432#   9104*   9105*   9106    9135   10833   10837   10840   10843   10846   10850   10853   10856
                        10859   10862   10866
$ERRTB  004454           1426#  11019
$ERTTL  001112            429#   9103*   9135
$ESCAP  001176            459#   2521*   3321*   3344*   3532*   3535*   3576*   4314*   4342*   4437*   4466*   4537*   5106*
                         5135*   5220*   5284*   5432*   5469*   5548*   5611*   5651*   5680*   5797*   6000*   6029*   6130*
                         6237*   6239*   6273*   6276*   6359*   6453*   6578*   6607*   6689*   6765*   6819*   6848*   6928*
                         6968*   6994*   7074*   7147*   7198*   7227*   7307*   7347*   7373*   7453*   7526*   7592*   7621*
                         7701*   7741*   7767*   7847*   7920*   8875*   8964*   8968*   9075*   9126    9128    9135
$ETABL  001230            478#
$ETEND  001334            413     531#
$FATAL  001212            471#   9321*
$FFLG   055032           9284*   9287*   9315    9324*   9332#
$FILLC  001156            450#   9183    9214
$FILLS  001155            449#   9214
$GDADR  001120            433#
$GDDAT  001124            435#
$GET42  047270           8112#
$GTSWR  055332           9437#   9859
$HD   =  000000             28
$HIBTS  001000            408#
$HIOCT  056334           9623*   9634#
$ICNT   001104            426#   9066*   9067    9069*   9080
$INTAG  001135            440#   9465    9582
$ITEMB  001114           4301    9106*   9114    9135   11013
$LF     001206            463#   9135    9214    9566    9576    9635
$LFLG   055031           9325*   9331#
$LPADR  001106            427#   2523*   3580    4538    5288    5615    5801    6134    6457    6769    7151    7530    7924
                         8911    8947    9057*   9073*   9078    9080
$LPERR  001110            428#   2524*   3579    5287    5614    5800    6133    6456    6768    7150    7529    7923    8860
                         8863*   8974    9057    9074*   9080    9125
$MADR1  001242            496#
```

```
$MADR2  001246            500#
$MADR3  001252            503#
$MADR4  001256            506#
$MAIL   001210            409     413     469#    2541    9072    9112    9158
$MAMS1  001240            490#
$MAMS2  001244            498#
$MAMS3  001250            501#
$MAMS4  001254            504#
$MBADR  001002            409#
$MFLG   055030           9285#    9291    9326#   9330#
$MNEW   056165           9440     9580#
$MSGAD  001224            476#    9301#   9304
$MSGLG  001226            477#    9306#
$MSGTY  001210            470#    9299    9307#   9319    9323#
$MSWR   056154           9437     9578#
$MTYP1  001241            491#
$MTYP2  001245            499#
$MTYP3  001251            502#
$MTYP4  001255            505#
$MXCNT  053656           9070     9080#
$NULL   001154            448#    9185    9214
$NWTST= 000001           2653#    2655    2693#   2695    2800#   2802    2891#   2893    2949#   2951    2993#   2995    3096#
                         3098     3136#   3138    3166#   3168    3199#   3201    3246#   3248    3293#   3295    3596#   3598
                         3754#    3756    3995#   3997    4203#   4205    4305#   4344#   4346    4561#   4563    4648#   4650
                         4625#    4827    4976#   5083#   5085    5304#   5410#   5412    5626#   5628    5817#   5819    5906#
                         5908     6157#   6159    6553#   6555    6788#   6790    7176#   7553#   7950#   7952
$OCNT   055256           9370#    9399#   9412#
$OCTVL  056440           3116     9648    9673#
$OMODE  055260           9365#    9369#   9374    9377#   9388#   9414#
$OVER   053642           9034     9050    9058    9068    9077#
$PASS   001216            473#    2541#   3107    3772    5897    8100#   8101#   8109    8122    9064    9081
$PASTM  001006            411#
$PWRCT  053376           9004#    9005#   9015#
$PWRON  053312           2517     8998#   9007
$PWRUP  053324           8998     9004#
$QUES   001204            461#    9135    9214    9483    9559    9576    9632    9635
$RDCHR  055544           9496#    9862
$RDDEC= ****** U         9865
$RDLIN  055664           9524#    9863
$RDOCT  056176           9596#    9864
$RDSZ = 000022           9517#
$RESRE  057004           9810#    9866
$RTNAD  047312           8121#
$R2A  = ****** U         9867
$SAVRE  056746           9794#    9865
$SB20   056456           9685#
$SCOPE  053400           2511     9031#
$SETUP= 000137           2466#    2510    2511    2513    2515    2517    2519    2520    2521    2523    8098    9032    9096
                         9122     9130    9420    9582
$STUP = 177777           2466#
$SUPRS  056706           3132     5902    9764#
$SVLAD  053606           9042     9071#
$SVPC = 000274            385#     390
$SWR  = 167400             13#      28      34      35      36      37      38      39      40      41     458     459     460
                         2520     2521    2523    2524    2664    2708    2816    2901    2959    3003    3104    3143    3173
                         3207     3259    3304    3608    3764    4009    4213    4309    4356    4571    4661    4835    4980
```

```
                    5098    5308    5424    5635    5836    5919    6167    6561    6806    7180    7557    7964    8086
                    8099    8114    8120    8122    9023    9024    9025    9026    9027    9033    9045    9047    9048
                    9051    9052    9053    9060    9061    9062    9074    9077    9080    9087    9088    9089    9090
                    9091    9100    9107    9119    9123    9135
$SWREG  001232       481#   2544
$SWRMK= 000000        41      42    9027    9028    9049
$TESTN  001214       472#   2961#   9072#  10833   10837   10840   10843   10846   10850   10853   10856   10859   10862
                   10865   10866
$TIMES  001174       458#   2520#   2664#   2708#   2816#   2901#   2959#   3003#   3104#   3143#   3173#   3207#   3259#
                    3304#   3608#   3764#   4009#   4213#   4309#   4356#   4571#   4661#   4835#   4980#   5098#   5308#
                    5424#   5635#   5836#   5919#   6167#   6561#   6806#   7180#   7557#   7964#   8099#   9060#   9067
                    9070#   9080
$TKB    001146       445#   9418    9429    9446    9500    9506
$TKS    001144       444#   9418    9427    9443    9467#   9498    9504
$TMP0   001160       452#   3563#   3569#   8676    8679#   8682#   8686    8688#   8689    8723#   8742#
$TMP1   001162       453#   8677    8680#   8683#   8684#   8690    8694    8696#   8697    8722#   8741#
$TMP2   001164       454#
$TMP3   001166       455#
$TMP4   001170       456#
$TMP5   001172       457#
$TN   = 000043        14#     28    2641    2653    2664#   2686    2693    2708#   2716    2800    2816#   2845    2891
                    2901#   2949    2959#   2961    2962    2993    3003#   3092    3096    3104#   3108    3136    3143#
                    3149    3163    3166    3173#   3196    3199    3207#   3243    3246    3259#   3290    3293    3304#
                    3545    3596    3608#   3751    3754    3764#   3989    3995    4009#   4203    4213#   4302    4305
                    4309#   4344    4356#   4532    4561    4571#   4648    4661#   4822    4825    4835#   4976    4980#
                    5083    5098#   5257    5304    5308#   5410    5424#   5583    5626    5635#   5761    5817    5836#
                    5877    5879    5898    5906    5919#   6095    6157    6167#   6462    6534    6553    6561#   6730
                    6788    6806#   7114    7176    7180#   7493    7553    7557#   7887    7950    7964#
$TNPWR  056622       9712    9713    9733#
$TPB    001152       447#   9203#   9214
$TPFLG  001157       451#   9152    9214
$TPS    001150       446#   9201    9214
$TRAP   057042       2515   9830#
$TRAP2  057064       9841#   9852
$TRP  = 000016       9845#   9854#   9855#   9856#   9857#   9858#   9859    9860#   9861    9862#   9863#   9864#   9865#
                    9866#   9867#   9868#
$TRPAD  057076       9835    9852#
$TSTM   001004        410#
$TSTNM  001102        424#   2962#   8098#   8884    9022    9049    9071#   9072    9077    9081    9099    9135
$TTYIN  056120       9526    9527    9539    9557    9571    9575#
$TYPBN= ****** U     9858
$TYPOS  054342       9226#   9857
$TYPE   054060       9152#   9312    9845    9853
$TYPEC  054272       9182    9189    9196    9201#   9202    9469
$TYPEX  054340       9207    9209    9212#
$TYPOC  055060       9368#   9854
$TYPON  055074       9367    9370#   9856
$TYPOS  055034       9363#   9855
$UNIT   001222        475#   2561#   2622#   2972#   2977    2981    3384    3623    3673    3789    3839    3908    3971
                    4023    4073    4174    4181    4270    4383    4501    5174    5205    5511    5718    5946    6064
                    6314    6645    6676    6886    7032    7265    7411    7659    7805    8046    8053    8274    8293
                    8319    8361    8381    8408    8789
$UNITM  001010        412#
$USWR   001234        482#
$VECT1  001260        507#
$VECT2  001262        508#
```

```
$XTSTR  053412          9036#
$$GET4= 000000          8114#
$OFILL  055257          9364*   9368*   9378    9413#
$$OCAT= ****** U        9033    9109
.       = 072466         358#    362#    367#    369#    371#    374#    376#    378#    385     386#    388#    390#    391#
                         397     398#    400#    402#    421#    464     603#    604#    605#    607#    608#    2508    2523
                        2524    8122    8126    9000    9080    9081    9135    9214    9280#   9333#   9418    9575#   9576
                        9582    9635    9673#   9753#  10832#  11086#  11087#  11232   11457   11461   11493   11517   11579
                       11587   11590#  11598#
.$ASTA= ****** U         9285    9288
.$X   = 001000           397#    402
```

```
CALIB     742#  3621  3787  4021
CIDAE     772#  4171  8043
CKWD12    647#  3349  3393  3639  3682  3805  3848  4039  4082  4240  4279  4392  4467  4510  4610
         4695  4759  4865  4928  5007  5144  5183  5335  5476  5520  5688  5727  5955  6030  6073
         6284  6323  6380  6615  6654  6856  6895  7002  7041  7235  7274  7381  7420  7629  7668
         7775  7814
CKWD3     675#  3370  3659  3825  4059  4487  4631  4716  4790  4949  5497  6050
COMMEN      1#   166#
DRCLR     720#  3382  3671  3837  3905  3968  4071  4179  4267  4381  4499  5171  5509  5715  5944
         6062  6311  6642  6883  7029  7262  7408  7656  7802  8051
ENDCOM      1#   166#
EOPGM    1339#  8090
ERROR      60#  2689  2714  2728  2733  2755  2777  2781  2792  2796  2827  2832  2848  2856  2874
         2878  2882  2886  2914  2921  2924  2927  2930  2933  3007  3019  3025  3033  3038  3042
         3048  3054  3058  3064  3070  3074  3080  3087  3093  3110  3155  3160  3164  3177  3180
         3187  3194  3197  3211  3214  3217  3220  3227  3232  3235  3238  3241  3244  3263  3266
         3269  3276  3282  3285  3288  3291  3308  3313  3318  3325  3330  3336  3340  3348  3355
         3358  3365  3368  3375  3379  3388  3391  3398  3401  3408  3411  3426  3436  3440  3446
         3450  3461  3469  3473  3479  3483  3494  3502  3506  3512  3516  3527  3531  3541  3551
         3556  3558  3561  3571  3574  3619  3629  3635  3638  3644  3647  3654  3657  3664  3668
         3677  3680  3687  3690  3697  3700  3708  3715  3717  3723  3727  3734  3737  3743  3746
         3785  3795  3801  3804  3810  3813  3820  3823  3830  3834  3843  3846  3853  3856  3863
         3866  3874  3890  3892  3898  3903  3912  3915  3923  3938  3940  3946  3955  3961  3966
         3975  3978  4018  4029  4035  4038  4044  4047  4054  4057  4064  4068  4077  4080  4087
         4090  4097  4100  4110  4117  4126  4128  4148  4154  4163  4178  4185  4188  4195  4200
         4217  4221  4227  4232  4236  4239  4246  4249  4256  4259  4265  4274  4277  4284  4287
         4294  4297  4303  4313  4319  4323  4333  4337  4359  4368  4370  4375  4378  4387  4390
         4397  4400  4407  4410  4417  4420  4426  4430  4434  4443  4446  4452  4454  4457  4460
         4462  4465  4472  4475  4482  4485  4492  4496  4505  4508  4515  4518  4525  4528  4542
         4547  4551  4581  4591  4594  4606  4615  4618  4625  4628  4636  4640  4671  4687  4691
         4700  4703  4710  4713  4721  4725  4738  4750  4753  4765  4774  4777  4784  4787  4795
         4799  4815  4839  4848  4852  4860  4863  4871  4874  4881  4884  4893  4900  4903  4915
         4923  4926  4934  4937  4944  4947  4954  4958  4968  4984  4999  5003  5012  5015  5022
         5025  5033  5042  5045  5056  5073  5101  5111  5114  5117  5124  5127  5132  5140  5143
         5150  5153  5160  5163  5169  5178  5181  5188  5191  5198  5201  5210  5218  5227  5230
         5242  5250  5254  5261  5266  5270  5273  5276  5281  5312  5327  5331  5340  5343  5350
         5353  5361  5370  5373  5384  5401  5428  5436  5442  5449  5453  5459  5462  5467  5472
         5475  5482  5485  5492  5495  5502  5506  5515  5518  5525  5528  5535  5538  5546  5554
         5557  5569  5576  5580  5587  5593  5597  5600  5603  5609  5639  5649  5657  5660  5663
         5670  5673  5678  5684  5687  5694  5697  5704  5707  5713  5722  5725  5732  5735  5742
         5745  5751  5770  5777  5781  5784  5787  5795  5840  5853  5857  5861  5864  5876  5878
         5884  5922  5931  5933  5938  5941  5950  5953  5960  5963  5970  5973  5980  5983  5989
         5993  5997  6006  6009  6015  6017  6020  6023  6025  6028  6035  6038  6045  6048  6055
         6059  6068  6071  6078  6081  6088  6091  6103  6110  6114  6117  6120  6128  6139  6144
         6148  6177  6181  6184  6203  6207  6233  6248  6251  6254  6261  6264  6269  6280  6283
         6290  6293  6300  6303  6309  6318  6321  6328  6331  6338  6341  6350  6357  6372  6376
         6385  6388  6395  6398  6426  6433  6437  6440  6443  6451  6490  6494  6497  6500  6508
         6511  6523  6530  6576  6584  6587  6590  6597  6600  6605  6611  6614  6621  6624  6631
         6634  6640  6649  6652  6659  6662  6669  6672  6680  6687  6698  6701  6713  6720  6738
         6745  6749  6752  6755  6763  6817  6830  6833  6836  6843  6846  6852  6855  6862  6865
         6872  6875  6881  6890  6893  6900  6903  6910  6913  6919  6926  6936  6939  6951  6958
         6966  6975  6978  6981  6988  6991  6998  7001  7008  7011  7018  7021  7027  7036  7039
         7046  7049  7056  7059  7065  7072  7082  7085  7097  7104  7120  7127  7131  7134  7137
         7145  7196  7209  7212  7215  7222  7225  7231  7234  7241  7244  7251  7254  7260  7269
         7272  7279  7282  7289  7292  7298  7305  7315  7318  7330  7337  7345  7354  7357  7360
         7367  7370  7377  7380  7387  7390  7397  7400  7406  7415  7418  7425  7428  7435  7438
```

```
        7444    7451    7461    7464    7476    7483    7499    7506    7510    7513    7516    7524    7562    7569    7573
        7576    7579    7590    7603    7606    7609    7616    7619    7625    7628    7635    7638    7645    7648    7654
        7663    7666    7673    7676    7683    7686    7692    7699    7709    7712    7724    7731    7739    7748    7751
        7754    7761    7764    7771    7774    7781    7784    7791    7794    7800    7809    7812    7813    7822    7829
        7832    7838    7845    7855    7858    7870    7877    7893    7900    7904    7907    7910    7918    7968    7973
        7977    7980    7983    7993    8001    8003    8007    8010    8013    8016    8021    8024    8030    8033    8037
        8041    8050    8057    8060    8067    8072    8365    8380    8415    8710    8966
ESCAPE     1#   166#
FSECA   1078#   8429    8449
FSECB   1104#   8469    8489
GETPRI     1#   166#
GETSWR     1#   166#
HDCHK3  1005#   6502    6692    6930    7076    7309    7455    7703    7849
HDTBL   1066#   4988    5316    6187
HEADER  1254#   4975    5303
LIMIT   1163#   4358    5921
LOOP     627#   2721    2820    2907    3020    3044    3060    3076    3151    3182    3222    3271    3614    3704    3781
        3870    3919    4013    4112    4222    4576    4667    4734    4888    5029    5213    5357    5541    5644    6228
        6352    6571    6682    6812    6921    6962    7067    7191    7300    7341    7446    7585    7694    7735    7840
        7988
LPCHK    691#   3576    5283    5610    5797    6130    6453    6765    7147    7526    7920
MSG     2653#   2655    2693#   2695    2800#   2802    2891#   2893    2949#   2951    2993#   2995    3096#   3098    3136#
        3138    3166#   3168    3199#   3201    3246#   3248    3293#   3295    3596#   3598    3754#   3756    3995#   3997
        4203#   4205    4344#   4346    4561#   4563    4647#   4650    4825#   4827    5082#   5085    5409#   5412    5626#
        5628    5817#   5819    5906#   5908    6157#   6159    6553#   6555    6788#   6790    7950#   7952
MULT       1#   166#
NEWTST     1#   166#   2653    2693    2800    2891    2949    2993    3096    3136    3166    3199    3246    3293    3596
        3754    3995    4203    4305    4344    4561    4648    4825    4976    5083    5304    5410    5626    5817    5906
        6157    6553    6788    7176    7553    7950
QWNTAG   415#   533
POP        1#   166#   9267    9327    9328    9624    9815
PUSH       1#   166#   9226    9288    9290    9311    9598    9795
QKRPSK   886#   5767    6100    6423    6735    7117    7496    7890
QKSEEK   862#   5262    5589    5773    6106    6429    6486    6741    7123    7502    7565    7896    7969
QKSRT    842#   4539    6136
QKUNLD   908#   3548
RALLHD  1015#   5036    5364
RD.QDR   969#   4586    4745    4895    5222    5549    6503    6693    6931    7077    7310    7456    7704    7850
REPORT     1#   166#
SCOPE     61#   2663    2707    2815    2900    2958    3002    3103    3142    3172    3206    3258    3303    3607    3763
        4008    4212    4308    4355    4570    4660    4834    4979    5097    5307    5423    5634    5835    5918    6166
        6560    6805    7179    7556    7963    8091    8097
SECTST  1131#   4317
SETPRI     1#   166#
SETTRA  9845#   9854    9855    9856    9857    9859    9861    9862    9863    9864    9865    9866    9867
SETUP      1#   166#   2502
SKATN    822#   4234    5138    5682    6278    6609    6850    6996    7229    7375    7623    7769
SKIP       1#   166#   2641    2686    2716    2845    3092    3108    3149    3163    3196    3243    3290    3545    3751
        3989    4302    4532    4822    5257    5583    5761    5877    5879    5898    6095    6462    6534    6730    7114
        7493    7887
SKOSC   1281#   6808    7187    7581
SKRDY    797#   5107    5653    6244    6580    6826    6971    7205    7350    7599    7744
SLASH      1#   166#
SPACE    166#
STARS      1#   166#    383     394     396     403     417     464     467    2653    2662    2693    2706    2800    2814
        2891    2899    2949    2957    2993    3001    3096    3102    3136    3141    3166    3171    3199    3205    3246
```

```
            3257    3293    3302    3596    3606    3754    3762    3995    4007    4203    4211    4305    4307    4344    4354
            4561    4569    4648    4659    4825    4833    4976    4978    5083    5096    5304    5306    5410    5422    5626
            5633    5817    5834    5906    5917    6157    6165    6553    6559    6788    6804    7176    7178    7553    7555
            7950    7962    8083    9019    9083    9137    9216    9283    9340    9417    9420    9488    9517    9584    9637
            9676    9694    9756    9779    9824   11002   11011
SWRSU         1#    166#   2525#
SW814       699#   3543    4531    5255    5581    5759    6094    6460    6533    6728    7112    7491    7885
TRMTRP     9845#
TSTSW9      709#   3585    3590    4553    5290    5295    5616    5621    5805    5811    6150    6466    6472    6537    6542
            6773    6779    7153    7158    7163    7168    7532    7537    7542    7547    7926    7931    7936    7941
TYPBIN        1#    166#
TYPDEC        1#    166#   8109
TYPNAM        1#    166#
TYPNUM        1#    166#
TYPOCS        1#    166#   2747    2787    2982    8348    8903    8938    8984
TYPOCT        1#    166#   9438
TYPTXT        1#    166#
WRHDR       946#   4683    4844    4995    5323    6199    6368
$$CMRE      415#
$$CMTM      415#    452     453     454     455     456     457
$$ESCA        1#    166#
$$NEWT        1#    166#   2653    2693    2800    2891    2949    2993    3096    3136    3166    3199    3246    3293    3596
            3754    3995    4203    4305    4344    4561    4648    4825    4976    5083    5304    5410    5626    5817    5906
            6157    6553    6788    7176    7553    7950
$$SET      9845#   9854    9855    9856    9857    9859    9861    9862    9863    9864    9865    9866    9867
$$SETM     2541#
$$SKIP        1#    166#   2641    2686    2716    2845    3092    3108    3149    3163    3196    3243    3290    3545    3751
            3989    4302    4532    4822    5257    5583    5761    5877    5879    5898    6095    6462    6534    6730    7114
            7493    7887
.EQUAT        1#      9#     56
.HEADE        1#      9#     18
.KTI1         1#
.SETUP        1#      9#   2466
.SWRHI        1#      9#     30
.SWRLO        9#     42#
.$ACTI        1#     12#    381
.$APTB        1#    465#
.$APTH        1#     12#    392
.$APTY        1#     12#   9281
.$ASTA        1#
.$CATC        1#      9#    356
.$CMTA        1#      9#    415
.$DB2D        1#     12#   9692
.$DB20        1#     11#   9635
.$DIV         1#
.$EOP         1#     10#   8081
.$ERRO        1#     10#   9081
.$ERRT        1#
.$MULT        1#
.$POWE        1#
.$RAND        1#
.$RDDE        1#
.$RDOC        1#     11#   9532
.$READ        1#     11#   9415
.$R2AZ        1#
.$SAVE        1#     10#   9777
```

```
.$S82D      1#      12#   9674
.$S820      1#
.$SCOP      1#      10#   9017
.$SIZE      1#
.$SUPR      1#      11#   9754
.$TRAP      1#      11#   9822
.$TYPB      1#
.$TYPD      1#      11#   9214
.$TYPE      1#      10#   9135
.$TYPO      1#      10#   9339
.$40CA      1#
.1170       1#
```

```
ADC      9722
ADD      3189  4325  5870  8075  8244  8278  8307  8331  8387  8447  8467  8487  8507  8551  8569
         8584  8634  8640  8642  8664  8744  8796  8817  8879  9171  9246  9298  9310  9322  9366
         9376  9454  9463  9619  9671  9688  9721  9723 11019 11061 11157 11167 11207 11328 11339
        11474 11482
ASL      4681  4810  7108  8597  8598  8599  8600  8601  8604  8661  8663  9477  9478  9479  9612
         9614  9616  9834 11016 11017 11018 11156 11170 11181 11204 11205 11206 11217 11241 11275
        11336 11348 11535
ASLB     9251
ASR      4140  6397  8398  8399  8400  8421  8422  8423  8424  8516  8517  8518  8519  8531  8532
         8533  8534  9305  9663 11168 11179 11239 11273 11285 11346 11476 11477 11478 11533
BCC      8839  9252
BCS     11169 11180 11240 11274 11286 11347 11534
BED      2543  2716  2773  2785  2840  2850  2866  2869  2871  2916  2929  2971  2974  2976  2978
         3014  3092  3179  3193  3196  3213  3216  3219  3234  3237  3265  3268  3374  3378  3425
         3458  3460  3493  3526  3568  3578  3663  3667  3722  3726  3768  3829  3833  3897  3902
         3945  3954  3960  3965  3984  3989  4063  4067  4109  4147  4168  4194  4220  4238  4264
         4302  4329  4332  4336  4491  4495  4593  4604  4635  4639  4690  4720  4724  4752  4763
         4794  4798  4814  4822  4851  4902  4913  4953  4957  4967  5002  5044  5054  5072  5131
         5142  5168  5209  5229  5240  5253  5272  5280  5286  5330  5372  5382  5400  5466  5474
         5501  5505  5556  5567  5579  5599  5608  5613  5677  5686  5712  5750  5756  5783  5799
         5856  5860  5863  5869  5883  5890  6054  6058  6116  6132  6172  6176  6180  6206  6213
         6236  6268  6272  6282  6308  6349  6375  6403  6439  6455  6481  6496  6510  6521  6529
         6565  6604  6613  6639  6679  6700  6711  6719  6724  67[illegible]  6767  6854  6880  6918  6938
         6949  6957  7000  7026  7064  7084  7095  7103  7110  71[illegible]3  7149  7184  7233  7259  7297
         7317  7328  7336  7379  7405  7443  7463  7474  7482  7489  7512  7528  7575  7627  7653
         7691  7711  7722  7730  7773  7799  7837  7857  7868  7876  7883  7906  7922  7979  8012
         8036  8040  8066  8077  8095  8113  8162  8170  8179  8201  8214  8276  8339  8385  8441
         8461  8478  8498  8687  8695  8703  8716  8718  8730  8738  8857  8859  8878  8883  8985
         8909  8914  8945  8950  9048  9050  9052  9056  9065  9098  9101  9124  9127  9161  9174
         9209  9292  9296  9316  9318  9393  9434  9461  9476  9545  9607  9658  9767  9769 11021
        11026 11031 11035 11041 11045 11052 11071 11151 11178 11212 11237 11298 11326 11332 11371
        11374 11409 11459 11497 11499 11501 11537
BGE      4134  4157  9068
BGT      8104  9260  9400  9473  9514  9609  9657 11419 11526
BHI      9054 11153 11200 11303 11480
BHIS    11215
BIC      2737  2836  3085  3090  3126  4107  4139  8101  8157  8177  8396  8420  8515  8530  8614
         8615  8682  8683  8728  8736  8809  9390  9430  9447  9474  9501  9507  9515  9618  9670
        11014 11154 11203 11245 11309 11310 11312 11333 11353 11416 11473 11481 11496
BIS      3127  4673  4840  8608  8609  8610  8616  8789  8810  8846  9254  9255  9395  9396  9481
         9725 11192 11251 11311 11380
BIT      2734  2766  2768  2770  2772  2784  2833  2859  2861  2863  2865  2868  2928  2931  3013
         3029  3148  3162  3178  3192  3195  3212  3215  3218  3233  3236  3239  3242  3264  3267
         3280  3283  3286  3289  3316  3422  3457  3490  3523  3539  3567  3586  3591  3633  3799
         4033  4193  4237  4432  4554  4592  4603  4689  4751  4762  4850  4861  4901  4912  4924
         5001  5043  5053  5141  5228  5239  5271  5292  5297  5329  5371  5381  5440  5473  5555
         5566  5598  5618  5623  5685  5782  5807  5813  5855  5859  5862  5995  6115  6151  6205
         6281  6374  6438  6468  6474  6495  6509  6520  6539  6544  6612  6699  6710  6750  6775
         6781  6853  6937  6948  6999  7083  7094  7132  7155  7160  7165  7170  7232  7316  7327
         7378  7462  7473  7511  7534  7539  7544  7549  7574  7626  7710  7721  7772  7856  7867
         7905  7928  7933  7938  7943  7978  8005  8008  8011  8014  8065  8238  8384  8411  8545
         8562  8579  8844  8856  8877  8882  8900  8908  8913  8935  8944  8949  8969  9033  9047
         9055  9062  9100  9107  9123 11070 11458
BITB     2542  2925  8275  8294  8320  9160  9165  9197  9295
BLE     11049
```

```
BLO     9540  11202  11509
BLOS    9528
BLT     9188   9243   9259   9401   9471   9512   9611   9718
BMI     9250
BNE     2508   2532   2568   2571   2586   2604   2735   2739   2741   2743   2760   2762   2767   2769   2771
        2834   2838   2844   2860   2862   2864   2923   2926   2932   2937   2965   3016   3018   3030   3108
        3130   3149   3163   3240   3243   3281   3284   3287   3290   3317   3343   3423   3491   3524   3540
        3570   3587   3592   3634   3750   3773   3800   3949   3951   4034   4339   4433   4555   4645   4731
        4818   4862   4925   4971   5061   5077   5249   5293   5298   5389   5405   5441   5575   5619   5624
        5792   5808   5814   5873   5875   5898   5996   6125   6152   6408   6410   6414   6448   6469   6475
        6540   6545   6760   6776   6782   7142   7156   7161   7166   7171   7521   7535   7540   7545   7550
        7915   7929   7934   7939   7944   8006   8009   8015   8135   8143   8166   8168   8181   8188   8239
        8241   8295   8297   8299   8321   8323   8412   8414   8438   8443   8458   8463   8481   8483   8501
        8503   8521   8546   8548   8563   8565   8567   8580   8582   8621   8647   8652   8754   8765   8768
        8778   8794   8815   8842   8845   8901   8936   8970   9006   9034   9063   9108   9113   9131   9159
        9166   9168   9176   9184   9198   9205   9248   9294   9300   9303   9320   9391   9426   9432   9452
        9459   9466   9503   9509   9532   9534   9550   9554   9564   9729  11054  11060  11065  11073  11306
       11313  11392  11520
BPL     3883   3931   9120   9153   9202   9234   9264   9389   9428   9444   9499   9505  11395  11457  11461
       11493  11517  11523
BR      2472   2477   2482   2487   2492   2534   2553   2594   2608   2631   2641   2644   2686   2752   2774
        2778   2782   2793   2797   2845   2853   2857   2867   2875   2879   2883   2887   3034   3390   3545
        3679   3751   3845   3914   3956   3962   3977   4079   4160   4187   4276   4389   4507   4532   5180
        5257   5517   5583   5724   5761   5865   5877   5879   5886   5895   5952   6070   6095   6217   6238
        6274   6320   6462   6534   6651   6730   6892   7038   7114   7271   7417   7493   7665   7811   7887
        8059   8182   8189   8219   8341   8692   8700   8708   8711   8720   8732   8739   9000   9036   9042
        9045   9058   9061   9118   9155   9181   9191   9200   9207   9245   9262   9286   9308   9367   9382
        9403   9455   9482   9484   9510   9543   9552   9558   9560   9620   9633   9672   9720  11047  11062
       11067  11120  11121  11129  11138  11143  11159  11161  11183  11185  11209  11216  11244  11246  11252
       11263  11269  11295  11324  11341  11381  11397  11417  11512  11547
CCC    11335  11399
CLC     8836
CLR     2470   2471   2474   2476   2481   2484   2485   2489   2494   2495   2506   2520   2521   2541   2557
        2560   2561   2621   2622   2647   2718   2719   2818   2852   2904   3040   3056   3068   3072   3146
        3328   3338   3356   3366   3399   3409   3432   3438   3448   3465   3471   3481   3498   3504   3514
        3521   3535   3537   3576   3611   3645   3655   3688   3698   3730   3744   3776   3777   3778   3779
        3811   3821   3854   3864   4012   4045   4055   4088   4098   4142   4150   4159   4247   4257   4285
        4295   4321   4342   4361   4380   4398   4408   4418   4439   4466   4473   4483   4516   4526   4537
        4574   4616   4626   4665   4701   4711   4733   4775   4785   4808   4872   4882   4887   4935   4945
        4963   5013   5023   5028   5065   5102   5115   5125   5151   5161   5189   5199   5220   5284   5341
        5351   5356   5393   5430   5444   5451   5460   5483   5493   5526   5536   5548   5611   5640   5661
        5671   5695   5705   5733   5743   5797   5841   5842   5843   5867   5892   5924   5943   5961   5971
        5981   6002   6029   6036   6046   6079   6089   6130   6186   6220   6221   6224   6252   6262   6291
        6301   6329   6339   6359   6386   6396   6415   6453   6484   6485   6588   6598   6622   6632   6660
        6670   6689   6765   6834   6844   6863   6873   6901   6911   6928   6979   6989   7009   7019   7047
        7057   7074   7147   7213   7223   7242   7252   7280   7290   7307   7358   7368   7388   7398   7426
        7436   7453   7526   7607   7617   7636   7646   7674   7684   7701   7752   7762   7782   7792   7820
        7830   7847   7920   7985   7986   8031   8098   8099   8133   8171   8300   8304   8324   8328   8382
        8522   8541   8595   8752   8763   8779   8781   8835   8875   8876   8968   9004   9060   9075   9237
        9240   9380   9441   9442   9525   9548   9604   9605   9653   9714  11032  11196  11197  11210  11296
       11427
CLRB    2493   3131   9059   9180   9206   9266   9324   9325   9326   9555   9565   9629   9730  11140  11345
       11352  11372  11448  11449  11524
CMP     2507   2531   2555   2633   2643   2646   2688   2738   2759   2837   2843   2936   3091   3721   3725
        3749   3896   3901   3944   3948   3959   3964   3983   3988   4108   4133   4144   4156   4167   4328
        4331   4335   4644   4730   4813   4817   4821   4966   4970   5060   5071   5076   5130   5208   5252
```

```
        5388  5399  5404  5574  5607  5676  5749  5755  5791  5872  5889  6124  6212  6267  6348
        6407  6409  6413  6447  6480  6528  6603  6678  6718  6759  6917  6356  7063  7102  7109
        7141  7296  7335  7442  7481  7488  7520  7690  7729  7836  7875  7882  7914  8035  8039
        8076  8094  8161  8165  8169  8178  8477  8480  8497  8500  8520  8620  8646  8651  8715
        8717  8729  8737  8793  8814  8910  8923  8946  8959  9043  9067  9130  9258  9425  9431
        9451  9458  9470  9472  9502  9508  9511  9513  9527  9539  9724  11072 11152 11199 11201
       11214 11302 11325 11331 11340 11408 11479 11508
CMPB    2570  2603  8884  9049  9053  9112  9158  9173  9175  9183  9204  9208  9293  9433  9465
        9531  9549  9553  9563  9608  9610  9768  11150 11211 11498 11500 11519
COM    11036 11053 11301
DEC     2851  3015  3017  3129  3569  4338  5790  6446  6723  6725  7140  7519  7881  8102  8240
        8296  8298  8322  8340  8413  8442  8462  8482  8502  8547  8564  8566  8581  8688  8696
        8704  8764  8767  8841  9538  9656  9727  9770  11015 11044 11048 11279 11280 11418
DECB    9187  9190  9388  9399  11525
EMT       60
HALT     362  8917  8953  8989  8999  9121  9132  9154
INC     2572  2576  2605  2609  2614  2625  2629  2636  2638  2744  2745  2758  2842  2935  2972
        3585  3590  3982  3987  4166  4553  4643  4729  4816  4820  4969  5075  5291  5296  5403
        5617  5622  5753  5754  5806  5812  5858  5871  5885  5888  6123  6150  6406  6416  6417
        6418  6467  6473  6479  6538  6543  6758  6774  6780  7154  7159  7164  7169  7487  7533
        7538  7543  7548  7913  7927  7932  7937  7942  8093  8100  8134  8144  8164  8174  8175
        8187  8191  8549  8619  8769  8840  9005  9066  9103  9244  9323  9394  9402  9480  9659
        9719 11158 11208 11213 11294 11329 11330 11337 11338
INCB    9071  9097  9210 11379 11421
IOT       61
JMP      366   368   370   372   375   377   379  2574  2575  2587  2690  2764  2798  2967  3319
        3414  3533  3542  3579  3580  3588  3593  3752  3770  3985  3991  4169  4435  4538  4556
        4596  4608  4693  4755  4767  4823  4854  4905  4917  5005  5047  5058  5232  5244  5287
        5288  5294  5299  5333  5375  5386  5559  5571  5614  5615  5620  5625  5757  5800  5801
        5609  5815  5998  6133  6134  6153  6178  6182  6209  6378  6404  6411  6419  6456  6457
        6470  6476  6482  6513  6525  6541  6546  6567  6703  6715  6726  6768  6769  6777  6783
        6941  6953  7087  7099  7111  7150  7151  7157  7162  7167  7172  7185  7320  7332  7466
        7478  7490  7529  7530  7536  7541  7546  7551  7714  7726  7860  7872  7884  7923  7924
        7930  7935  7940  7945  8078  8096  8120  8185  8911  8924  8947  8960  8974  9013 11075
       11125 11137 11173 11218 11288 11428 11541
JSR     2559  2569  2589  2591  2593  2606  2613  2713  2727  2732  2754  2776  2780  2795  2826
        2831  2847  2855  2873  2877  2881  2885  2913  2920  3006  3012  3024  3028  3032  3037
        3041  3047  3051  3053  3057  3063  3067  3069  3073  3079  3083  3086  3109  3114  3132
        3147  3154  3159  3176  3186  3190  3210  3226  3231  3262  3275  3279  3307  3312  3315
        3324  3329  3333  3335  3339  3347  3352  3354  3357  3362  3364  3367  3372  3376  3387
        3389  3395  3397  3400  3405  3407  3410  3419  3421  3428  3431  3433  3435  3439  3443
        3445  3449  3454  3456  3463  3466  3468  3472  3476  3478  3482  3487  3489  3496  3499
        3501  3505  3509  3511  3515  3520  3522  3529  3530  3538  3544  3550  3555  3557  3560
        3566  3573  3584  3589  3618  3628  3632  3637  3641  3643  3646  3651  3653  3656  3661
        3665  3676  3678  3684  3686  3689  3694  3696  3699  3707  3714  3716  3720  3724  3731
        3733  3736  3740  3742  3745  3784  3794  3798  3803  3807  3809  3812  3817  3819  3822
        3827  3831  3842  3844  3850  3852  3855  3860  3862  3865  3873  3889  3891  3895  3900
        3911  3913  3922  3937  3939  3943  3952  3958  3963  3974  3976  4017  4028  4032  4037
        4041  4043  4046  4051  4053  4056  4061  4065  4076  4078  4084  4086  4089  4094  4096
        4099  4105  4116  4125  4127  4137  4151  4153  4162  4177  4184  4196  4192  4199  4216
        4218  4226  4231  4235  4243  4245  4248  4253  4255  4258  4262  4267  4275  4281  4283
        4286  4291  4293  4296  4300  4312  4318  4322  4327  4330  4358  4367  4369  4372  4374
        4377  4386  4388  4394  4396  4399  4404  4406  4409  4414  4416  4419  4425  4428  4429
        4440  4442  4445  4449  4451  4453  4456  4459  4461  4464  4469  4471  4474  4479  4481
        4484  4489  4493  4504  4506  4512  4514  4517  4522  4524  4527  4531  4541  4546  4550
        4580  4590  4605  4612  4614  4617  4622  4624  4627  4633  4637  4670  4686  4688  4697
```

```
                4699    4702    4707    4709    4712    4718    4722    4737    4749    4764    4771    4773    4776    4781    4783
                4786    4792    4796    4838    4847    4849    4859    4868    4870    4873    4878    4880    4883    4892    4899
                4914    4922    4931    4933    4936    4941    4943    4946    4951    4955    4983    4992    4998    5000    5009
                5011    5014    5019    5021    5024    5032    5041    5055    5063    5100    5110    5113    5116    5121    5123
                5126    5129    5139    5147    5149    5152    5157    5159    5162    5166    5177    5179    5185    5187    5190
                5195    5197    5200    5206    5217    5226    5241    5256    5260    5265    5269    5275    5278    5311    5320
                5326    5328    5337    5339    5342    5347    5349    5352    5360    5369    5383    5391    5427    5435    5439
                5445    5448    5452    5456    5458    5461    5464    5471    5479    5481    5484    5489    5491    5494    5499
                5503    5514    5516    5522    5524    5527    5532    5534    5537    5545    5553    5568    5582    5586    5592
                5596    5602    5606    5638    5648    5656    5659    5662    5667    5669    5672    5675    5683    5691    5693
                5696    5701    5703    5706    5710    5721    5723    5729    5731    5734    5739    5741    5744    5748    5760
                5769    5776    5780    5786    5794    5839    5852    5854    5901    5902    5921    5930    5932    5935    5937
                5940    5949    5951    5957    5959    5962    5967    5969    5972    5977    5979    5982    5988    5991    5992
                6003    6005    6008    6012    6014    6016    6019    6022    6024    6027    6032    6034    6037    6042    6044
                6047    6052    6056    6067    6069    6075    6077    6080    6085    6087    6090    6094    6102    6109    6113
                6119    6127    6138    6143    6147    6183    6191    6202    6204    6216    6232    6247    6250    6253    6258
                6260    6263    6266    6279    6287    6289    6292    6297    6299    6302    6306    6317    6319    6325    6327
                6330    6335    6337    6340    6347    6356    6360    6370    6373    6382    6384    6387    6392    6394    6397
                6425    6432    6436    6442    6450    6461    6489    6493    6499    6507    6522    6533    6575    6583    6586
                6589    6594    6596    6599    6602    6610    6618    6620    6623    6628    6630    6633    6637    6648    6650
                6656    6658    6661    6666    6668    6671    6677    6686    6697    6712    6729    6737    6744    6748    6754
                6762    6816    6829    6832    6835    6840    6842    6845    6851    6859    6861    6864    6869    6871    6874
                6878    6889    6891    6897    6899    6902    6907    6909    6912    6916    6925    6935    6950    6965    6974
                6977    6980    6985    6987    6990    6997    7005    7007    7010    7015    7017    7020    7024    7035    7037
                7043    7045    7048    7053    7055    7058    7062    7071    7081    7096    7113    7119    7126    7130    7136
                7144    7195    7208    7211    7214    7219    7221    7224    7230    7238    7240    7243    7248    7250    7253
                7257    7268    7270    7276    7278    7281    7286    7288    7291    7295    7304    7314    7329    7344    7353
                7356    7359    7364    7366    7369    7376    7384    7386    7389    7394    7396    7399    7403    7414    7416
                7422    7424    7427    7432    7434    7437    7441    7450    7460    7475    7492    7498    7505    7509    7515
                7523    7561    7568    7572    7578    7589    7602    7605    7608    7613    7615    7618    7624    7632    7634
                7637    7642    7644    7647    7651    7662    7664    7670    7672    7675    7680    7682    7685    7689    7698
                7708    7723    7738    7747    7750    7753    7758    7760    7763    7770    7778    7780    7783    7788    7790
                7793    7797    7808    7810    7816    7818    7821    7826    7828    7831    7835    7844    7854    7869    7886
                7892    7899    7903    7909    7917    7967    7972    7976    7982    7992    8000    8002    8020    8023    8027
                8029    8032    8034    8038    8049    8056    8058    8064    8071    8115    8216    8242    8245    8301    8305
                8325    8329    8364    8379    8383    8394    8436    8439    8456    8459    8476    8479    8496    8499    8513
                8528    8544    8561    8578    8633    8635    8791    8812    9109    9115    9163    9182    9189    9196    9312
                9469    9687   11130   11148   11165   11194   11195   11198   11243   11250   11256   11258   11260   11262   11267
               11277   11283   11287   11315   11317   11319   11321   11351   11377   11390   11410   11411   11414   11422   11424
               11426   11450   11455   11485   11495   11511   11550
MOV             2469    2475    2479    2480    2486    2490    2491    2497    2498    2499    2505    2509    2511    2512    2513
                2514    2515    2516    2517    2518    2519    2523    2524    2527    2528    2529    2530    2535    2537    2538
                2539    2544    2546    2547    2549    2551    2552    2556    2611    2612    2616    2617    2618    2620    2623
                2626    2627    2630    2634    2637    2639    2640    2664    2665    2667    2668    2685    2708    2709    2711
                2712    2720    2723    2725    2726    2729    2730    2731    2736    2747    2787    2816    2817    2819    2822
                2824    2825    2828    2829    2830    2835    2901    2902    2905    2909    2911    2912    2917    2918    2919
                2959    2960    2961    2962    2969    2979    2981    2982    3003    3004    3009    3010    3011    3022    3031
                3036    3045    3050    3052    3061    3066    3077    3082    3084    3089    3104    3105    3113    3115    3116
                3117    3118    3125    3143    3144    3152    3157    3158    3173    3174    3184    3207    3208    3224    3229
                3230    3259    3260    3273    3278    3304    3305    3310    3311    3321    3322    3323    3332    3334    3344
                3345    3346    3351    3353    3360    3361    3363    3371    3383    3384    3385    3386    3394    3396    3403
                3404    3406    3416    3417    3418    3420    3429    3430    3434    3442    3444    3452    3453    3455    3464
                3467    3475    3477    3485    3486    3488    3497    3500    3508    3510    3518    3519    3532    3536    3553
                3554    3563    3564    3565    3608    3609    3612    3616    3622    3623    3624    3627    3631    3636    3640
                3642    3649    3650    3652    3660    3672    3673    3674    3675    3683    3685    3692    3693    3695    3705
                3710    3711    3712    3713    3718    3719    3729    3732    3735    3738    3739    3741    3764    3765    3782
```

3788	3789	3790	3793	3797	3802	3806	3808	3815	3816	3818	3826	3838	3839	3840
3841	3849	3851	3858	3859	3861	3871	3877	3878	3879	3880	3881	3885	3886	3887
3888	3893	3894	3907	3908	3909	3910	3920	3925	3926	3927	3928	3929	3933	3934
3935	3936	3941	3942	3970	3971	3972	3973	3990	4009	4010	4015	4022	4023	4024
4027	4031	4036	4040	4042	4049	4050	4052	4060	4072	4073	4074	4075	4083	4085
4092	4093	4095	4104	4106	4114	4120	4123	4124	4129	4131	4136	4138	4152	4161
4173	4174	4175	4176	4180	4181	4182	4183	4197	4198	4213	4214	4224	4229	4230
4233	4242	4244	4251	4252	4254	4269	4270	4271	4272	4280	4282	4289	4290	4292
4309	4310	4314	4316	4324	4326	4356	4357	4363	4364	4365	4366	4371	4373	4376
4382	4383	4384	4385	4393	4395	4402	4403	4405	4413	4415	4422	4423	4424	4427
4437	4438	4441	4444	4447	4448	4450	4458	4463	4468	4470	4477	4478	4480	4488
4500	4501	4503	4503	4511	4513	4520	4521	4523	4544	4545	4549	4571	4572	4578
4584	4587	4588	4589	4598	4599	4600	4611	4613	4620	4621	4623	4632	4661	4662
4668	4674	4675	4677	4680	4682	4684	4685	4696	4698	4705	4706	4708	4717	4735
4743	4746	4747	4748	4757	4758	4759	4770	4772	4779	4780	4782	4791	4803	4806
4809	4811	4812	4835	4836	4841	4842	4845	4846	4857	4858	4867	4869	4876	4877
4879	4890	4896	4897	4898	4907	4908	4909	4920	4921	4930	4932	4939	4940	4942
4950	4961	4964	4965	4980	4981	4985	4986	4987	4989	4990	4991	4994	4996	4997
5008	5010	5017	5018	5020	5030	5035	5037	5039	5040	5049	5050	5051	5066	5067
5069	5070	5098	5099	5103	5104	5105	5106	5108	5109	5112	5119	5120	5122	5135
5136	5146	5148	5155	5156	5158	5173	5174	5175	5176	5184	5186	5193	5194	5196
5204	5205	5215	5221	5223	5224	5225	5234	5235	5236	5263	5264	5268	5308	5309
5313	5314	5315	5317	5318	5319	5322	5324	5325	5336	5338	5345	5346	5348	5358
5363	5365	5367	5368	5377	5378	5379	5394	5395	5397	5398	5424	5425	5429	5431
5432	5433	5434	5437	5438	5447	5455	5457	5469	5470	5478	5480	5487	5488	5490
5498	5510	5511	5512	5513	5521	5523	5530	5531	5533	5543	5550	5551	5552	5561
5562	5563	5588	5590	5591	5595	5635	5636	5641	5642	5645	5651	5652	5654	5655
5658	5665	5666	5668	5680	5681	5690	5692	5699	5700	5702	5717	5718	5719	5720
5728	5730	5737	5738	5740	5772	5774	5775	5779	5836	5837	5845	5846	5847	5849
5850	5851	5881	5882	5893	5894	5900	5919	5920	5926	5927	5928	5929	5934	5936
5939	5945	5946	5947	5948	5956	5958	5965	5966	5968	5976	5978	5985	5986	5987
5990	6000	6001	6004	6007	6010	6011	6013	6021	6026	6031	6033	6040	6041	6043
6051	6063	6064	6065	6066	6074	6076	6083	6084	6086	6105	6107	6108	6112	6141
6142	6146	6167	6170	6188	6189	6190	6194	6195	6197	6200	6201	6215	6219	6223
6225	6230	6237	6239	6240	6242	6245	6246	6249	6256	6257	6259	6273	6276	6277
6286	6288	6295	6296	6298	6313	6314	6315	6316	6324	6326	6333	6334	6336	6344
6345	6354	6362	6364	6365	6366	6369	6370	6381	6383	6390	6391	6393	6428	6430
6431	6435	6487	6488	6492	6504	6505	6506	6515	6516	6517	6561	6562	6568	6569
6573	6578	6579	6581	6582	6585	6592	6593	6595	6607	6608	6617	6619	6626	6627
6629	6644	6645	6646	6647	6655	6657	6664	6665	6667	6675	6676	6684	6691	6694
6695	6696	6705	6706	6707	6740	6742	6743	6747	6806	6807	6809	6810	6814	6819
6820	6821	6823	6835	6827	6828	6831	6838	6839	6841	6848	6849	6858	6860	6867
6868	6870	6885	6886	6887	6888	6896	6898	6905	6906	6908	6923	6929	6932	6933
6934	6943	6944	6953	6963	6968	6969	6970	6972	6973	6976	6983	6984	6986	6994
6995	7004	7006	7013	7014	7016	7031	7032	7033	7034	7042	7044	7051	7052	7054
7069	7075	7078	7079	7080	7089	7090	7091	7122	7124	7125	7129	7180	7181	7188
7189	7193	7198	7199	7200	7202	7204	7206	7207	7210	7217	7218	7220	7227	7228
7237	7239	7246	7247	7249	7264	7265	7266	7267	7275	7277	7284	7285	7287	7302
7308	7311	7312	7313	7322	7323	7324	7342	7347	7348	7349	7351	7352	7355	7362
7363	7365	7373	7374	7383	7385	7392	7393	7395	7410	7411	7412	7413	7421	7423
7430	7431	7433	7448	7454	7457	7458	7459	7468	7469	7470	7501	7503	7504	7508
7557	7558	7564	7566	7567	7571	7582	7583	7587	7592	7593	7594	7596	7598	7600
7601	7604	7611	7612	7614	7621	7622	7631	7633	7640	7641	7643	7658	7659	7660
7661	7669	7671	7678	7679	7681	7696	7702	7705	7706	7707	7716	7717	7718	7736
7741	7742	7743	7745	7746	7749	7756	7757	7759	7767	7768	7777	7779	7786	7787
7789	7804	7805	7806	7807	7815	7817	7824	7825	7827	7842	7848	7851	7852	7853

```
        7862  7863  7864  7895  7897  7898  7902  7964  7965  7970  7971  7975  7990  7995  7996
        7997  7998  7999  8018  8019  8022  8025  8026  8028  8045  8046  8047  8048  8052  8053
        8054  8055  8069  8070  8092  8105  8109  8112  8131  8132  8154  8155  8158  8159  8199
        8202  8204  8212  8215  8218  8225  8226  8227  8252  8253  8254  8255  8256  8257  8258
        8259  8260  8261  8262  8263  8264  8265  8273  8274  8277  8280  8291  8292  8293  8302
        8306  8318  8319  8326  8330  8348  8360  8361  8362  8363  8366  8377  8378  8381  8393
        8395  8406  8407  8409  8409  8410  8416  8419  8426  8434  8435  8444  8446  8454  8455
        8464  8466  8474  8475  8484  8486  8494  8495  8504  8506  8512  8514  8527  8529  8542
        8543  8559  8560  8576  8577  8592  8593  8594  8596  8602  8606  8607  8612  8613  8617
        8623  8624  8631  8632  8637  8639  8643  8645  8649  8650  8654  8655  8662  8675  8676
        8677  8679  8680  8689  8690  8691  8697  8698  8699  8705  8706  8707  8713  8714  8722
        8723  8724  8727  8734  8741  8742  8743  8749  8750  8755  8757  8758  8766  8774  8775
        8788  8790  8792  8795  8798  8807  8808  8811  8813  8816  8819  8832  8833  8834  8847
        8848  8860  8863  8898  8903  8933  8938  8964  8965  8973  8981  8984  8998  9007  9008
        9009  9010  9038  9039  9041  9044  9057  9069  9070  9073  9074  9077  9078  9099  9104
        9125  9128  9156  9157  9162  9170  9185  9227  9228  9229  9230  9231  9232  9233  9238
        9241  9261  9267  9268  9269  9270  9271  9273  9274  9289  9290  9297  9301  9306  9307
        9309  9311  9321  9327  9328  9363  9371  9372  9373  9379  9386  9404  9405  9406  9407
        9408  9438  9462  9467  9496  9497  95??  952?  9537  9568  9569  9570  9571  9596  9597
        9598  9599  9600  9602  9603  9622  9623  962?  9625  9626  9647  9648  9649  9650  9651
        9652  9655  9660  9685  9686  9689  9706  9707  9708  9709  9710  9711  9712  9713  9764
        9765  9771  9774  9775  9795  9796  9797  9798  9799  9800  9801  9802  9803  9804  9811
        9812  9813  9814  9815  9816  9817  9818  9819  9820  9830  9831  9835  9841  9842 11020
       11025 11030 11033 11034 11042 11056 11074 11122 11123 11124 11127 11128 11131 11139 11141
       11142 11149 11155 11166 11172 11182 11184 11193 11238 11242 11259 11261 11276 11281 11282
       11284 11299 11300 11304 11307 11308 11314 11316 11318 11320 11322 11327 11349 11354 11355
       11356 11357 11358 11359 11360 11361 11362 11363 11364 11375 11388 11389 11393 11405 11407
       11412 11413 11415 11420 11423 11425 11432 11433 11434 11435 11436 11437 11438 11439 11440
       11469 11470 11471 11472 11484 11486 11538 11546 11548 11549
MOVB    2522  3128  4143  8156  8176  9072  9076  9106  9114  9167  9195  9203  9236  9239  9253
        9256  9265  9284  9285  9287  9364  9365  9368  9369  9370  9374  9377  9378  9397  9429
        9446  9500  9506  9530  9535  95??  9546  9561  9606  9654  9726  9833 11013 11038 11132
       11136 11334 11350 11376 11378 11396 11398 11404 11446 11447 11462 11463 11475 11483 11494
       11510 11518 11521
NEG     3884  3932  9235  9375
NOP     8116  8117  8118  8978  8979  8980  8990  8991
RESET   2496  8114  9012
ROL     3119  3120  3121  3122  3123  3124  3747  3748  8838  9381  9383  9384  9385  9387  9613
        9615  9617
ROR     9664  9665  9666  9667  9668  9669
RORB   11133 11134 11135 11400 11401 11402 11403
RTI     2500  2536  2619  8751  8770  8776  8861  8864  8915  8951  8971  8992  9079  9134  9172
        9275  9409  9468  9516  9572  9627  9805  9821  9843 11139
RTS     8136  8146  8172  8192  8203  8205  8217  8228  8243  8246  8266  8279  8281  8303  8308
        8327  8332  8342  8353  8367  8386  8388  8401  8417  8427  8445  8448  8465  8468  8485
        8488  8505  8508  8523  8535  8550  8552  8568  8570  8583  8585  8625  8656  8665  8725
        8745  8756  8759  8780  8782  8797  8799  8818  8820  8849  8880  8886  9212  9329  9662
        9690  9732  9776  9836 11076 11442 11451 11464 11487 11502 11528 11539 11551
RTT    11365
SBC     9716
SUB     3882  3930  6822  7201  7595  8638  9105  9242  9304  9715  9717 11160 11278
SWAB    4119  4121  4130  4132  4141  4583  4585  4676  4678  4742  4744  4802  4804  6196  6198
        6241  6243  6361  6363  8603  8684  8735
TRAP    9845  9854  9855  9856  9857  9859  9861  9862  9863  9864  9865  9866  9867
TST     2567  2585  2624  2628  2635  2669  2670  2671  2672  2673  2674  2675  2676  2677  2678
        2679  2680  2681  2682  2683  2715  2740  2742  2757  2761  2839  2841  2849  2870  2915
```

```
                2964    2970    2973    2975    2977    3107    3342    3373    3377    3424    3459    3492    3525    3577    3662
                3666    3767    3772    3828    3832    3950    3953    4062    4066    4219    4263    4301    4490    4494    4634
                4638    4719    4723    4793    4797    4952    4956    5167    5207    5248    5279    5285    5465    5500    5504
                5578    5612    5711    5798    5868    5874    5897    6053    6057    6131    6171    6175    6179    6235    6271
                6307    6402    6454    6564    6638    6766    6879    7025    7148    7183    7258    7404    7527    7652    7798
                7921    8142    8163    8167    8180    8200    8213    8338    8437    8440    8457    8460    8686    8694    8702
                8719    8731    8753    8777    8899    8934    8982    9040    9064    9119    9126    9169    9177    9199    9247
                9257    9299    9317    9319    9392    9460    9475    9533    9544    9567    9621    9628    9832   11040   11051
               11064   11171   11177   11236   11257   11268   11297   11305   11323   11373   11406   11441   11527   11536   11540
TSTB            2922    8858    9051    9152    9201    9249    9263    9291    9302    9315    9427    9443    9498    9504    9766
               11039   11059   11370   11391   11394   11456   11460   11492   11516   11522
.ASCII           461     462    9873    9878    9881    9885    9893    9901    9911    9919    9924    9929    9934    9940   10015
               10049   10057   10337   10429
.ASCIZ           460     463    8123    9576    9577    9578    9580    9946    9955    9962    9967    9973    9981    9986    9991
                9999   10003   10007   10013   10020   10025   10029   10031   10035   10039   10045   10066   10070   10083   10091
               10102   10110   10118   10121   10124   10134   10142   10150   10159   10166   10172   10176   10180   10185   10189
               10192   10195   10198   10201   10205   10211   10220   10225   10229   10232   10235   10239   10246   10251   10255
               10259   10265   10269   10276   10282   10288   10295   10302   10307   10314   10322   10328   10347   10354   10357
               10360   10363   10376   10387   10400   10406   10412   10418   10423   10436   10442   10450   10456   10461   10466
               10469   10472   10479   10483   10488   10494   10500   10505   10508   10512   10516   10520   10523   10528   10537
               10541   10545   10549   10556   10560   10567   10576   10579   10585   10587   10589   10596   10605   10612   10616
               10624   10628   10634   10641   10648   10653   10660   10668   10672   10676   10680   10685   10689   10694   10697
               10701   10705   10709   10714   10719   10725   10729   10733   10737   10746   10753   10757   10762   10770   10774
               10781   10788   10796   10804   10811   10818   10823   11077
.BLKB           9575    9673    9753
.BLKW            603     604     605     607     608    9280
.BYTE            424     425     430     431     439     440     448     449     450     451     479     480     490     491     498
                 499     501     502     504     505     617    2750    2751    2790    2791    2985    2986    8122    8351    8352
                8906    8907    8941    8942    8987    8988    9116    9117    9330    9331    9332    9410    9411    9412    9413
                9573    9574   10873   10875   10878   10880   10882   10885   10887   10889   10892   10894   10896   10899   10901
               10903   10906   10908   10911   10913   10915   10917   10920   10922   10924   10926   10929   10931   10933   10936
               10938   10940   10942   10944   10946   10948   10950   10952   10954   10957   10959   10963   10965   10967   10970
               10972   10975   10977   10979   10981   10983   10985   10987   10989   10991   10993   10995   10997   10999   11001
               11554   11555   11557   11558   11563   11564   11565   11567   11568   11569   11570   11571   11572   11573   11574
               11575   11576   11577   11578   11581   11582   11583   11584   11585   11586   11591
.DSABL          9485
.ENABL             1       5    9418   11085
.END           11619
.ENDC             23      38      40      41      42      60     152     166     367     384     388     390     395     397     404
                 418     422     424     452     458     459     460     461     465     468     490     498     501     504     507
                 508     509     510     511     512     513     514     515     516     517     518     519     520     521     522
                 523     524     525     526     527     528     529     533    1412    2466    2509    2510    2513    2515    2517
                2519    2520    2521    2523    2525    2546    2642    2654    2655    2662    2663    2664    2665    2687    2694
                2695    2706    2707    2708    2709    2717    2725    2751    2752    2791    2792    2801    2802    2814    2815
                2816    2817    2824    2846    2892    2893    2899    2900    2901    2902    2911    2950    2951    2957    2958
                2959    2960    2986    2987    2994    2995    3001    3002    3003    3004    3027    3050    3066    3082    3093
                3097    3098    3102    3103    3104    3105    3109    3137    3138    3141    3142    3143    3144    3150    3157
                3164    3167    3168    3171    3172    3173    3174    3189    3197    3200    3201    3205    3206    3207    3208
                3229    3244    3247    3248    3257    3258    3259    3260    3278    3291    3294    3295    3302    3303    3304
                3305    3413    3546    3597    3598    3606    3607    3608    3609    3621    3671    3702    3710    3752    3755
                3756    3762    3763    3764    3765    3787    3837    3868    3876    3917    3925    3980    3990    3996    3997
                4007    4008    4009    4010    4020    4071    4102    4119    4190    4204    4205    4211    4212    4213    4214
                4229    4299    4303    4306    4307    4308    4309    4310    4345    4346    4354    4355    4356    4357    4412
                4530    4533    4562    4563    4569    4570    4571    4572    4583    4630    4649    4650    4659    4660    4661
                4662    4673    4715    4740    4789    4823    4826    4827    4833    4834    4835    4836    4856    4895    4919
                4977    4978    4979    4980    4981    5027    5035    5084    5085    5096    5097    5098    5099    5203    5220
```

```
                5246  5258  5305  5306  5307  5308  5309  5355  5363  5411  5412  5422  5423  5424  5425
                5540  5548  5573  5584  5627  5628  5633  5634  5635  5636  5651  5747  5762  5818  5819
                5834  5835  5836  5837  5878  5880  5899  5907  5908  5917  5918  5919  5920  5975  6093
                6096  6158  6159  6165  6166  6167  6168  6211  6235  6343  6359  6400  6463  6527  6535
                6554  6555  6559  6560  6561  6562  6578  6674  6689  6717  6731  6789  6790  6804  6805
                6806  6807  6819  6915  6928  6955  6968  7061  7074  7101  7115  7177  7178  7179  7180
                7181  7198  7294  7307  7334  7347  7440  7453  7480  7494  7554  7555  7556  7557  7558
                7592  7688  7701  7728  7741  7834  7847  7874  7888  7951  7952  7962  7963  7964  7965
                7995  8062  8084  8085  8086  8088  8098  8104  8107  8108  8112  8114  8120  8122  8123
                8126  8352  8353  8907  8908  8942  8943  8988  8989  9020  9023  9028  9033  9035  9046
                9049  9050  9051  9053  9055  9062  9066  9071  9073  9077  9080  9081  9084  9087  9097
                9104  9109  9110  9111  9119  9130  9134  9135  9138  9167  9217  9284  9285  9288  9315
                9330  9341  9418  9419  9421  9449  9485  9489  9517  9518  9526  9528  9531  9559  9576
                9582  9585  9591  9635  9638  9677  9695  9757  9780  9825  9831  9834  9853  9854  9855
                9856  9857  9858  9859  9860  9861  9862  9863  9864  9865  9866  9867  9952 11003 11012
.EQUIV            60    61    69    84    85   114   115   116   117   118   119   120   121   122   123
                 142   143   144   145   146   147   148   149   150   151
.EVEN            468   620  9333 10832 11086 11560 11592
.IF               19    37    39    40    41    42    58   124   152   365   383   386   388   394   396
                 403   417   421   423   452   458   459   460   464   465   467   490   498   501   504
                 507   508   509   510   511   512   513   514   515   516   517   518   519   520   521
                 522   523   524   525   526   527   528   529   533  2466  2504  2509  2511  2513  2515
                2517  2519  2520  2521  2523  2541  2641  2653  2655  2662  2664  2665  2686  2693  2695
                2706  2708  2709  2716  2725  2750  2751  2790  2791  2800  2802  2814  2816  2817  2824
                2845  2891  2893  2899  2901  2902  2911  2949  2951  2957  2959  2960  2985  2986  2993
                2995  3001  3003  3004  3024  3047  3063  3079  3092  3096  3098  3102  3104  3105  3108
                3136  3138  3141  3143  3144  3149  3154  3163  3166  3168  3171  3173  3174  3186  3196
                3199  3201  3205  3207  3208  3226  3243  3246  3248  3257  3259  3260  3275  3290  3293
                3295  3302  3304  3305  3393  3545  3596  3598  3606  3608  3609  3618  3639  3682  3707
                3751  3754  3756  3762  3764  3765  3784  3805  3848  3873  3917  3922  3980  3989  3995
                3997  4007  4009  4010  4017  4039  4082  4116  4190  4203  4205  4211  4213  4214  4226
                4279  4302  4305  4307  4309  4310  4344  4346  4354  4356  4357  4392  4510  4532  4561
                4563  4569  4571  4572  4580  4610  4648  4650  4659  4661  4662  4670  4695  4737  4769
                4822  4825  4827  4833  4835  4836  4856  4892  4919  4976  4978  4980  4981  5007  5032
                5083  5085  5096  5098  5099  5183  5217  5246  5257  5304  5306  5308  5309  5335  5360
                5410  5412  5422  5424  5425  5520  5545  5573  5583  5626  5628  5633  5635  5636  5648
                5727  5761  5817  5819  5834  5836  5837  5877  5879  5898  5906  5908  5917  5919  5920
                5955  6073  6095  6157  6159  6165  6167  6168  6211  6232  6323  6356  6380  6462  6527
                6534  6553  6555  6559  6561  6562  6575  6654  6686  6717  6730  6788  6790  6804  6806
                6807  6816  6895  6925  6955  6965  7041  7071  7101  7114  7176  7178  7180  7181  7195
                7274  7304  7334  7344  7420  7450  7480  7493  7553  7555  7557  7558  7589  7668  7698
                7728  7738  7814  7844  7874  7887  7950  7952  7962  7964  7965  7992  8062  8083  8084
                8085  8086  8087  8088  8090  8103  8106  8108  8112  8114  8120  8122  8123  8351  8352
                8906  8907  8941  8942  8987  8988  9019  9022  9027  9033  9045  9047  9048  9049  9051
                9052  9053  9062  9064  9072  9074  9079  9080  9081  9083  9086  9097  9100  9107  9109
                9110  9112  9119  9123  9130  9134  9135  9137  9158  9216  9283  9285  9288  9315  9330
                9340  9417  9419  9420  9421  9449  9488  9489  9517  9525  9527  9531  9532  9575  9576
                9582  9584  9587  9603  9637  9676  9694  9756  9779  9824  9830  9834  9845  9854  9855
                9856  9857  9858  9859  9861  9862  9863  9864  9865  9866  9867  9872 11002 11011
.IFF              37    40    41    42    58   384   388   395   397   404   418   421   424   452   465
                 468  2509  2642  2654  2655  2663  2664  2665  2687  2694  2695  2707  2708  2709  2717
                2751  2791  2801  2802  2815  2816  2817  2846  2892  2893  2900  2901  2902  2950  2951
                2958  2959  2960  2986  2994  2995  3002  3003  3004  3093  3097  3098  3103  3104  3105
                3109  3137  3138  3142  3143  3144  3150  3164  3167  3168  3172  3173  3174  3197  3200
                3201  3206  3207  3208  3244  3247  3248  3258  3259  3260  3291  3294  3295  3303  3304
                3305  3546  3597  3598  3607  3608  3609  3752  3755  3756  3763  3764  3765  3990  3996
```

```
            3997  4008  4009  4010  4204  4205  4212  4213  4214  4303  4306  4307  4308  4309  4310
            4345  4346  4355  4356  4357  4533  4562  4563  4570  4571  4572  4649  4650  4660  4661
            4662  4823  4826  4827  4834  4835  4836  4977  4978  4979  4980  4981  5084  5085  5097
            5098  5099  5258  5305  5306  5307  5308  5309  5411  5412  5423  5424  5425  5584  5627
            5628  5634  5635  5636  5762  5818  5819  5835  5836  5837  5878  5880  5899  5907  5908
            5918  5919  5920  6096  6158  6159  6166  6167  6168  6463  6535  6554  6555  6560  6561
            6562  6731  6789  6790  6805  6806  6807  7115  7177  7178  7179  7180  7181  7494  7554
            7555  7556  7557  7558  7888  7951  7952  7963  7964  7965  8084  8087  8090  8104  8107
            8122  8352  8907  8942  8988  9020  9046  9049  9050  9053  9080  9081  9084  9086  9100
            9130  9135  9138  9217  9284  9341  9418  9421  9489  94[illegible]  9496  9517  9518  9527  9559
            9575  9585  9638  9677  9695  9757  9780  9825  9831  9[illegible]52  11003  11012
.IFT        9061  9110  9491  9496  9608  9628  9635
.IFTF       9059  9109  9436  9489  9492  9604  9612  9634
.IIF          18    23    28    34    35    36    38    41    42   362   464   468  2510  2513  2519
            2520  2521  2523  2524  2748  2788  2983  8085  8098  8099  8110  8122  8126  8349  8904
            8939  8985  9023  9024  9025  9026  9027  9028  9032  9060  9061  9077  9080  9081  9087
            9088  9089  9090  9091  9096  9122  9130  9135  9214  9418  9439  9567  9576  9582  9635
            9853  9854  9855  9856  9857  9859  9861  9862  9863  9864  9865  9866  9867
.IRP         533  2466  2653  2693  2800  2891  2949  2993  3096  3136  3166  3199  3246  3293  3596
            3754  3995  4203  4305  4344  4561  4643  4825  4976  5083  5304  5410  5626  5817  5906
            6157  6553  6788  7176  7553  7950  8090  9227  9267  9289  9290  9311  9327  9328  9598
            9624  9795  9815
.LIST          1     4    41   166   362   452   453   454   455   456   457   458   465   468   622
            2466  2525  2653  2664  2693  2708  2800  2816  2891  2901  2949  2959  2993  3003  3096
            3104  3136  3143  3166  3173  3199  3207  3246  3259  3293  3304  3596  3608  3754  3764
            3995  4009  4203  4213  4305  4309  4344  4356  4561  4571  4648  4661  4825  4835  4976
            4980  5083  5098  5304  5308  5410  5424  5626  5635  5817  5836  5906  5919  6157  6167
            6553  6561  6788  6806  7176  7180  7553  7557  7950  7964  8098  8114  9027  9130  9517
            9845  9853  9854  9855  9856  9857  9858  9859  9860  9861  9862  9863  9864  9865  9866
            9667  9868
.MACRO         1    42   415   627   647   675   691   699   709   720   742   772   797   822   842
             862   886   908   946   969  1005  1015  1066  1078  1104  1131  1163  1254  1281  1339
            2541  2653  2693  2800  2891  2949  2993  3096  3136  3166  3199  3246  3293  3596  3754
            3995  4203  4344  4561  4647  4825  5082  5409  5626  5817  5906  6157  6553  6788  7950
            9845
.MCALL         9    10    11    12   166   465  2525
.NLIST         1     3    41   166   362   452   453   454   455   456   457   458   465   468  1350
            2466  2525  2653  2664  2693  2708  2800  2816  2891  2901  2949  2959  2993  3003  3096
            3104  3136  3143  3166  3173  3199  3207  3246  3259  3293  3304  3596  3608  3754  3764
            3995  4009  4203  4213  4305  4309  4344  4356  4561  4571  4648  4661  4825  4835  4976
            4980  5083  5098  5304  5308  5410  5424  5626  5635  5817  5836  5906  5919  6157  6167
            6553  6561  6788  6806  7176  7180  7553  7557  7950  7964  8098  8114  9027  9130  9517
            9845  9853  9854  9855  9856  9857  9858  9859  9860  9861  9862  9863  9864  9865  9866
            9867  9868
.PAGE         56   355   415  1412  2466  2651  8081  8126  9017  9869  11002  11078
.REPT        362   452
.SBTTL        30    44    56   167   187   218   235   254   270   286   299   313   326   340   347
            356   365   381   392   415   465  1412  2467  2503  2651  2653  2693  2800  2891  2949
            2991  2993  3096  3136  3166  3199  3246  3293  3594  3596  3754  3995  4203  4305  4344
            4561  4648  4825  4976  5083  5304  5410  5626  5817  5906  6157  6553  6788  7176  7553
            7950  8081  8126  8889  8926  8962  8976  8994  9017  9081  9135  9214  9281  9338  9415
            9582  9635  9674  9692  9754  9777  9822  9845  9870  10081  10574  10830  10870  11003
.TITLE        18
.WORD        362   363   364   389   408   409   410   411   412   413   423   426   427   428   429
            432   433   434   435   436   437   438   441   442   443   452   453   454   455   456
            457   470   471   472   473   474   475   476   477   481   482   483   496   500   503
```

```
  506   507   508   509   510   511   512   513   514   515   516   517   518   519   520
  521   522   523   524   525   526   527   528  8103  8106  8121  9164  9211  9313  9414
 9631  9634  9691  9728  9773  9852 11024 11029 11058 11553 11561

ERRORS DETECTED:  0 HARD     2 SOFT
DEFAULT GLOBALS GENERATED:  0

*,DZR6HB/CRF/SOL=SYSMAC.C1,DZR6HB.P11
RUN-TIME: 72 110 20 SECONDS
RUN-TIME RATIO: 800/203=3.9
CORE USED:  44K  (87 PAGES)
```