

RP04/5/6

FUNCTIONAL CONTROLLER 2
CZRJJCO

AH-9225C-MC
COPYRIGHT © 74-78
FICHE 1 OF 2

DEC 1978
digital
MADE IN USA

RP04/5/6

FUNCTIONAL CONTROLLER 2
CZRJJCO

AH-9225C-MC

COPYRIGHT © 74-78

FICHE 2 OF 2

DEC 1978

digital

MADE IN USA

.REM @

IDENTIFICATION

PRODUCT CODE: AC-9223C-MC
PRODUCT NAME: CZRJJCO RP04/5/6 FUNCTIONAL CONTROLLER TEST PART II
DATE CREATED: MAY 1975
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: PETE BLACKSTONE

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1974,1978 DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION

DIGITAL PDP UNIBUS MASSBUSS

DEC DECUS DECTAPE

CONTENTS

1. ABSTRACT
2. REQUIREMENTS
 - 2.1 EQUIPMENT
 - 2.2 STORAGE
 - 2.3 PRELIMINARY PROGRAMS
3. LOADING PROCEDURE
 - 3.1 METHOD
4. STARTING PROCEDURE
 - 4.1 CONTROL SWITCH SETTINGS
 - 4.2 STARTING ADDRESS OR ADDRESSES
 - 4.3 PROGRAM AND/OR OPERATOR ACTION
5. OPERATING PROCEDURE
 - 5.1 OPERATIONAL SWITCH SETTINGS
 - 5.2 SUB-ROUTINE ABSTRACTS
6. ERRORS
 - 6.1 'FATAL' ERRORS
7. RESTRICTIONS
8. MISCELLANEOUS
 - 8.1 EXECUTION TIME
 - 8.2 STACK POINTER
 - 8.3 OPERATOR SELECTABLE SCOPE LOOPS
 - 8.4 PROGRAM REVISION HISTORY
9. PROGRAM DESCRIPTION

1.0 ABSTRACT

THIS DIAGNOSTIC TESTS THE DCL OF THE RP04/5/6 DISK SUBSYSTEM WHEN CONNECTED TO EITHER AN RH11 OR RH70 CONTROLLER.

IT USES THE DISK SURFACE AND THE DRIVE MECHANICS TO PROVE THE PROPER WORKING OF THE SUBSYSTEM. IT DOES NOT NEED A FORMATTED DISK PACK. A DISK PACK WITH NO VITAL INFORMATION WRITTEN ON IT IS ESSENTIAL. AFTER A SUCCESSFUL RUN (WITH NO ERRORS) OF THIS DIAGNOSTIC IT CAN BE ASSERTED THAT THE DCL IN THE RP04/5/6 SUBSYSTEM WORKS SUCCESSFULLY WHILE STANDING ALONE. SYSTEMS INTERACTION AND DRIVE TIMING IS LEFT TO OTHER DIAGNOSTICS. THIS IS WITH THE ASSUMPTION THAT STATIC 1 (DZRPS AND DZRPT) HAS BEEN RUN SUCCESSFULLY.

2.0 REQUIREMENTS

2.1 EQUIPMENT

PDP-11 COMPUTER WITH CONSOLE TELETYPE, AND A RP04/5/6 DISK SYSTEM. THE RP04/5/6 DISK SYSTEM WILL CONSIST OF AN RH11 CONTROLLER, A DISK CONTROL LOGIC (DCL), A DEC 733 DISK DRIVE, AND ITS APPROPRIATE DISK PACK. THE DISK PACK NEED NOT BE FORMATTED. USED SECTION OF THE DISK SURFACE SHALL BE GOOD (HOLE FREE). THE SURFACE FOR THE FOLLOWING SECTORS MUST BE GOOD, THAT IS, FREE OF ANY HOLES OR SURFACE IRREGULARITY BEFORE ANY DATA ERROR CAN BE ATTRIBUTED TO THE LOGIC.

CYLINDER 00, TRACK 00, SECTOR 00
CYLINDER 00, TRACK 00, SECTOR 01
CYLINDER 00, TRACK 18, SECTOR 21
CYLINDER 01, TRACK 00, SECTOR 00
CYLINDER 02, TRACK 00, SECTOR 00
CYLINDER 03, TRACK 00, SECTOR 00
CYLINDER 04, TRACK 00, SECTOR 00
CYLINDER 05, TRACK 00, SECTOR 00
CYLINDER 05, TRACK 07, SECTOR 04
CYLINDER 06, TRACK 00, SECTOR 00
CYLINDER 07, TRACK 00, SECTOR 00
CYLINDER 08, TRACK 00, SECTOR 00
CYLINDER 09, TRACK 18, SECTOR 21
CYLINDER 410, TRACK 18, SECTOR 21

2.2 STORAGE

THIS PROGRAM REQUIRES 16K WORDS OF MEMORY

2.3 PRELIMINARY PROGRAMS

THIS PROGRAM ASSUMES THAT MAINDEC-11-DZRJG-(LATEST REV) HAS BEEN RUN WITHOUT ERRORS.

IT ASSUMES THAT MAINDEC-11-DZRJH-(LATEST REV) HAS BEEN RUN WITHOUT ERRORS.

AND IT ASSUMES THAT MAINDEC-11-DZRJI-(LATEST REV) HAS BEEN RUN WITHOUT ERRORS.

3.0 LOADING PROCEDURE

USE STANDARD PROCEDURE FOR LOADING .ABS TAPES

4.0 STARTING PROCEDURE

SWITCH 12 MUST BE SET WHEN THIS PROGRAM IS TO BE RUN USING AN RH70 CONTROLLER. IT CAN BE SET AT THE FRONT PANEL, OR IN THE SOFTWARE SWITCH REGISTER IF THE OPERATOR SO DESIRES. SEE PARAGRAPH 5.1 FOR A DESCRIPTION OF SOFTWARE SWITCH REGISTER OPERATION.

4.1 CONTROL SWITCH SETTINGS

SEE SECTION 5.1

4.2 STARTING ADDRESS

START AT ADDRESS 200---FOR NORMAL RUN
START AT ADDRESS 210---FOR UNIT SELECTION

200 START

ALL SWITCHES MUST BE DOWN FOR WORST CASE RUN. WITH THIS STARTING ADDRESS ALL THE RP04/5/6S ON THE SYSTEM WILL BE TESTED ONE AT A TIME BEFORE 'END PASS' IS PRINTED OUT. TESTING WILL START WITH THE LOWEST UNIT NUMBER DRIVE THAT IS POWERED UP (THAT IS THE LOWEST UNIT NUMBER RHAS REGISTER THAT RESPONDS) THEN GO ON TO THE NEXT HIGHER UNIT NUMBER THAT IS POWERED UP.

204 RESTART

SAME AS 200 START, WITH THE FOLLOWING EXCEPTIONS: THE PROGRAM WILL INTERROGATE THE OPERATOR FOR THE NON-DEFAULT C.S.R. AND VECTOR ADDRESS FOR THE RHXX CONTROLLER. WHEN THESE QUESTIONS HAVE CORRECTLY BEEN ANSWERED, THE PROGRAM WILL AUTOMATICALLY RESTART FROM ADDRESS 200.

210 START

ALL SWITCHES MUST BE DOWN FOR WORST CASE RUN. WITH THIS STARTING ADDRESS THE CONSOLE TELETYPE WILL ASK FOR THE UNIT NUMBER TO BE TESTED. THEN ONLY THAT UNIT WILL BE TESTED FOR EACH PASS OF THE PROGRAM.

4.3 PROGRAM AND/OR OPERATOR ACTION

1. LOAD THE PROGRAM INTO MEMORY.
2. SET STARTING ADDRESS ON THE SWITCH REGISTER
3. PRESS 'LOAD ADDRESS'.

4. SET 'OPERATIONAL SWITCH SETTINGS' (SEE SECTION 5.1)
WORST CASE IS ALL SWITCHES DOWN.
5. PRESS "START".
6. FOR THE FIRST PASS EACH TEST WILL BE EXECUTED ONCE
ON THE DRIVES PRESENT OR DRIVE SELECTED BEFORE 'END
PASS' IS PRINTED. THE FIRST PASS WILL REQUIRE OPERATOR
INTERVENTION IF THE PROGRAM IS NOT RUN UNDER AN 'ACT-11'
MONITOR. THE SECOND AND SUBSEQUENT PASSES WILL EXECUTE
EACH TEST FOUR TIMES ON EACH DRIVES PRESENT OR DRIVE
SELECTED BEFORE 'END PASS' IS PRINTED. THE SECOND
AND SUBSEQUENT PASSED DO NOT NEED ANY OPERATOR INTERVENTION.

5.0 OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

IF THE PROGRAM IS BEING RUN ON A SWITCHLESS PROCESSOR (I. E.
AN 11/34) IT WILL DETERMINE THAT A HARDWARE SWITCH REGISTER
IS NOT PRESENT, AND WILL USE A 'SOFTWARE' SWITCH REGISTER.
THE SETTINGS OF THE 'SOFTWARE' SWITCHES ARE CONTROLLED
THROUGH A KEYBOARD ROUTINE WHICH IS CALED BY TYPING A
'CONTROL G'. THE PROGRAM WILL RECOGNIZE A 'CONTROL G' AT ANY
TIME EXCEPT WHEN IT IS AT A HIGHER PRIORITY PROCESSING AN
RP04/5/6 INTERRUPT. THE 'SOFTWARE' SWITCH VALUES ARE ENTERED AS
AN OCTAL NUMBER IN RESPONSE TO PROMPTING FROM THE SWITCH
ENTRY ROUTINE:

'SWR = NNNNNN NEW ='

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

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

SWITCH DEFINITIONS ARE GIVEN IN SECTION 9 'OPERATIONAL
SWITCH SETTINGS' HOWEVER THE DETAIL DESCRIPTION ARE GIVEN
HERE.

SWITCH 15 - HALT ON ERROR
WHEN THIS SWITCH IS SET, IF THE PROGRAM FINDS AN ERROR
THEN THE APPROPRIATE INFORMATION WILL BE PRINTED OUT
AND THEN THE PROGRAM WILL HALT. AFTER THIS HALT, PRESSING
'CONTINUE' WILL CONTINUE WITH THE PROGRAM TILL THE NEXT
ERROR IS FOUND WHEN THE SAME THING WILL HAPPEN.

SWITCH 14 - LOOP ON TEST
WHEN THIS SWITCH IS SET THE PROGRAM WILL BEGIN TO LOOP

ON THE CURRENT TEST BEING EXECUTED. FOR EXAMPLE IF THIS SWITCH IS SET WHEN THE PROGRAM IS IN TEST 10 THEN THE PROGRAM WILL KEEP EXECUTING ALL OF TEST 10 REPEATEDLY. ONE WAY TO BE SURE THAT THE PROGRAM IS IN THE EXPECTED TEST IS TO SET THIS SWITCH DURING AN ERROR PRINTOUT OR DURING A PROGRAM HALT.

SWITCH 13 - INHIBIT ERROR TYPEOUTS
WHEN THIS SWITCH IS SET FURTHER ERROR PRINTOUTS WILL CEASE, HOWEVER OPERATOR INSTRUCTIONS SUCH AS "STOP DRIVE X" WILL CONTINUE. AT THE END OF PASS "TOTAL NUMBER OF ERRORS ON THIS PASS ON DRIVE X" WILL BE TRUE, THAT IS, ALTHOUGH PRINTOUTS WERE INHIBITED IF THAT PASS FOUND 6 ERRORS, IT WILL SAY SO.

SWITCH 12 - RH70 CONTROLLER SELECT
THIS SWITCH MUST BE SET AT THE START OF THE PROGRAM WHEN THE DISK DRIVES TO TESTED ARE CONNECTED TO AN RH70 CONTROLLER. IT MUST NOT BE SET WHEN DISK DRIVES TO BE TESTED ARE CONNECTED TO AN RH11 CONTROLLER.

SWITCH 11 - INHIBIT ITERATIONS
WHEN THIS SWITCH IS SET THE PROGRAM ON SECOND PASS WILL NOT REPEAT EACH TEST FOUR TIMES BUT WILL DO EACH TEST ONCE ONLY.

SWITCH 10 - BELL ON ERROR
WHEN THIS SWITCH IS SET, IF THE PROGRAM FINDS AN ERROR THE "BELL" OR "ALARM" WILL BE SOUNDED. THIS SWITCH IS USEFUL WHEN SWITCH 11 IS SET YET INFORMATION IS NEEDED WHEN ANY ERROR IS DETECTED. TAKE THE EXAMPLE OF A PROGRAM LOOPING ON A TEST WITH SWITCH 11 SET TO HELP SCOPING. THEN IF THIS SWITCH IS SET AND THE BELL OR ALARM SOUNDS IT MEANS THAT THE ERROR IS PRESENT BUT IF THE BELL OR ALARM STOPS IT MEANS THAT THE ERROR IS NOT PRESENT.

SWITCH 9 - LOOP ON ERROR
WHEN THIS SWITCH IS SET, IF THE PROGRAM FINDS AN ERROR THEN GENERALLY THE PROGRAM WILL LOOP BACK TO THE LAST EXECUTED "SCOPE" STATEMENT. IF ON THE SECOND TIME THROUGH AN ERROR IS FOUND IT WILL AGAIN LOOP BACK TO THAT "SCOPE" STATEMENT. THIS LOOPING WILL CONTINUE AS LONG AS THE ERROR IS PRESENT AND THIS SWITCH IS SET. HOWEVER IF THE ERROR IS NOT PRESENT AT ANY TIME THEN IT WILL CONTINUE NORMALLY WITH THE PROGRAM. EACH TIME THE ERROR IS ENCOUNTERED PRINTOUT WILL TAKE PLACE UNLESS SWITCH 11 IS ALSO SET. DURING BEGUG, USING A SCOPE, IT IS RECOMMENDED THAT SWITCH 11 IS ALSO SET.

NOTE: SEE SECTION 8.3

SWITCH 8 - LOOP ON TEST IN SWR <7:0>
THIS IS A SPECIAL SWITCH. WHEN SET SWITCHES 0 THRU 7
HAVE ONE MEANING AND WHEN RESET SWITCHES 0 THRU 7 HAVE
ANOTHER MEANING. THIS MEANS THAT ANY SETTING OF SWITCH
0 THRU 7 MUST BE DONE WITH SWITCH 8 IN THE APPROPRIATE
POSITION. WHEN THIS SWITCH IS SET THEN SWITCHES 0 THRU
7 GIVE THE TEST NUMBER TO BE LOOPED ON. FOR EXAMPLE
WITH SWITCH 8 SET AND SWITCH 3 SET THE PROGRAM WILL LOOP
ON TEST 10. HOWEVER THIS SETTING MUST BE DONE AT THE
BEGINNING OF THE PROGRAM THEN ALL THE TESTS FROM 1 TO 10
WILL BE EXECUTED AND THEN TEST 10 WILL BE REPEATED OVER
AND OVER AGAIN. WHEN THIS SWITCH IS NOT SET THEN SWITCHES
0 THRU 7 HAVE THE MEANING ITS NAME INDICATES.
FOR EXAMPLE SWITCH 7 IS "STOP FURTHER COMPARES: THAT IS
IF SWITCH 8 IS NOT SET AND SWITCH 7 IS SET THEN WHEN A
DATA ERROR IS DETECTED NO FURTHER COMPARES WILL BE DONE.
FOR EXAMPLE IN A 256 WORD BUFFER IF ALL THE WORDS ARE IN
ERROR THEN AFTER SEEING THE PRINTOUT FOR THE FIRST FEW
WORDS SETTING SWITCH 7 ONLY WILL STOP FURTHER PRINTOUTS
OF THIS ERROR AND GO ON WITH THE TEST RATHER THAN PRINT
ALL THE 256 WORDS. HOWEVER IF THIS WAS DONE WITH SWITCH 11
THEN THE NEXT ERROR THAT THE PROGRAM DETECTS IN A SUB-
SEQUENT TEST WILL ALSO BE LOST. BUT WITH SWITCH 7, ONLY
THIS GROUP OF DATA ERRORS ARE NOT PRINTED OUT. ANOTHER
EXAMPLE OF SWITCH 8 BEING LOW IS WITH SWITCH 6, WHICH
IS "ECC TEST-COMPARE END RESULT ONLY". THAT IS IF SWITCH
8 IS NOT SET AND SWITCH 6 IS SET THEN ON ECC TESTS (TEST
120 THRU TEST 134) INSTEAD OF COMPARING CONTENTS OF THE
POSITION REGISTER AND PATTERN REGISTER AFTER EVERY CLOCK,
COMPARES WILL ONLY BE DONE AT THE END OF ALL THE CLOCKS.

NOTE: SEE SECTION 8.3

SWITCH 7 - STOP FURTHER COMPARES IF SW08 IS LOW.
IF SWITCH 8 IS SET AND THIS SWITCH IS ALSO SET THEN THIS
SWITCH GIVES THE TEST NUMBER TO BE LOOPED ON AS INDICATED
IN THE DESCRIPTION OF SWITCH 8. IF SWITCH 8 IS NOT SET
AND THIS SWITCH IS SET THEN THE PROGRAM WILL DO AS THE
NAME INDICATES. FOR EXAMPLE IN A 256 WORD BUFFER IF ALL
THE WORDS ARE IN ERROR THEN AFTER SEEING THE ERROR PRINTOUTS
FOR THE FIRST FEW WORDS THEN SETTING SWITCH 7 WITH SWITCH 8
NOT SET WILL STOP THE PRINTOUT OF ALL 256 WORDS BUT WILL
NOT STOP THE PRINTOUT OF ANOTHER ERROR IN ANY SUBSEQUENT
TEST. IT IS EXPECTED THAT SWITCH 7 AFTER BEING SET FOR
A WHILE TO STOP PRINTING ALL THE 256 WORDS WILL BE RESET
AGAIN TO ENABLE THE PRINTING OF OTHER DATA ERRORS.

SWITCH 6 - TYPE ALL REGISTERS WITH ERROR IF SW08 IS LOW
IF SWITCH 8 IS SET AND THIS SWITCH IS ALSO SET THEN
THIS SWITCH GIVES THE TEST NUMBER TO BE LOOPED ON AS
INDICATED IN THE DESCRIPTION OF SWITCH 8. IF SWITCH 8 IS

NOT SET AND THIS SWITCH IS SET THEN THE PROGRAM WILL DO AS THE NAME INDICATES. THAT IS ON FINDING AN ERROR INSTEAD OF ONLY GIVING THE ERROR MESSAGE AND RELEVANT REGISTERS AS WILL BE DONE IF SWITCH 11 IS NOT SET BUT WILL ALSO GIVE ALL THE REGISTER CONTENTS (EXCEPT 'DATA BUFFER' RHDB).

5.2 SUB-ROUTINE ABSTRACTS

SEE SECTION 9 'SUBROUTINES'.

6.0 ERRORS

ERROR PRINTOUTS CONTAIN THE ERROR ADDRESS AND OTHER PERTINENT INFORMATION CONCERNING THE PARTICULAR FAILURE. THIS INFORMATION MAY BE THE CONTENTS OF RELEVANT RP04/5/6 REGISTERS OR GOOD/RECEIVED DATA. IF THE ERROR OCCURRED IN A SUBROUTINE, THE ADDRESS OF THE SUBROUTINE CALL IS ALSO GIVEN. REFER TO THE PROGRAM LISTING AT THE STATED ADDRESS TO DETERMINE THE CAUSE OF THE ERROR.

6.1 'FATAL' ERRORS

IN THE EVENT THAT THE DISK DRIVE BECOMES UNAVAILABLE TO THE CONTROLLER, POWERS DOWN, OR CERTAIN CRITICAL STATUS BITS CANNOT BE CLEARED PRIOR TO THE START OF A TEST SEQUENCE - THIS INFORMATION WILL BE COMMUNICATED TO THE OPERATOR. IN ADDITION, THE TTY BELL WILL RING AND THE PROGRAM WILL HALT. IT IS SUGGESTED THAT IF THIS HAPPENS, THE OPERATOR LOAD ADDRESS 200 (210) AND RESTART THE PROGRAM AS A FIRST ATTEMPT TO SOLVE THE PROBLEM. IF THE FAILURE CONTINUES TO OCCUR, THERE ARE TWO OPTIONS FOR THE OPERATOR:

1. LOOK IN THE TEST LISTING FOR THE 'HALT' INSTRUCTION AND REPLACE IT, PLUS THE TWO WORDS ('TYPE ,CPHALT') ABOVE WITH 'NOP'S. WITH TTY ERROR PRINTOUTS INHIBITED, A SCOPE LOOP CAN BE INITIATED FOR THE TEST IN QUESTION.

2. GO BACK AND RERUN THE DZRPS DIAGNOSTIC AS IT IS QUITE POSSIBLE THAT A HARD FAILURE HAS OCCURRED IN ONE OF THE HARDWARE REGISTERS.

IT IS ALSO POSSIBLE TO CONTINUE FROM THE 'HALT' POINT, BUT THIS IS NOT RECOMMENDED AS ALL FOLLOWING TESTS WILL EXHIBIT THE SAME SYMPTOMS AND GIVE MISLEADING ERROR PRINTOUTS.

7.0 RESTRICTIONS

BEFORE STARTING THE PROGRAM THE OPERATOR MUST HAVE THE DRIVE PORT SWITCH LOCKED EITHER ON PORT A OR PORT B BUT MUST NEVER LEAVE IT IN THE PROGRAMMABLE STATE.

SWITCH 12 MUST BE SET WHEN RUNNING ON AN RH70 CONTROLLER AND IT MUST NOT BE SET WHEN RUNNING ON AN RH11 CONTROLLER. BECAUSE OF THE REQUIREMENT FOR IT TO BE SET WHEN USING AN RH70, THE PROGRAM CANNOT BE RUN IN CHAIN MODE WHEN USING THE

SOFTWARE REGISTER FEATURE WHILE RUNNING ON AN RH70. THIS IS BECAUSE THE ROUTINE WHICH GETS SOFTWARE SWITCH SETTINGS IS NOT OPERABLE WHEN IN CHAIN MODE.

8. MISCELLANEOUS

8.1 EXECUTION TIME

THE FIRST PASS OF THE PROGRAM WILL TAKE APPROXIMATELY 20 SECONDS. SUBSEQUENT PASSES WILL TAKE 60 SECONDS .

8.2 STACK POINTER

THE STACK IS INITIALLY SET TO 1000

8.3 OPERATOR SELECTABLE SCOPE LOOPS

HERE IS A DETAILED EXPLANATION OF HOW THE LOOP ON ERROR WORKS. FOR INSTRUCTIONS REGARDING THE USAGE OF THIS TECHNIQUE, HIT ^C ANY TIME WHILE THE PROGRAM IS RUNNING. ON HITTING AN ERROR IF THE LOOP ON ERROR SWITCH IS SET, THE PROGRAM GOES BACK - USUALLY BACK TO THE BEGINNING OF THE TEST.

WHEN THIS OPERATOR SELECTABLE SCOPE LOOP IS USED THEN THE POINT THE PROGRAM GOES BACK TO CAN BE CHANGED.

THE RESTRICTIONS TO THE POINT WHERE THE PROGRAM CAN GO ARE: -

1. IT MUST BE WITHIN THE TEST UNDER CONSIDERATION
 2. LOOP ON ERROR SWITCH MUST BE SET
 3. THE ERROR MUST OCCUR WITHIN THE TEST UNDER CONSIDERATION
- IF THE ERROR DOES NOT OCCUR WITHIN THE TEST UNDER CONSIDERATION THE PROGRAM WILL REVERT TO NORMAL OPERATION. HOWEVER, IF LOOP ON TEST SWITCH IS SET AND THIS OPERATOR SELECTABLE SCOPE LOOP IS USED THEN THE PROGRAM WILL LOOP BACK TO THE SELECTED POINT WHEN IT COMES TO THE END OF THE TEST UNDER CONSIDERATION.

AFTER LOOPING FOR SOME TIME IF THE LOOP SWITCH IS PUT DOWN THEN NORMAL OPERATION WILL CONTINUE.

8.4 PROGRAM REVISION HISTORY

9.0 PROGRAM DESCRIPTION

9.1 LOGIC DIVISION IN HARDWARE MODULES

REGISTER BOARD (RG) - ERROR REGISTER 1 STATUS REGISTERS
MUX FOR REGISTERS GO HANDLING REGISTER
DECODE COMMAND DECODE EXECUTION OF
MECH. COMMANDS

SYNC. DATA BOARD (SN) - DATA CONTROL PARALLEL TO SERIAL
SYNC. BYTE DETECT.

SEEK AND SEARCH (SS) - SEEK LOGIC SEARCH LOGIC HEADER

HANDLING.

ERROR CORRECTION (EC) - ECC LOGIC ERROR REGISTER 2 & 3
MUX FOR ERROR REG. 2 & 3 LOOK AHEAD
REG. SECTOR COUNTER DATA FORMATION
RING COUNTER.

DUAL PORT (DP) - DUAL PORT ARBITRATION ATTENTION LOGIC
SERIAL NO REGISTER MASS BUS REGISTER
STORAGE

9.2 DISK SURFACE USAGE

SYMBOLS USED

C = CYLINDER

T = TRACK

S = SECTOR

W = WRITE

R = READ

TT = TEST NUMBER

C0, T0, S0

TT22-W,R, TT23-R, TT24-W,R, TT25-W,R, TT26-W,R, TT35-W,R, TT37-W, TT50-W, TT51-W,R, TT52-W,R, TT55-W,R

C0, T0, S1

TT27-W,R, TT37-W,R, TT40-R, TT41-W,R, TT42-W,R, TT43-W,R

C0, T18, S21

TT30-W, TT31-W,R

C1, T0, S0

TT30-W,R, TT31-W,R, TT53-W,R, TT54-W,R

C1, T18, S21

TT31-W

C2, T0, S0

TT31-W,R

C2, T18, S21

TT31-W

C3, T0, S0

TT31-W,R

C3, T18, S21

TT31-W

C4, T0, S0

TT31-W,R

C4, T18, S21

TT31-W

C5, T0, S0

TT31-W,R

C5, T7, S4
TT33-W,R, TT34-W,R

C5, T18, S21
TT31-W

C6, T0, S0
TT31-W,R

C6, T18, S21
TT31-W

C7, T0, S0
TT31-W,R

C7, T18, S18
TT31-W

C8, T0, S0
TT31-W,R

C8, T18, S21
TT31-W

C9, T0, S0
TT31-W

C9, T18, S21
TT31-W, TT32-R

C10, T0, S0
TT31-W,R

C410, T18, S21
TT36-W,R, TT50-W,R

- 9.3 THE FOLLOWING SECTION DESCRIBES EACH TEST AND SUBROUTINES
IN DETAIL AND CAN BE USED AS AN INDEX TO THE LISTING.
THE LEFT MOST COLUMN IS THE LINE NUMBER WITHIN THE LISTING
WHERE THAT ITEM WILL BE FOUND.
@

603
604
605

;DRIVE MUST BE LOCKED ON PORT A OR PORT B

611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627

;*INTERNAL PROGRAM MACROS BEGIN HERE
;*****

;*
;*NOTE: ALL MACRO CALLS BEGINNING WITH '\$' ARE SUPPLIED FROM AN
;* EXTERNAL SYSMAC.SML PACKAGE WHICH MUST BE MADE AVAILABLE
;* TO THE SOURCE PROGRAM AT ASSEMBLY TIME.
;*

628

CZF
CZF


```
629 000174 000000      DISPREG: .WORD 0      ;;SOFTWARE DISPLAY REGISTER
630 000176 000000      SWREG:   .WORD 0      ;;SOFTWARE SWITCH REGISTER
631
632 000046 032662      $ENDAD      ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
633 000052 020000      .WORD 20000      ;;2)SET LOC.52 TO 20000
634
635      .SBTTL  STARTING ADDRESS
636
637      .=200
638 000200 000137 004710      RA:      JMP @#BEGIN      ;NORMAL START
639 000204 000137 035236      ADDMOD:  JMP @#BASECH    ;GET DEVICE PARAMETERS
640 000210 000137 004674      JMP @#BEGIN2      ;JUMP TO SELECT DRIVE START
641      .=220
642 000220 000137 004660      JMP @#BEGIN1      ;JUMP TO NO OPERATOR TESTS START
643
644      ;*STARTING ADDRESS 200 FOR NORMAL STARTS
645      ;*THIS WILL TEST ALL RP04'S ON THE SYSTEM A SINGLE DRIVE AT A TIME
646      ;*
647      ;*STARTING ADDRESS 210 WILL TEST ONLY ONE SPECIFIED DRIVE
648      ;*
649      ;*STARTING ADDRESS 220 WILL JUMP OVER THE TESTS REQUIRING AN OPERATOR
650      ;*AT THE DRIVE
651
```


652
653

001110

.=1110

; ?

654	001100	000000	\$PASS:	.WORD	0	::CONTAINS	PASS COUNT
655	001102	000	\$TSTNM:	.BYTE	0	::CONTAINS	THE TEST NUMBER
656	001103	000	\$ERFLG:	.BYTE	0	::CONTAINS	ERROR FLAG
657	001104	000000	\$ICNT:	.WORD	0	::CONTAINS	SUBTEST ITERATION COUNT
658	001106	000000	\$LPADR:	.WORD	0	::CONTAINS	SCOPE LOOP ADDRESS
659	001110	000000	\$LPERR:	.WORD	0	::CONTAINS	SCOPE RETURN FOR ERRORS
660	001112	000000	\$ERTTL:	.WORD	0	::CONTAINS	TOTAL ERRORS DETECTED
661	001114	000	\$ITEMB:	.BYTE	0	::CONTAINS	ITEM CONTROL BYTE
662	001115	001	\$ERMAX:	.BYTE	1	::CONTAINS	MAX. ERRORS PER TEST
663	001116	000000	\$ERRPC:	.WORD	0	::CONTAINS	PC OF LAST ERROR INSTRUCTION
664	001120	000000	\$GDADR:	.WORD	0	::CONTAINS	ADDRESS OF 'GOOD' DATA
665	001122	000000	\$BDADR:	.WORD	0	::CONTAINS	ADDRESS OF 'BAD' DATA
666	001124	000000	\$GDDAT:	.WORD	0	::CONTAINS	'GOOD' DATA
667	001126	000000	\$BDDAT:	.WORD	0	::CONTAINS	'BAD' DATA
668	001130	000000		.WORD	0	::RESERVED--NOT TO BE USED	
669	001132	000000		.WORD	0		
670	001134	000	\$AUTOB:	.BYTE	0	::AUTOMATIC	MODE INDICATOR
671	001135	000	\$INTAG:	.BYTE	0	::INTERRUPT	MODE INDICATOR
672	001136	000000		.WORD	0		
673	001140	177570	SWR:	.WORD	DSWR	::ADDRESS	OF SWITCH REGISTER
674	001142	177570	DISPLAY:	.WORD	DDISP	::ADDRESS	OF DISPLAY REGISTER
675	001144	177560	\$TKS:	177560		::TTY KBD	STATUS
676	001146	177562	\$TKB:	177562		::TTY KBD	BUFFER
677	001150	177564	\$TPS:	177564		::TTY PRINTER	STATUS REG. ADDRESS
678	001152	177566	\$TPB:	177566		::TTY PRINTER	BUFFER REG. ADDRESS
679	001154	000	\$NULL:	.BYTE	0	::CONTAINS	NULL CHARACTER FOR FILLS
680	001155	002	\$FILLS:	.BYTE	2	::CONTAINS	# OF FILLER CHARACTERS REQUIRED
681	001156	012	\$FILLC:	.BYTE	12	::INSERT	FILL CHARS. AFTER A 'LINE FEED'
682	001157	000	\$TPFLG:	.BYTE	0	::'TERMINAL	AVAILABLE' FLAG (BIT<07>=0=YES)
683	001160	000000	\$REGAD:	.WORD	0	::CONTAINS	THE ADDRESS FROM
684	001162	000000	\$REG0:	.WORD	0	::CONTAINS	((REGAD)+0)
685	001164	000000	\$REG1:	.WORD	0	::CONTAINS	((REGAD)+2)
686	001166	000000	\$REG2:	.WORD	0	::CONTAINS	((REGAD)+4)
687	001170	000000	\$REG3:	.WORD	0	::CONTAINS	((REGAD)+6)
688	001172	000000	\$REG4:	.WORD	0	::CONTAINS	((REGAD)+10)
689	001174	000000	\$REG5:	.WORD	0	::CONTAINS	((REGAD)+12)
690	001176	000000	\$TMP0:	.WORD	0	::USER	DEFINED
691	001200	000000	\$TMP1:	.WORD	0	::USER	DEFINED
692	001202	000000	\$TMP2:	.WORD	0	::USER	DEFINED
693	001204	000000	\$TMP3:	.WORD	0	::USER	DEFINED
694	001206	000000	\$TMP4:	.WORD	0	::USER	DEFINED
695	001210	000000	\$TMP5:	.WORD	0	::USER	DEFINED
696	001212	000000	\$TIMES:	0		::MAX.	NUMBER OF ITERATIONS
697	001214	000000	\$ESCAPE:	0		::ESCAPE	ON ERROR ADDRESS
698	001216	177607	\$BELL:	.ASCIZ	<207><377><377>	::CODE	FOR BELL
699	001222	077	\$QUES:	.ASCII	/?/	::QUESTION	MARK
700	001223	015	\$CRLF:	.ASCII	<15>	::CARRIAGE	RETURN
701	001224	000012	\$LF:	.ASCIZ	<12>	::LINE	FEED

000377

702				
703				
704				
705			:ITEM1	
706	001226	042572	EM1	:RP04 DID NOT INTERRUPT
707				:WAITED ON BIT DID NOT OCCUR
708	001230	057450	DH1	:PC
709				:WAT PC
710				:BIT WAITED
711				:REG ADDRESS
712				:REG CONTENTS
713				:RHCS1 CONTENTS
714	001232	061720	DT1	:\$ERRPC, WAITPC, WAITBT, WAITRE, \$BDDAT, CS1
715	001234	062240	DF1	:0,0,0,0,0,0
716				
717			:ITEM2	
718	001236	042621	EM2	: INTERRUPT ENABLE BIT DOWN BUT
719				:WAITED ON BIT DID NOT OCCUR
720	001240	057450	DH1	:PC
721				:WAT PC
722				:BIT WAITED
723				:REG ADDRESS
724				:REG CONTENTS
725				:RHCS1 CONTENTS
726	001242	061720	DT1	:\$ERRPC, WAITPC, WAITBT, WAITRE, \$BDDAT, CS1
727	001244	062240	DF1	:0,0,0,0,0,0
728				
729			:ITEM3	
730	001246	042710	EM3	:RP04 DID NOT INTERRUPT WHEN
731				:WAITED ON BIT DID SET
732	001250	057450	DH1	:PC
733				:WAT PC
734				:BIT WAITED
735				:REG ADDRESS
736				:RHCS1 CONTENTS
737	001252	061720	DT1	:\$ERRPC, WAITPC, WAITBT, WAITRE, \$BDDAT, CS1
738	001254	062240	DF1	:0,0,0,0,0,0
739				
740			:ITEM4	
741	001256	042771	EM4	:WAITED ON BIT DID SET BUT
742				:TIME IS IN ERROR
743				:TIME IS GIVEN IN 10 MICRO SEC.
744				:(DECIMAL)
745	001260	057630	DH4	:PC
746				:WAT PC
747				:BIT WAITED
748				:REG ADDRESS
749				:TIME IN 10 MSEC
750	001262	061740	DT4	:\$ERRPC, WAITPC, WAITBT, WAITRE, \$BDDAT, WAITIM
751	001264	062247	DF4	:0,0,0,0,0,1
752				
753			:ITEM5	
754	001266	043102	EM5	:RHAS DOES NOT CLEAR BY
755				:MOVING IN ALL ONES
756	001270	057771	DH5	:PC
757				:REG. ADDR.

758				:GOOD DATA
759				:RECEIVED DATA
760	001272	061762	DT5	:\$ERRPC,REGADR,\$GDDAT,\$BDDAT
761	001274	062256	DF5	:0,0,0,0
762				
763			:ITEM6	
764	001276	043154	EM6	:LOADING RHER1 FOR ALL
765				:UNITS DID NOT SET ANY BITS
766				:IN RHAS-NO UNITS PRESENT
767	001300	060110	DH6	:PC
768				:REG ADDR
769				:RECEIVED DATA
770	001302	061776	DT6	:\$ERRPC,REGADR,\$BDDAT
771	001304	062263	DF6	:0,0,0
772				
773			:ITEM7	
774	001306	043242	EM7	:SPECIFIED REGISTER NONEXISTANT
775				:SO ABORT PROGRAM
776	001310	060207	DH7	:PC
777				:ADDR. OF REG.
778	001312	062010	DT7	:\$ERRPC,TEMP1
779	001314	062267	DF7	:0,0
780				
781			:ITEM10	
782	001316	043312	EM10	:STOPED DRIVE HAS MOL BIT
783				:IN RHDS1 = 1
784	001320	060247	DH10	:PC
785				:TEST NO
786				:FAILING REG ADDR
787				:CONTENTS OF RHCS1
788				:CONTENTS OF RHCS2
789				:CONTENTS OF RHDS1
790				:CONTENTS OF RHER1
791	001322	062020	DT10	:\$ERRPC,\$STSTM,\$BDADR,CS1,CS2,DS1,ER1
792	001324	062272	DF10	:0,0,0,0,0,0,0
793				
794			:ITEM11	
795	001326	043361	EM11	:WITH SPINDLE POWERED DOWN
796				:RHCS2 SHOULD HAVE ONLY
797				:UNIT NUMBER AND IR HIGH
798	001330	060247	DH10	:PC
799				:TEST NO
800				:FAILING REG. ADR
801				:CONTENTS OF RHCS1
802				:CONTENTS OF RHCS2
803				:CONTENTS OF RHDS1
804				:CONTENTS OF RHER1
805	001332	062020	DT10	:\$ERRPC,\$STSTM,\$BDADR,CS1,CS2,DS1,ER1
806	001334	062272	DF10	:0,0,0,0,0,0,0
807				
808			:ITEM12	
809	001336	043466	EM12	:AFTER A POWER UP WITH
810				:NO PACK ACKNOWLEDGE COMMAND
811				:RHDS1 SHOULD HAVE MOL=1, VV=0
812	001340	060247	DH10	:PC
813				:TEST NO

814					: FAILING REGISTER ADDR.
815					: CONTENTS OF RHCS1
816					: CONTENTS OF RHCS2
817					: CONTENTS OF RHDS1
818					: CONTENTS OF RHER1
819	001342	062020		DT10	: \$ERRPC, \$TSTNM, \$BDADR, CS1, CS2, DS1, ER1
820	001344	062272		DF10	: 0,0,0,0,0,0,0
821					
822					: ITEM13
823	001346	043574		EM13	: AFTER A POWER UP WITHOUT
824					: ANY INIT RHCS1 SHOULD
825					: HAVE GO=0, DVA=1, RDY=1
826					: IE=0, DISREGARD
827					: ALL OTHER BITS
828	001350	060247		DH10	: PC
829					: TEST NO
830					: FAILING REGISTER ADDR.
831					: CONTENTS OF RHCS1
832					: CONTENTS OF RHCS2
833					: CONTENTS OF RHDS1
834					: CONTENTS OF RHER1
835	001352	062020		DT10	: \$ERRPC, \$TSTNM, \$BDADR, CS1, CS2, DS1, ER1
836	001354	062272		DF10	: 0,0,0,0,0,0,0
837					
838					: ITEM14
839	001356	043713		EM14	: AFTER POWER UP RHCC
840					: SHOULD BE=0
841	001360	057771		DH5	: PC
842					: REG. ADDR.
843					: GOOD DATA
844					: RECEIVED DATA
845	001362	061762		DT5	: \$ERRPC, REGADR, \$GDDAT, \$BDDAT
846	001364	062256		DF5	: 0,0,0,0
847					

848			:ITEM15		
849	001366	043765		EM15	:PACK ACKNOWLEDGE CAUSED
850					:AN ERROR
851					:GOOD DATA IS BEFORE COMMAND
852					:RECEIVED DATA IS AFTER COMMAND
853	001370	057771		DH5	:PC
854					:REG. ADDR.
855					:GOOD DATA
856					:RECEIVED DATA
857	001372	061762		DT5	:\$ERRPC,REGADR,\$GDDAT,\$BDDAT
858	001374	062256		DF5	:0,0,0,0
859					
860			:ITEM16		
861	001376	044126		EM16	:GIVING A NO-OP COMMAND CAUSED
862					:AN ERROR
863					:GOOD DATA GIVES REGISTER
864					:CONTENTS BEFORE COMMAND
865					:RECEIVED DATA GIVES REGISTER
866					:CONTENTS AFTER COMMAND
867	001400	057771		DH5	:PC
868					:REG. ADDR.
869					:GOOD DATA
870					:RECEIVED DATA
871	001402	061762		DT5	:\$ERRPC,REGADR,\$GDDAT,\$BDDAT
872	001404	062256		DF5	:0,0,0,0
873					
874			:ITEM17		
875	001406	044254		EM17	:DRIVE CLEAR COMMAND
876					:CAUSED AN ERROR
877					:GOOD DATA GIVES WHAT SHOULD
878					:BE THERE
879					:RECEIVED DATA GIVES WHAT WAS
880					:THERE AFTER COMMAND
881	001410	057771		DH5	:PC
882					:REG. ADDR.
883					:GOOD DATA
884					:RECEIVED DATA
885	001412	061762		DT5	:\$ERRPC,REGADR,\$GDDAT,\$BDDAT
886	001414	062256		DF5	:0,0,0,0
887					
888			:ITEM20		
889	001416	044411		EM20	:READ-IN COMMAND GAVE AN ERROR
890					:GOOD DATA HAS WHAT SHOULD BE THERE
891					:RECEIVED DATA HAS WHAT WAS
892					:AFTER COMMAND
893	001420	057771		DH5	:PC
894					:REG. ADDR.
895					:GOOD DATA
896					:RECEIVED DATA
897	001422	061762		DT5	:\$ERRPC,REGADR,\$GDDAT,\$BDDAT
898	001424	062256		DF5	:0,0,0,0
899					
900					
901			:ITEM 21		
902	001426	044560		EM21	:RHCS1 CONTENTS DURING
903					:COMMAND WAS IN ERROR

904	001430	057771		DH5	
905	001432	061762		DT5	
906	001434	062256		DF5	
907					
908			:ITEM 22		
909	001436	044633		EM22	:RHDS1 CONTENTS DURING
910					:COMM ANS WAS IN ERROR
911	001440	057771		DH5	
912	001442	061762		DT5	
913	001444	062256		DF5	
914					
915			:ITEM 23		
916	001446	044706		EM23	:UNLOAD COMMAND GAVE AN ERROR
917					:GOOD DATA GIVES WHAT SHOULD
918					:BE THERE
919					:RECEIVED DATA GIVES WHAT WAS
920					:THERE AFTER COMMAND
921	001450	057771		DH5	
922	001452	061762		DT5	
923	001454	062256		DF5	
924					
925			:ITEM 24		
926	001456	045055		EM24	:OFFSET COMMAND CAUSED AN ERROR
927					:GOOD DATA IS WHAT SHOULD BE THERE
928					:RECEIVED DATA GIVES WHAT WAS THERE
929					:AFTER AN OFFSET COMMAND
930	001460	057771		DH5	
931	001462	061762		DT5	
932	001464	062256		DF5	
933					
934			:ITEM 25		
935	001466	045220		EM25	:RETURN TO CENTER LINE COMMAND
936					:CAUSED AN ERROR
937					:GOOD DATA GIVES WHAT SHOULD BE
938					:THERE
939					:RECEIVED DATA GIVES WHAT WAS
940					:THERE AFTER COMMAND
941	001470	057771		DH5	
942	001472	061762		DT5	
943	001474	062256		DF5	
944					
945			:ITEM 26		
946	001476	045402		EM26	:500 OFFSETS CAUSED AN ERROR
947	001500	060426		DH26	:PC
948					:CONT. OF RHCS1
949					:CONT. OF RHCS2
950					:CONT. OF RHDS1
951					:CONT. OF RHER1
952					:CONT. OF RHER2
953					:CONT. OF RHER3
954	001502	062040		DT26	:\$ERRPC,CS1,CS2,DS1,ER1,ER2,ER3
955	001504	062301		DF26	:0,0,0,0,0,0,0
956					
957			:ITEM 27		
958	001506	045472		EM27	:WRITE HEADER AND DATA
959					:CAUSED IMPROPER REGISTER CHANGE

960				:GOOD DATA GIVES WHAT
961				:SHOULD BE THERE
962				:RECEIVED DATA GIVES WHAT
963				:WAS THERE AFTER COMMAND
964	001510	057771	DH5	
965	001512	061762	DT5	
966	001514	062256	DF5	
967				
968				:ITEM 30
969	001516	045710	EM30	:WRITE HEADER AND DATA
970				:CHANGED WRITE FROM BUFFER
971	001520	060626	DH30	:PC
972				:WORD NO
973				:GOOD DATA
974				:BAD DATA
975	001522	062062	DT30	:\$ERRPC,ERWORD,\$GDDAT,\$BDDAT
976	001524	062311	DF30	:0,0,0,0
977				
978				:ITEM 31
979	001526	045770	EM31	:READ HEADER AND DATA CAUSED
980				:IMPROPER REGISTER CHANGE
981				:GOOD DATA HAS WHAT SHOULD
982				:BE THERE
983				:RECEIVED DATA GIVES WHAT
984				:WAS THERE AFTER COMMAND
985	001530	057771	DH5	
986	001532	061762	DT5	
987	001534	062256	DF5	
988				
989				:ITEM 32
990	001536	046205	EM32	:WRITE HEADER AND DATA FOLLOWED
991				:BY A READ HEADER AND DATA
992				:CAUSED A READ/WRITE ERROR
993	001540	060626	DH30	
994	001542	062062	DT30	
995	001544	062311	DF30	
996				
997				:ITEM 33
998	001546	046312	EM33	:READ DATA CAUSED IMPROPER REGISTER
999				:CHANGE
1000				:GOOD DATA GIVES WHAT SHOULD BE THERE
1001				:RECEIVED DATA GIVES WHAT WAS THERE AFTER
1002				:COMMAND
1003	001550	057771	DH5	
1004	001552	061762	DT5	
1005	001554	062256	DF5	
1006				
1007				:ITEM 34
1008	001556	046514	EM34	:READ DATA INCORRECT
1009	001560	060626	DH30	
1010	001562	062062	DT30	
1011	001564	062311	DF30	
1012				
1013				:ITEM 35
1014	001566	046540	EM35	:WRITE DATA COMMAND CAUSED
1015				:IMPROPER REGISTER CHANGE

1016					:GOOD DATA GIVES WHAT SHOULD BE THERE
1017					:RECEIVED DATA GIVES REGISTER
1018					:CONTENTS AFTER WRITE DATA
1019	001570	057771		DH5	
1020	001572	061762		DT5	
1021	001574	062256		DF5	
1022					
1023					:ITEM 36
1024	001576	046756		EM36	:WRITE DATA COMMAND CHANGED
1025					:WRITE FROM BUFFER
1026	001600	060626		DH30	
1027	001602	062062		DT30	
1028	001604	062311		DF30	
1029					
1030					:ITEM 37
1031	001606	047033		EM37	:SEEK COMMAND CAUSED AN
1032					:ERROR
1033					:GOOD DATA GIVES WHAT SHOULD
1034					:BE THERE
1035					:RECEIVED DATA GIVES WHAT
1036					:WAS THERE AFTER SEEK COMMAND
1037	001610	057771		DH5	:
1038	001612	061762		DT5	:
1039	001614	062256		DF5	:
1040					:
1041					:ITEM 40
1042	001616	047250		EM40	:WRITE CHECK CAUSED AN
1043					:IMPROPER REGISTER CHANGE
1044					:GOOD DATA GIVES WHAT SHOULD
1045					:BE THERE
1046					:RECEIVED DATA GIVES WHAT WAS
1047					:THERE AFTER COMMAND
1048	001620	057771		DH5	
1049	001622	061762		DT5	
1050	001624	062256		DF5	
1051					
1052					:ITEM 41
1053	001626	047457		EM41	:LOCKING OUT WRITES BY WRITE
1054					:LOCK BUTTON CAUSED IMPROPER
1055					:REGISTER CHANGE
1056					:GOOD DATA GIVES WHAT SHOULD
1057					:BE THERE
1058					:RECEIVED DATA GIVES WHAT
1059					:WAS THERE AFTER WRITES
1060					:WERE LOCKED OUT BY
1061					:BUTTON
1062	001630	057771		DH5	
1063	001632	061762		DT5	
1064	001634	062256		DF5	
1065					
1066					:ITEM 42
1067	001636	047740		EM42	:ATTEMPTING TO WRITE WITH WRITE
1068					:LOCKED OUT CAUSED IMPROPER
1069					:REGISTER CHANGE
1070					:GOOD DATA GIVES WHAT SHOULD
1071					:BE THERE

1072					:RECEIVED DATA GIVES WHAT WAS
1073					:THERE AFTER ATTEMPT
1074	001640	057771		DH5	
1075	001642	061762		DT5	
1076	001644	062256		DF5	
1077					
1078					
1079	001646	050216	:ITEM 43	EM43	:WRITING WITH WRITE LOCKED
1080					:OUT CHANGED DISK DATA
1081					:GOOD DATA GIVES WHAT WAS
1082					:ON DISK BEFORE WRITE WITH
1083					:WRITE LOCK WAS ATTEMPTED
1084					:RECEIVED DATA GIVES WHAT WAS
1085					:READ BACK AFTER WRITE WITH
1086					:WRITE LOCK WAS ATTEMPTED
1087	001650	060626		DH30	
1088	001652	062062		DT30	
1089	001654	062311		DF30	
1090					
1091			:ITEM 44		
1092	001656	050554		EM44	:ENABLING WRITES BY WRITE LOCK
1093					:BUTTON CAUSED AN ERROR
1094					:GOOD DATA GIVES WHAT SHOULD
1095					:BE THERE
1096					:RECEIVED DATA GIVES WHAT WAS
1097					:THERE AFTER WRITE LOCK
1098					:BUTTON ENABLED WRITES
1099	001660	057771		DH5	:
1100	001662	061762		DT5	:
1101	001664	062256		DF5	:
1102					:
1103			:ITEM 45		
1104	001666	051046		EM45	:TRANSFERRING ON LAST BLOCK IE. CYLINDER
1105					:410, SECTOR 21, TRACK 18
1106					:CAUSED IMPROPER REGISTER
1107					:CHANGE
1108					:GOOD DATA GIVES WHAT SHOULD
1109					:BE THERE
1110					:RECEIVED DATA GIVES WHAT WAS
1111					:THERE AFTER TRANSFER
1112	001670	057771		DH5	
1113	001672	061762		DT5	
1114	001674	062256		DF5	
1115					
1116			:ITEM 46		
1117	001676	051354		EM46	:DATA READ FROM LAST
1118					:BLOCK IE. CYLINDER 410
1119					:SECTOR 21, TRACK 18 IS IN
1120					:ERROR
1121	001700	060626		DH30	
1122	001702	062062		DT30	
1123	001704	062311		DF30	
1124					
1125			:ITEM 47		
1126	001706	051500		EM47	:TRANSFERRING FROM NONEXISTANT
1127					:SECTOR CAUSED IMPROPER

1128					:REGISTER CHANGE
1129					:GOOD DATA GIVES WHAT SHOULD
1130					:BE THERE
1131					:RECEIVED DATA GIVES WHAT WAS
1132					:THERE AFTER ATTEMPTED
1133					:TRANSFER
1134	001710	057771		DH5	
1135	001712	061762		DT5	
1136	001714	062256		DF5	
1137					
1138					:ITEM 50
1139	001716	051762		EM50	:TRANSFERRING FROM NONEXISTANT
1140					:SECTOR CAUSED DATA ERROR
1141					:GOOD DATA GIVES WHAT
1142					:SHOULD BE IN BUFFER
1143					:RECEIVED DATA GIVES WHAT WAS
1144					:IN BUFFER AFTER TRANSFER
1145	001720	060626		DH30	
1146	001722	062062		DT30	
1147	001724	062311		DF30	
1148					
1149					:ITEM 51
1150	001726	052201		EM51	:GIVING ILLEGAL FUNCTION CAUSED
1151					:IMPROPER REGISTER CHANGE
1152					:GOOD DATA GIVES WHAT SHOULD BE
1153					:THERE
1154					:RECEIVED DATA GIVES REGISTER
1155					:CONTENTS AFTER ILLEGAL FUNCTION
1156	001730	060740		DH51	:PC
1157					:REG. ADDR.
1158					:GOOD DATA
1159					:RECEIVED DATA
1160					:ILLEGAL FUNCTION
1161	001732	062076		DT51	:\$ERRPC,REGADR,\$GDDAT,\$BDDAT,ILLEGL
1162	001734	062316		DF51	:0,0,0,0,0
1163					
1164					
1165					:ITEM 52
1166	001736	052446		EM52	:WRITE DATA ON NONEXISTANT
1167					:SECTOR CAUSED IMPROPER
1168					:REGISTER CHANGE
1169					:GOOD DATA GIVES WHAT SHOULD
1170					:BE THERE
1171					:RECEIVED DATA GIVES WHAT
1172					:WAS THERE AFTER ATTEMPTED
1173					:WRITE DATA
1174	001740	057771		DH5	
1175	001742	061762		DT5	
1176	001744	062256		DF5	
1177					
1178					:ITEM 53
1179	001746	052717		EM53	:READ HEADER AND DATA AFTER
1180					:A SEARCH CAUSED AN ERROR
1181	001750	060626		DH30	
1182	001752	062062		DT30	
1183	001754	062311		DF30	

1184					
1185			:ITEM 54		
1186	001756	053005	EM54		:ATTEMPTED OPERATION WITH
1187					:INVALID ADDRESS CAUSED
1188					:IMPROPER REGISTER CHANGE
1189					:GOOD DATA GIVES WHAT SHOULD
1190					:BE THERE
1191					:RECEIVED DATA GIVES WHAT WAS
1192					:THERE AFTER OPERATION
1193	001760	057771		DH5	
1194	001762	061762		DT5	
1195	001764	062256		DF5	
1196					
1197			:ITEM 55		
1198	001766	053252	EM55		:WRITING/READING WITH EXPECTED
1199					:ADDRESS OVERFLOW ERROR CAUSED
1200					:IMPROPER REGISTER CHANGE
1201					:GOOD DATA GIVES WHAT SHOULD
1202					:BE THERE
1203					:RECEIVED DATA GIVES WHAT
1204					:WAS THERE AFTER OPERATION
1205	001770	057771		DH5	
1206	001772	061762		DT5	
1207	001774	062256		DF5	
1208					
1209			:ITEM 56		
1210	001776	053540	EM56		:DATA READ WITH AN EXPECTED
1211					:ADDRESS OVERFLOW ERROR IS
1212					:INCORRECT
1213					:WORD NO 1 TO 260 SHOULD
1214					:BE READ
1215					:WORD NOS 261 TO 266 SHOULD
1216					:NOT CHANGE DUE TO READ
1217	002000	060626		DH30	
1218	002002	062062		DT30	
1219	002004	062311		DF30	
1220					
1221			:ITEM 57		
1222	002006	053750	EM57		:ATTEMPTING DATA COMMAND
1223					:WITH WRONG FORMAT BIT CAUSED
1224					:IMPROPER REGISTER CHANGE
1225					:GOOD DATA GIVES WHAT SHOULD BE
1226					:THERE
1227					:RECEIVED DATA GIVES WHAT WAS
1228					:THERE AFTER ATTEMPTED DATA
1229					:TRANSFER
1230	002010	057771		DH5	
1231	002012	061762		DT5	
1232	002014	062256		DF5	
1233					
1234			:ITEM 60		
1235	002016	054242	EM60		:ATTEMPTING TO MODIFY REGISTER
1236					:DURING AN OPERATION CAUSED
1237					:IMPROPER REGISTER CHANGE
1238					:GOOD DATA GIVES WHAT SHOULD
1239					:BE THERE

1240				:RECEIVED DATA GIVES WHAT WAS
1241				:THERE AFTER OPERATION
1242				:WAS COMPLETE
1243	002020	061077	DH60	:PC
1244				:REG. ADDR.
1245				:GOOD DATA
1246				:RECEIVED DATA
1247				:MODFING REGISTER
1248	002022	062114	DT60	:\$ERRPC,REGADR,\$GDDAT,\$BDDAT,\$BDADR
1249	002024	062324	DF60	:0,0,0,0,0
1250				
1251			:ITEM 61	
1252	002026	054651	EM61	:DEVICE NOT AVAIBLE BEFOR COMMAND WAS TO BE GIVEN
1253	002030	061234	DH61	:PC
1254				:PC OF JSR
1255				:RHDS1
1256	002032	062132	DT61	:\$ERRPC,PCJSR,\$BDADR
1257	002034	062332	DF61	:0,0,0
1258				
1259			:ITEM 62	
1260	002036	054651	EM61	:DEVICE NOT AVAIBLE BEFOR COMMAND WAS TO BE GIVEN
1261	002040	061326	DH62	:PC
1262				:PC OF JSR
1263				:RHCS1 WAS
1264	002042	062144	DT62	:\$ERRPC,PCJSR,\$BDADR
1265	002044	062336	DF62	:0,0,0
1266				
1267				
1268			:ITEM 63	
1269	002046	054735	EM63	:RHDS1 CONTENTS DURING
1270				:COMMAND WAS IN ERROR
1271	002050	057771	DH5	
1272	002052	061762	DT5	
1273	002054	062256	DF5	
1274				
1275				
1276			:ITEM 64	
1277	002056	055010	EM64	:RECALIBRATE COMMAND CAUSED
1278				:IMPROPER REGISTER CHANGE.
1279				:GOOD DATA GIVES WHAT SHOULD BE
1280				:THERE.
1281				:RECEIVED DATA GIVES WHAT WAS THERE
1282				:AFTER COMMAND
1283	002060	057771	DH5	
1284	002062	061762	DT5	
1285	002064	062256	DF5	
1286				
1287				
1288			:ITEM65	
1289				
1290	002066	055227	EM65	:INTERRUPT FAILING
1291	002070	061401	DH65	:PC
1292				:TEST NO
1293				:CONTENTS OF RHCS1
1294				:CONTENTS OF RHAS
1295				:CONTENTS OF RHDS1

1296	002072	062156		DT65	;\$ERRPC,TSTNM,CS1,AS,DS1
1297	002074	062342		DF65	:0,0,0,0,0
1298					
1299					
1300			:ITEM66		
1301	002076	055251		EM66	:HEADER AND DATA COMMAND
1302					:FOR HEAD SELECTION TEST
1303					:CAUSED AN ERROR
1304					:RHDST GIVES WHAT TRACK
1305					:WAS BEING WRITTEN ON CYLINDER 0
1306					:SECTOR 0
1307	002100	061515		DH66	:PC
1308					:RHDST
1309					:RHER1
1310					:RHER2
1311					:RHER3
1312					:RHCS1
1313					:RHCS2
1314	002102	062172		DT66	;\$ERRPC,DST,ER1,ER2,ER3,CS1,CS2
1315	002104	062347		DF66	:0,0,0,0,0,0,0
1316			:ITEM67		
1317	002106	055462		EM67	:READ HEADER AND DATA ERROR
1318					:IN HEAD SELECTION TEST
1319					:FIRST FOUR WORDS GIVE HEADER
1320					:NEXT WORDS ARE DATA
1321					:GOOD DATA WORDS GIVE
1322					:THE TRACK NUMBER IN
1323					:BITS 4,5,6,7,8
1324	002110	060626		DH30	
1325	002112	062062		DT30	
1326	002114	062311		DF30	
1327			:ITEM70		
1328	002116	055736		EM70	:READ HEADER AND DATA ERROR
1329					:IN DIFFERENCE LINE TEST
1330					:WORD NOS. 1-4 GIVE
1331					:HEADER
1332					:WORD NOS. 5-260 GIVE DATA
1333					:WHICH IS THE CYLINDER
1334					:ADDRESS
1335	002120	060626		DH30	
1336	002122	062062		DT30	
1337	002124	062311		DF30	
1338					
1339			:ITEM 71		
1340	002126	056144		EM71	:FORCING OPI CAUSED IMPROPER REGISTER
1341					:CHANGE
1342					:GOOD DATA GIVES WHAT SHOULD
1343					:BE THERE
1344					:RECEIVED DATA GIVES WHAT WAS
1345					:THERE AFTER 3 INDEX PULSES
1346	002130	057771		DH5	:PC
1347					:REG. ADDR.
1348					:GOOD DATA
1349					:RECEIVED DATA
1350	002132	061762		DT5	;\$ERRPC,REGADR,\$GDDAT,\$BDDAT
1351	002134	062256		DF5	:0,0,0,0

1352					
1353			:ITEM72		
1354	002136	056405	EM72		:THERE WAS AN ERROR
1355					:AFTER A WRITE HEADER
1356					:AND DATA COMMAND
1357					
1358	002140	061613	DH72		:PC
1359					:RHCS1
1360					:RHCS2
1361					:RHDS1
1362					:RHDST
1363					:RHCA
1364					:RHER1
1365					:RHCW
1366	002142	062214	DT72		:\$ERRPC,CS1,CS2,DS1,DST,CA,ER1,WC
1367	002144	062360	DF72		:0,0,0,0,0,0,0,0
1368					
1369					
1370					
1371					
1372					
1373			:ITEM73		
1374	002146	056653	EM73		:READING OVER 3 INDEX
1375					:PULSES CAUSED SC
1376	002150	061613	DH72		
1377	002152	062214	DT72		
1378	002154	062360	DF72		
1379					
1380			:ITEM74		
1381	002156	057023	EM74		:READING OVER 3 INDEX
1382					:PULSES CAUSED OPI
1383	002160	061613	DH72		
1384	002162	062214	DT72		
1385	002164	062360	DF72		
1386					

1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431

002166 000254

;RH11 REGISTER BITS

RPVEC: 254

;RP04 VECTOR ADDRESS

;WORD COUNT REGISTER (RHWC)
;EACH BIT IS CALLED BY BIT NUMBER

;BUS ADDRESS REGISTER (RHBA)
;EACH BIT IS CALLED BY BIT NUMBER

;CONTROL AND STATUS REGISTER 2 (RHCS2)

000001
000002
000004
000010
000020
000040
000100
000200
000400
001000
002000
004000
010000
020000
040000
100000

US1= 1
US2= 2
US4= 4
BAI= 10
UNIB= 20
CLR= 40
IR= 100
OR= 200
MPE= 400
MXF= 1000
PGE= 2000
NEM= 4000
NED= 10000
UPE= 20000
WCE= 40000
DLT= 100000

;UNIT SELECT (BIT #0)
;UNIT SELECT (BIT #1)
;UNIT SELECT (BIT #2)
;BUS ADDRESS INCREMENT INHIBIT (BIT #3)
;UNIBUS B DC LO (BIT #4)
;CLEAR (BIT #5)
;INPUT READY (BIT #6)
;OUTPUT READY (BIT #7)
;MASS BUS PARITY ERROR (BIT #8)
;MISSED TRANSFER ERROR (BIT #9)
;PROGRAM ERROR (BIT #10)
;NON EXISTANT MEMORY (BIT #11)
;NON EXISTANT DRIVE (BIT #12)
;UNIBUS PARITY ERROR (BIT #13)
;WRITE CHECK ERROR (BIT #14)
;DATA LATE (BIT #15)

;DATA BUFFER REGISTER (RHDB)
;EACH BIT IS CALLED BY BIT NUMBER


```

1432      ;RP04 REGISTER BITS
1433
1434
1435
1436      ;CONTROL AND STATUS 1 REGISTER. (#00)
1437
1438      000001      GO=      1      ;GO (BIT #0)
1439      000100      IE=      100     ;INTERRUPT ENABLE (BIT #6)
1440      000200      RDY=     200     ;READY (BIT #7)
1441      000400      A16=     400     ;HIGH ORDER UNIBUS BITS (BIT #8)
1442      001000      A17=    1000     ;HIGH ORDER UNIBUS BITS (BIT #9)
1443      002000      PSEL=    2000     ;PORT SELECT (BIT #10)
1444      004000      DVA=     4000     ;DEVICE AVAILABLE (BIT #11)
1445      020000      MCPE=   20000     ;MASSBUSS PARITY ERROR (BIT #13)
1446      040000      TRE=     40000     ;TRANSFER ERROR (BIT #14)
1447      100000      SC=    100000     ;SPECIAL CONDITION (BIT #15)
1448
1449      ;STATUS REGISTER (RHDS1) (#01)
1450
1451      000001      DFF5=      1      ;DRIVE FORWARD 5'/SEC. (BIT #0)
1452      000002      DFF20=     2      ;DRIVE FORWARD 20'/SEC. (BIT #1)
1453      000004      DIGB=      4      ;DRIVE TO INNER GAVRD BAND (BIT #2)
1454      000010      GRV=     10      ;GO REVERSE (BIT #3)
1455      000020      DL64=     20      ;DIFFERENCE LESS THAN 64 (BIT #4)
1456      000040      DE1=      40      ;DIFFERENCE EQUALS 1 (BIT #5)
1457      000100      VV=     100      ;VOLUME VALID (BIT #6)
1458      000200      DRY=     200      ;DRIVE READY (BIT #7)
1459      000400      DPR=     400      ;DRIVE PRESENT (BIT #8)
1460      001000      PROG=    1000      ;PROGRAMABLE (BIT #9)
1461      002000      LBT=     2000      ;LAST SECTOR TRANSFERRED (BIT #10)
1462      004000      WRL=     4000      ;WRITE LOCK (BIT #11)
1463      010000      MOL=    10000      ;MEDIUM ON-LINE (BIT #12)
1464      020000      PIP=    20000      ;POSITIONING OPERATION IN PROGRESS (BIT #13)
1465      040000      ERR=     40000      ;COMPOSIT ERROR. (BIT #14)
1466      100000      ATA=    100000      ;ATTENTION ACTIVE (BIT #15)
1467
1468      ;ERROR REGISTER #01 (RHER1) (#02)
1469      000001      ILF=      1      ;ILLEGAL FUNCTION (BIT #0)
1470      000002      ILR=      2      ;ILLEGAL REGISTER (BIT #1)
1471      000004      RMR=      4      ;REGISTER MODIFICATION REFUSED (BIT #2)
1472      000010      PAR=     10      ;PARITY ERROR (BIT #3)
1473      000020      FER=     20      ;FORMAT ERROR (BIT #4)
1474      000040      WCF=     40      ;WRITE CLOCK FAIL (BIT #5)
1475      000100      ECH=     100      ;ECC HARD ERROR (BIT #6)
1476      000200      HCE=     200      ;HEADER COMPARE ERROR (BIT #7)
1477      000400      HCRC=    400      ;HEADER CRC ERROR (BIT #8)
1478      001000      AOE=    1000      ;ADDRESS OVERFLOW ERROR (BIT #9)
1479      002000      IAE=    2000      ;INVALID ADDRESS ERROR (BIT #10)
1480      004000      WLE=    4000      ;WRITE LOCK ERROR (BIT #11)
1481      010000      DTE=   10000      ;DRIVE TIMING ERROR (BIT #12)
1482      020000      OPI=   20000      ;OPERATION INCOMPLETE (BIT #13)
1483      040000      UNS=   40000      ;DRIVE UNSAFE (BIT #14)
1484      100000      DCK=  100000      ;DATA CHECK ERROR (BIT 15)
1485
1486      ;MAINTAINABILITY REGISTER (RHMR) (#03)
1487
    
```


1488	000001	DMD= 1	:DIAGINOSTIC MODE (BIT #0)
1489	000002	MCLK= 2	:MAINTAINABILITY CLOCK (BIT #1)
1490	000004	MINX= 4	:MAINTAINABILITY INDEX (BIT #2)
1491	000010	MSTCK= 10	:MAINTAINABILITY SECTOR CLOCK (BIT #3)
1492	000020	MRD= 20	:MAINTAINABILITY READ (BIT #4)
1493	000040	MWR= 40	:MAINTAINABILITY WRITE (BIT #5)
1494	001000	DTSY= 1000	:MAINTAINABILITY SYNC DETECTED (BIT #9)

:ATTENTION SUMMARY PSEUDO-REGISTER (RHAS) (#04)

1497			
1498	000001	AT0= 1	:DEVICE 0 (BIT #0)
1499	000002	AT1= 2	:DEVICE 1 (BIT #1)
1500	000004	AT2= 4	:DEVICE 2 (BIT #2)
1501	000010	AT3= 10	:DEVICE 3 (BIT #3)
1502	000020	AT4= 20	:DEVICE 4 (BIT #4)
1503	000040	AT5= 40	:DEVICE 5 (BIT #5)
1504	000100	AT6= 100	:DEVICE 6 (BIT #6)
1505	000200	AT7= 200	:DEVICE 7 (BIT #7)

:DESIRED SECTOR/TRACK ADDRESS REGISTER (RHDST) (#1)
 :EACH BIT IS CALLED BY BIT NUMBER

1512
1513
1514
1515
1516
1517
1518

:DRIVE TYPE REGISTER (RHDT) (#06)
 :EACH BIT IS CALLED BY BIT NUMBER

1519
1520
1521
1522
1523
1524
1525

:LOOK-AHEAD REGISTER (RHLA) (#07)

1526			
1527			
1528	000001	EXT1= 1	:EXTENSION 1 (BIT #0)
1529	000002	EXT2= 2	:EXTENSION 2 (BIT #1)
1530	000004	EXT4= 4	:EXTENSION 3 (BIT #2)
1531	000010	EXT10= 10	:EXTENSION 4 (BIT #3)
1532	000020	EXT20= 20	:EXTENSION 5 (BIT #4)
1533	000040	EXT40= 40	:EXTENSION 6 (BIT #5)
1534	000100	SC1= 100	:SECTOR COUNT FIELD 0 (BIT #6)
1535	000200	SC2= 200	:SECTOR COUNT FIELD 1 (BIT #7)
1536	000400	SC4= 400	:SECTOR COUNT FIELD 2 (BIT #8)
1537	001000	SC10= 1000	:SECTOR COUNT FIELD 3 (BIT #9)
1538	002000	SC20= 2000	:SECTOR COUNT FIELD 4 (BIT #10)
1539	004000	TRK1= 4000	:TRACK FIELD 1 (BIT #11)
1540	010000	TRK2= 10000	:TRACK FIELD 2 (BIT #12)
1541	020000	TRK4= 20000	:TRACK FIELD 3 (BIT #13)
1542	040000	TRK10= 40000	:TRACK FIELD 4 (BIT #14)
1543	100000	TRK20= 100000	:TRACK FIELD 5 (BIT #15)


```

1544
1545           ;ERROR REGISTER #2 (RHCR2) (#10)
1546
1547           000001      WCU= 1           ;WRITE CURRENT UNSAFE (BIT #0)
1548           000002      CSF= 2           ;CURRENT SINK FAILURE (BIT #1)
1549           000004      WSU= 4           ;WRITE SELECT UNSAFE (BIT #2)
1550           000010      CSU= 10          ;CURRENT SWITCH UNSAFE (BIT #3)
1551           000020      MSE= 20          ;MOTOR SEQUENCE ERROR (BIT #4)
1552           000040      TDF= 40          ;TRANSITIONS DETECTOR FAILURE (BIT #5)
1553           000100      TUF= 100         ;TRANSITIONS UNSAFE (BIT #6)
1554           000200      FEN= 200         ;FAILSAFE ENABLED (BIT #7)
1555           000400      WRU= 400         ;WRITE READY UNSAFE (BIT #8)
1556           001000      MHS= 1000        ;MULTIPLE HEAD SELECT (BIT #9)
1557           002000      NHS= 2000        ;NO HEAD SELECTION (BIT #10)
1558           004000      IXE= 4000        ;INDEX ERROR (BIT #11)
1559           010000      VU30= 10000      ;30VOLT UNSAFE (BIT #12)
1560           020000      PLU= 20000       ;PLO UNSAFE (BIT #13)
1561           100000      ACU= 100000      ;ACUNSAFE (BIT #15)
1562
1563           ;OFFSET REGISTER (RHOF) (#11)
1564
1565           000001      OF25= 1           ;OFFSET 25 MICRO INCHES (BIT #0)
1566           000002      OF50= 2           ;OFFSET 50 MICRO INCHES (BIT #1)
1567           000004      OF100= 4          ;OFFSET 100 MICRO INCHES (BIT #2)
1568           000010      OF200= 10         ;OFFSET 200 MICRO INCHES (BIT #3)
1569           000020      OF400= 20         ;OFFSET 400 MICRO INCHES (BIT #4)
1570           000040      OF800= 40         ;OFFSET 800 MICRO INCHES (BIT #5)
1571
1572           000200      OFREV= 200         ;OFFSET NEGATIVE (REVERSE) (BIT #5)
1573           002000      HCI= 2000         ;HEADER COMPARE INHIBIT (BIT #10)
1574           004000      ECI= 4000         ;ERROR CORRECTION CODE INHIBIT (BIT #11)
1575           010000      FMT22= 10000      ;FORMAT BIT (BIT #12)
1576
1577           ;DESIRED CYLINDER ADDRESS (RHCA) (#12)
1578           ;EACH BIT IS CALLED BY BIT NUMBER.
1579
1580
1581
1582
1583           ;CURRENT CYLINDER ADDRESS (RHCC) (#13)
1584           ;EACH BIT IS CALLED BY BIT NUMBER
1585
1586
1587
1588
1589           ;SERIAL NUMBER REGISTER (RHSN) (#14)
1590           ;EACH IS CALLED BY BIT NUMBER
1591
1592
1593
1594
1595           ;ERROR REGISTER #03 (RHER3) (#15)
1596
1597           000001      PSU= 1             ;PACK SPEED UNSAFE (BIT #0)
1598           000002      VUF= 2           ;VELOCITY UNSAFE (BIT #1)
1599           000010      UWR= 10          ;ANY UNSAFE EXCEPT READ/WRITE (BIT #3)
    
```


1600	000020	PRE=	20	:DISK PACK ROTATION ERROR (BIT #4)
1601	000040	ACL=	40	:AC LOW (BIT #5)
1602	000100	DCL=	100	:DC LOW (BIT #6)
1603	040000	SKI=	40000	:SEEK INCOMPLETE (BIT #14)
1604	100000	OCYL=	100000	:OFF CYLINDER (BIT #15)
1605				
1606				
1607				
1608		:ECC POSITION REGISTER (RHEC1) (#16)		
1609		:EACH BIT IS CALLED BY BIT NUMBER		
1610				
1611				
1612				
1613				
1614		:ECC PATTERN REGISTER (RHEC2) (#17)		
1615		:EACH BIT IS CALLED BY BIT NUMBER		
1616				


```
1617 .SBTTL REGISTER ADDRESSES
1618
1619
1620
1621
1622 ;RP04/5/6 DISK I/O REGISTERS LOCATED IN THE RH11 CONTROLLER
1623
1624 002170 176722 RHDB: 176722 ;DATA BUFFER
1625 002172 176702 RHWC: 176702 ;WORD COUNT
1626 002174 176704 RHBA: 176704 ;BUS ADDRESS
1627 002176 176710 RHCS2: 176710 ;CONTROL AND STATUS 2
1628
1629 ;RP04/5/6 DISK I/O REGISTERS LOCATED IN THE DEVICE CONTROL LOGIC (DCL)
1630
1631 002200 176700 RHCS1: 176700 ;CONTROL AND STATUS 1
1632 002202 176714 RHER1: 176714 ;ERROR #1
1633 002204 176706 RHDST: 176706 ;DESIRED SECTOR/TRACK ADDRESS
1634 002206 176740 RHER2: 176740 ;ERROR #2
1635 002210 176732 RHOF: 176732 ;OFFSET
1636 002212 176734 RHCA: 176734 ;DESIRED CYLINDER ADDRESS
1637 002214 176742 RHER3: 176742 ;ERROR #3
1638 002216 176716 RHAS: 176716 ;ATTENTION SUMMARY
1639 002220 176724 RHMR: 176724 ;MAINTAINABILITY
1640 002222 176712 RHDS1: 176712 ;DRIVE STATUS
1641 002224 176726 RHDT: 176726 ;DRIVE TYPE
1642 002226 176730 RHSN: 176730 ;SERIAL NUMBER
1643 002230 176744 RHEC1: 176744 ;ECC POSITION
1644 002232 176746 RHEC2: 176746 ;ECC PATTERN
1645 002234 176736 RHCC: 176736 ;CURRENT CYLINDER ADDRESS
1646 002236 176720 RHLA: 176720 ;LOOK-AHEAD
1647
1648 ;ADDITIONAL REGISTERS LOCATED IN THE RH70 CONTROLLER LOGIC
1649
1650 002240 176750 RHBAE: 176750 ;BUS ADDRESS EXTENSION REGISTER
1651 002242 176752 RHCS3: 176752 ;CONTROL AND STATUS REGISTER #3
1652
1653
1654 ;P-CLOCK (KW11-P) I/O REGISTERS
1655
1656 002244 172540 PCLCSR: 172540 ;CONTROL AND STATUS REGISTERS
1657 002246 172542 PCLBUF: 172542 ;COUNT SET BUFFER
1658 002250 172544 PCLCTR: 172544 ;COUNTER
1659
```



```
1660
1661
1662           ;THE FOLLOWING LOCATIONS ARE RESERVED FOR REGISTER SAVES
1663           ;ANY TIME THERE IS AN ERROR ALL THESE WILL BE FILLED
1664           ;ONLY SOME MAY BE PRINTED BUT ALL WILL BE FILLED TRUE
1665           ;FOR THE TIME JUST AFTER THE 'ERROR' ERROR COMMAND
1666
1667 002252 000000      DB:      0           ;DATA BUFFER
1668 002254 000000      WC:      0           ;WORD COUNT
1669 002256 000000      BA:      0           ;BUS ADDRESS
1670 002260 000000      CS2:     0           ;CONTROL AND STATUS 2
1671
1672
1673 002262 000000      CS1:     0           ;CONTROL AND STATUS 1
1674 002264 000000      ER1:     0           ;ERROR #1
1675 002266 000000      DST:     0           ;DESIRED SECTOR/TRACK ADDRESS
1676 002270 000000      ER2:     0           ;ERROR #2
1677 002272 000000      OF:      0           ;OFFSET
1678 002274 000000      CA:      0           ;DESIRED CYLINDER ADDRESS
1679 002276 000000      ER3:     0           ;ERROR #3
1680 002300 000000      AS:      0           ;ATTENTION SUMMARY
1681 002302 000000      MR:      0           ;MAINTAINABILITY
1682 002304 000000      DS1:     0           ;DRIVE STATUS
1683 002306 000000      DT:      0           ;DRIVE TYPE
1684 002310 000000      SN:      0           ;SERIAL NUMBER
1685 002312 000000      EC1:     0           ;ECC POSITION
1686 002314 000000      EC2:     0           ;ECC PATTERN
1687 002316 000000      CC:      0           ;CURRENT CYLINDER ADDRESS
1688 002320 000000      LA:      0           ;LOOK-AHEAD
1689
1690
1691
```



```
1692
1693
1694
1695           ;FUNCTION EQUATES
1696
1697           ;*TABLE OF FUNCTIONS FOR RHCS1 THEN 'GO' BIT HAS TO BE SET
1698 002322 FUTABL:
1699 002322 000000 NOPERA: 0           ;NO OPERATION
1700 002324 000002 UNLOAD: 2        ;UNLOAD (STAND BY)
1701 002326 000006 RECALI: 6        ;RECALIBRATE
1702 002330 000010 DCLEAR: 10       ;DRIVE CLEAR
1703 002332 000012 RELEAS: 12       ;RELEASE (DUAL-PORT OPERATION)
1704 002334 000030 SERCH: 30        ;SEARCH COMMAND
1705 002336 000050 WRCKEK: 50       ;WRITE CHECK DATA
1706 002340 000052 WRCHDT: 52       ;WRITE CHECK HEADER AND DATA
1707 002342 000060 WRIDAT: 60       ;WRITE DATA
1708 002344 000062 WRIFOR: 62       ;WRITE HEADER AND DATA (FORMAT)
1709 002346 000070 READAT: 70       ;READ DATA
1710 002350 000072 REFOR: 72        ;READ HEADER AND DATA
1711 002352 000004 SEECOM: 4        ;SEEK COMMAND
1712 002354 000014 OFSETC: 14       ;OFFSET COMMAND
1713 002356 000016 RETCL: 16        ;RETURN TO CENTERLINE
1714 002360 000022 PKACK: 22        ;PACK ACKNOWLEDGE
1715 002362 000020 READIN: 20       ;READ IN
1716 002364 000000 ILLEGL: .WORD 0    ;COMPUTED ILLEGAL FUNCTION
1717
1718
1719
1720           ;*DATA BUFFER FOR READ WRITE
1721
1722
1723 002370 000422 WRFROM: .BLKW 274.    ;WRITE FROM THIS BUFFER
1724 003434 000422 REINTO: .BLKW 274.    ;READ INTO THIS BUFFER
1725
```



```

1726
1727
1728 ;RESERVED LOCATIONS FOR FLAGS AND INTERNAL PROGRAM CONTROL WORDS
1729
1730 004500 000000 REGADR: 0 ;SAVE REGISTER ADDRESS HERE
1731 004502 000000 ERWORD: 0 ;SAVE ERROR WORD NUMBER HERE
1732 004504 000000 TSTNM: 0 ;TEST NUMBER
1733 004506 000000 RP4VEC: 0 ;CONTAINS ADDRESS OF LOCATION
;WHERE AN RP04 INTERRUPT IS TO VECTOR TO
1734 ;THIS MUST BE MOVED INTO 'RPVEC' TO BE
1735 ;EFFECTIVE.
1736
1737
1738 004510 000000 OFSTVL: 0 ;OFFSET VALUE USED IN OFFSET TEST
1739
1740
1741 004512 000024 SAVERE: .BLKW 20. ;BLOCK TO SAVE REGISTERS
1742 004562 000000 FINALA: 0 ;SAVE LOOK AHEAD REGISTER AT END OF OPERATION
1743 004564 000000 FINACC: 0 ;SAVE CURRENT CYLINDER REGISTER AT END OF OPERATION
1744
1745
1746 ;TABLE FOR ATTENTION BITS
1747 ;ATTENTION TABLE
1748
1749 004566 001 002 004 ATABLE: .BYTE 1,2,4,10,20,40,100,200
1750 004571 010 020 040
1751 004574 100 200
1752
1753
1754 ;RESERVED LOCATIONS FOR UNIT SELECT
1755
1756
1757 004576 000010 UNITS: .BLKW 8. ;THIS IS FILLED WITH -1
1758 004616 000000 UNIT: .WORD 0 ;UNIT UNDER TEST
1759 004620 000000 NOUNIT: .WORD 0 ;NUMBER OF UNITS PRESENT
1760 ;USED TO KEEP TRACK OF UNIT UNDER TEST
1761 004622 000000 NUNIT: .WORD 0 ;USED TO DETERMIN IF THERE ARE MORE
1762 ;THAN ONE UNIT
1763 004624 000000 NOPUSH: 0 ;ALL ONES INDICATE NONE OF THE OPERATOR
1764 ;INTERVENTION TESTE WILL BE PERFORMED
1765 004626 000000 SELECT: .WORD 0 ;ALL ONES INDICATE UNIT TO BE SELECTED
1766 004630 000000 UNITSL: .WORD 0 ;UNIT NO. SELECTED
1767
1768
1769
1770 004632 000000 ERFLG$: 0 ;ERROR FLAG
1771
1772 004634 000000 FIRST: 0 ;IF ZERO WILL TYPE HEADER
1773 ;IF ONES WILL NOT TYPE HEADER
1774
1775 004636 000000 RP06: 0 ;DEVICE TYPE FLAG
1776
1777 004640 000000 RH70: 0 ;IF 1, PROGRAM IS RUNNING ON AN RH70
1778 ;IF 0, PROGRAM IS ON AN RH11
1779
1780 004642 000000 RUNCTR: .WORD 0 ;'RUN' LINE DELAY COUNTER TO BE USED
1781 ;WHILE THE SILO IS FILLING

```


1782					
1783	004644	000000	ATTENT: 0		:ATTENTION BIT FOR PRESENT UNIT
1784	004646	000000	TOTALAT: 0		:TOTAL ATTENTION BITS
1785					
1786	004650	000000	TMP0: .WORD 0		:TEMP STORAGE
1787	004652	000000	TMP1: .WORD 0		
1788	004654	000000	TMP4: .WORD 0		:TEMP STORAGE
1789	004656	000000	TMP5: .WORD 0		:TEMP STORAGE


```

1790 .SBTTL
1791 .SBTTL *** DIAGNOSTIC CODE ***
1792 .SBTTL
1793
1794
1795 .SBTTL SETUP TESTS
1796
1797
1798 004660 012737 177777 004624 BEGIN1: MOV #-1,@#NOPUSH ;JUMP OVER OPERATOR REQUIRED TESTS
1799 004666 005037 004626 CLR @#SELECT ;DO NOT SELECT UNIT
1800 004672 000412 BR START
1801 004674 012737 177777 C04626 BEGIN2: MOV #-1,@#SELECT ;SELECT UNIT
1802 004702 005037 004624 CLR @#NOPUSH ;DO NOT JUMP OVER ANY TEST
1803 004706 000404 BR START
1804 004710 005037 004626 BEGIN: CLR @#SELECT ;DO NOT SELECT UNIT
1805 004714 005037 004624 CLR @#NOPUSH ;DO NOT JUMP OVER ANY TEST
1806 ;NORMAL RUN
1807
1808 START:
1809 004720 000005 RESET
1810 004722 012706 001100 MOV #SCMTAG,R6 ;:FIRST LOCATION TO BE CLEARED
1811 004726 005026 CLR (R6)+ ;:CLEAR MEMORY LOCATION
1812 004730 022706 001140 CMP #SWR,R6 ;:DONE?
1813 004734 001374 BNE .-6 ;:LOOP BACK IF NO
1814 004736 012706 001000 MOV #STACK,SP ;:SETUP THE STACK POINTER
1815 004742 012737 036402 000020 MOV #SCOPE,@#IOTVEC ;:IOT VECTOR FOR SCOPE ROUTINE
1816 004750 012737 000340 000022 MOV #340,@#IOTVEC+2 ;:LEVEL 7
1817 004756 012737 040600 000030 MOV #ERROR,@#EMTVEC ;:EMT VECTOR FOR ERROR ROUTINE
1818 004764 012737 000340 000032 MOV #340,@#EMTVEC+2 ;:LEVEL 7
1819 004772 012737 042324 000034 MOV #TRAP,@#TRAPVEC ;:TRAP VECTOR FOR TRAP CALLS
1820 005000 012737 000340 000036 MOV #340,@#TRAPVEC+2;LEVEL 7
1821 005006 012737 042410 000024 MOV #SPWRDN,@#PWRVEC ;:POWER FAILURE VECTOR
1822 005014 012737 000340 000026 MOV #340,@#PWRVEC+2 ;:LEVEL 7
1823 005022 005037 001212 CLR $TIMES ;:INITIALIZE NUMBER OF ITERATIONS
1824 005026 005037 001214 CLR $ESCAPE ;:CLEAR THE ESCAPE ON ERROR ADDRESS
1825 005032 112737 000001 001115 MOVB #1,$ERMAX ;:ALLOW ONE ERROR PER TEST
1826 005040 012737 005040 001106 MOV #,$LPADR ;:INITIALIZE THE LOOP ADDRESS FOR SCOPE
1827 005046 012737 005046 001110 MOV #,$LPERR ;:SETUP THE ERROR LOOP ADDRESS
1828 005054 013746 000004 MOV @#ERRVEC,-(SP) ;:SAVE ERROR VECTOR
1829 005060 012737 005114 000004 MOV #64$,@#ERRVEC ;:SET UP ERROR VECTOR
1830 005066 012737 177570 001140 MOV #DSWR,SWR ;:SETUP FOR A HARDWARE SWICH REGISTER
1831 005074 012737 177570 001142 MOV #DDISP,DISPLAY ;:AND A HARDWARE DISPLAY REGISTER
1832 005102 022777 177777 174030 CMP #-1,@SWR ;:TRY TO REFERENCE HARDWARE SWR
1833 005110 001012 BNE 66$ ;:BRANCH IF NO TIMEOUT TRAP OCCURRED
1834 005112 000403 BR 65$ ;:BRANCH IF NO TIMEOUT
1835 005114 012716 005122 64$: MOV #65$,(SP) ;:SET UP FOR TRAP RETURN
1836 005120 000002 RTI
1837 005122 012737 000176 001140 65$: MOV #SWREG,SWR ;:POINT TO SOFTWARE SWR
1838 005130 012737 000174 001142 MOV #DISPREG,DISPLAY
1839 005136 012637 000004 66$: MOV (SP)+,@#ERRVEC ;:RESTORE ERROR VECTOR
1840
1841
1842 005142 012737 000000 177776 MOV #0,PS ;:SET PROCESSOR STATUS TO 0
1843 005150 012737 000200 000036 MOV #200,@#TRAPVEC+2 ;:TRAP PRIORITY = 4
1844 005156 013700 002166 MOV @#RPVEC,R0 ;:GET RP VECTOR ADDRESS
1845 005162 012720 036312 MOV #RPVECT,(R0)+ ;:THIS IS FOR UNTIMELY INTERRUPTS

```



```

1846 005166 012710 000340      MOV      #340,(R0)      ;RP04 INTERRUPT SERVICE ROUTINE
1847                                     ;PRIORITY = 7
1848
1849 005172 004737 037340      JSR      PC,@#STKINT   ;INITIALIZE THE TTY KEYBOARD
1850 005176 005737 004634      TST     @#FIRST       ;IS THIS FIRST TIME ROUND ?
1851 005202 001001                BNE     1$            ;DO NOT GIVE HEADER IF NOT
1852 005204 000402                BR      2$            ;GIVE HEADER IF SO
1853 005206 000137 005774      1$:     JMP     @#SND1    ;SKIP OVERALL PROGRAM HEADER
1854
1855                                     2$:
1856 005212 104401 005220      TYPE    ,68$          ;;TYPE ASCIZ STRING
1857 005216 000435                BR      67$          ;;GET OVER THE ASCIZ
1858 005312 104401 005320      TYPE    ,70$          ;;TYPE ASCIZ STRING
1859 005316 000414                BR      69$          ;;GET OVER THE ASCIZ
1860
1861 005350 104401 005356      TYPE    ,72$          ;;TYPE ASCIZ STRING
1862 005354 000433                BR      71$          ;;GET OVER THE ASCIZ
1863 005444 104401 005452      TYPE    ,74$          ;;TYPE ASCIZ STRING
1864 005450 000433                BR      73$          ;;GET OVER THE ASCIZ
1865 005540 104401 005546      TYPE    ,76$          ;;TYPE ASCIZ STRING
1866 005544 000436                BR      75$          ;;GET OVER THE ASCIZ
1867
1868 005642 104401 005650      TYPE    ,78$          ;;TYPE ASCIZ STRING
1869 005646 000430                BR      77$          ;;GET OVER THE ASCIZ
1870 005730 104401 005736      TYPE    ,80$          ;;TYPE ASCIZ STRING
1871 005734 000417                BR      79$          ;;GET OVER THE ASCIZ
1872
1873 005774 012737 177777 004634  SND1:  MOV     #-1,@#FIRST   ;NEXT TIME DO NOT GIVE HEADER
1874
1875 006002 005737 000042                TST     @#42         ;;ARE WE RUNNING UNDER XXDP/ACT?
1876 006006 001006                BNE     64$          ;;BRANCH IF YES
1877 006010 023727 001140 000176      CMP     SWR,#SWREG   ;;SOFTWARE SWITCH REG SELECTED?
1878 006016 001005                BNE     65$          ;;BRANCH IF NO
1879 006020 104406                GTSWR                ;;GET SOFT-SWR SETTINGS
1880 006022 000403                BR      65$
1881 006024 112737 000001 001134 64$:  MOVB   #1,$AUTOB    ;;SET AUTO-MODE INDICATOR
1882
1883 006032 032777 010000 173100  RH70CK: BIT    #SW12,@SWR    ;LOOK TO SEE IF USING RH70
1884 006040 001403                BEQ     3$           ;IF SW12 = 0, SKIP NEXT
1885 006042 012737 000001 004640      MOV     #1,@#RH70   ;IF SW12 = 1, CU IS AN RH70
1886 006050                                     3$:

```



```

1887                                     ;*IS THERE A P-CLOCK (KW11-P) ON THE SYSTEM ?
1888                                     ;*IF SO MAKE 'WAT' TRAPS GO TO 'WAIT.P'
1889                                     ;*IF SO MAKE RPO4 INTERRUPTS GO TO 'TIME 1'
1890                                     ;*IF NOT MAKE 'WAT' TRAPS GO TO 'WAIT.T'
1891                                     ;*IF NOT MAKE RPO4 INTERRUPTS GO TO 'TIME 2'
1892
1893                                     ;*THE NEXT LINE IS TO BE ADDED LATER
1894                                     ;*AND THE JUMP AND NOP REMOVED
1895                                     ;*FOR NOW NO CLOCK WILL BE USED
1896
1897                                     ;*MOV @#1$,@#ERRVEC ;SET TIME-OUT VECTOR
1898
1899                                     : JMP @#1$ ;DO NOT USE CLOCK
1900                                     : NOP
1901                                     : TST @#PCLCSR ;REFERENCE P-CLOCK STATUS REGISTER
1902                                     : ;ADDRESS = 172540
1903                                     : MOV #WAIT.P,@#$TRPAD+20 ;THERE IS A P-CLOCK
1904                                     : MOV #TIME1,@#RP4VEC ;THERE IS A P CLOCK SO
1905                                     : ;VECTOR TO TIME1
1906                                     : BR 2$
1907                                     :1$: MOV #WAIT.T,@#$TRPAD+20 ;THERE IS NO P-CLOCK
1908
1909
1910 006050 012737 033304 004506 MOV #TIME2,@#RP4VEC ;MAKE RPO4/5/6 INTERRUPTS GO TO 'TIME 2'
1911 006056 012737 177777 040746 2$: MOV #-1,@#PRITEM ;CLEAR PREVIOUS ITEM NUMBER
1912
1913 006064 005737 004626 TST @#SELECT ;WAS IT A 200 START
1914 006070 001442 BEQ TST1 ;BRANCH IF STARTING FROM 200
1915 006072 104401 006100 TYPE ,65$ ;:TYPE ASCIZ STRING
1916 006076 000424 BR 64$ ;:GET OVER THE ASCIZ
1917 006150 104412 RDOCT
1918 006152 042716 177770 BIC #177770,(SP) ;ONLY KEEP LAST 3 BITS
1919 006156 011637 004616 MOV (SP),@#UNIT ;SAVE UNIT TO BE TESTED
1920 006162 012637 004630 MOV (SP)+,@#UNITSL ;SAVE UNIT TO BE TESTED
1921 006166 001403 BEQ TST1 ;BRANCH IF STARTING FROM 200
1922 006170 013737 004630 004616 MOV @#UNITSL,@#UNIT ;SET UNIT NUMBER
1923
  
```



```

1924
1925 006176 000004          TST1:  SCOPE
1926 006200 012737 000001 001212  MOV    #1,$TIMES      ;;DO 1 ITERATION
1927 006206 012737 000001 004504  MOV    #2-1,@#TSTNM  ;;THIS SAVES TEST NUMBER
1928 006214 012706 001000          MOV    #STACK, SP    ;;SET UP STACK POINTER
1929 006220 012737 040610 000030  MOV    #REGSA1,@#EMTVEC;ERROR VECTOR SO THAT
1930                                ;;NO REGISTERS ARE SAVED
1931 006226 012737 006254 000004  MOV    #2$,@#ERRVEC  ;;SET UP FOR BUS TIMEOUT
1932
1933 006234 012700 000024          MOV    #24,R0        ;;THERE ARE 24 REG TO TEST
1934 006240 012701 002170          MOV    #RHDB,R1      ;;R1 NOW HAS ADDR OF ADDR OF FIRST REG.
1935 006244 013102          1$:  MOV    @(R1)+, R2    ;;READ HARDWARE REG.
1936 006246 005300          DEC    R0            ;;COUNT DOWN
1937 006250 001375          BNE   1$            ;;BRANCH IF 24 NOT DONE
1938 006252 000454          BR    3$            ;;BRANCH IF 24 DONE
1939 006254 012737 000006 000004  2$:  MOV    #ERRVEC+2,@#ERRVEC ;RESTORE TRAP CATCHER
1940 006262 022626          CMP    (SP)+,(SP)+  ;;CLEAN STACK
1941 006264 016137 177776 001200  MOV    -2(R1), $TMP1 ;STORE FAILING REG ADDR
1942 006272 104007          ERROR 7            ;;REGISTER NON EXISTANT
1943 006274 032777 020000 172636  BIT    #SW13,@SWR   ;;INHIBIT ERROR PRINTOUT ?
1944 006302 001036          BNE   4$            ;;BRANCH IF YES
1945
1946 006304 104401 006312          TYPE  ,65$         ;;TYPE ASCIZ STRING
1947 006310 000427          BR    64$         ;;GET OVER THE ASCIZ
1948
1949 006370 012746 000204          MOV    #ADDMOD,-(SP) ;GET READY TO TYPE STARTING ADDRESS
1950                                ;;OF 'CHANGE OF BASE ADDRESS' ROUTINE
1951 006374 104402          TYPOC
1952 006376 000000          HALT
1953 006400 000137 032574          4$:  JMP    @#$EOP      ;GO TO END OF PROGRAM ----->
1954
1955 006404 012737 006464 000004  3$:  MOV    #TRP,@#4     ;INITIALIZE VECTOR
1956 006412 005037 004640          CLR    RH70        ;INIT RH INDICATOR ++ C.W
1957 006416 005777 173616          TST   @RHBAB       ;ADDRESS RPBAE(RH11/RH70?)
1958 006422 005237 004640          INC   RH70        ;FOUND AN RH70-SET MASK
1959 006426 104401 006434          TYPE  ,67$         ;;TYPE ASCIZ STRING
1960 006432 000413          BR    66$         ;;GET OVER THE ASCIZ
1961 006462 000417          BR    RTN         ;GET OUT
1962 006464 022626          TRP:  CMP    (SP)+,(SP)+ ;ADJUST THE STACK
1963 006466 104401 006474          TYPE  ,65$         ;;TYPE ASCIZ STRING
1964 006472 000413          BR    64$         ;;GET OVER THE ASCIZ
1965 006522 012737 040600 000030  RTN:  MOV    #ERROR,@#EMTVEC;RESTORE ERROR VECTOR
1966                                ;;SO THAT REGISTERS ARE SAVED
1967 006530 012737 000006 000004  MOV    #ERRVEC+2,@#ERRVEC ;RESTORE TRAP CATCHER
1968
  
```



```

1969
1970 006536 000004
1971 006540 012737 000001 001212 TST2: SCOPE
1972 006546 012706 001000 MOV #1,$TIMES ;;DO 1 ITERATION
1973 MOV #STACK,SP ;SET STACK POINTER
1974 006552 013701 002216 MOV @#RHAS,R1 ;R1 HAS ADDRESS OF RHAS
1975 006556 012711 177777 MOV #-1,@R1 ;WRITE ALL ONES INTO RHAS
1976 006562 105711 TSTB @R1 ;TEST IT FOR ALL 0'S
1977 006564 001407 BEQ TST3 ;BRANCH IF GOOD
1978 006566 011137 001126 MOV @R1,@#$BDDAT ;BAD DATA
1979 006572 005037 001124 CLR @#$GDDAT ;GOOD DATA
1980 006576 010137 004500 MOV R1,@#REGADR ;FAILING REG. RHAS
1981 006602 104005 ERROR 5 ;RHAS DOES NOT CLEAR BY WRITING ALL
1982 ;ONES INTO IT
1983
    
```



```

1984
1985 006604 000004
1986 006606 012737 000001 001212 TST3: SCOPE
1987
1988 006614 012737 000003 004504 MOV #1,$TIMES ;:DO 1 ITERATION
1989 006622 000005 RESET #4-1,@#TSTNM ;:THIS SAVES TEST NUMBER
1990 006624 004737 037340 JSR PC,@#$TKINT ;:START WITH AN INIT
1991
1992 006630 032777 020000 172302 BIT #SW13,@SWR ;:INHIBIT ERROR TYPEOUT?
1993 006636 001026 BNE 4$ ;:BRANCH IF YES
1994 006640 104401 006646 TYPE ,65$ ;:TYPE ASCIZ STRING
1995 006644 000423 BR 64$ ;:GET OVER THE ASCIZ
1996 006714 013701 002216 4$: MOV @#RHAS,R1 ;:R1 HAS ADDR. OF RHAS
1997 006720 013702 002176 MOV @#RHCS2,R2 ;:R2 HAS ADDR. OF RHCS2
1998 006724 005012 CLR @R2 ;:CLEAR RHCS2
1999 006726 012700 000010 MOV #8.,R0 ;:COUNT
2000 006732 013704 002202 MOV @#RHER1,R4 ;:R4 HAS ADDR. OF RHER1
2001 006736 012714 177777 1$: MOV #-1,@R4 ;:MOVE ERRORS INTO RHER1
2002 006742 005212 INC @R2 ;:INCREMENT UNIT NO.
2003 006744 005300 DEC R0 ;:COUNT
2004 006746 001373 BNE 1$ ;:BRANCH IF 8 NOT DONE
2005 006750 111137 004646 MOVB @R1,@#TOTALAT ;:SAVE TOTAL ATTENTION
2006
2007 006754 105037 004647 CLRB @#TOTALAT+1 ;:USED IN DRIVE CLEAR TEST
2008 006760 105711 TSTB @R1 ;:CLEAR UPPER BYTE
2009 006762 001402 BEQ 2$ ;:TEST FOR ANY DRIVES PRESENT
2010 006764 000137 007346 JMP XE2 ;:IF NONE THERE - SAY SO
2011
2012 006770 032777 020000 172142 2$: BIT #SW13,@SWR ;:INHIBIT ERROR TYPE OUT?
2013 006776 001402 BEQ 3$ ;:BRANCH IF NO
2014 007000 000137 007704 JMP SELTST ;:CHECK FOR SELECTED UNIT START AND LOAD
2015
2016
2017 007004 3$:
2018 007004 104401 007012 TYPE ,67$ ;:TYPE ASCIZ STRING
2019 007010 000421 BR 66$ ;:GET OVER THE ASCIZ
2020 007054 104401 007062 TYPE ,69$ ;:TYPE ASCIZ STRING
2021 007060 000430 BR 68$ ;:GET OVER THE ASCIZ
2022 007142 104401 007150 TYPE ,71$ ;:TYPE ASCIZ STRING
2023 007146 000430 BR 70$ ;:GET OVER THE ASCIZ
2024 007230 104401 007236 TYPE ,73$ ;:TYPE ASCIZ STRING
2025 007234 000442 BR 72$ ;:GET OVER THE ASCIZ
2026
2027 007342 000137 032574 JMP @#$EOP ;:GO OUT ----->
2028
2029
2030 ;*SET UP THE UNITS TABLE
2031
2032 007346 XE2:
2033 007346 012700 000010 2$: MOV #8.,R0 ;:COUNTER
2034 007352 012703 004576 MOV #UNITS,R3 ;:POINTER
2035 007356 012723 177777 3$: MOV #-1,(R3)+ ;:PRESET BLOCK TO ALL ONES
2036 007362 005300 DEC R0 ;:COUNT
2037 007364 001374 BNE 3$ ;:BRANCH IF 8 NOT DONE
2038 007366 012703 004576 MOV #UNITS,R3 ;:POINTER
2039 007372 005005 CLR R5
    
```



```

2040 007374 005037 004620 CLR @#NUNIT ;NO. OF UNITS PRESENT
2041 007400 012700 000010 MOV #8,R0 ;COUNTER
2042 007404 011137 001176 MOV @R1,@#STMP0 ;TEMPORARY STORAGE
2043 007410 006037 001176 4$: ROR @#STMP0 ;SET CARRY IF ONE IN 0 BIT
2044 007414 103120 BCC 5$ ;CHECK NEXT UNIT IF ONE NOT IN BIT 0
2045
2046 007416 010577 172554 MOV R5,@RHCS2 ;INSERT UNIT NUMBER INTO UA BITS
2047 007422 022777 024020 172574 CMP #24020,@RHDT ;IS THIS A DUAL PORT RP04 ?
2048 007430 001503 BEQ 6$ ;TYPE DRIVE NO. IF SO
2049 007432 022777 020020 172564 CMP #20020,@RHDT ;IS THIS A SINGLE PORT RP04 ?
2050 007440 001477 BEQ 6$ ;TYPE NO. IF SO
2051
2052 007442 022777 024021 172554 CMP #24021,@RHDT ;IS THIS A DUAL PORT RP05 ?
2053 007450 001473 BEQ 6$ ;TYPE DRIVE NO. IF SO
2054 007452 022777 020021 172544 CMP #20021,@RHDT ;IS THIS A SINGLE PORT RP05 ?
2055 007460 001467 BEQ 6$ ;TYPE THE NO. IF SO
2056
2057 007462 022777 024022 172534 CMP #24022,@RHDT ;IS THIS A DUAL PORT RP06 ?
2058 007470 001463 BEQ 6$ ;TYPE DRIVE NO. IF SO
2059 007472 022777 020022 172524 CMP #20022,@RHDT ;IS THIS A SINGLE PORT RP06 ?
2060 007500 001457 BEQ 6$ ;TYPE DRIVE NO. IF SO
2061
2062
2063 007502 104401 007510 TYPE ,65$ ;:TYPE ASCIZ STRING
2064 007506 000410 BR 64$ ;:GET OVER THE ASCIZ
2065 007530 010546 MOV R5,-(SP) ;GET READY TO TYPE UNIT NUMBER
2066 007532 104405 TYPDS
2067 007534 104401 007542 TYPE ,67$ ;:TYPE ASCIZ STRING
2068 007540 000406 BR 66$ ;:GET OVER THE ASCIZ
2069 007556 017746 172442 MOV @RHDT,-(SP) ;GET READY TO TYPE RHDT
2070 007562 104402 TYPOC
2071 007564 104401 007572 TYPE ,69$ ;:TYPE ASCIZ STRING
2072 007570 000422 BR 68$ ;:GET OVER THE ASCIZ
2073 007636 000407 BR 5$ ;NO RP04/RP05/RP06 FOUND SO INCR TABLE
2074
2075 007640 010523 6$: MOV R5,(R3)+
2076 007642 104401 001223 TYPE ,5CRLF
2077 007646 010546 MOV R5,-(SP)
2078 007650 104405 TYPDS
2079 007652 005237 004620 INC @#NUNIT ;TYPE DRIVE NO.
;NUMBER OF DRIVES
2080
2081 007656 005205 5$: INC R5 ;INCR UNIT NUMBER
2082 007660 005300 DEC R0 ;DECR NO. OF UNITS LOOKED AT
2083 007662 001252 BNE 4$ ;TEST THE NEXT UNIT
2084
2085 007664 013737 004576 004616 MOV @#UNITS,@#UNIT ;SET UNIT NO. TO FIRST ONE FOUND/OR 0
2086 007672 013737 004620 004622 MOV @#NUNIT,@#NUNIT ;SAVE NO. OF UNITS
2087 007700 005337 004622 DEC @#NUNIT ;IF NUNIT = 0 THEN ONLY ONE UNIT
;IF NUNIT > 0 THEN MORE THAN ONE UNIT
2088
2089
2090 007704 005737 004626 SELTST: TST @#SELECT ;STARTING ADDRESS 200 ?
2091 007710 001403 BEQ TST4 ;BRANCH IF STARTING FROM 200
2092 007712 013737 004630 004616 MOV @#UNITSL,@#UNIT ;SET UNIT NUMBER
    
```



```

2093
2094
2095 007720 000004          TST4: SCOPE
2096 007722 012737 000001 001212  MOV    #1,$TIMES      ;;DO 1 ITERATION
2097 007730 012737 010622 001106  MOV    #1$,$LPADR     ;;SET SCOPE LOOP ADDRESS
2098
2099 007736 012706 001000          MOV    #STACK,SP      ;RESET STACK
2100 007742 012737 000004 004504  MOV    #4,@#TSTNM     ;SAVE TEST NUMBER
2101 007750 004737 033072          JSR    PC,@#CLDISK    ;SET R1-RHCS1, R2-RHCS2
2102 007754 005037 004644          CLR    @#ATTENT       ;CLEAR UNIT UNDER TEST ATTENTION
2103
2104 007760 005737 004616          TST    @#UNIT         ;IS THE 'UNIT' = 0 ?
2105 007764 001107          BNE    20$           ;IF NOT, SKIP NEXT MODS
2106 007766 012700 000041          MOV    #41,R0        ;IF SO, CHECK THE LOAD MEDIA LOCATION
2107 007772 122710 000011          CMPB   #11,(R0)      ;WAS IT AN RP04/5/6 ?
2108 007776 001102          BNE    20$           ;IF NOT, GO AHEAD AND TEST UNIT #0
2109 010000 005737 004626          TST    @#SELECT       ;WAS UNIT #0 SELECTED ?
2110                          ;(IE. 210 START ?)
2111 010004 001006          BNE    19$           ;IF SO, CHANGE PACK
2112
2113                          ;*INCREMENT THE UNITS TABLE TO NEXT DRIVE (IF ANY)
2114                          ;*& DECREMENT 'NOUNITS' PRESENT TO BE TESTED
2115
2116 010006 012700 004576          MOV    #UNITS,R0     ;IF NOT, LOAD THE UNITS TABLE POINTER
2117 010012 005720          TST    (R0)+         ;SELECT THE NEXT UNIT IN TABLE
2118                          ;(DOUBLE INCREMENT THE POINTER, R0)
2119 010014 022710 177777          CMP    #-1,(R0)     ;IS THERE ANOTHER TABLE ENTRY PRESENT ?
2120 010020 001065          BNE    18$           ;IF SO, USE NEXT DRIVE & DECR 'NOUNITS'
2121                          ;IF NOT, CHANGE PACK ON UNIT #0
2122
2123                          19$:
2124 010022 104401 010030          TYPE   ,65$         ;;TYPE ASCIZ STRING
2125 010026 000434          BR     64$           ;;GET OVER THE ASCIZ
2126 010120 104401 010126          TYPE   ,67$         ;;TYPE ASCIZ STRING
2127 010124 000421          BR     66$           ;;GET OVER THE ASCIZ
2128 010170 000000          HALT
2129 010172 000404          BR     20$           ;CONTINUE, USING SCRATCH PACK ON UNIT #0
2130
2131 010174 011037 004616          18$: MOV    (R0),@#UNIT ;SET UP NEW UNIT UNDER TEST
2132 010200 005337 004620          DEC    @#NOUNITS     ;DECR BECAUSE UNIT #0 WON'T BE TESTED
2133
2134 010204 013700 004616          20$: MOV    @#UNIT,R0 ;R0 NOW CONTAINS UNIT NO
2135
2136
2137
2138
2139 010210 005037 004636          CLR    @#RP06        ;CLEAR RP06 DEVICE TYPE FLAG
2140 010214 010077 171756          MOV    R0,@#RHCS2    ;SET UP UNIT ADDRESSING
2141 010220 022777 024022 171776          CMP    #24022,@#RHDT ;DUAL PORT RP06 ?
2142 010226 001405          BEQ    2$            ;YES..SET FLAG
2143 010230 022777 020022 171766          CMP    #20022,@#RHDT ;SINGLE PORT RP06 ?
2144 010236 001401          BEQ    2$            ;YES...SET FLAG
2145 010240 000403          BR     3$            ;DON'T SET RP06 FLAG
2146 010242 012737 177777 004636 2$: MOV    #-1,@#RP06    ;SET THE FLAG
2147
2148 010250          3$:                ;ASSUME THE NEXT UNIT IS AN RP04/RP05

```



```
2149
2150
2151 010250 116037 004566 004644      MOVB  ATABLE(RO),@#ATTENT ;SET APPROPRIATE ATTENTION BIT
2152 010256 104401 010264              TYPE  ,69$                ;;TYPE ASCIZ STRING
2153 010262 000414                      BR    68$                ;;GET OVER THE ASCIZ
2154 010314 013746 004616      MOV   @#UNIT,-(SP)      ;UNIT NO. TO STACK
2155 010320 104405                      TYPDS ;TYPE DRIVE NO.
2156 010322 104401 010330      TYPE  ,71$                ;;TYPE ASCIZ STRING
2157 010326 000410                      BR    70$                ;;GET OVER THE ASCIZ
2158 010350 017746 171652      MOV   @RHSN,-(SP)     ;;SAVE @RHSN FOR TYPEOUT
2159 010354 104402                      TYPOC ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
2160 010356 104401 010364      TYPE  ,73$                ;;TYPE ASCIZ STRING
2161 010362 000410                      BR    72$                ;;GET OVER THE ASCIZ
2162 010404 017746 171614      MOV   @RHDT,-(SP)     ;;SAVE @RHDT FOR TYPEOUT
2163 010410 104402                      TYPOC ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
```



```

2164
2165
2166
2167
2168 010412 022777 024020 171604      CMP      #24020,@RHDT      ;DUAL PORT RP04 ?
2169 010420 001424                      BEQ      4$           ;TYPE ASCII MESSAGE OUT
2170 010422 022777 020020 171574      CMP      #20020,@RHDT ;SINGLE PORT RP04 ?
2171 010430 001420                      BEQ      4$           ;TYPE THE MESSAGE
2172
2173 010432 022777 024021 171564      CMP      #24021,@RHDT ;DUAL PORT RP05 ?
2174 010440 001433                      BEQ      5$           ;TYPE THE MESSAGE
2175 010442 022777 020021 171554      CMP      #20021,@RHDT ;SINGLE PORT RP05 ?
2176 010450 001427                      BEQ      5$           ;TYPE THE MESSAGE
2177
2178 010452 022777 024022 171544      CMP      #24022,@RHDT ;DUAL PORT RP06 ?
2179 010460 001442                      BEQ      6$           ;TYPE THE MESSAGE
2180 010462 022777 020022 171534      CMP      #20022,@RHDT ;SINGLE PORT RP06 ?
2181 010470 001436                      BEQ      6$           ;TYPE THE MESSAGE
2182
2183 010472                      4$:
2184 010472 104401 010500      TYPE      ,75$        ;;TYPE ASCIZ STRING
2185 010476 000413      BR        74$        ;;GET OVER THE ASCIZ
2186 010526 000435      BR        1$         ;SKIP NEXT ONES
2187 010530                      5$:
2188 010530 104401 010536      TYPE      ,77$        ;;TYPE ASCIZ STRING
2189 010534 000413      BR        76$        ;;GET OVER THE ASCIZ
2190 010564 000416      BR        1$         ;SKIP NEXT
2191 010566                      6$:
2192 010566 104401 010574      TYPE      ,79$        ;;TYPE ASCIZ STRING
2193 010572 000413      BR        78$        ;;GET OVER THE ASCIZ
2194
2195
2196
2197
2198 010622 005777 171400      1$:      TST      @RHSN      ;READ SERIAL NO. AND DRIVE TYPE
2199 010626 005777 171372      TST      @RHDT      ;THESE TWO ARE TO HELP SCOPE LOOPS
2200 010632 017737 171370 002310      MOV      @RHSN,@#SN ;SAVE TO CHECK IF DRIVE CLEAR CLEARS ANY BITS
2201 010640 017737 171360 002306      MOV      @RHDT,@#DT ;SAVE TO CHECK IF DRIVE CLEAR CLEARS ANY BITS
  
```



```
2202
2203 010646 000004
2204 010650 012737 000005 004504 TST5: SCOPE
2205
2206
2207
2208 010656 004737 033072 JSR PC,@#CLDISK ;GIVE INITILIZE
2209 010662 032713 010000 BIT #MOL,@R3 ;CHECK MOL IN RHDS1
2210 010666 001144 BNE TST6 ;BRANCH IF MOL HIGH
2211 010670 104401 010676 TYPE ,65$ ;;TYPE ASCIZ STRING
2212 010674 000420 BR 64$ ;;GET OVER THE ASCIZ
2213 010736 104401 010744 TYPE ,67$ ;;TYPE ASCIZ STRING
2214 010742 000424 BR 66$ ;;GET OVER THE ASCIZ
2215 011014 104401 011022 TYPE ,69$ ;;TYPE ASCIZ STRING
2216 011020 000431 BR 68$ ;;GET OVER THE ASCIZ
2217 011104 032713 010000 1$: BIT #MOL,@R3 ;CHECK MOL IN RHDS1
2218 011110 001775 BEQ 1$ ;WAIT IF MOL IS STILL LOW
2219 011112 104401 011120 TYPE ,71$ ;;TYPE ASCIZ STRING
2220 011116 000430 BR 70$ ;;GET OVER THE ASCIZ
2221
2222
2223
```



```

2224
2225 011200 000004          TST6:  SCOPE
2226
2227 011202 012737 000006 004504    MOV  #7-1,@#TSTNM    ;THIS SAVES TEST NUMBER
2228 011210 012706 001000          MOV  #STACK,SP      ;RESET STACK
2229 011214 004737 033072          JSR  PC,@#CLDISK    ;SET R1-RHCS1, R2-RHCS2
2230
2231 011220 013700 002166          MOV  @#RPVEC,R0     ;GET RP VECTOR ADDRESS
2232 011224 012720 011272          MOV  #RPTRP1,(R0)+ ;THIS IS FOR TIMELY INTERRUPTS
2233 011230 012710 000340          MOV  #340,(R0)     ;RP04 INTERRUPT SERVICE ROUTINE
2234                          ;PRIORITY = 7
2235 011234 012737 000200 177776    MOV  #200,PS        ;SET PROCESSOR PRIORITY @ 4
2236 011242 012711 000300          MOV  #RDY!IE,@R1   ;RDY, IE IN RHSC1 SHOULD CAUSE INTERRUPT
2237
2238 011246 013737 033570 001200    MOV  @#TIMCNT,@#$TMP1;COUNTER
2239 011254 005337 001200    1$:  DEC  @#$TMP1      ;WAIT FOR INTERRUPT
2240 011260 001375          BNE  1$            ;BRANCH IF NOT ZERO
2241                          ;BEFORE THIS IS ZERO INTERRUPT SHOULD OCCUR
2242 011262 104065          ERROR 65          ;INTERRUPT DID NOT OCCUR
2243 011264 012712 000040          MOV  #CLR,@R2      ;CLEAR CONTROLLER VIA CS2
2244 011270 000407          BR   TST7          ;BRANCH TO NEXT TEST -----)
2245
2246 011272 022626          RPTRP1: CMP  (SP)+,(SP)+   ;RESTORE STACK
2247 011274 022711 004200          CMP  #DVA!RDY,@R1 ;IE SHOULD BE LOW
2248 011300 001403          BEQ  TST7          ;CONTINUE IF GOOD -----)
2249 011302 104065          ERROR 65          ;INTERRUPT OCCURED BUT
2250                          ;IE FAILED TO RESET
2251 011304 012712 000040          MOV  #CLR,@R2     ;CLEAR CONTROLLER
    
```



```

2252
2253
2254 011310 000004          TST7:  SCOPE
2255
2256 011312 012737 000007 004504      MOV  #10-1,@#TSTNM  ;THIS SAVES TEST NUMBER
2257 011320 012706 001000              MOV  #STACK,SP      ;RESET STACK
2258 011324 004737 033072              JSR  PC,@#CLDISK   ;SET R1-RHCS1, R2-RHCS2
2259
2260 011330 013700 002166              MOV  @#RPVEC,R0    ;GET RP VECTOR ADDRESS
2261 011334 012720 011400              MOV  #RPTRP2,(R0)+;THIS IS FOR UNTIMELY INTERRUPTS
2262 011340 012710 000340              MOV  #340,(R0)    ;RPO4 INTERRUPT SERVICE ROUTINE
2263                                     ;PRIORITY = 7
2264 011344 012737 000240 177776      MOV  #240,PS      ;SET PROCESSOR PRIORITY = 5
2265 011352 012711 000300              MOV  #RDY!IE,@R1  ;RDY, IE IN RHSC1 WHOULD CAUSE INTERRUPT
2266
2267 011356 013737 033570 001200      MOV  @#TIMCNT,@#$TMP1;COUNTER
2268 011364 005337 001200      1$:  DEC  @#$TMP1    ;WAIT FOR INTERRUPT
2269 011370 001375              BNE  1$          ;BRANCH IF NOT ZERO
2270                                     ;BEFORE THIS IS ZERO INTERRUPT WHOULD OCCUR
2271 011372 012712 000040              MOV  #CLR,@R2     ;CLEAR CONTROLLER
2272 011376 000404              BR   TST10       ;NO INTERRUPT SO CONTINUE -----)
2273
2274 011400 022626          RPTRP2: CMP  (SP)+,(SP)+ ;RESTORE STACK
2275 011402 104065          ERROR  65        ;INTERRUPT OCCURRED WITH
2276                                     ;PROCESSOR PRIORITY SAME AS DISK
2277 011404 012712 000040              MOV  #CLR,@R2     ;CLEAR CONTROLLER
2278
2279
2280
2281
  
```

2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281


```

2282
2283 011410 000004          TST10: SCOPE
2284
2285                      ;*IN CASE THERE IS ANY DRIVE ERRORS DURING POWER UP
2286                      ;*OR POWER DOWN OR ANY PARITY ERRORS A RESET IS GIVEN
2287 011412 000005          RESET
2288 011414 004737 037340  JSR    PC,@#STKINT      ;INITILIZE TK
2289 011420 012737 000000 177776  MOV    #0,PS
2290
2291 011426 012706 001000  MOV    #STACK,SP      ;RESET STACK
2292 011432 012737 000010 004504  MOV    #10,@#TSTNM    ;SAVE TEST NUMBER
2293 011440 004737 033072  JSR    PC,@#CLDISK    ;SET R1-RHCS1, R2-RHCS2
2294 011444 004737 033130  JSR    PC,@#CHECK     ;CHECK THAT DVA,RDY,MOL,DPR,DRY = 1
2295 011450 104401 057176  TYPE   ,CPHALT       ;CANNOT CONTINUE TESTS IF THEY DON'T
2296 011454 000000  HALT
2297 011456 013777 002360 170514  MOV    @#PKACK,@RHCS1 ;GET READY FOR PKACK
2298
2299                      ;*NOW SAVE REGISTERS FOR COMPARISON AFTER PACK ACKNOWLEDGE
2300
2301 011464 004037 033240  JSR    R0,@#SAVER     ;SAVE REGISTERS
2302 011470 002172  RHWC
2303 011472 004512  SAVERE                ;RHWC IS THE FIRST REGISTER SAVED
2304 011474 000022  18.                  ;STARTING ADDRES OF WHERE
2305                      ;NUMBER OF REGISTERS
2306 011476 013777 004506 170462  MOV    @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
2307
2308 011504 013746 002360  MOV    @#PKACK,-(SP)  ;GET READY TO MOVE COMMAND
2309 011510 052716 000001  BIS    #GO,(SP)      ;GET READY TO SET GO
2310 011514 012677 170460  MOV    (SP)+,@RHCS1  ;GO WITH
2311 011520 011100  MOV    @R1,R0        ;SAVE RHCS1 DURING ABOVE OPERATION
2312 011522 011305  MOV    @R3,R5        ;SAVE RHDS1 DURING ABOVE OPERATION
2313
2314 011524 104413  WAT
2315 011526 002222  RHDS1                ;WAIT FOR VV BIT TO SET
2316 011530 000100  VV                  ;WAIT FOR RHDS1 REGISTER
2317 011532 000001  1.                  ;WAIT FOR VV BIT IN RHDS1 REGISTER
2318 011534 000001  1.                  ;ALLOW 10 MICRO SECONDS
2319 011536 013746 002360  MOV    @#PKACK,-(SP)  ;VV MUST SET BETWEEN
2320 011542 052716 004200  BIS    #DVA!RDY,(SP) ;SAVE COMMAND
2321 011546 011637 001124  MOV    (SP),@#$GDDAT ;INCLUDE DVA!RDY
2322 011552 022600  CMP    (SP)+,R0      ;SAVE FOR PRINTOUT
2323 011554 001405  BEQ    64$           ;DURING ABOVE OPERATION ONLY DVA!RDY
2324 011556 010037 001126  MOV    R0,@#$BDDAT   ;BRANCH IF GOOD
2325 011562 010137 004500  MOV    R1,@#REGADR   ;BAD DATA
2326 011566 104021  ERROR 21            ;FAILING REGISTER RHCS1
2327 011570 012746 010700 64$: MOV    #MOL!DPR!DRY!VV,-(SP) ;DURING ABOVE OPERATION ONLY
2328 011574 011637 001124  MOV    (SP),@#$GDDAT ;SAVE BITS SET DURING OPERATION IN RHDS1
2329 011600 022605  CMP    (SP)+,R5      ;SAVE FOR PRINTOUT
2330 011602 001405  BEQ    66$           ;DURING ABOVE OPERATION ONLY MOL!DPR!DRY!VV
2331 011604 010537 001126  MOV    R5,@#$BDDAT   ;BRANCH IF GOOD
2332 011610 010337 004500  MOV    R3,@#REGADR   ;BAD DATA
2333 011614 104063  ERROR 63            ;FAILING REGISTER RHDS1
2334 011616 004037 033762  JSR    R0,@#CHREG    ;DURING ABOVE OPERATION ONLY
2335 011622 002222  RHDS1                ;CHANGE BITS IN SAVED REGISTER
2336 011624 000001  1                    ;CHANGE RHDS1 REGISTER
2337 011626 000001  1                    ;1 BIT/BITS TO BE CHANGED
                      ;NEW VALUE OF VV IS 1
    
```



```
2338 011630 000100          VV          ;CHANGE VV BIT
2339
2340          ;*NOW COMPARE REGISTERS BEFORE PACK ACKNOWLEDGE
2341          ;*WITH AFTER PACK ACKNOWLEDGE
2342
2343
2344 011632 004037 034070    JSR      R0,@#COMREG    ;COMPARE SAVED REGISTERS WITH
2345 011636 004512          SAVERE    ;GOOD DATA SAVED IN 'SAVERE'
2346 011640 002254          WC          ;TEST DATA STARTING FROM 'RHWC'
2347 011642 000022          18.        ;18. REGISTERS TO BE COMPARED
2348 011644 011650          1$          ;RETURN TO 1$ ON ERROR
2349 011646 011654          2$          ;RETURN TO 2$ ON NO ERROR
2350
2351 011650 104015          1$:      ERROR 15      ;GIVING A PACK ACKNOWLEDGE
2352 011652 000207          RTS      PC      ;CAUSED AN ERROR
2353          ;PACK ACKNOWLEDGE SHOULD
2354          ;SET VV IN RHDS1
2355          ;INTERRUPT SHOULD MAKE
2356          ;IE = 0
2357          ;NO OTHER REGISTERS SHOULD
2358          ;CHANGE
2359          ;GOOD DATA GIVES CONTENTS
2360          ;OF REGISTER BEFORE COMMAND
2361          ;RECEIVED DATA GIVES CONTENTS
2362          ;OF REGISTER AFTER COMMAND
2363 011654          2$:
2364
2365
2366
2367
```


.SBTTL DATA TRANSFER RELATED ERRORS (USING MEDIA)

```
2368
2369
2370
2371 011654 000004          TST11: SCOPE
2372 011656 012706 001000  MOV      #STACK,SP      ;RESET STACK
2373 011662 012737 000011 004504  MOV      #11,@#TSTNM    ;SAVE TEST NUMBER
2374 011670 004737 033072  JSR      PC,@#CLDISK    ;SET R1-RHCS1, R2-RHCS2
2375
2376      ;*CHECK THE DRIVE TYPE AND THEN FILL THE
2377      ;*WRITE FROM BUFFER WITH APPROPRIATE HEADER
2378 011674 005737 004636  TST      @#RP06 ;TEST FOR RP06 DRIVE
2379 011700 001412          BEQ      11$           ;TREAT UNIT AS AN RP04
2380                          ;TREAT UNIT AS AN RP06
2381
2382 011702 004037 032716  JSR      R0,@#FLHEAD    ;SAVE HEADER DATA IN WRFROM
2383 011706 002370          WRFROM                ;LOCATION WHERE SAVED
2384 011710 000005          5                     ;NUMBER OF WORDS SAVED
2385 011712 011456          11456                ;FIRST DATA WORD
2386 011714 011025          <18.*400>!<21.>      ;SECOND DATA WORD
2387 011716 000000          0                     ;THIRD DATA WORD
2388 011720 000000          0                     ;FOURTH DATA WORD
2389 011722 065125          <26.*2000>!<18.*40>!<21.> ;FIFTH DATA WORD
2390 011724 000411          BR       12$           ;CONTINUE WITH SET UP
2391
2392 011726          11$:
2393 011726 004037 032716  JSR      R0,@#FLHEAD    ;SAVE HEADER DATA IN WRFROM
2394 011732 002370          WRFROM                ;LOCATION WHERE SAVED
2395 011734 000005          5                     ;NUMBER OF WORDS SAVED
2396 011736 010632          10632                ;FIRST DATA WORD
2397 011740 011025          <18.*400>!<21.>      ;SECOND DATA WORD
2398 011742 000000          0                     ;THIRD DATA WORD
2399 011744 000000          0                     ;FOURTH DATA WORD
2400 011746 065125          <26.*2000>!<18.*40>!<21.> ;FIFTH DATA WORD
2401 011750          12$:
2402
2403      ;*FILL READ INTO BUFFER WITH ALL ONES
2404
2405 011750 004037 032742  JSR      R0,@#CLAREA    ;CLEAR 256 WORDS, FROM REINTO
2406 011754 003434          REINTO                ;STARTING FROM REINTO
2407 011756 000256          256                  ;256 WORDS
2408 011760 177777          -1                    ;FILL WITH -1
2409
```



```

2410
2411
2412
2413
2414 011762 005737 004636
2415 011766 001412
2416
2417 011770 004037 035054
2418 011774 001456
2419 011776 025
2420 011777 022
2421 012000 177773
2422 012002 002370
2423 012004 000000
2424 012006 010000
2425 012010 002344
2426 012012 000411
2427
2428 012014
2429 012014 004037 035054
2430 012020 000632
2431 012022 025
2432 012023 022
2433 012024 177773
2434 012026 002370
2435 012030 000000
2436 012032 010000
2437 012034 002344
2438 012036
2439
2440
2441
2442
2443 012036 004037 033240
2444 012042 002172
2445 012044 004512
2446 012046 000022
2447
2448 012050 004737 033152
2449 012054 104401 057176
2450 012060 000000
2451 012062 013777 004506 170076
2452 012070 013746 002344
2453 012074 052716 000101
2454 012100 012677 170074
2455
2456
2457 012104 104413
2458 012106 002222
2459 012110 002000
2460 012112 004704
2461 012114 004704
2462
2463
2464 012116 004037 032774
2465 012122 002172

```

```

;*DRIVE TYPE IS CHECKED AND THEN THE APPROPRIATE
;*WRITE HEADER AND DATA COMMAND IS LOADED

TST @#RP06 ;TEST FOR RP06 DRIVE
BEQ 7$ ;TREAT UNIT AS RP04

JSR R0,@#RUN ;SETUP TO RUN FOR DATA COMMAND
814. ;CYLINDER 814.
.BYTE 21. ;SECTOR 21.
.BYTE 18. ;TRACK 18.
-1-4 ;WORD COUNT (DATA) = 1 +
WRFROM ;BUS ADDRESS
0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
FMT22 ;16 BITS PER WORD FORMAT
WRIFOR ;GET READY TO DO A WRIFOR
BR 8$ ;CONTINUE WITH TEST

7$:
JSR R0,@#RUN ;SETUP TO RUN FOR DATA COMMAND
410. ;CYLINDER 410.
.BYTE 21. ;SECTOR 21.
.BYTE 18. ;TRACK 18.
-1-4 ;WORD COUNT (DATA) = 1 +
WRFROM ;BUS ADDRESS
0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
FMT22 ;16 BITS PER WORD FORMAT
WRIFOR ;GET READY TO DO A WRIFOR

8$:
;*NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE

JSR R0,@#SAVER ;SAVE REGISTERS
RHWC ;RHWC IS THE FIRST REGISTER SAVED
SAVERE ;STARTING ADDRES OF WHERE
18. ;NUMBER OF REGISTERS

JSR PC,@#CHECKT ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF NOT
HALT ;STOP TEST
MOV @#RP4VEC,@#RVEC ;SET RP04 VECTOR ADDRESS
MOV @#WRIFOR,-(SP) ;GET READY TO MOVE COMMAND
BIS #GO!IE,(SP) ;GET READY TO SET GO AND
MOV (SP)+,@#RHCS1 ;GO WITH

;*TIME IS NOT CRITICAL HERE
WAT ;WAIT FOR LBT BIT TO SET
RHDS1 ;WAIT FOR RHDS1 REGISTER
LBT ;WAIT FOR LBT BIT IN RHDS1 REGISTER
2500. ;ALLOW 25000 MICRO SECONDS
2500. ;LBT MUST SET BETWEEN

;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
JSR R0,@#FILLRE ;MOV 0 INTO SAVED RHWC
RHWC ;SAVED REGISTER TO CHANGE

```



```

2466 012124 000000 0 ;DATA
2467 012126 004037 032774 JSR RO,@#FILLRE ;MOV WRFROM+<5*2> INTO SAVED RHBA
2468 012132 002174 RHBA ;SAVED REGISTER TO CHANGE
2469 012134 002402 WRFROM+<5*2> ;DATA
2470 012136 004037 033762 JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
2471 012142 002222 RHDS1 ;CHANGE RHDS1 REGISTER
2472 012144 000001 1 ;1 BIT/BITS TO BE CHANGED
2473 012146 000001 1 ;NEW VALUE OF LBT IS 1
2474 012150 002000 LBT ;CHANGE LBT BIT
2475 012152 004037 033762 JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
2476 012156 002212 RHCA ;CHANGE RHCA REGISTER
2477 012160 000001 1 ;1 BIT/BITS TO BE CHANGED
2478 012162 000001 1 ;NEW VALUE OF BIT0 IS 1
2479 012164 000001 BIT0 ;CHANGE BIT0 BIT
2480 012166 004037 032774 JSR RO,@#FILLRE ;MOV 0 INTO SAVED RHDST
2481 012172 002204 RHDST ;SAVED REGISTER TO CHANGE
2482 012174 000000 0 ;DATA
2483
2484 ;*COMPARE ALL REGISTERS
2485 012176 004037 034070 JSR RO,@#COMREG ;COMPARE SAVED REGISTERS WITH
2486 012202 004512 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
2487 012204 002254 WC ;TEST DATA STARTING FROM 'RHWC'
2488 012206 000021 17. ;17. REGISTERS TO BE COMPARED
2489 012210 012214 1$ ;RETURN TO 1$ ON ERROR
2490 012212 012220 2$ ;RETURN TO 2$ ON NO ERROR
2491
2492 012214 104045 1$: ERROR 45 ;WRITING ON THE LAST BLOCK
2493 012216 000207 RTS PC ;IE. CYLINDER 410./814., SECTOR 21
2494 ;TRACK 18 CAUSED
2495 ;IMPROPER REGISTER CHANGE
2496 ;GOOD DATA GIVES WHAT
2497 ;SHOULD BE THERE
2498 ;RECEIVED DATA GIVES WHAT
2499 ;WAS THERE AFTER WRITE
2500 ;ON LAST BLOCK
    
```



```

2501
2502
2503                ;*NOW A READ DATA WILL BE DONE ON SAME CYLINDER, SECTOR & TRACK
2504
2505                ;*CLEAR ANY PREVIOUS ERRORS
2506 012220          2$: JSR    PC,@#CLDISK    ;SET R1-RHCS1, R2-RHCS2
2507 012220 004737 033072
2508
2509                ;*FILL WRITE FROM BUFFER WITH EXPECTED DATA
2510 012224 004037 032716 JSR    RO,@#FLHEAD    ;SAVE HEADER DATA IN WRFROM
2511 012230 002370 WRFROM    ;LOCATION WHERE SAVED
2512 012232 000001 1 ;NUMBER OF WORDS SAVED
2513 012234 065125 <26.*2000>!<18.*40>!<21.> ;FIRST DATA WORD
2514 012236 004037 032742 JSR    RO,@#CLAREA    ;CLEAR 256. WORDS, FROM WRFROM+2
2515 012242 002372 WRFROM+2 ;STARTING FROM WRFROM+2
2516 012244 000400 256. ;256. WORDS
2517 012246 000000 0 ;FILL WITH 0
2518
2519                ;*FIRST THE DRIVE TYPE IS CHECKED AND THEN THE APPROPRIATE
2520                ;*READ COMMAND IS LOADED
2521
2522 012250 005737 004636 TST    @#RP06 ;TEST FOR RP06 DRIVE
2523 012254 001412 BEQ    9$ ;TREAT UNIT AS RP04
2524
2525 012256 004037 035054 JSR    RO,@#RUN ;SETUP TO RUN FOR DATA COMMAND
2526 012262 001456 814. ;CYLINDER 814.
2527 012264 025 .BYTE 21. ;SECTOR 21.
2528 012265 022 .BYTE 18. ;TRACK 18.
2529 012266 177400 -256. ;WORD COUNT = 256.
2530 012270 003434 REINTO ;BUS ADDRESS
2531 012272 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
2532 012274 010000 FMT22 ;16 BITS PER WORD FORMAT
2533 012276 002346 READAT ;GET READY TO DO A READAT
2534 012300 000411 BR 10$ ;CONTINUE WITH TEST
2535
2536 012302          9$: JSR    RO,@#RUN ;SETUP TO RUN FOR DATA COMMAND
2537 012302 004037 035054 410. ;CYLINDER 410.
2538 012306 000632 .BYTE 21. ;SECTOR 21.
2539 012310 025 .BYTE 18. ;TRACK 18.
2540 012311 022 -256. ;WORD COUNT = 256.
2541 012312 177400 REINTO ;BUS ADDRESS
2542 012314 003434 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
2543 012316 000000 FMT22 ;16 BITS PER WORD FORMAT
2544 012320 010000 READAT ;GET READY TO DO A READAT
2545 012322 002346
2546 012324          10$:
2547
2548                ;*SAVE REGISTERS FOR COMPARISON AFTER READ DATA
2549 012324 004037 033240 JSR    RO,@#SAVER ;SAVE REGISTERS
2550 012330 002172 RHWC ;RHWC IS THE FIRST REGISTER SAVED
2551 012332 004512 SAVERE ;STARTING ADDRESS OF WHERE
2552 012334 000022 18. ;NUMBER OF REGISTERS
2553 012336 004737 033152 JSR    PC,@#CHECKT ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
2554 012342 104401 057176 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF NOT
2555 012346 000000 HALT ;STCP TEST
2556 012350 013777 004506 167610 MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
    
```



```

2557 012356 013746 002346      MOV    @#READAT,-(SP)  ;GET READY TO MOVE COMMAND
2558 012362 052716 000101      BIS    #GO!IE,(SP)    ;GET READY TO SET GO AND
2559 012366 012677 167606      MOV    (SP)+,@RHCS1   ;GO WITH
2560
2561                               ;*TIME IS NOT CRITICAL HERE
2562 012372 104413      WAT    ;WAIT FOR RDY BIT TO SET
2563 012374 002200      RHCS1  ;WAIT FOR RHCS1 REGISTER
2564 012376 000200      RDY    ;WAIT FOR RDY BIT IN RHCS1 REGISTER
2565 012400 001614      908.   ;ALLOW 9080 MICRO SECONDS
2566 012402 001502      834.   ;RDY MUST SET BETWEEN
2567
2568                               ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
2569
2570 012404 004037 032774      JSR    RO,@#FILLRE    ;MOV 0 INTO SAVED RHWC
2571 012410 002172      RHWC   ;SAVED REGISTER TO CHANGE
2572 012412 000000      0      ;DATA
2573 012414 004037 032774      JSR    RO,@#FILLRE    ;MOV REINTO+<256.*2> INTO SAVED RHBA
2574 012420 002174      RHBA   ;SAVED REGISTER TO CHANGE
2575 012422 004434      REINTO+<256.*2>      ;DATA
2576 012424 004037 033762      JSR    RO,@#CHREG     ;CHANGE BITS IN SAVED REGISTER
2577 012430 002222      RHDS1  ;CHANGE RHDS1 REGISTER
2578 012432 000001      1      ;1 BIT/BITS TO BE CHANGED
2579 012434 000001      1      ;NEW VALUE OF LBT IS 1
2580 012436 002000      LBT    ;CHANGE LBT BIT
2581 012440 004037 033762      JSR    RO,@#CHREG     ;CHANGE BITS IN SAVED REGISTER
2582 012444 002212      RHCA   ;CHANGE RHCA REGISTER
2583 012446 000001      1      ;1 BIT/BITS TO BE CHANGED
2584 012450 000001      1      ;NEW VALUE OF BIT0 IS 1
2585 012452 000001      BIT0   ;CHANGE BIT0 BIT
2586 012454 004037 032774      JSR    RO,@#FILLRE    ;MOV 0 INTO SAVED RHDST
2587 012460 002204      RHDST  ;SAVED REGISTER TO CHANGE
2588 012462 000000      0      ;DATA
2589
2590                               ;*COMPARE ALL REGISTERS
2591 012464 004037 034070      JSR    RO,@#COMREG    ;COMPARE SAVED REGISTERS WITH
2592 012470 004512      SAVERE ;GOOD DATA SAVED IN 'SAVERE'
2593 012472 002254      WC     ;TEST DATA STARTING FROM 'RHWC'
2594 012474 000022      18.   ;18. REGISTERS TO BE COMPARED
2595 012476 012502      3$    ;RETURN TO 3$ ON ERROR
2596 012500 012506      4$    ;RETURN TO 4$ ON NO ERROR
2597
2598 012502 104045      3$:   ERROR 45      ;READING ON LAST BLOCK IE.
2599 012504 000207      RTS   PC          ;CYLINDER 410./814., SECTOR 21, TRACK 18
2600
2601                               ;CAUSED AN ERROR
2602                               ;GOOD DATA GIVES WHAT SHOULD
2603                               ;BE THERE
2604                               ;RECEIVED DATA GIVES WHAT
2605                               ;WAS THERE AFTER READ
2606                               ;FROM LAST BLOCK
2607
2608 012506      4$:   ;*READ DATA WILL BE COMPARED
2609 012506 004037 035120      JSR    RO,@#COMPAR    ;COMPARE TWO BLOCKS OF MEMORY
2610 012512 002370      WRFROM ;GOOD DATA STARTS FROM WRFROM
2611 012514 003434      REINTO ;TEST DATA STARTS FROM REINTO
2612 012516 000400      256.  ;256., WORDS TO BE COMPARED
  
```



```

2623
2624 012530 000004          TST12: SCOPE
2625 012532 012706 001000  MOV      #STACK,SP      ;RESET STACK
2626 012536 012737 000012 004504  MOV      #12,@#TSTNM    ;SAVE TEST NUMBER
2627 012544 004737 033072  JSR      PC,@#CLDISK    ;SET R1-RHCS1, R2-RHCS2
2628                      ;*GET HEADS TO CYLINDER 0
2629 012550 004737 033152  JSR      PC,@#CHECKT    ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
2630 012554 104401 057176  TYPE     ,CPHALT        ;CANNOT CONTINUE TESTING IF NOT
2631 012560 000000          HALT                    ;STOP TEST
2632
2633 012562 013777 004506 167376  MOV      @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
2634
2635 012570 013746 002326  MOV      @#RECALI,-(SP)  ;GET READY TO MOVE COMMAND
2636 012574 052716 000101  BIS      #GO!IE,(SP)    ;GET READY TO SET GO AND
2637 012600 012677 167374  MOV      (SP)+,@RHCS1   ;GO WITH
2638 012604 104413          WAT                    ;WAIT FOR RDY BIT TO SET
2639 012606 002222          RHDS1                 ;WAIT FOR RHDS1 REGISTER
2640 012610 000200          RDY                   ;WAIT FOR RDY BIT IN RHDS1 REGISTER
2641 012612 012740          5600.                 ;ALLOW 56000 MICRO SECONDS
2642 012614 012737          5599.                 ;RDY MUST SET BETWEEN
2643
2644 012616 004737 033072  JSR      PC,@#CLDISK    ;SET R1-RHCS1, R2-RHCS2
2645
2646                      ;*FILL WRITE FROM BUFFER WITH HEADER
2647 012622 004037 032716  JSR      RO,@#FLHEAD    ;SAVE HEADER DATA IN WRFROM
2648 012626 002370          WRFROM                ;LOCATION WHERE SAVED
2649 012630 000004          4                      ;NUMBER OF WORDS SAVED
2650 012632 010000          10000                 ;FIRST DATA WORD
2651 012634 000000          0                      ;SECOND DATA WORD
2652 012636 000000          0                      ;THIRD DATA WORD
2653 012640 000000          0                      ;FOURTH DATA WORD
2654
2655                      ;*FILL WRITE FROM BUFFER WITH DATA
2656 012642 004037 032742  JSR      RO,@#CLAREA    ;CLEAR 256. WORDS, FROM WRFROM+10
2657 012646 002400          WRFROM+10             ;STARTING FROM WRFROM+10
2658 012650 000400          256.                  ;256. WORDS
2659 012652 000000          0                      ;FILL WITH 0
2660
2661                      ;*FILL WRITE FROM BUFFER WITH NEXT SECTOR HEADER
2662 012654 004037 032716  JSR      RO,@#FLHEAD    ;SAVE HEADER DATA IN WRFROM+<260.*2>
2663 012660 003400          WRFROM+<260.*2>      ;LOCATION WHERE SAVED
2664 012662 000004          4                      ;NUMBER OF WORDS SAVED
2665 012664 010000          10000                 ;FIRST DATA WORD
2666 012666 000001          1                      ;SECOND DATA WORD
2667 012670 000000          0                      ;THIRD DATA WORD
2668 012672 000000          0                      ;FOURTH DATA WORD
2669
2670                      ;*FILL WRITE FROM BUFFER WITH NEXT SECTOR DATA
2671 012674 004037 032742  JSR      RO,@#CLAREA    ;CLEAR 4 WORDS, FROM WRFROM+<264.*2>
2672 012700 003410          WRFROM+<264.*2>      ;STARTING FROM WRFROM+<264.*2>
2673 012702 000004          4                      ;4 WORDS
2674 012704 000001          1                      ;FILL WITH 1
2675
2676                      ;*CLEAR READ INTO BUFFER WITH DATA OTHER THAN EXPECTED DATA
2677 012706 004037 032742  JSR      RO,@#CLAREA    ;CLEAR 260. WORDS, FROM REINTO
2678 012712 003434          REINTO                ;STARTING FROM REINTO
  
```



```

2679 012714 000404          260.          :260. WORDS
2680 012716 000377          377           :FILL WITH 377
2681
2682
2683 012720 004037 035054    ;*THE WRITE HEADER AND DATA WILL BE LOADED
      JSR    RO,@#RUN        :SETUP TO RUN FOR DATA COMMAND
2684 012724 000000          0             :CYLINDER 0
2685 012726 000           .BYTE 0        :SECTOR 0
2686 012727 000           .BYTE 0        :TRACK 0
2687 012730 177364          -264.-4      :WORD COUNT (DATA) = 264. +
2688 012732 002370          WRFROM       :BUS ADDRESS
2689 012734 000000          0            :DO NOT INHIBIT BUS ADDRESS INCREMENT
2690 012736 010000          FMT22        :16 BITS PER WORD FORMAT
2691 012740 002344          WRIFOR       :GET READY TO DO A WRIFOR
2692
2693
2694 012742 004037 033240    ;*SAVE REGISTERS FOR COMPARISON AFTER WRITE HEADER AND DATA
      JSR    RO,@#SAVER     :SAVE REGISTERS
2695 012746 002172          RHWC         :RHWC IS THE FIRST REGISTER SAVED
2696 012750 004512          SAVERE       :STARTING ADDRESS OF WHERE
2697 012752 000022          18.         :NUMBER OF REGISTERS
2698
2699 012754 004737 033152    JSR    PC,@#CHECKT      :CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
2700 012760 104401 057176    TYPE    ,CPHALT        :CANNOT CONTINUE TESTING IF NOT
2701 012764 000000          HALT           :STOP TEST
2702
2703 012766 013777 004506 167172  MOV    @#RP4VEC,@RPVEC  :SET RP04 VECTOR ADDRESS
2704
2705 012774 013746 002344    MOV    @#WRIFOR,-(SP)   :GET READY TO MOVE COMMAND
2706 013000 052716 000101    BIS    #GO!IE,(SP)     :GET READY TO SET GO AND
2707 013004 012677 167170    MOV    (SP)+,@RHCS1    :GO WITH
2708 013010 011100          MOV    @R1,R0          :SAVE RHCS1 DURING ABOVE OPERATION
2709 013012 011305          MOV    @R3,R5          :SAVE RHDS1 DURING ABOVE OPERATION
2710
2711
2712 013014 104413          ;*ONE REVOLUTION=16670 MICRO SEC, ONE SECTOR = 760 MICRO SEC
      WAT                    :WAIT FOR RDY BIT TO SET
2713 013016 002200          RHCS1         :WAIT FOR RHCS1 REGISTER
2714 013020 000200          RDY           :WAIT FOR RDY BIT IN RHCS1 REGISTER
2715 013022 001614          908.         :ALLOW 9080 MICRO SECONDS
2716 013024 001507          839.         :RDY MUST SET BETWEEN
2717 013026 013746 002344    MOV    @#WRIFOR,-(SP)   :SAVE COMMAND
2718 013032 052716 004101    BIS    #IE!DVA!GO,(SP) :INCLUDE IE!DVA!GO
2719 013036 011637 001124    MOV    (SP),@#$GDDAT   :SAVE FOR PRINTOUT
2720 013042 022600          CMP    (SP)+,R0        :DURING ABOVE OPERATION ONLY IE!DVA!GO
2721 013044 001405          BEQ    67$           :BRANCH IF GOOD
2722 013046 010037 001126    MOV    R0,@#$BDDAT     :BAD DATA
2723 013052 010137 004500    MOV    R1,@#REGADR     :FAILING REGISTER RHCS1
2724 013056 104021          ERROR 21         :DURING ABOVE OPERATION ONLY
2725 013060 012746 010500    MOV    #MOL!DPR!VV,-(SP) :SAVE BITS SET DURING OPERATION IN RHDS1
2726 013064 011637 001124    MOV    (SP),@#$GDDAT   :SAVE FOR PRINTOUT
2727 013070 022605          CMP    (SP)+,R5        :DURING ABOVE OPERATION ONLY MOL!DPR!VV
2728 013072 001405          BEQ    69$           :BRANCH IF GOOD
2729 013074 010537 001126    MOV    R5,@#$BDDAT     :BAD DATA
2730 013100 010337 004500    MOV    R3,@#REGADR     :FAILING REGISTER RHDS1
2731 013104 104063          ERROR 63         :DURING ABOVE OPERATION ONLY
2732
2733
2734 013106 004037 032774    ;*NOW CHANGE SAVE REGISTERS TO EXPECTED VALUES
      JSR    RO,@#FILLRE   :MOV 0 INTO SAVED RHWC
  
```



```

2735 013112 002172          RHWC          :SAVED REGISTER TO CHANGE
2736 013114 000000          0              :DATA
2737 013116 004037 032774  JSR    RO,@#FILLRE :MOV WRFROM+<268.*2> INTO SAVED RHBA
2738 013122 002174          RHBA          :SAVED REGISTER TO CHANGE
2739 013124 003420          WRFROM+<268.*2> :DATA
2740 013126 004037 032774  JSR    RO,@#FILLRE :MOV 2 INTO SAVED RHDST
2741 013132 002204          RHDST         :SAVED REGISTER TO CHANGE
2742 013134 000002          2              :DATA
2743
2744          :*NOW COMPARE REGISTERS BEFORE WRITE HEADER AND DATA
2745          :*WITH REGISTERS AFTER COMMAND
2746 013136 004037 034070  JSR    RO,@#COMREG :COMPARE SAVED REGISTERS WITH
2747 013142 004512          SAVERE        :GOOD DATA SAVED IN 'SAVERE'
2748 013144 002254          WC           :TEST DATA STARTING FROM 'RHWC'
2749 013146 000022          18.          :18. REGISTERS TO BE COMPARED
2750 013150 013154          1$           :RETURN TO 1$ ON ERROR
2751 013152 013160          2$           :RETURN TO 2$ ON NO ERROR
2752
2753          1$:      ERROR 27          :WRITE HEADER AND DATA
2754 013156 000207          RTS    PC     :CAUSED IMPROPER REGISTER
2755          :CHANGE
2756          :GOOD DATA GIVES WHAT SHOULD
2757          :BE THERE
2758          :RECEIVED DATA GIVES WHAT
2759          :WAS THERE AFTER COMMAND
2760
2761          :*NOW A SEARCH COMMAND WILL BE GIVEN
2762          :*BUT BEFORE THAT ALL POSSIBLE REGISTERS
2763          :*WILL BE FILLED FOR THE READ HEADER AND DATA SECTOR 1
2764          :*AS THERE WILL NOT BE MUCH TIME BETWEEN THE
2765          :*COMPLETION OF THE SEARCH AND THE SECTOR 1 COMING.
2766
2767          :*FILL FOR THE READ HEADER AND DATA COMMAND WHICH WILL NOT
2768          :*BE EXECUTED TILL AFTER THE SEARCH
2769          :*THE SEARCH WILL ONLY LEAVE RHCS1 AND RHDST
2770          :*CHANGED ALL THE REST WILL BE UNCHANGED
2771
2772          2$:
2773 013160 004737 033072  JSR    PC,@#CLDISK :SET R1-RHCS1, R2-RHCS2
2774 013164 004037 035054  JSR    RO,@#RUN    :SETUP TO RUN FOR DATA COMMAND
2775 013170 000000          0              :CYLINDER 0
2776 013172 000          .BYTE 0         :SECTOR 0
2777 013173 000          .BYTE 0         :TRACK 0
2778 013174 177770          -8.           :WORD COUNT = 8.
2779 013176 003434          REINTO        :BUS ADDRESS
2780 013200 000000          0              :DO NOT INHIBIT BUS ADDRESS INCREMENT
2781 013202 014000          ECI!FMT22    :16 BITS PER WORD FORMAT
2782 013204 002334          SERCH        :GET READY TO DO A SERCH
2783
2784          :*SAVE REGISTERS FOR COMPARISON NOT AFTER THE
2785          :*SEARCH COMMAND BUT AFTER THE READ HEADER AND DATA
2786 013206 004037 033240  JSR    RO,@#SAVER :SAVE REGISTERS
2787 013212 002172          RHWC          :RHWC IS THE FIRST REGISTER SAVED
2788 013214 004512          SAVERE        :STARTING ADDRESS OF WHERE
2789 013216 000022          18.          :NUMBER OF REGISTERS
2790

```



```

2791                                     ;*NOW SAVE VALUES FOR RHCS1 AND RHDST WHICH
2792                                     ;*WILL CHANGE AFTER THE SEARCH
2793
2794 013220 013746 002350                MOV    @#REFOR,-(SP)    ;SAVE READ HEADER AND DATA
2795 013224 052716 000101                BIS    #IE!GO,(SP)    ;INTERRUPT ENABLE AND GO
2796 013230 012637 004650                MOV    (SP)+,@#TMP0   ;SAVE IN R0 FOR RHCS1
2797 013234 012737 000001 004656        MOV    #1,@#TMP5     ;SAVE TRACK 0 SECTOR 1 FOR RHDST
2798
2799                                     ;*THE INTERRUPT VECTOR WILL BE SET TO GO TO 2$
2800                                     ;*AFTER THE SEARCH
2801
2802 013242 012777 013310 166716          MOV    #7$,@RPVEC    ;SET INTERRUPT VECTOR TO 2$
2803 013250 004737 033152                  JSR    PC,@#CHECKT   ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
2804 013254 104401 057176                  TYPE   ,CPHALT      ;CANNOT CONTINUE TESTING IF NOT
2805 013260 000000                          HALT                ;STOP TEST
2806 013262 013746 002334                MOV    @#SERCH,-(SP) ;GET READY TO MOVE COMMAND
2807 013266 052716 000101                BIS    #GO!IE,(SP)  ;GET READY TO SET GO AND
2808 013272 012677 166702                MOV    (SP)+,@RHCS1 ;GO WITH
2809
2810                                     ;*TIME IS NOT CRITICAL THIS ONLY WAITS FOR SEARCH COMPLETION
2811 013276 104413                          WAT                ;WAIT FOR DRY BIT TO SET
2812 013300 002222                          RHDS1              ;WAIT FOR RHDS1 REGISTER
2813 013302 000200                          DRY                ;WAIT FOR DRY BIT IN RHDS1 REGISTER
2814 013304 001614                          908.              ;ALLOW 9080 MICRO SECONDS
2815 013306 001507                          839.              ;DRY MUST SET BETWEEN
2816
2817 013310 012737 000000 177776 7$:      MOV    #0,PS        ;SET PROSESSOR STATUS TO
2818                                     ;PRIORITY 0 IN CASE IT WAS
2819                                     ;TAKEN OUT OF WAT ROUTINE
2820                                     ;BEFORE RTI
2821 013316 013777 004656 166660          MOV    @#TMP5,@RHDST ;SET DESIRED SECTOR/TRACK
2822                                     ;REGISTER TO SECTOR 1,TRACK 0
2823 013324 013777 004506 166634          MOV    @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
2824
2825 013332 013777 004650 166640          MOV    @#TMP0,@RHCS1 ;FILL RHCS1 WITH READ COMMAND
2826                                     ;TOGETHER WITH INTERRUPT ENABLE
2827                                     ;AND GO
2828
2829                                     ;*TIME ALLOWED HERE IS CRITICAL ANY TIME ERROR
2830                                     ;*INDICATES WRONG SEARCH IN THE SEARCH COMMAND
2831 013340 104413                          WAT                ;WAIT FOR RDY BIT TO SET
2832 013342 002200                          RHCS1              ;WAIT FOR RHCS1 REGISTER
2833 013344 000200                          RDY                ;WAIT FOR RDY BIT IN RHCS1 REGISTER
2834 013346 000225                          149.              ;ALLOW 1490 MICRO SECONDS
2835 013350 000002                          2.                ;RDY MUST SET BETWEEN
2836
2837                                     ;*WRITE FROM BUFFER WILL BE FILLED WITH EXPECTED DATA
2838 013352 004037 032716                  JSR    R0,@#FLHEAD  ;SAVE HEADER DATA IN WRFROM
2839 013356 002370                          WRFROM            ;LOCATION WHERE SAVED
2840 013360 000004                          4                ;NUMBER OF WORDS SAVED
2841 013362 010000                          10000            ;FIRST DATA WORD
2842 013364 000001                          1                ;SECOND DATA WORD
2843 013366 000000                          0                ;THIRD DATA WORD
2844 013370 000000                          0                ;FOURTH DATA WORD
2845 013372 004037 032742                  JSR    R0,@#CLAREA ;CLEAR 4 WORDS, FROM WRFROM+<4*2>
2846 013376 002400                          WRFROM+<4*2>    ;STARTING FROM WRFROM+<4*2>
  
```



```

2847 013400 000004      4      :4 WORDS
2848 013402 000001      1      :FILL WITH 1
2849
2850      :*CHANGE SAVED REGISTERS TO EXPECTED VALUES
2851 013404 004037 032774 JSR    RO,@#FILLRE  :MOV 0 INTO SAVED RHWC
2852 013410 002172      RHWC    :SAVED REGISTER TO CHANGE
2853 013412 000000      0      :DATA
2854 013414 004037 032774 JSR    RO,@#FILLRE  :MOV REINTO+<8.*2> INTO SAVED RHBA
2855 013420 002174      RHBA    :SAVED REGISTER TO CHANGE
2856 013422 003454      REINTO+<8.*2> :DATA
2857 013424 004037 032774 JSR    RO,@#FILLRE  :MOV 4272 INTO SAVED RHCS1
2858 013430 002200      RHCS1   :SAVED REGISTER TO CHANGE
2859 013432 004272      4272    :DATA
2860 013434 004037 032774 JSR    RO,@#FILLRE  :MOV 2 INTO SAVED RHDST
2861 013440 002204      RHDST   :SAVED REGISTER TO CHANGE
2862 013442 000002      2      :DATA
2863
2864      :*COMPARE REGISTER BEFORE READ HEADER AND DATA
2865      :*WITH REGISTERS AFTER COMMAND
2866
2867 013444 004037 034070 JSR    RO,@#COMREG  :COMPARE SAVED REGISTERS WITH
2868 013450 004512      SAVERE   :GOOD DATA SAVED IN 'SAVERE'
2869 013452 002254      WC      :TEST DATA STARTING FROM 'RHWC'
2870 013454 000022      18.     :18. REGISTERS TO BE COMPARED
2871 013456 013462      3$      :RETURN TO 3$ ON ERROR
2872 013460 013466      4$      :RETURN TO 4$ ON NO ERROR
2873 013462 104031      3$:     ERROR 31 :READ HEADER AND DATA CAUSED
2874 013464 000207      RTS     PC  :IMPROPER REGISTER CHANGE
2875      :GOOD DATA GIVES WHAT SHOULD
2876      :BE THERE
2877      :RECEIVED DATA GIVES WHAT WAS
2878      :THERE AFTER COMMAND
2879
2880      :*NOW READ INTO BUFFER WILL BE CHECKED TO SEE
2881      :*THE READ WAS GOOD
2882
2882 013466      4$:
2883 013466 004037 035120 JSR    RO,@#COMPAR  :COMPARE TWO BLOCKS OF MEMORY
2884 013472 002370      WRFROM  :GOOD DATA STARTS FROM WRFROM
2885 013474 003434      REINTO  :TEST DATA STARTS FROM REINTO
2886 013476 000010      8.     :8. WORDS TO BE COMPARED
2887 013500 013504      5$     :RETURN TO 5$ ON ERROR
2888 013502 013510      6$     :RETURN TO 6$ ON NO ERROR
2889
2890 013504 104053      5$:     ERROR 53 :READ HEADER AND DATA
2891 013506 000207      RTS     PC  :AFTER A SEARCH CAUSED
2892      :AN ERROR
2893 013510      6$:
2894

```



```

2895
2896 013510 000004
2897 013512 012706 001000
2898 013516 012737 000013 004504
2899 013524 004737 033072
2900
2901 ;*GET THE HEADS TO CYLINDER 10
2902 013530 004737 033152 JSR PC,@#CHECKT ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
2903 013534 104401 057176 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF NOT
2904 013540 000000 HALT ;STOP TEST
2905 013542 013777 004506 166416 MOV @#RP4VEC,@RPVEC ;SET RPO4 VECTOR ADDRESS
2906 013550 004037 033042 JSR RO,@#SEEKCY ;SEEK FOR
2907 013554 000012 10. ;CYLINDER 10.
2908 013556 013746 002352 MOV @#SEECOM,-(SP) ;GET READY TO MOVE COMMAND
2909 013562 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET GO AND
2910 013566 012677 166406 MOV (SP)+,@RHCS1 ;GO WITH
2911 013572 104413 WAT ;WAIT FOR DRY BIT TO SET
2912 013574 002222 RHDS1 ;WAIT FOR RHDS1 REGISTER
2913 013576 000200 DRY ;WAIT FOR DRY BIT IN RHDS1 REGISTER
2914 013600 015530 7000. ;ALLOW 70000 MICRO SECONDS
2915 013602 000043 35. ;DRY MUST SET BETWEEN
2916
2917 ;*FILL REGISTERS FOR READ HEADER AND DATA TO BE DONE AFTER SEARCH
2918 013604 004737 033072 JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
2919 013610 004037 035054 JSR RO,@#RUN ;SETUP TO RUN FOR DATA COMMAND
2920 013614 000000 0 ;CYLINDER 0
2921 013616 000 .BYTE 0 ;SECTOR 0
2922 013617 000 .BYTE 0 ;TRACK 0
2923 013620 177770 -8. ;WORD COUNT = 8.
2924 013622 003434 REINTO ;BUS ADDRESS
2925 013624 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
2926 013626 014000 ECI!FMT22 ;16 BITS PER WORD FORMAT
2927 013630 002334 SERCH ;GET READY TO DO A SERCH
2928
2929 ;*SAVE REGISTERS FOR COMPARISON AFTER SEARCH
2930 ;*AND READ HEADER AND DATA
2931 013632 004037 033240 JSR RO,@#SAVER ;SAVE REGISTERS
2932 013636 002172 RHWC ;RHWC IS THE FIRST REGISTER SAVED
2933 013640 004512 SAVERE ;STARTING ADDRESS OF WHERE
2934 013642 000022 18. ;NUMBER OF REGISTERS
2935
2936 ;*NOW GIVE THE SEARCH COMMAND
2937 013644 004737 033152 JSR PC,@#CHECKT ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
2938 013650 104401 057176 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF NOT
2939 013654 000000 HALT ;STOP TEST
2940 013656 012777 014042 166302 MOV #3$,@RPVEC ;INTERRUPT VECTOR SET TO 3$
2941 013664 004037 033022 JSR RO,@#SRCH ;SEARCH FOR
2942 013670 000000 0 ;CYLINDER 0
2943 013672 000 .BYTE 0 ;SECTOR 0
2944 013673 000 .BYTE 0 ;TRACK 0
2945 013674 013700 002334 MOV @#SERCH,RO ;EXPECTED CONTENTS OF RHCS1
2946 ;IMMEDIATELY AFTER GO
2947 013700 052700 004301 BIS #DVA!RDY!IE!GO,RO ;EXPECTED BITS IN RHCS1
2948 013704 012705 010500 MOV #MOL!DPR!VV,R5 ;EXPECTED BITS IN RHDS1
2949 ;IMMEDIATELY AFTER GO
2950
    
```



```

2951 013710 013746 002334      MOV    @#SERCH,-(SP)    ;GET READY TO MOVE COMMAND
2952 013714 052716 000101      BIS    #GO!IE,(SP)    ;GET READY TO SET GO AND
2953 013720 012677 166254      MOV    (SP)+,@RHCS1   ;GO WITH
2954 013724 021100              CMP    @R1,R0         ;IS RHCS1 GOOD
2955 013726 001413              BEQ    1$             ;BRANCH IF GOOD
2956 013730 011137 001126      MOV    @R1,@#$BDDAT   ;BAD DATA FOR RHCS1
2957 013734 010037 001124      MOV    R0,@#$GDDAT   ;GOOD DATA
2958 013740 010137 004500      MOV    R1,@#REGADR   ;FAILING REGISTER RHCS1
2959 013744 012737 000340 000036  MOV    #340,@#TRAPVEC+2 ;TRAP PRIORITY = 7
2960 013752 104021              ERROR  21            ;DURING SEARCH COMMAND
2961                          ;CONTENTS OF RHCS1 WAS
2962                          ;NOT AS EXPECTED
2963 013754 000414              BR     2$             ;IF LAST ERROR 21 OCCURRED
2964                          ;THEN DO NOT CHECK RHDS1
2965                          ;AS TOO MUCH TIME HAS
2966                          ;PASSED
2967
2968 013756 021305              1$:  CMP    @R3,R5         ;IS RHDS1 GOOD
2969 013760 001412              BEQ    2$             ;BRANCH IF GOOD
2970 013762 011337 001126      MOV    @R3,@#$BDDAT   ;BAD DATA FOR RHDS1
2971 013766 010537 001124      MOV    R5,@#$GDDAT   ;GOOD DATA
2972 013772 010337 004500      MOV    R3,@#REGADR   ;FAILING REGISTER RHDS1
2973 013776 012737 000340 000036  MOV    #340,@#TRAPVEC+2 ;TRAP PRIORITY = 7
2974 014004 104063              ERROR  63            ;DURING SEARCH COMMAND
2975                          ;CONTENTS OF RHDS1 WAS
2976                          ;IN CORRECT
2977
2978 014006 013737 002350 004650 2$:  MOV    @#REFOR,@#TMP0 ;SAVE READ HEADER AND DATA
2979 014014 052737 000101 004650  BIS    #IE!GO,@#TMP0 ;INCLUDE INTERRUPT ENABLE, GO
2980 014022 012737 000001 004656  MOV    #1,@#TMP5     ;SAVE TRACK 0, SECTOR 1
2981
2982                          ;*THIS IS ONLY A WAIT LOOP
2983 014030 104413              WAT                    ;WAIT FOR RDY BIT TO SET
2984 014032 002222              RHDS1                 ;WAIT FOR RHDS1 REGISTER
2985 014034 000200              RDY                   ;WAIT FOR RDY BIT IN RHDS1 REGISTER
2986 014036 015530              7000.                 ;ALLOW 70000 MICRO SECONDS
2987 014040 000043              35.                   ;RDY MUST SET BETWEEN
2988
2989 014042 012737 000200 000036 3$:  MOV    #200,@#TRAPVEC+2 ;TRAP PRIORITY = 4
2990 014050 012737 000000 177776  MOV    #0,PS          ;SET PROSESSOR STATUS TO 0
2991 014056 013777 004656 166120  MOV    @#TMP5,@RHDS1 ;SET DESIRED SECTOR/TRACK
2992                          ;REGISTER TO SECTOR 1, TRACK 0
2993
2994 014064 013777 004506 166074  MOV    @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
2995 014072 013711 004650              MOV    @#TMP0,@R1    ;FILL RHCS1 WITH READ COMMAND
2996                          ;TOGETHER WITH INTERRUPT ENABLE
2997                          ;AND GO
2998
2999                          ;*TIME ALLOWED HERE IS CRITICAL ANY TIME ERROR
3000                          ;*INDICATES WRONG SEARCH IN THE SEARCH COMMAND
3001 014076 104413              WAT                    ;WAIT FOR RDY BIT TO SET
3002 014100 002200              RHCS1                 ;WAIT FOR RHCS1 REGISTER
3003 014102 000200              RDY                   ;WAIT FOR RDY BIT IN RHCS1 REGISTER
3004 014104 000225              149.                 ;ALLOW 1490 MICRO SECONDS
3005 014106 000002              2.                   ;RDY MUST SET BETWEEN
3006
    
```



```

3007 ;*WRITE FROM BUFFER WILL BE FILLED WITH EXPECTED DATA
3008 014110 004037 032716 JSR RO,@#FLHEAD ;SAVE HEADER DATA IN WRFROM
3009 014114 002370 WRFROM ;LOCATION WHERE SAVED
3010 014116 000004 4 ;NUMBER OF WORDS SAVED
3011 014120 010000 10000 ;FIRST DATA WORD
3012 014122 000001 1 ;SECOND DATA WORD
3013 014124 000000 0 ;THIRD DATA WORD
3014 014126 000000 0 ;FOURTH DATA WORD
3015 014130 004037 032742 JSR RO,@#CLAREA ;CLEAR 4 WORDS, FROM WRFROM+<4*2>
3016 014134 002400 WRFROM+<4*2> ;STARTING FROM WRFROM+<4*2>
3017 014136 000004 4 ;4 WORDS
3018 014140 000001 1 ;FILL WITH 1
3019
3020 ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
3021 014142 004037 032774 JSR RO,@#FILLRE ;MOV 0 INTO SAVED RHWC
3022 014146 002172 RHWC ;SAVED REGISTER TO CHANGE
3023 014150 000000 0 ;DATA
3024 014152 004037 032774 JSR RO,@#FILLRE ;MOV REINTO+<8.*2> INTO SAVED RHBA
3025 014156 002174 RHBA ;SAVED REGISTER TO CHANGE
3026 014160 003454 REINTO+<8.*2> ;DATA
3027 014162 004037 032774 JSR RO,@#FILLRE ;MOV 4272 INTO SAVED RHCS1
3028 014166 002200 RHCS1 ;SAVED REGISTER TO CHANGE
3029 014170 004272 4272 ;DATA
3030 014172 004037 032774 JSR RO,@#FILLRE ;MOV 2 INTO SAVED RHDST
3031 014176 002204 RHDST ;SAVED REGISTER TO CHANGE
3032 014200 000002 2 ;DATA
3033 014202 004037 032774 JSR RO,@#FILLRE ;MOV 0 INTO SAVED RHCC
3034 014206 002234 RHCC ;SAVED REGISTER TO CHANGE
3035 014210 000000 0 ;DATA
3036
3037 ;*COMPARE REGISTER BEFORE READ HEADER AND DATA
3038 ;*WITH REGISTERS AFTER COMMAND
3039
3040 014212 004037 034070 JSR RO,@#COMREG ;COMPARE SAVED REGISTERS WITH
3041 014216 004512 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
3042 014220 002254 WC ;TEST DATA STARTING FROM 'RHWC'
3043 014222 000022 18. ;18. REGISTERS TO BE COMPARED
3044 014224 014230 4$ ;RETURN TO 4$ ON ERROR
3045 014226 014234 5$ ;RETURN TO 5$ ON NO ERROR
3046
3047 014230 104031 4$: ERROR 31 ;READ HEADER AND DATA CAUSED
3048 014232 000207 RTS PC ;IMPROPER REGISTER CHANGE
3049 ;GOOD DATA GIVES WHAT SHOULD
3050 ;BE THERE
3051 ;RECEIVED DATA GIVES WHAT WAS
3052 ;THERE AFTER COMMAND
3053
3054 ;*NOW READ INTO BUFFER WILL BE CHECKED TO SEE
3055 ;*THE READ WAS GOOD
3056 014234 5$:
3057 014234 004037 035120 JSR RO,@#COMPAR ;COMPARE TWO BLOCKS OF MEMORY
3058 014240 002370 WRFROM ;GOOD DATA STARTS FROM WRFROM
3059 014242 003434 REINTO ;TEST DATA STARTS FROM REINTO
3060 014244 000010 8. ;8. WORDS TO BE COMPARED
3061 014246 014252 6$ ;RETURN TO 6$ ON ERROR
3062 014250 014256 7$ ;RETURN TO 7$ ON NO ERROR
  
```


3063					
3064	014252	104053	6\$:	ERROR	53
3065	014254	000207		RTS	PC
3066					
3067	014256		7\$:		
3068					
3069					
3070					
3071					

:READ HEADER AND DATA
:AFTER A SEARCH CAUSED
:AN ERROR


```

3072
3073      :*      THE NEXT TEST REMOVES SECTOR 1 ON CYLINDER 0
3074      :*      TRACK0 AND PUTS SECTOR 0 THERE.
3075      :*      HENCE THE PACK IS UNFORMATTED FROM
3076      :*      THIS POINT ON TO THE TEST WHEN SECTOR
3077      :*      1 IS REPLACED. IF TESTING IS STOPPED WITH
3078      :*      AN ERROR IN THE SECTION OF THE PROGRAM BETWEEN
3079      :*      THIS AND WHEN SECTOR 1 IS REPLACED THEN THE
3080      :*      DISK BEING USED MAY HAVE BEEN UNFORMATTED
3081      :*      IF THE LAST PASS OF THIS PROGRAM GIVES
3082      :*      NO ERRORS IN THIS SECTION THEN THE DISK
3083      :*      MAY NOT HAVE BEEN UNFORMATTED. HOWEVER IT
3084      :*      IS RECOMMENDED THAT AFTER A PASS OF THIS
3085      :*      PROGRAM THE DISK BE REFORMATTED.
3086
3087
3088
3089
3090
3091
3092 014256 000004          TST14: SCOPE
3093 014260 012706 001000  MOV      #STACK,SP      ;RESET STACK
3094 014264 012737 000014 004504  MOV      #14,@#TSTNM    ;SAVE TEST NUMBER
3095 014272 004737 033072  JSR      PC,@#CLDISK    ;SET R1-RHCS1, R2-RHCS2
3096
3097      ;*FILL WRITE FROM BUFFER WITH HEADER
3098 014276 004037 032716  JSR      RO,@#FLHEAD    ;SAVE HEADER DATA IN WRFROM
3099 014302 002370          WRFROM      ;LOCATION WHERE SAVED
3100 014304 000005          5              ;NUMBER OF WORDS SAVED
3101 014306 010000          10000         ;FIRST DATA WORD
3102 014310 000000          0              ;SECOND DATA WORD
3103 014312 000000          0              ;THIRD DATA WORD
3104 014314 000000          0              ;FOURTH DATA WORD
3105 014316 000001          1              ;FIFTH DATA WORD
3106
3107      ;*FILL READ INTO BUFFER WITH ALL ONES
3108
3109 014320 004037 032742  JSR      RO,@#CLAREA    ;CLEAR 256. WORDS, FROM REINTO
3110 014324 003434          REINTO      ;STARTING FROM REINTO
3111 014326 000400          256.         ;256. WORDS
3112 014330 177777          -1          ;FILL WITH -1
3113
3114      ;*WRITE HEADER AND DATA IS LOADED
3115 014332 004037 035054  JSR      RO,@#RUN      ;SETUP TO RUN FOR DATA COMMAND
3116 014336 000000          0.           ;CYLINDER 0.
3117 014340          001         ;SECTOR 1.
3118 014341          000         ;TRACK 0.
3119 014342 177773          -1-4        ;WORD COUNT (DATA) = 1 +
3120 014344 002370          WRFROM      ;BUS ADDRESS
3121 014346 000000          0           ;DO NOT INHIBIT BUS ADDRESS INCREMENT
3122 014350 010000          FMT22      ;16 BITS PER WORD FORMAT
3123 014352 002344          WRIFOR     ;GET READY TO DO A WRIFOR
3124
3125      ;*NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE
3126
3127 014354 004037 033240  JSR      RO,@#SAVER    ;SAVE REGISTERS
    
```



```

3128 014360 002172          RHWC          ;RHWC IS THE FIRST REGISTER SAVED
3129 014362 004512          SAVERE        ;STARTING ADDRES OF WHERE
3130 014364 000021          17.          ;NUMBER OF REGISTERS
3131
3132 014366 004737 033152    JSR    PC,@#CHECKT ;CHECKS DVA, RDY, MOL, DPR, DRY AND VV = 1
3133 014372 104401 057176    TYPE    ,CPHALT   ;CANNOT CONTINUE TESTING IF NOT
3134 014376 000000          HALT          ;STOP TEST
3135
3136 014400 013777 004506 165560  MOV    @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
3137
3138 014406 013746 002344    MOV    @#WRIFOR,-(SP) ;GET READY TO MOVE COMMAND
3139 014412 052716 000101    BIS    #GO!IE,(SP)  ;GET READY TO SET GO AND
3140 014416 012677 165556    MOV    (SP)+,@RHCS1 ;GO WITH
3141
3142          ;*TIME IS NOT CRITICAL
3143 014422 104413          WAT          ;WAIT FOR RDY BIT TO SET
3144 014424 002200          RHCS1        ;WAIT FOR RHCS1 REGISTER
3145 014426 000200          RDY          ;WAIT FOR RDY BIT IN RHCS1 REGISTER
3146 014430 004704          2500.        ;ALLOW 25000 MICRO SECONDS
3147 014432 004704          2500.        ;RDY MUST SET BETWEEN
3148
3149          ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
3150 014434 004037 032774    JSR    R0,@#FILLRE ;MOV 0 INTO SAVED RHWC
3151 014440 002172          RHWC          ;SAVED REGISTER TO CHANGE
3152 014442 000000          0            ;DATA
3153 014444 004037 032774    JSR    R0,@#FILLRE ;MOV WRFROM+<5*2> INTO SAVED RHBA
3154 014450 002174          RHBA          ;SAVED REGISTER TO CHANGE
3155 014452 002402          WRFROM+<5*2> ;DATA
3156 014454 004037 032774    JSR    R0,@#FILLRE ;MOV 2 INTO SAVED RHDST
3157 014460 002204          RHDST        ;SAVED REGISTER TO CHANGE
3158 014462 000002          2            ;DATA
3159
3160          ;*COMPARE ALL REGISTERS
3161
3162 014464 004037 034070    JSR    R0,@#COMREG ;COMPARE SAVED REGISTERS WITH
3163 014470 004512          SAVERE        ;GOOD DATA SAVED IN 'SAVERE'
3164 014472 002254          WC          ;TEST DATA STARTING FROM 'RHWC'
3165 014474 000021          17.          ;17. REGISTERS TO BE COMPARED
3166 014476 014502          1$          ;RETURN TO 1$ ON ERROR
3167 014500 014506          2$          ;RETURN TO 2$ ON NO ERROR
3168 014502 104027          1$: ERROR 27 ;WRITING HEADER AND DATA CAUSED
3169 014504 000207          RTS    PC
3170
3171          ;IMPROPER REGISTER CHANGE
3172          ;GOOD DATA GIVES WHAT
3173          ;SHOULD BE THERE
3174          ;RECEIVED DATA GIVES WHAT
3174          ;WAS THERE AFTER WRITE
    
```



```

3175
3176
3177
3178
3179
3180
3181 014506
3182 014506 004737 033072
3183 014512 004037 032742
3184 014516 002370
3185 014520 000400
3186 014522 177777
3187
3188
3189 014524 004037 035054
3190 014530 000000
3191 014532 001
3192 014533 000
3193 014534 177777
3194 014536 003434
3195 014540 000000
3196 014542 010000
3197 014544 002346
3198
3199
3200 014546 004037 033240
3201 014552 002172
3202 014554 004512
3203 014556 000022
3204
3205 014560 004737 033152
3206 014564 104401 057176
3207 014570 000000
3208
3209 014572 013777 004506 165366
3210
3211 014600 013746 002346
3212 014604 052716 000101
3213 014610 012677 165364
3214
3215
3216 014614 104413
3217 014616 002200
3218 014620 000200
3219 014622 010110
3220 014624 001502
3221
3222
3223 014626 004037 033762
3224 014632 002222
3225 014634 000002
3226 014636 000001
3227 014640 100000
3228 014642 000001
3229 014644 040000
3230 014646 004037 033762
    
```

2\$:

```

;*NOW A READ DATA WILL BE DONE ON CYLINDER=0, SECTOR=1,
;*TRACK=0
;*FILL WRITE FROM BUFFER WITH EXPECTED DATA

JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
JSR RO,@#CLAREA ;CLEAR 256. WORDS, FROM WRFROM
WRFROM ;STARTING FROM WRFROM
256. ;256. WORDS
-1 ;FILL WITH -1

;*READ COMMAND IS LOADED
JSR RO,@#RUN ;SETUP TO RUN FOR DATA COMMAND
0 ;CYLINDER 0
.BYTE 1. ;SECTOR 1.
.BYTE 0 ;TRACK 0
-1 ;WORD COUNT = 1
REINTO ;BUS ADDRESS
0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
FMT22 ;16 BITS PER WORD FORMAT
READAT ;GET READY TO DO A READAT

;*SAVE REGISTERS FOR COMPARISON AFTER READ DATA
JSR RO,@#SAVER ;SAVE REGISTERS
RHWC ;RHWC IS THE FIRST REGISTER SAVED
SAVERE ;STARTING ADDRESS OF WHERE
18. ;NUMBER OF REGISTERS

JSR PC,@#CHECKT ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF NOT
HALT ;STOP TEST

MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS

MOV @#READAT,-(SP) ;GET READY TO MOVE COMMAND
BIS #GO!IE,(SP) ;GET READY TO SET GO AND
MOV (SP)+,@RHCS1 ;GO WITH

;*TIME IS NOT CRITICAL
WAT ;WAIT FOR RDY BIT TO SET
RHCS1 ;WAIT FOR RHCS1 REGISTER
RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
4168. ;ALLOW 41680 MICRO SECONDS
834. ;RDY MUST SET BETWEEN

;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
RHDS1 ;CHANGE RHDS1 REGISTER
2 ;2 BIT/BITS TO BE CHANGED
1 ;NEW VALUE OF ATA IS 1
ATA ;CHANGE ATA BIT
1 ;NEW VALUE OF ERR IS 1
ERR ;CHANGE ERR BIT
JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
    
```



```

3231 014652 002200          RHCS1          ;CHANGE RHCS1 REGISTER
3232 014654 000002          2              ;2 BIT/BITS TO BE CHANGED
3233 014656 000001          1              ;NEW VALUE OF SC IS 1
3234 014660 100000          SC              ;CHANGE SC BIT
3235 014662 000001          1              ;NEW VALUE OF TRE IS 1
3236 014664 040000          TRE            ;CHANGE TRE BIT
3237 014666 004037 033762  JSR      RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
3238 014672 002202          RHER1         ;CHANGE RHER1 REGISTER
3239 014674 000001          1              ;1 BIT/BITS TO BE CHANGED
3240 014676 000001          1              ;NEW VALUE OF HCE IS 1
3241 014700 000200          HCE            ;CHANGE HCE BIT
3242 014702 004037 032774  JSR      RO,@#FILLRE ;MOV 2 INTO SAVED RHDST
3243 014706 002204          RHDST         ;SAVED REGISTER TO CHANGE
3244 014710 000002          2              ;DATA
3245 014712 053737 004644 004536 BIS      @#ATTENT,@#SAVERE+24 ;SET APPROPRIATE 'ATA' BITS
3246
3247          ;*COMPARE ALL REGISTERS
3248 014720 004037 034070  JSR      RO,@#COMREG ;COMPARE SAVED REGISTERS WITH
3249 014724 004512          SAVERE        ;GOOD DATA SAVED IN 'SAVERE'
3250 014726 002254          WC           ;TEST DATA STARTING FROM 'RHWC'
3251 014730 000022          18.          ;18. REGISTERS TO BE COMPARED
3252 014732 014736          3$           ;RETURN TO 3$ ON ERROR
3253 014734 014742          4$           ;RETURN TO 4$ ON NO ERROR
3254 014736 104047          3$: ERROR 47   ;READING ON NON EXISTANT SECTOR
3255 014740 000207          RTS      PC
3256          ; CAUSED AN ERROR
3257          ; GOOD DATA GIVES WHAT SHOULD
3258          ; BE THERE
3259          ; RECEIVED DATA GIVES WHAT
3260          ; WAS THERE AFTER READ
3261
3262          ;*READ DATA WILL BE COMPARED
3263
3264 014742          4$:
3265 014742 004037 035120  JSR      RO,@#COMPAR ;COMPARE TWO BLOCKS OF MEMORY
3266 014746 002370          WRFROM       ;GOOD DATA STARTS FROM WRFROM
3267 014750 003434          REINTO       ;TEST DATA STARTS FROM REINTO
3268 014752 000400          256.        ;256., WORDS TO BE COMPARED
3269 014754 014760          5$           ;RETURN TO 5$ ON ERROR
3270 014756 014764          6$           ;RETURN TO 6$ ON NO ERROR
3271 014760 104050          5$: ERROR 50   ;DATA READ FROM NON
3272 014762 000207          RTS      PC   ;EXISTANT SECTOR CAUSED AN ERROR
3273 014764          6$:
3274

```



```

3275 014764 000004          TST15: SCOPE
3276 014766 012706 001000      MOV      #STACK,SP      ;RESET STACK
3277 014772 012737 000015 004504  MOV      #15,@#TSTNM    ;SAVE TEST NUMBER
3278 015000 004737 033072      JSR      PC,@#CLDISK    ;SET R1-RHCS1, R2-RHCS2
3279
3280                          ;*FILL WRITE FROM BUFFER WITH HEADER AND DATA
3281 015004 004037 032716      JSR      RO,@#FLHEAD    ;SAVE HEADER DATA IN WRFROM
3282 015010 002370              WRFROM                  ;LOCATION WHERE SAVED
3283 015012 000006              6                       ;NUMBER OF WORDS SAVED
3284 015014 010000              10000                  ;FIRST DATA WORD
3285 015016 000000              0                       ;SECOND DATA WORD
3286 015020 000000              0                       ;THIRD DATA WORD
3287 015022 000000              0                       ;FOURTH DATA WORD
3288 015024 000001              1                       ;FIFTH DATA WORD
3289 015026 000001              1                       ;SIXTH DATA WORD
3290
3291                          ;*FILL READ INTO BUFFER WITH ALL ONES
3292 015030 004037 032742      JSR      RO,@#CLAREA    ;CLEAR 256. WORDS, FROM REINTO
3293 015034 003434              REINTO                  ;STARTING FROM REINTO
3294 015036 000400              256.                   ;256. WORDS
3295 015040 177777              -1                      ;FILL WITH -1
3296
3297                          ;*WRITE HEADER AND DATA IS LOADED
3298 015042 004037 035054      JSR      RO,@#RUN       ;SETUP TO RUN FOR DATA COMMAND
3299 015046 000000              0                       ;CYLINDER 0
3300 015050          001          .BYTE 1                 ;SECTOR 1
3301 015051          000          .BYTE 0                 ;TRACK 0
3302 015052 177772              -2-4                   ;WORD COUNT (DATA) = 2 +
3303 015054 002370              WRFROM                  ;BUS ADDRESS
3304 015056 000000              0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
3305 015060 010000              FMT22                   ;16 BITS PER WORD FORMAT
3306 015062 002344              WRIFOR                  ;GET READY TO DO A WRIFOR
3307
3308                          ;*NOW SAVE REGISTERS FOR COMPARISON AFTER
3309                          ;*WRITE HEADER AND DATA
3310
3311 015064 004037 033240      JSR      RO,@#SAVER     ;SAVE REGISTERS
3312 015070 002172              RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
3313 015072 004512              SAVERE                  ;STARTING ADDRESS OF WHERE
3314 015074 000021              17.                    ;NUMBER OF REGISTERS
3315
3316 015076 004737 033152      JSR      PC,@#CHECKT    ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
3317 015102 104401 057176      TYPE      ,CPHALT      ;CANNOT CONTINUE TESTING IF NOT
3318 015106 000000              HALT                   ;STOP TEST
3319
3320 015110 013777 004506 165050  MOV      @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
3321
3322 015116 013746 002344      MOV      @#WRIFOR,-(SP) ;GET READY TO MOVE COMMAND
3323 015122 052716 000101      BIS      #GO!IE,(SP)    ;GET READY TO SET GO AND
3324 015126 012677 165046      MOV      (SP)+,@RHCS1   ;GO WITH
3325
3326                          ;*TIME IS NOT CRITICAL
3327 015132 104413              WAT                     ;WAIT FOR RDY BIT TO SET
3328 015134 002200              RHCS1                   ;WAIT FOR RHCS1 REGISTER
3329 015136 000200              RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
3330 015140 004704              2500.                  ;ALLOW 25000 MICRO SECONDS
    
```



```
3331 015142 004704      2500.          :RDY MUST SET BETWEEN
3332
3333                      ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
3334 015144 004037 032774 JSR      RO,@#FILLRE  :MOV 0 INTO SAVED RHWC
3335 015150 002172          RHWC          :SAVED REGISTER TO CHANGE
3336 015152 000000          0            :DATA
3337 015154 004037 032774 JSR      RO,@#FILLRE  :MOV WRFROM+<6*2> INTO SAVED RHBA
3338 015160 002174          RHBA          :SAVED REGISTER TO CHANGE
3339 015162 002404          WRFROM+<6*2> :DATA
3340 015164 004037 032774 JSR      RO,@#FILLRE  :MOV 2 INTO SAVED RHDST
3341 015170 002204          RHDST         :SAVED REGISTER TO CHANGE
3342 015172 000002          2            :DATA
3343
3344                      ;*COMPARE ALL REGISTERS
3345 015174 004037 034070 JSR      RO,@#COMREG  :COMPARE SAVED REGISTERS WITH
3346 015200 004512          SAVERE        :GOOD DATA SAVED IN 'SAVERE'
3347 015202 002254          WC            :TEST DATA STARTING FROM 'RHWC'
3348 015204 000021          17.          :17. REGISTERS TO BE COMPARED
3349 015206 015212          1$           :RETURN TO 1$ ON ERROR
3350 015210 015216          2$           :RETURN TO 2$ ON NO ERROR
3351 015212 104027          1$: ERROR 27    :WRITING HEADER AND DATA CAUSED
3352 015214 000207          RTS      PC     :IMPROPER REGISTER CHANGE
3353                      :GOOD DATA GIVES WHAT
3354                      :SHOULD BE THERE
3355                      :RECEIVED DATA GIVES WHAT
3356                      :WAS THERE AFTER WRITE
3357
```



```

3358
3359
3360
3361
3362 015216          2$:
3363
3364 015216 004037 032742
3365 015222 002370
3366 015224 000106
3367 015226 177400
3368
3369
3370 015230 004037 035054
3371 015234 000000
3372 015236      001
3373 015237      000
3374 015240 177672
3375 015242 002370
3376 015244 000000
3377 015246 010000
3378 015250 002342
3379
3380
3381 015252 004037 033240
3382 015256 002172
3383 015260 004512
3384 015262 000022
3385
3386 015264 004737 033152
3387 015270 104401 057176
3388 015274 000000
3389
3390 015276 013777 004506 164662
3391
3392 015304 013746 002342
3393 015310 052716 000101
3394 015314 012677 164660
3395
3396
3397 015320 104413
3398 015322 002200
3399 015324 000200
3400 015326 001614
3401 015330 001502
3402
3403
3404 015332 005737 004640
3405 015336 001421
3406
3407 015340 004037 032774
3408 015344 002172
3409 015346 177702
3410 015350 004037 032774
3411 015354 002174
3412 015356 002410
3413 015360 004037 033762
  
```

:*NOW A WRITE DATA WILL BE DONE ON SAME CYLINDER, SECTOR
 :*TRACK

:*FILL WRITE FROM BUFFER WITH DATA
 JSR RO,@#CLAREA :CLEAR 70. WORDS, FROM WRFROM
 WRFROM :STARTING FROM WRFROM
 70. :70. WORDS
 177400 :FILL WITH 177400

:*WRITE DATA COMMAND IS LOADED
 JSR RO,@#RUN :SETUP TO RUN FOR DATA COMMAND
 0 :CYLINDER 0
 .BYTE 1 :SECTOR 1
 .BYTE 0 :TRACK 0
 -70. :WORD COUNT = 70.
 WRFROM :BUS ADDRESS
 0 :DO NOT INHIBIT BUS ADDRESS INCREMENT
 FMT22 :16 BITS PER WORD FORMAT
 WRIDAT :GET READY TO DO A WRIDAT

:*SAVE REGISTERS FOR COMPARISON AFTER WRITE DATA
 JSR RO,@#SAVER :SAVE REGISTERS
 RHWC :RHWC IS THE FIRST REGISTER SAVED
 SAVERE :STARTING ADDRESS OF WHERE
 18. :NUMBER OF REGISTERS

JSR PC,@#CHECKT :CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
 TYPE ,CPHALT :CANNOT CONTINUE TESTING IF NOT
 HALT :STOP TEST

MOV @#RP4VEC,@RPVEC :SET RP04 VECTOR ADDRESS

MOV @#WRIDAT,-(SP) :GET READY TO MOVE COMMAND
 BIS #GO!IE,(SP) :GET READY TO SET GO AND
 MOV (SP)+,@RHCS1 :GO WITH

:*TIME IS NOT CRITICAL
 WAT :WAIT FOR RDY BIT TO SET
 RHCS1 :WAIT FOR RHCS1 REGISTER
 RDY :WAIT FOR RDY BIT IN RHCS1 REGISTER
 908. :ALLOW 9080 MICRO SECONDS
 834. :RDY MUST SET BETWEEN

:*NOW CHANGE REGISTERS TO EXPECTED VALUE
 TST @#RH70 :RUNNING ON RH70 ?
 BEQ 9\$:IF NOT, SKIP NEXT RH70 CODE

JSR RO,@#FILLRE :MOV -62. INTO SAVED RHWC
 RHWC :SAVED REGISTER TO CHANGE
 -62. :DATA

JSR RO,@#FILLRE :MOV WRFROM+<8.*2> INTO SAVED RHBA
 RHBA :SAVED REGISTER TO CHANGE
 WRFROM+<8.*2> :DATA
 JSR RO,@#CHREG :CHANGE BITS IN SAVED REGISTER


```

3414 015364 002176          RHCS2          :CHANGE RHCS2 REGISTER
3415 015366 000002          2          :2 BIT/BITS TO BE CHANGED
3416 015370 000001          1          :NEW VALUE OF OR IS 1
3417 015372 000200          OR          :CHANGE OR BIT
3418 015374 000000          0          :NEW VALUE OF IR IS 0
3419 015376 000100          IR          :CHANGE IR BIT
3420 015400 000416          BR          10$      :SKIP NEXT RH11 CODE
3421
3422 015402          9$:
3423 015402 004037 032774          JSR          RO,@#FILLRE :MOV -4. INTO SAVED RHWC
3424 015406 002172          RHWC        :SAVED REGISTER TO CHANGE
3425 015410 177774          -4.         :DATA
3426 015412 004037 032774          JSR          RO,@#FILLRE :MOV WRFROM+<66.*2> INTO SAVED RHBA
3427 015416 002174          RHBA        :SAVED REGISTER TO CHANGE
3428 015420 002574          WRFROM+<66.*2> :DATA
3429 015422 004037 033762          JSR          RO,@#CHREG  :CHANGE BITS IN SAVED REGISTER
3430 015426 002176          RHCS2        :CHANGE RHCS2 REGISTER
3431 015430 000001          1          :1 BIT/BITS TO BE CHANGED
3432 015432 000001          1          :NEW VALUE OF OR IS 1
3433 015434 000200          OR          :CHANGE OR BIT
3434
3435
3436 015436          10$:
3437 015436 004037 033762          JSR          RO,@#CHREG  :CHANGE BITS IN SAVED REGISTER
3438 015442 002200          RHCS1        :CHANGE RHCS1 REGISTER
3439 015444 000002          2          :2 BIT/BITS TO BE CHANGED
3440 015446 000001          1          :NEW VALUE OF SC IS 1
3441 015450 100000          SC          :CHANGE SC BIT
3442 015452 000001          1          :NEW VALUE OF TRE IS 1
3443 015454 040000          TRE          :CHANGE TRE BIT
3444 015456 004037 032774          JSR          RO,@#FILLRE :MOV 2 INTO SAVED RHDST
3445 015462 002204          RHDST        :SAVED REGISTER TO CHANGE
3446 015464 000002          2          :DATA
3447 015466 053737 004644 004536          BIS          @#ATTENT,@#SAVERE+24 :SET APPROPRIATE 'ATA' BITS
3448 015474 004037 033762          JSR          RO,@#CHREG  :CHANGE BITS IN SAVED REGISTER
3449 015500 002202          RHER1        :CHANGE RHER1 REGISTER
3450 015502 000001          1          :1 BIT/BITS TO BE CHANGED
3451 015504 000001          1          :NEW VALUE OF HCE IS 1
3452 015506 000200          HCE          :CHANGE HCE BIT
3453 015510 004037 033762          JSR          RO,@#CHREG  :CHANGE BITS IN SAVED REGISTER
3454 015514 002222          RHDS1        :CHANGE RHDS1 REGISTER
3455 015516 000002          2          :2 BIT/BITS TO BE CHANGED
3456 015520 000001          1          :NEW VALUE OF ATA IS 1
3457 015522 100000          ATA          :CHANGE ATA BIT
3458 015524 000001          1          :NEW VALUE OF ERR IS 1
3459 015526 040000          ERR          :CHANGE ERR BIT
3460
3461          ;*COMPARE ALL REGISTERS
3462 015530 004037 034070          JSR          RO,@#COMREG :COMPARE SAVED REGISTERS WITH
3463 015534 004512          SAVERE       :GOOD DATA SAVED IN 'SAVERE'
3464 015536 002254          WC          :TEST DATA STARTING FROM 'RHWC'
3465 015540 000022          18.         :18. REGISTERS TO BE COMPARED
3466 015542 015546          3$          :RETURN TO 3$ ON ERROR
3467 015544 015552          4$          :RETURN TO 4$ ON NO ERROR
3468 015546 104052          3$:         :WRITE DATA ON NON EXISTANT SECTOR
3469 015550 000207          ERROR       52      :CAUSED IMPROPER REGISTER CHANGE
          RTS          PC
  
```



```

3526 015704 000000 HALT ;STOP TEST
3527
3528 015706 013777 004506 164252 MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
3529
3530 015714 013746 002350 MOV @#REFOR,-(SP) ;GET READY TO MOVE COMMAND
3531 015720 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET GO AND
3532 015724 012677 164250 MOV (SP)+,@RHCS1 ;GO WITH
3533
3534 ;*TIME IS NOT CRITICAL
3535 015730 104413 WAT ;WAIT FOR RDY BIT TO SET
3536 015732 002200 RHCS1 ;WAIT FOR RHCS1 REGISTER
3537 015734 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
3538 015736 001614 908. ;ALLOW 9080 MICRO SECONDS
3539 015740 001507 839. ;RDY MUST SET BETWEEN
3540
3541 ;*CHANGE SAVED REGISTERS TO EXPECTED VALUE
3542 015742 004037 032774 JSR RO,@#FILLRE ;MOV 0 INTO SAVED RHWC
3543 015746 002172 RHWC ;SAVED REGISTER TO CHANGE
3544 015750 000000 0 ;DATA
3545 015752 004037 032774 JSR RO,@#FILLRE ;MOV REINTO+<204.*2> INTO SAVED RHBA
3546 015756 002174 RHBA ;SAVED REGISTER TO CHANGE
3547 015760 004264 REINTO+<204.*2> ;DATA
3548 015762 004037 033762 JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
3549 015766 002202 RHER1 ;CHANGE RHER1 REGISTER
3550 015770 000001 1 ;1 BIT/BITS TO BE CHANGED
3551 015772 000001 1 ;NEW VALUE OF HCE IS 1
3552 015774 000200 HCE ;CHANGE HCE BIT
3553 015776 004037 033762 JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
3554 016002 002222 RHDS1 ;CHANGE RHDS1 REGISTER
3555 016004 000002 2 ;2 BIT/BITS TO BE CHANGED
3556 016006 000001 1 ;NEW VALUE OF ATA IS 1
3557 016010 100000 ATA ;CHANGE ATA BIT
3558 016012 000001 1 ;NEW VALUE OF ERR IS 1
3559 016014 040000 ERR ;CHANGE ERR BIT
3560 016016 004037 032774 JSR RO,@#FILLRE ;MOV 2 INTO SAVED RHDST
3561 016022 002204 RHDST ;SAVED REGISTER TO CHANGE
3562 016024 000002 2 ;DATA
3563 016026 004037 033762 JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
3564 016032 002200 RHCS1 ;CHANGE RHCS1 REGISTER
3565 016034 000001 1 ;1 BIT/BITS TO BE CHANGED
3566 016036 000001 1 ;NEW VALUE OF SC!TRE IS 1
3567 016040 140000 SC!TRE ;CHANGE SC!TRE BIT
3568 016042 053737 004644 004536 BIS @#ATTENT,@#SAVERE+24 ;SET APPROPRIATE 'ATA' BITS
3569
3570 ;*COMPARE REGISTERS BEFORE READ HEADER AND DATA
3571 ;*WITH AFTER
3572 016050 004037 034070 JSR RO,@#COMREG ;COMPARE SAVED REGISTERS WITH
3573 016054 004512 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
3574 016056 002254 WC ;TEST DATA STARTING FROM 'RHWC'
3575 016060 000022 18. ;18. REGISTERS TO BE COMPARED
3576 016062 016066 5$ ;RETURN TO 5$ ON ERROR
3577 016064 016072 6$ ;RETURN TO 6$ ON NO ERROR
3578
3579 016066 104031 5$: ERROR 31 ;READ HEADER AND DATA WITH
3580 016070 000207 RTS PC ;FORCED HEADER COMPARE ERROR
3581 ;CAUSED ERROR

```



```
3582 ;GOOD DATA GIVES WHAT SHOULD
3583 ;BE THERE
3584 ;RECEIVED DATA GIVES WHAT
3585 ;WAS THERE AFTER READ
3586
3587 ;*NOW COMPARE READ DATA
3588 ;*THE COMMAND READ ONLY 204 WORDS, 4 HEADER WORDS
3589 ;*AND 200 DATA WORDS
3590
3591 016072 6$: JSR R0,@#COMPAR ;COMPARE TWO BLOCKS OF MEMORY
3592 016072 004037 035120 WRFROM ;GOOD DATA STARTS FROM WRFROM
3593 016076 002370 REINTO ;TEST DATA STARTS FROM REINTO
3594 016100 003434 256. ;256. WORDS TO BE COMPARED
3595 016102 000400 7$ ;RETURN TO 7$ ON ERROR
3596 016104 016110 8$ ;RETURN TO 8$ ON NO ERROR
3597 016106 016114 7$: ERROR 34 ;DATA READ FROM A FORCED
3598 016110 104034 RTS PC ;HEADER COMPARE ERROR IS
3599 016112 000207 ;INCORRECT
3600 ;GOOD DATA GIVES WHAT
3601 ;THE READ HEADER AND DATA
3602 ;SHOULD HAVE READ
3603 ;BAD DATA GIVES WHAT
3604 ;WAS IN BUFFER AFTER
3605 ;READ COMMAND
3606
3607 016114 8$:
3608
3609
3610
3611
3612
3613
```



```

3614
3615 016114 000004          TST16: SCOPE
3616 016116 012706 001000  MOV      #STACK,SP      ;RESET STACK
3617 016122 012737 000016 004504  MOV      #16,@#TSTNM    ;SAVE TEST NUMBER
3618 016130 004737 033072  JSR      PC,@#CLDISK    ;SET R1-RHCS1, R2-RHCS2
3619
3620          ;*GET HEADS TO CYLINDER 0
3621 016134 004737 033152  JSR      PC,@#CHECKT    ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
3622 016140 104401 057176  TYPE     ,CPHALT        ;CANNOT CONTINUE TESTING IF NOT
3623 016144 000000          HALT                    ;STOP TEST
3624
3625 016146 013777 004506 164012  MOV      @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
3626
3627 016154 013746 002326  MOV      @#RECALI,-(SP)  ;GET READY TO MOVE COMMAND
3628 016160 052716 000101  BIS      #GO!IE,(SP)    ;GET READY TO SET GO AND
3629 016164 012677 164010  MOV      (SP)+,@RHCS1   ;GO WITH
3630 016170 011100          MOV      @R1,R0         ;SAVE RHCS1 DURING ABOVE OPERATION
3631 016172 011305          MOV      @R3,R5         ;SAVE RHDS1 DURING ABOVE OPERATION
3632
3633 016174 104413          WAT                        ;WAIT FOR DRY BIT TO SET
3634 016176 002222          RHDS1                    ;WAIT FOR RHDS1 REGISTER
3635 016200 000200          DRY                      ;WAIT FOR DRY BIT IN RHDS1 REGISTER
3636 016202 012740          5600.                   ;ALLOW 56000 MICRO SECONDS
3637 016204 012737          5599.                   ;DRY MUST SET BETWEEN
3638
3639 016206 004737 033072  JSR      PC,@#CLDISK    ;SET R1-RHCS1, R2-RHCS2
3640
3641          ;*FILL REGISTERS FOR SEARCH
3642 016212 004037 033022  JSR      R0,@#SRCH      ;SEARCH FOR
3643 016216 000000          0                        ;CYLINDER 0
3644 016220          001                    .BYTE 1                  ;SECTOR 1
3645 016221          000                    .BYTE 0                  ;TRACK 0
3646
3647          ;*SAVE REGISTERS FOR COMPARISON AFTER SEARCH
3648 016222 004037 033240  JSR      R0,@#SAVER     ;SAVE REGISTERS
3649 016226 002172          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
3650 016230 004512          SAVERE                  ;STARTING ADDRESS OF WHERE
3651 016232 000022          18.                    ;NUMBER OF REGISTERS
3652
3653
3654 016234 004737 033152  JSR      PC,@#CHECKT    ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
3655 016240 104401 057176  TYPE     ,CPHALT        ;CANNOT CONTINUE TESTING IF NOT
3656 016244 000000          HALT                    ;STOP TEST
3657
3658 016246 013777 004506 163712  MOV      @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
3659
3660 016254 013746 002334  MOV      @#SERCH,-(SP)  ;GET READY TO MOVE COMMAND
3661 016260 052716 000101  BIS      #GO!IE,(SP)    ;GET READY TO SET GO AND
3662 016264 012677 163710  MOV      (SP)+,@RHCS1   ;GO WITH
3663 016270 011100          MOV      @R1,R0         ;SAVE RHCS1 DURING ABOVE OPERATION
3664 016272 011305          MOV      @R3,R5         ;SAVE RHDS1 DURING ABOVE OPERATION
3665
3666 016274 104413          WAT                        ;WAIT FOR DRY BIT TO SET
3667 016276 002222          RHDS1                    ;WAIT FOR RHDS1 REGISTER
3668 016300 000200          DRY                      ;WAIT FOR DRY BIT IN RHDS1 REGISTER
3669 016302 001614          908.                   ;ALLOW 9080 MICRO SECONDS
  
```



```

3670 016304 001507      839.      ;DRY MUST SET BETWEEN
3671 016306 013746 002334  MOV      @#SERCH,-(SP) ;SAVE COMMAND
3672 016312 052716 004301  BIS      #IE!GO!DVA!RDY,(SP) ;INCLUDE IE!GO!DVA!RDY
3673 016316 011637 001124  MOV      (SP),@#$GDDAT ;SAVE FOR PRINTOUT
3674 016322 022600      CMP      (SP)+,R0 ;DURING ABOVE OPERATION ONLY IE!GO!DVA!RDY
3675 016324 001405      BEQ      67$ ;BRANCH IF GOOD
3676 016326 010037 001126  MOV      R0,@#$BDDAT ;BAD DATA
3677 016332 010137 004500  MOV      R1,@#REGADR ;FAILING REGISTER RHCS1
3678 016336 104021      ERROR   21 ;DURING ABOVE OPERATION ONLY
3679 016340 012746 010500  67$:    MOV      #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
3680 016344 011637 001124  MOV      (SP),@#$GDDAT ;SAVE FOR PRINTOUT
3681 016350 022605      CMP      (SP)+,R5 ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
3682 016352 001405      BEQ      69$ ;BRANCH IF GOOD
3683 016354 010537 001126  MOV      R5,@#$BDDAT ;BAD DATA
3684 016360 010337 004500  MOV      R3,@#REGADR ;FAILING REGISTER RHDS1
3685 016364 104063      ERROR   63 ;DURING ABOVE OPERATION ONLY
3686
3687      ;*CHANGE SAVED REGISTERS TO EXPECTED VALUE
3688 016366 004037 033762  JSR      RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
3689 016372 002200      RHCS1   ;CHANGE RHCS1 REGISTER
3690 016374 000001      1       ;1 BIT/BITS TO BE CHANGED
3691 016376 000001      1       ;NEW VALUE OF SC IS 1
3692 016400 100000      SC      ;CHANGE SC BIT
3693 016402 004037 033762  JSR      RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
3694 016406 002222      RHDS1   ;CHANGE RHDS1 REGISTER
3695 016410 000001      1       ;1 BIT/BITS TO BE CHANGED
3696 016412 000001      1       ;NEW VALUE OF ATA IS 1
3697 016414 100000      ATA     ;CHANGE ATA BIT
3698 016416 053737 004644 004536  BIS      @#ATTENT,@#SAVERE+24 ;SET APPROPRIATE 'ATA' BITS
3699
3700      ;*COMPARE REGISTERS BEFORE SEARCH WITH AFTER SEARCH
3701 016424 004037 034070  JSR      RO,@#COMREG ;COMPARE SAVED REGISTERS WITH
3702 016430 004512      SAVERE  ;GOOD DATA SAVED IN 'SAVERE'
3703 016432 002254      WC      ;TEST DATA STARTING FROM 'RHWC'
3704 016434 000022      18.    ;18. REGISTERS TO BE COMPARED
3705 016436 016442      1$     ;RETURN TO 1$ ON ERROR
3706 016440 016446      2$     ;RETURN TO 2$ ON NO ERROR
3707
3708 016442 104047      1$:    ERROR 47 ;SEARCH TO A NON-EXISTANT
3709
3710 016444 000207      RTS     PC ;SECTOR CAUSED IMPROPER
3711      ;REGISTER CHANGE
3712      ;GOOD DATA GIVES WHAT SHOULD
3713      ;BE THERE
3714      ;RECEIVED DATA GIVES
3715      ;WHAT WAS THERE AFTER
3716      ;SEARCH
3717 016446      2$:
3718

```



```

3719 016446 000004          TST17: SCOPE
3720 016450 012706 001000  MOV      #STACK,SP      ;RESET STACK
3721 016454 012737 000017 004504  MOV      #17,@#TSTNM    ;SAVE TEST NUMBER
3722 016462 004737 033072  JSR      PC,@#CLDISK    ;SET R1-RHCS1, R2-RHCS2
3723
3724                      ;*FILL WRITE FROM BUFFER WITH HEADER
3725 016466 004037 032716  JSR      RO,@#FLHEAD    ;SAVE HEADER DATA IN WRFROM
3726 016472 002370          WRFROM          ;LOCATION WHERE SAVED
3727 016474 000004          4                ;NUMBER OF WORDS SAVED
3728 016476 010000          10000           ;FIRST DATA WORD
3729 016500 000001          1                ;SECOND DATA WORD
3730 016502 000000          0                ;THIRD DATA WORD
3731 016504 000000          0                ;FOURTH DATA WORD
3732
3733                      ;*FILL WRITE FROM BUFFER WITH DATA
3734 016506 004037 032742  JSR      RO,@#CLAREA    ;CLEAR 256. WORDS, FROM WRFROM+10
3735 016512 002400          WRFROM+10       ;STARTING FROM WRFROM+10
3736 016514 000400          256.            ;256. WORDS
3737 016516 000000          0                ;FILL WITH 0
3738
3739                      ;*NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA
3740                      ;*AS WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
3741                      ;*CAN BE MADE TO MAKE SURE THAT WRITE DID NOT
3742                      ;*CHANGE WRITE FROM BUFFER
3743
3744 016520 004037 032716  JSR      RO,@#FLHEAD    ;SAVE HEADER DATA IN REINTO
3745 016524 003434          REINTO          ;LOCATION WHERE SAVED
3746 016526 000004          4                ;NUMBER OF WORDS SAVED
3747 016530 010000          10000           ;FIRST DATA WORD
3748 016532 000001          1                ;SECOND DATA WORD
3749 016534 000000          0                ;THIRD DATA WORD
3750 016536 000000          0                ;FOURTH DATA WORD
3751 016540 004037 032742  JSR      RO,@#CLAREA    ;CLEAR 256. WORDS, FROM REINTO+10
3752 016544 003444          REINTO+10       ;STARTING FROM REINTO+10
3753 016546 000400          256.            ;256. WORDS
3754 016550 000000          0                ;FILL WITH 0
3755
3756                      ;*NOW THE WRITE HEADER AND DATA COMMAND WILL BE FILLED
3757 016552 004037 035054  JSR      RO,@#RUN      ;SETUP TO RUN FOR DATA COMMAND
3758 016556 000000          0                ;CYLINDER 0
3759 016560 001          .BYTE 1          ;SECTOR 1
3760 016561 000          .BYTE 0          ;TRACK 0
3761 016562 177374          -256.-4         ;WORD COUNT (DATA) = 256. +
3762 016564 002370          WRFROM          ;BUS ADDRESS
3763 016566 000000          0                ;DO NOT INHIBIT BUS ADDRESS INCREMENT
3764 016570 010000          FMT22           ;16 BITS PER WORD FORMAT
3765 016572 002344          WRIFOR          ;GET READY TO DO A WRIFOR
3766
3767                      ;*NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE HEADER AND DATA
3768 016574 004037 033240  JSR      RO,@#SAVER    ;SAVE REGISTERS
3769 016600 002172          RHWC            ;RHWC IS THE FIRST REGISTER SAVED
3770 016602 004512          SAVERE         ;STARTING ADDRES OF WHERE
3771 016604 000021          17.            ;NUMBER OF REGISTERS
3772
3773 016606 004737 033152  JSR      PC,@#CHECKT   ;CHECKS DVA,RDY,MOL,DPP,DRY AND VV = 1
3774 016612 104401 057176  TYPE          .CPHALT ;CANNOT CONTINUE TESTING IF NOT
  
```



```

3775 016616 000000 HALT ;STOP TEST
3776
3777 016620 013777 004506 163340 MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
3778
3779 016626 013746 002344 MOV @#WRIFOR,-(SP) ;GET READY TO MOVE COMMAND
3780 016632 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET GO AND
3781 016636 012677 163336 MOV (SP)+,@RHCS1 ;GO WITH
3782 016642 011100 MOV @R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
3783 016644 011305 MOV @R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
3784 ;*ONE REVOLUTION=16670 MICRO SEC, ONE SECTOR = 760 MICRO SEC
3785 016646 104413 WAT ;WAIT FOR RDY BIT TO SET
3786 016650 002200 RHCS1 ;WAIT FOR RHCS1 REGISTER
3787 016652 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
3788 016654 001614 908. ;ALLOW 9080 MICRO SECONDS
3789 016656 001507 839. ;RDY MUST SET BETWEEN
3790 016660 013746 002344 MOV @#WRIFOR,-(SP) ;SAVE COMMAND
3791 016664 052716 004101 BIS #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
3792 016670 011637 001124 MOV (SP),@#$GDDAT ;SAVE FOR PRINTOUT
3793 016674 022600 CMP (SP)+,R0 ;DURING ABOVE OPERATION ONLY IE!GO!DVA
3794 016676 001405 BEQ 64$ ;BRANCH IF GOOD
3795 016700 010037 001126 MOV R0,@#$BDDAT ;BAD DATA
3796 016704 010137 004500 MOV R1,@#REGADR ;FAILING REGISTER RHCS1
3797 016710 104021 ERROR 21 ;DURING ABOVE OPERATION ONLY
3798 016712 012746 010500 64$: MOV #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
3799 016716 011637 001124 MOV (SP),@#$GDDAT ;SAVE FOR PRINTOUT
3800 016722 022605 CMP (SP)+,R5 ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
3801 016724 001405 BEQ 66$ ;BRANCH IF GOOD
3802 016726 010537 001126 MOV R5,@#$BDDAT ;BAD DATA
3803 016732 010337 004500 MOV R3,@#REGADR ;FAILING REGISTER RHDS1
3804 016736 104063 ERROR 63 ;DURING ABOVE OPERATION ONLY
3805
3806 ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
3807 016740 004037 032774 JSR R0,@#FILLRE ;MOV 0 INTO SAVED RHWC
3808 016744 002172 RHWC ;SAVED REGISTER TO CHANGE
3809 016746 000000 0 ;DATA
3810 016750 004037 032774 JSR R0,@#FILLRE ;MOV WRFROM+<260.*2> INTO SAVED RHBA
3811 016754 002174 RHBA ;SAVED REGISTER TO CHANGE
3812 016756 003400 WRFROM+<260.*2> ;DATA
3813 016760 004037 032774 JSR R0,@#FILLRE ;MOV 2 INTO SAVED RHDST
3814 016764 002204 RHDST ;SAVED REGISTER TO CHANGE
3815 016766 000002 2 ;DATA
3816
3817 ;*NOW COMPARE REGISTERS BEFORE WRITE HEADER AND DATA
3818 ;*WITH REGISTERS AFTER COMMAND
3819
3820 016770 004037 034070 JSR R0,@#COMREG ;COMPARE SAVED REGISTERS WITH
3821 016774 004512 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
3822 016776 002254 WC ;TEST DATA STARTING FROM 'RHWC'
3823 017000 000021 17. ;17. REGISTERS TO BE COMPARED
3824 017002 017006 1$ ;RETURN TO 1$ ON ERROR
3825 017004 017012 2$ ;RETURN TO 2$ ON NO ERROR
3826
3827 017006 104027 1$: ERROR 27 ;WRITE HEADER AND DATA
3828 017010 000207 RTS PC ;CAUSED IMPROPER REGISTER
3829 ;CHANGE
3830 ;GOOD DATA GIVES WHAT SHOULD
    
```



```

3887 017150 002200          RHCS1          :WAIT FOR RHCS1 REGISTER
3888 017152 000200          RDY           :WAIT FOR RDY BIT IN RHCS1 REGISTER
3889 017154 001614          908.         :ALLOW 9080 MICRO SECONDS
3890 017156 001507          839.         :RDY MUST SET BETWEEN
3891 017160 013746 002350  MOV @#REFOR,-(SP) :SAVE COMMAND
3892 017164 052716 004101  BIS #IE!GO!DVA,(SP) :INCLUDE IE!GO!DVA
3893 017170 011637 001124  MOV (SP),@#$GDDAT :SAVE FOR PRINTOUT
3894 017174 022600          CMP (SP)+,R0   :DURING ABOVE OPERATION ONLY IE!GO!DVA
3895 017176 001405          BEQ 67$       :BRANCH IF GOOD
3896 017200 010037 001126  MOV R0,@#$BDDAT :BAD DATA
3897 017204 010137 004500  MOV R1,@#REGADR :FAILING REGISTER RHCS1
3898 017210 104021          ERROR 21      :DURING ABOVE OPERATION ONLY
3899 017212 012746 010500  67$: MOV #MOL!DPR!VV,-(SP) :SAVE BITS SET DURING OPERATION IN RHDS1
3900 017216 011637 001124  MOV (SP),@#$GDDAT :SAVE FOR PRINTOUT
3901 017222 022605          CMP (SP)+,R5   :DURING ABOVE OPERATION ONLY MOL!DPR!VV
3902 017224 001405          BEQ 69$       :BRANCH IF GOOD
3903 017226 010537 001126  MOV R5,@#$BDDAT :BAD DATA
3904 017232 010337 004500  MOV R3,@#REGADR :FAILING REGISTER RHDS1
3905 017236 104063          ERROR 63      :DURING ABOVE OPERATION ONLY
3906
3907          ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
3908 017240 004037 032774  JSR R0,@#FILLRE :MOV 0 INTO SAVED RHWC
3909 017244 002172          RHWC         :SAVED REGISTER TO CHANGE
3910 017246 000000          0           :DATA
3911 017250 004037 032774  JSR R0,@#FILLRE :MOV REINTO+<260.*2> INTO SAVED RHBA
3912 017254 002174          RHBA         :SAVED REGISTER TO CHANGE
3913 017256 004444          REINTO+<260.*2> :DATA
3914 017260 004037 032774  JSR R0,@#FILLRE :MOV 2 INTO SAVED RHDST
3915 017264 002204          RHDST        :SAVED REGISTER TO CHANGE
3916 017266 000002          2           :DATA
3917
3918          ;*COMPARE REGISTER BEFORE READ HEADER AND DATA
3919          ;*WITH REGISTERS AFTER COMMAND
3920
3921 017270 004037 034070  JSR R0,@#COMREG :COMPARE SAVED REGISTERS WITH
3922 017274 004512          SAVERE       :GOOD DATA SAVED IN 'SAVERE'
3923 017276 002254          WC          :TEST DATA STARTING FROM 'RHWC'
3924 017300 000022          18.         :18. REGISTERS TO BE COMPARED
3925 017302 017306          5$          :RETURN TO 5$ ON ERROR
3926 017304 017312          6$          :RETURN TO 6$ ON NO ERROR
3927
3928 017306 104031 5$: ERROR 31      :READ HEADER AND DATA CAUSED
3929 017310 000207          RTS PC       :IMPROPER REGISTER CHANGE
3930          :GOOD DATA GIVES WHAT SHOULD
3931          :BE THERE
3932          :RECEIVED DATA GIVES WHAT WAS
3933          :THERE AFTER COMMAND
3934
3935          ;*NOW READ INTO BUFFER WILL BE CHECKED TO SEE
3936          ;*THE READ WAS GOOD
3937
3938 017312 6$: JSR R0,@#COMPAR :COMPARE TWO BLOCKS OF MEMORY
3939 017312 004037 035120  WRFROM        :GOOD DATA STARTS FROM WRFROM
3940 017316 002370          REINTO       :TEST DATA STARTS FROM REINTO
3941 017320 003434          260.        :260., WORDS TO BE COMPARED
3942 017322 000404
    
```



```

3954
3955 017334 000004          TST20: SCOPE
3956 017336 012706 001000  MOV      #STACK,SP      ;RESET STACK
3957 017342 012737 000020 004504  MOV      #20,@#TSTNM    ;SAVE TEST NUMBER
3958 017350 004737 033072  JSR      PC,@#CLDISK    ;SET R1-RHCS1, R2-RHCS2
3959
3960
3961          ;*CHECK THE DRIVE TYPE AND THEN FILL THE
3962          ;*WRITE FROM BUFFER WITH APPROPRIATE HEADER
3963
3964 017354 005737 004636  TST      @#RP06 ;TEST FOR RP06 DRIVE
3965 017360 001411  BEQ      5$          ;TREAT UNIT AS AN RP04
3966          ;TREAT AS AN RP06
3967
3968 017362 004037 032716  JSR      R0,@#FLHEAD    ;SAVE HEADER DATA IN WRFROM
3969 017366 002370  WRFROM    ;LOCATION WHERE SAVED
3970 017370 000004  4          ;NUMBER OF WORDS SAVED
3971 017372 011457  11457     ;FIRST DATA WORD
3972 017374 000000  0          ;SECOND DATA WORD
3973 017376 000000  0          ;THIRD DATA WORD
3974 017400 000000  0          ;FOURTH DATA WORD
3975 017402 000410  BR       6$          ;CONTINUE WITH SET UP
3976
3977 017404          5$:
3978 017404 004037 032716  JSR      R0,@#FLHEAD    ;SAVE HEADER DATA IN WRFROM
3979 017410 002370  WRFROM    ;LOCATION WHERE SAVED
3980 017412 000004  4          ;NUMBER OF WORDS SAVED
3981 017414 010633  10633     ;FIRST DATA WORD
3982 017416 000000  0          ;SECOND DATA WORD
3983 017420 000000  0          ;THIRD DATA WORD
3984 017422 000000  0          ;FOURTH DATA WORD
3985 017424          6$:
3986          ;*FILL WRITE FROM BUFFER WITH DATA
3987 017424 004037 032742  JSR      R0,@#CLAREA    ;CLEAR 256. WORDS, FROM WRFROM+10
3988 017430 002400  WRFROM+10 ;STARTING FROM WRFROM+10
3989 017432 000400  256.      ;256. WORDS
3990 017434 177777  0-1      ;FILL WITH 0-1
3991
3992
3993          ;*THE DRIVE TYPE IS CHECKED AND THE APPROPRIATE
3994          ;*WRITE HEADER AND DATA COMMAND IS LOADED
3995
3996
3997 017436 005737 004636  TST      @#RP06 ;TEST FOR RP06 DRIVE
3998 017442 001412  BEQ      3$          ;TREAT UNIT AS RP04
3999          ;TREAT UNIT AS RP06
4000
4001
4002 017444 004037 035054  JSR      R0,@#RUN      ;SETUP TO RUN FOR DATA COMMAND
4003 017450 001457  815.      ;CYLINDER 815.
4004 017452 000      .BYTE 0          ;SECTOR 0
4005 017453 000      .BYTE 0          ;TRACK 0
4006 017454 177374  -256.-4    ;WORD COUNT (DATA) = 256. +
4007 017456 002370  WRFROM    ;BUS ADDRESS
4008 017460 000000  0          ;DO NOT INHIBIT BUS ADDRESS INCREMENT
4009 017462 010000  FMT22     ;16 BITS PER WORD FORMAT

```



```

4101
4102 017714 000004          TST21: SCOPE
4103 017716 012706 001000  MOV      #STACK,SP      ;RESET STACK
4104 017722 012737 000021 004504  MOV      #21,@#TSTNM    ;SAVE TEST NUMBER
4105 017730 004737 033072          JSR      PC,@#CLDISK    ;SET R1-RHCS1, R2-RHCS2
4106
4107          ;*FILL WRITE FROM BUFFER WITH DATA
4108 017734 004037 032742  JSR      RO,@#CLAREA    ;CLEAR 256. WORDS, FROM WRFROM
4109 017740 002370          WRFROM          ;STARTING FROM WRFROM
4110 017742 000400          256.           ;256. WORDS
4111 017744 000377          377            ;FILL WITH 377
4112
4113          ;*WRITE DATA COMMAND WILL BE FILLED
4114 017746 004037 035054  JSR      RO,@#RUN       ;SETUP TO RUN FOR DATA COMMAND
4115 017752 000000          0              ;CYLINDER 0
4116 017754 000          .BYTE 0          ;SECTOR 0
4117 017755 002          .BYTE 19.        ;TRACK 19.
4118 017756 177400          -256.         ;WORD COUNT = 256.
4119 017760 002370          WRFROM        ;BUS ADDRESS
4120 017762 000000          0             ;DO NOT INHIBIT BUS ADDRESS INCREMENT
4121 017764 010000          FMT22        ;16 BITS PER WORD FORMAT
4122 017766 002342          WRIDAT       ;GET READY TO DO A WRIDAT
4123
4124          ;*SAVE REGISTERS FOR COMPARISON AFTER ATTEMPTED WRITE DATA
4125 017770 004037 033240  JSR      RO,@#SAVER    ;SAVE REGISTERS
4126 017774 002172          RHWC         ;RHWC IS THE FIRST REGISTER SAVED
4127 017776 004512          SAVERE      ;STARTING ADDRES OF WHERE
4128 020000 000022          18.         ;NUMBER OF REGISTERS
4129
4130 020002 004737 033152  JSR      PC,@#CHECKT   ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
4131 020006 104401 057176  TYPE      ,CPHALT     ;CANNOT CONTINUE TESTING IF NOT
4132 020012 000000          HALT        ;STOP TEST
4133
4134 020014 013777 004506 162144  MOV      @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
4135
4136 020022 013746 002342  MOV      @#WRIDAT,-(SP) ;GET READY TO MOVE COMMAND
4137 020026 052716 000101  BIS      #GO!IE,(SP)   ;GET READY TO SET GO AND
4138 020032 012677 162142  MOV      (SP)+,@RHCS1  ;GO WITH
4139
4140 020036 104413          WAT         ;WAIT FOR IAE BIT TO SET
4141 020040 002202          RHER1      ;WAIT FOR RHER1 REGISTER
4142 020042 002000          IAE       ;WAIT FOR IAE BIT IN RHER1 REGISTER
4143 020044 000011          9.         ;ALLOW 90 MICRO SECONDS
4144 020046 000011          9.         ;IAE MUST SET BETWEEN
4145
4146          ;*CHANGE          SAVED REGISTERS TO EXPECTED VALUES
4147
4148
4149
4150          ;*AS EXCEPTION IS ASSERTED BEFORE RUN IS LATCHED
4151          ;*RHWC,RHBA,RHCS1,RHCS2, CANNOT BE PEREDETERMINED
4152          ;*THEY WILL VARY DEPENDING ON GATE DELAYS ON DIFFRENT UNITS
4153
4154 020050 017737 162116 004512  MOV      @RHWC,@#SAVERE ;RHWC IS UNPREDICTABLE
4155          ;AS EXPLAINED ABOVE
4156 020056 017737 162112 004514  MOV      @RHBA,@#SAVERE+2;RHBA IS UNPREDICTABLE
  
```



```

4196
4197 020172 000004          TST22: SCOPE
4198 020174 012706 001000  MOV      #STACK,SP      ;RESET STACK
4199 020200 012737 000022 004504  MOV      #22,@#TSTNM    ;SAVE TEST NUMBER
4200 020206 004737 033072  JSR      PC,@#CLDISK    ;SET R1-RHCS1, R2-RHCS2
4201
4202          ;*GET THE HEADS TO CYLINDER 10
4203
4204 020212 004737 033152  JSR      PC,@#CHECKT    ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
4205 020216 104401 057176  TYPE     ,CPHALT        ;CANNOT CONTINUE TESTING IF NOT
4206 020222 000000          HALT                    ;STOP TEST
4207
4208 020224 013777 004506 161734  MOV      @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
4209
4210 020232 004037 033042  JSR      RO,@#SEEKCY    ;SEEK FOR
4211 020236 000010          10                     ;CYLINDER 10
4212
4213 020240 013746 002352  MOV      @#SEECOM,-(SP)  ;GET READY TO MOVE COMMAND
4214 020244 052716 000101  BIS      #GO!IE,(SP)    ;GET READY TO SET GO AND
4215 020250 012677 161724  MOV      (SP)+,@RHCS1   ;GO WITH
4216
4217 020254 104413          WAT                    ;WAIT FOR DRY BIT TO SET
4218 020256 002222          RHDS1                 ;WAIT FOR RHDS1 REGISTER
4219 020260 000200          DRY                   ;WAIT FOR DRY BIT IN RHDS1 REGISTER
4220 020262 015530          7000.                 ;ALLOW 70000 MICRO SECONDS
4221 020264 015530          7000.                 ;DRY MUST SET BETWEEN
4222
4223 020266 004737 033072  JSR      PC,@#CLDISK    ;SET R1-RHCS1, R2-RHCS2
4224
4225          ;*FILL READ INTO BUFFER WITH 125252
4226 020272 004037 032742  JSR      RO,@#CLAREA    ;CLEAR 260 WORDS, FROM REINTO
4227 020276 003434          REINTO                ;STARTING FROM REINTO
4228 020300 000260          260                   ;260 WORDS
4229 020302 125252          125252                 ;FILL WITH 125252
4230
4231          ;*THE READ HEADER AND DATA COMMAND IS FILLED
4232 020304 004037 035054  JSR      RO,@#RUN       ;SETUP TO RUN FOR DATA COMMAND
4233 020310 000000          0                       ;CYLINDER 0
4234 020312          026                 ;SECTOR 22.
4235 020313          000                 ;TRACK 0
4236 020314 177374          -256.-4                ;WORD COUNT (DATA) = 256. +
4237 020316 003434          REINTO                 ;BUS ADDRESS
4238 020320 000000          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
4239 020322 014000          FMT22!ECI             ;16 BITS PER WORD FORMAT
4240 020324 002350          REFOR                  ;GET READY TO DO A REFOR
4241
4242          ;*SAVE REGISTERS FOR COMPARISON AFTER ATTEMPTED READ
4243 020326 004037 033240  JSR      RO,@#SAVER     ;SAVE REGISTERS
4244 020332 002172          RHWC                   ;RHWC IS THE FIRST REGISTER SAVED
4245 020334 004512          SAVERE                 ;STARTING ADDRES OF WHERE
4246 020336 000022          18.                    ;NUMBER OF REGISTERS
4247
4248 020340 004737 033152  JSR      PC,@#CHECKT    ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
4249 020344 104401 057176  TYPE     ,CPHALT        ;CANNOT CONTINUE TESTING IF NOT
4250 020350 000000          HALT                    ;STOP TEST
4251

```



```

4252 020352 013777 004506 161606 MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
4253 020360 013746 002350 MOV @#REFOR,-(SP) ;GET READY TO MOVE COMMAND
4254 020364 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET GO AND
4255 020370 012677 161604 MOV (SP)+,@RHCS1 ;GO WITH
4256
4257 020374 104413 WAT ;WAIT FOR IAE BIT TO SET
4258 020376 002202 RHER1 ;WAIT FOR RHER1 REGISTER
4259 020400 002000 IAE ;WAIT FOR IAE BIT IN RHER1 REGISTER
4260 020402 000002 2. ;ALLOW 20 MICRO SECONDS
4261 020404 000002 2. ;IAE MUST SET BETWEEN
4262
4263 ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
4264
4265
4266
4267 ;*AS EXCEPTION IS ASSERTED BEFORE RUN IS LATCHED
4268 ;*RHWC,RHBA,RHCS1,RHCS2, CANNOT BE PEREDETERMINED
4269 ;*THEY WILL VARY DEPENDING ON GATE DELAYS ON DIFFRENT UNITS
4270
4271 020406 017737 161560 004512 MOV @RHWC,@#SAVERE ;RHWC IS UNPREDICTABLE
4272 ;AS EXPLAINED ABOVE
4273 020414 017737 161554 004514 MOV @RHBA,@#SAVERE+2 ;RHBA IS UNPREDICTABLE
4274 ;AS EXPLAINED ABOVE
4275 020422 017737 161550 004516 MOV @RHCS2,@#SAVERE+4 ;RHCS2 IS UNPREDICTABLE
4276 ;AS EXPLAINED ABOVE
4277 020430 017737 161544 004520 MOV @RHCS1,@#SAVERE+6 ;RHCS1 IS UNPREDICTABLE
4278 ;AS EXPLAINED ABOVE
4279 020436 004037 033762 JSR R0,@#CHREG ;CHANGE BITS IN SAVED REGISTER
4280 020442 002222 RHDS1 ;CHANGE RHDS1 REGISTER
4281 020444 000002 2 ;2 BIT/BITS TO BE CHANGED
4282 020446 000001 1 ;NEW VALUE OF ATA IS 1
4283 020450 100000 ATA ;CHANGE ATA BIT
4284 020452 000001 1 ;NEW VALUE OF ERR IS 1
4285 020454 040000 ERR ;CHANGE ERR BIT
4286 020456 017737 161522 004524 MOV @RHDST,@#SAVERE+12 ;RHDST IS INDETERMINATE SO IT IS NOT CHECKED
4287
4288 020464 004037 033762 JSR R0,@#CHREG ;CHANGE BITS IN SAVED REGISTER
4289 020470 002202 RHER1 ;CHANGE RHER1 REGISTER
4290 020472 000001 1 ;1 BIT/BITS TO BE CHANGED
4291 020474 000001 1 ;NEW VALUE OF IAE IS 1
4292 020476 002000 IAE ;CHANGE IAE BIT
4293 020500 053737 004644 004536 BIS @#ATTENT,@#SAVERE+24 ;SET APPROPRIATE 'ATA' BITS
4294
4295 ;*COMPARE REGISTERS BEFORE ATTEMPTED READ HEADER
4296 ;*AND DATA WITH AFTER ATTEMPTED READ
4297 020506 004037 034070 JSR R0,@#COMREG ;COMPARE SAVED REGISTERS WITH
4298 020512 004512 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
4299 020514 002254 WC ;TEST DATA STARTING FROM 'RHWC'
4300 020516 000022 18. ;18. REGISTERS TO BE COMPARED
4301 020520 020524 1$ ;RETURN TO 1$ ON ERROR
4302 020522 020530 2$ ;RETURN TO 2$ ON NO ERROR
4303
4304 020524 104054 1$: ERROR 54 ;ATTEMPTED READ HEADER
4305 020526 000207 RTS PC ;AND DATA WITH INVALID
4306 ;ADDRESS CAUSED IMPROPER
4307 ;REGISTER CHANGE

```


4308
4309
4310
4311
4312
4313
4314 020530
4315

2\$:

;GOOD DATA GIVES WHAT
;SHOULD BE THERE
;RECEIVED DATA GIVES
;REGISTER CONTENTS
;AFTER ATTEMPTED
;READ


```

4316
4317 020530 000004          TST23: SCOPE
4318 020532 012706 001000  MOV      #STACK,SP      ;RESET STACK
4319 020536 012737 000023 004504  MOV      #23,@#TSTNM    ;SAVE TEST NUMBER
4320 020544 004737 033072          JSR      PC,@#CLDISK    ;SET R1-RHCS1, R2-RHCS2
4321
4322          ;*FILL READ INTO BUFFER WITH 125252
4323 020550 004037 032742  JSR      RO,@#CLAREA    ;CLEAR 260 WORDS, FROM REINTO
4324 020554 003434          REINTO   ;STARTING FROM REINTO
4325 020556 000260          260      ;260 WORDS
4326 020560 125252          125252   ;FILL WITH 125252
4327
4328          ;*THE READ HEADER AND DATA COMMAND IS FILLED
4329 020562 004037 035054  JSR      RO,@#RUN      ;SETUP TO RUN FOR DATA COMMAND
4330 020566 000000          0          ;CYLINDER 0
4331 020570          024      .BYTE    20.          ;SECTOR 20.
4332 020571          000      .BYTE    0          ;TRACK 0
4333 020572 177400          -256.     ;WORD COUNT = 256.
4334 020574 003434          REINTO   ;BUS ADDRESS
4335 020576 000000          0          ;DO NOT INHIBIT BUS ADDRESS INCREMENT
4336 020600 004000          ECI      ;18 BITS PER WORD FORMAT
4337 020602 002346          READAT   ;GET READY TO DO A READAT
4338
4339          ;*SAVE REGISTERS FOR COMPARISON AFTER ATTEMPTED READ
4340 020604 004037 033240  JSR      RO,@#SAVER    ;SAVE REGISTERS
4341 020610 002172          RHWC     ;RHWC IS THE FIRST REGISTER SAVED
4342 020612 004512          SAVERE   ;STARTING ADDRES OF WHERE
4343 020614 000022          18.     ;NUMBER OF REGISTERS
4344
4345 020616 004737 033152  JSR      PC,@#CHECKT   ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
4346 020622 104401 057176  TYPE    ,CPHALT       ;CANNOT CONTINUE TESTING IF NOT
4347 020626 000000          HALT    ;STOP TEST
4348
4349 020630 013777 004506 161330  MOV      @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
4350
4351 020636 013746 002346  MOV      @#READAT,-(SP) ;GET READY TO MOVE COMMAND
4352 020642 052716 000101  BIS      #GO!IE,(SP)   ;GET READY TO SET GO AND
4353 020646 012677 161326  MOV      (SP)+,@RHCS1  ;GO WITH
4354
4355 020652 104413          WAT      ;WAIT FOR IAE BIT TO SET
4356 020654 002202          RHER1   ;WAIT FOR RHER1 REGISTER
4357 020656 002000          IAE     ;WAIT FOR IAE BIT IN RHER1 REGISTER
4358 020660 000002          2.     ;ALLOW 20 MICRO SECONDS
4359 020662 000002          2.     ;IAE MUST SET BETWEEN
4360
4361          ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
4362 020664 004037 033762  JSR      RO,@#CHREG    ;CHANGE BITS IN SAVED REGISTER
4363 020670 002200          RHCS1   ;CHANGE RHCS1 REGISTER
4364 020672 000002          2       ;2 BIT/BITS TO BE CHANGED
4365 020674 000001          1       ;NEW VALUE OF SC IS 1
4366 020676 100000          SC      ;CHANGE SC BIT
4367 020700 000001          1       ;NEW VALUE OF TRE IS 1
4368 020702 040000          TRE    ;CHANGE TRE BIT
4369 020704 017737 161274 004524  MOV      @RHDST,@#SAVERE+12 ;RHDST IS UNPREDICTABLE
4370 020712 004037 033762  JSR      RO,@#CHREG    ;CHANGE BITS IN SAVED REGISTER
4371 020716 002222          RHDS1   ;CHANGE RHDS1 REGISTER
  
```



```

4372 020720 000002          2          ;2 BIT/BITS TO BE CHANGED
4373 020722 000001          1          ;NEW VALUE OF ATA IS 1
4374 020724 100000          ATA        ;CHANGE ATA BIT
4375 020726 000001          1          ;NEW VALUE OF ERR IS 1
4376 020730 040000          ERR        ;CHANGE ERR BIT
4377 020732 004037 033762  JSR      RO,@#CHREG  ;CHANGE BITS IN SAVED REGISTER
4378 020736 002202          RHER1     ;CHANGE RHER1 REGISTER
4379 020740 000001          1          ;1 BIT/BITS TO BE CHANGED
4380 020742 000001          1          ;NEW VALUE OF IAE IS 1
4381 020744 002000          IAE       ;CHANGE IAE BIT
4382 020746 053737 004644 004536  BIS      @#ATTENT,@#SAVERE+24 ;SET APPROPRIATE 'ATA' BITS
4383
4384          ;*COMPARE REGISTERS BEFORE ATTEMPTED READ
4385          ;*DATA WITH AFTER ATTEMPTED READ DATA
4386 020754 004037 034070  JSR      RO,@#COMREG  ;COMPARE SAVED REGISTERS WITH
4387 020760 004512          SAVERE    ;GOOD DATA SAVED IN 'SAVERE'
4388 020762 002254          WC         ;TEST DATA STARTING FROM 'RHWC'
4389 020764 000022          18.      ;18. REGISTERS TO BE COMPARED
4390 020766 020772          1$        ;RETURN TO 1$ ON ERROR
4391 020770 020776          2$        ;RETURN TO 2$ ON NO ERROR
4392
4393 020772 104054          1$:      ERROR   54      ;ATTEMPTED READ
4394 020774 000207          RTS      PC      ;DATA WITH INVALID
4395          ;ADDRESS CAUSED IMPROPER
4396          ;REGISTER CHANGE
4397          ;GOOD DATA GIVES WHAT
4398          ;SHOULD BE THERE
4399          ;RECEIVED DATA GIVES
4400          ;REGISTERS CONTENTS
4401          ;AFTER ATTEMPTED
4402          ;READ
4403 020776          2$:
4404
    
```



```

4405
4406 020776 000004          TST24: SCOPE
4407 021000 012706 001000  MOV      #STACK,SP      ;RESET STACK
4408 021004 012737 000024 004504  MOV      #24,@#TSTNM    ;SAVE TEST NUMBER
4409 021012 004737 033072  JSR      PC,@#CLDISK    ;SET R1-RHCS1, R2-RHCS2
4410
4411          ;*FILL WRITE FROM BUFFER WITH HEADER
4412 021016 004037 032716  JSR      RO,@#FLHEAD    ;SAVE HEADER DATA IN WRFROM
4413 021022 002370          WRFROM    ;LOCATION WHERE SAVED
4414 021024 000004          4          ;NUMBER OF WORDS SAVED
4415 021026 010000          10000     ;FIRST DATA WORD
4416 021030 000000          0          ;SECOND DATA WORD
4417 021032 000000          0          ;THIRD DATA WORD
4418 021034 000000          0          ;FOURTH DATA WORD
4419
4420          ;*FILL WRITE FROM BUFFER WITH DATA
4421 021036 004037 032742  JSR      RO,@#CLAREA    ;CLEAR 256. WORDS, FROM WRFROM+<4*2>
4422 021042 002400          WRFROM+<4*2> ;STARTING FROM WRFROM+<4*2>
4423 021044 000400          256.      ;256. WORDS
4424 021046 000000          0          ;FILL WITH 0
4425
4426
4427          ;*THE FIRST WRITE OPERATION IS DONE
4428          ;*FILL WRITE HEADER AND DATA COMMAND
4429 021050 004037 035054  JSR      RO,@#RUN      ;SETUP TO RUN FOR DATA COMMAND
4430 021054 000000          0          ;CYLINDER 0
4431 021056 000          .BYTE 0          ;SECTOR 0
4432 021057 000          .BYTE 0          ;TRACK 0
4433 021060 177374          -256.-4     ;WORD COUNT (DATA) = 256. +
4434 021062 002370          WRFROM      ;BUS ADDRESS
4435 021064 000000          0          ;DO NOT INHIBIT BUS ADDRESS INCREMENT
4436 021066 010000          FMT22      ;16 BITS PER WORD FORMAT
4437 021070 002344          WRIFOR     ;GET READY TO DO A WRIFOR
4438
4439 021072 004737 033152  JSR      PC,@#CHECKT   ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
4440 021076 104401 057176  TYPE      ,CPHALT     ;CANNOT CONTINUE TESTING IF NOT
4441 021102 000000          HALT       ;STOP TEST
4442
4443 021104 013777 004506 161054  MOV      @#RP4VEC,@RPVEC ;SET RPO4 VECTOR ADDRESS
4444
4445 021112 013746 002344  MOV      @#WRIFOR,-(SP) ;GET READY TO MOVE COMMAND
4446 021116 052716 000101  BIS      #GO!IE,(SP)   ;GET READY TO SET GO AND
4447 021122 012677 161052  MOV      (SP)+,@RHCS1  ;GO WITH
4448
4449 021126 104413          WAT        ;WAIT FOR RDY BIT TO SET
4450 021130 002200          RHCS1     ;WAIT FOR RHCS1 REGISTER
4451 021132 000200          RDY      ;WAIT FOR RDY BIT IN RHCS1 REGISTER
4452 021134 004704          2500.    ;ALLOW 25000 MICRO SECONDS
4453 021136 004704          2500.    ;RDY MUST SET BETWEEN
4454 021140 004737 033072  JSR      PC,@#CLDISK   ;SET R1-RHCS1, R2-RHCS2
4455
4456
4457          ;*CHECK THE DRIVE TYPE AND DO THE
4458          ;*APPROPRIATE SECOND WRITE OPERATION
4459
4460          ;*FILL WRITE FROM BUFFER WITH HEADER
  
```



```
4461
4462
4463 021144 005737 004636      TST   @#RP06 ;TEST FOR RP06 DRIVE
4464 021150 001411              BEQ   15$    ;TREAT DRIVE AS AN RP04
4465                          ;TREAT AS AN RP06
4466
4467 021152 004037 032716      JSR   R0,@#FLHEAD ;SAVE HEADER DATA IN WRFROM
4468 021156 002370              WRFROM ;LOCATION WHERE SAVED
4469 021160 000004              4      ;NUMBER OF WORDS SAVED
4470 021162 011456              11456 ;FIRST DATA WORD
4471 021164 011025              <18.*400>!<21.> ;SECOND DATA WORD
4472 021166 000000              0      ;THIRD DATA WORD
4473 021170 000000              0      ;FOURTH DATA WORD
4474 021172 000410              BR    16$    ;CONTINUE WITH THE SECOND WRITE
4475
4476
4477 021174              15$:
4478 021174 004037 032716      JSR   R0,@#FLHEAD ;SAVE HEADER DATA IN WRFROM
4479 021200 002370              WRFROM ;LOCATION WHERE SAVED
4480 021202 000004              4      ;NUMBER OF WORDS SAVED
4481 021204 010632              10632 ;FIRST DATA WORD
4482 021206 011025              <18.*400>!<21.> ;SECOND DATA WORD
4483 021210 000000              0      ;THIRD DATA WORD
4484 021212 000000              0      ;FOURTH DATA WORD
4485 021214              16$:
4486
4487 ;*FILL WRITE FROM BUFFER WITH DATA - 65125
4488 021214 004037 032742      JSR   R0,@#CLAREA ;CLEAR 256. WORDS, FROM WRFROM+<4*2>
4489 021220 002400              WRFROM+<4*2> ;STARTING FROM WRFROM+<4*2>
4490 021222 000400              256. ;256. WORDS
4491 021224 065125              <26.*2000>!<18.*40>!<21.> ;FILL WITH <26.*2000>!<18.*40>!<
4492
4493
4494 ;*CHECK THE DRIVE TYPE AND
4495 ;*FILL WRITE FROM BUFFER WITH APPROPRIATE NEXT HEADER
4496
4497 ;*THIS IS A NON EXISTANT HEADER AND SHOULD NOT BE WRITTEN
4498 ;*SINCE 'AOE' SHOULD INHIBIT THE WRITE OPERATION
4499
4500
4501 021226 005737 004636      TST   @#RP06 ;TEST FOR RP06 DRIVE
4502 021232 001411              BEQ   17$    ;TREAT UNIT AS AN RP04
4503                          ;TREAT AS AN RP06
4504
4505 021234 004037 032716      JSR   R0,@#FLHEAD ;SAVE HEADER DATA IN WRFROM+<260.*2>
4506 021240 003400              WRFROM+<260.*2> ;LOCATION WHERE SAVED
4507 021242 000004              4      ;NUMBER OF WORDS SAVED
4508 021244 011457              11457 ;FIRST DATA WORD
4509 021246 000000              0      ;SECOND DATA WORD
4510 021250 000000              0      ;THIRD DATA WORD
4511 021252 000000              0      ;FOURTH DATA WORD
4512 021254 000410              BR    18$    ;CONTINUE WITH TEST
4513
4514
4515 021256              17$:
4516 021256 004037 032716      JSR   R0,@#FLHEAD ;SAVE HEADER DATA IN WRFROM+<260.*2>
```



```

4517 021262 003400          WRFROM+<260.*2>          ;LOCATION WHERE SAVED
4518 021264 000004          4                          ;NUMBER OF WORDS SAVED
4519 021266 010633          10633                      ;FIRST DATA WORD
4520 021270 000000          0                          ;SECOND DATA WORD
4521 021272 000000          0                          ;THIRD DATA WORD
4522 021274 000000          0                          ;FOURTH DATA WORD
4523 021276                0                          ;CONTINUE
4524                18$:    ;*FILL WRITE FROM BUFFER WITH DATA FOR NEXT SECTOR
4525 021276 004037 032742  JSR    R0,@#CLAREA        ;CLEAR 2 WORDS, FROM WRFROM+<264.*2>
4526 021302 003410          WRFROM+<264.*2>          ;STARTING FROM WRFROM+<264.*2>
4527 021304 000002          2                          ;2 WORDS
4528 021306 066000          <27.*2000>                ;FILL WITH <27.*2000>
4529
4530
4531                ;*CHECK THE DRIVE TYPE AND DO THE APPROPRIATE
4532                ;*FILL WRITE HEADER AND DATA COMMAND
4533
4534
4535 021310 005737 004636    TST    @#RP06            ;TEST FOR RP06 DRIVE
4536 021314 001412          BEQ    7$                ;TREAT UNIT AS AN RP04
4537
4538
4539 021316 004037 035054    JSR    R0,@#RUN          ;SETUP TO RUN FOR DATA COMMAND
4540 021322 001456          814.                      ;CYLINDER 814.
4541 021324          025      .BYTE 21.                  ;SECTOR 21.
4542 021325          022      .BYTE 18.                  ;TRACK 18.
4543 021326 177373          -257.-4                   ;WORD COUNT (DATA) = 257. +
4544 021330 002370          WRFROM                    ;BUS ADDRESS
4545 021332 000000          0                          ;DO NOT INHIBIT BUS ADDRESS INCREMENT
4546 021334 010000          FMT22                     ;16 BITS PER WORD FORMAT
4547 021336 002344          WRIFOR                    ;GET READY TO DO A WRIFOR
4548 021340 000411          BR    8$
4549 021342                7$:
4550 021342 004037 035054    JSR    R0,@#RUN          ;SETUP TO RUN FOR DATA COMMAND
4551 021346 000632          410.                      ;CYLINDER 410.
4552 021350          025      .BYTE 21.                  ;SECTOR 21.
4553 021351          022      .BYTE 18.                  ;TRACK 18.
4554 021352 177373          -257.-4                   ;WORD COUNT (DATA) = 257. +
4555 021354 002370          WRFROM                    ;BUS ADDRESS
4556 021356 000000          0                          ;DO NOT INHIBIT BUS ADDRESS INCREMENT
4557 021360 010000          FMT22                     ;16 BITS PER WORD FORMAT
4558 021362 002344          WRIFOR                    ;GET READY TO DO A WRIFOR
4559 021364                8$:
4560
4561                ;*SAVE REGISTERS FOR COMPARISON AFTER WIRTE HEADER AND DATA
4562 021364 004037 033240    JSR    R0,@#SAVER        ;SAVE REGISTERS
4563 021370 002172          RHWC                      ;RHWC IS THE FIRST REGISTER SAVED
4564 021372 004512          SAVERE                    ;STARTING ADDRES OF WHERE
4565 021374 000022          18.                      ;NUMBER OF REGISTERS
4566
4567 021376 004737 033152    JSR    PC,@#CHECKT       ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
4568 021402 104401 057176    TYPE  ,CPHALT            ;CANNOT CONTINUE TESTING IF NOT
4569 021406 000000          HALT                      ;STOP TEST
4570
4571 021410 013777 004506 160550  MOV    @#RP4VEC,@RPVEC   ;SET RP04 VECTOR ADDRESS
4572
    
```



```

4573 021416 013746 002344      MOV    @#WRIFOR,-(SP)  ;GET READY TO MOVE COMMAND
4574 021422 052716 000101      BIS    #GO!IE,(SP)   ;GET READY TO SET GO AND
4575 021426 012677 160546      MOV    (SP)+,@RHCS1  ;GO WITH
4576
4577 021432 104413              WAT                      ;WAIT FOR RDY BIT TO SET
4578 021434 002200      RHCS1                    ;WAIT FOR RHCS1 REGISTER
4579 021436 000200      RDY                      ;WAIT FOR RDY BIT IN RHCS1 REGISTER
4580 021440 004704      2500.                   ;ALLOW 25000 MICRO SECONDS
4581 021442 004704      2500.                   ;RDY MUST SET BETWEEN
4582
4583                          ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
4584 021444 004037 032774      JSR    R0,@#FILLRE      ;MOV WRFROM+<260.*2>+<1.*2> INTO SAVED RHBA
4585 021450 002174      RHBA                    ;SAVED REGISTER TO CHANGE
4586 021452 003402      WRFROM+<260.*2>+<1.*2> ;DATA
4587 021454 004037 032774      JSR    R0,@#FILLRE      ;MOV 0 INTO SAVED RHWC
4588 021460 002172      RHWC                    ;SAVED REGISTER TO CHANGE
4589 021462 000000      0                        ;DATA
4590 021464 004037 033762      JSR    R0,@#CHREG       ;CHANGE BITS IN SAVED REGISTER
4591 021470 002200      RHCS1                   ;CHANGE RHCS1 REGISTER
4592 021472 000002      2                        ;2 BIT/BITS TO BE CHANGED
4593 021474 000001      1                        ;NEW VALUE OF SC IS 1
4594 021476 100000      SC                       ;CHANGE SC BIT
4595 021500 000001      1                        ;NEW VALUE OF TRE IS 1
4596 021502 040000      TRE                      ;CHANGE TRE BIT
4597 021504 004037 033762      JSR    R0,@#CHREG       ;CHANGE BITS IN SAVED REGISTER
4598 021510 002176      RHCS2                   ;CHANGE RHCS2 REGISTER
4599 021512 000002      2                        ;2 BIT/BITS TO BE CHANGED
4600 021514 000001      1                        ;NEW VALUE OF OR IS 1
4601 021516 000200      OR                       ;CHANGE OR BIT
4602 021520 000001      1                        ;NEW VALUE OF IR IS 1
4603 021522 000100      IR                       ;CHANGE IR BIT
4604 021524 004037 032774      JSR    R0,@#FILLRE      ;MOV AOE INTO SAVED RHER1
4605 021530 002202      RHER1                   ;SAVED REGISTER TO CHANGE
4606 021532 001000      AOE                     ;DATA
4607 021534 004037 033762      JSR    R0,@#CHREG       ;CHANGE BITS IN SAVED REGISTER
4608 021540 002222      RHDS1                   ;CHANGE RHDS1 REGISTER
4609 021542 000003      3                        ;3 BIT/BITS TO BE CHANGED
4610 021544 000001      1                        ;NEW VALUE OF ATA IS 1
4611 021546 100000      ATA                     ;CHANGE ATA BIT
4612 021550 000001      1                        ;NEW VALUE OF ERR IS 1
4613 021552 040000      ERR                     ;CHANGE ERR BIT
4614 021554 000001      1                        ;NEW VALUE OF LBT IS 1
4615 021556 002000      LBT                     ;CHANGE LBT BIT
4616 021560 053737 004644 004536  BIS    @#ATTENT,@#SAVERE+24 ;SET APPROPRIATE 'ATA' BITS
4617
4618                          ;*CHECK DEVICE TYPE BEFORE SETTING UP 'RHCA' & 'RHCC'
4619
4620 021566 005737 004636      TST    @#RP06 ;TEST FOR RP06 DRIVE
4621 021572 001411      BEQ    9$                ;TREAT AS RP04
4622                          ;TREAT AS RP06
4623
4624 021574 004037 032774      JSR    R0,@#FILLRE      ;MOV 815. INTO SAVED RHCA
4625 021600 002212      RHCA                    ;SAVED REGISTER TO CHANGE
4626 021602 001457      815.                   ;DATA
4627 021604 004037 032774      JSR    R0,@#FILLRE      ;MOV 814. INTO SAVED RHCC
4628 021610 002234      RHCC                    ;SAVED REGISTER TO CHANGE
    
```



```

4629 021612 001456      814.      :DATA
4630 021614 000410      BR        10$      :CONTINUE WITH TEST
4631 021616              9$:
4632 021616 004037 032774 JSR        RO,@#FILLRE ;MOV 411. INTO SAVED RHCA
4633 021622 002212      RHCA      :SAVED REGISTER TO CHANGE
4634 021624 000633      411.      :DATA
4635 021626 004037 032774 JSR        RO,@#FILLRE ;MOV 410. INTO SAVED RHCC
4636 021632 002234      RHCC      :SAVED REGISTER TO CHANGE
4637 021634 000632      410.      :DATA
4638 021636              10$:      :CONTINUE WITH TEST
4639
4640
4641
4642 021636 017737 160342 004524 MOV        @RHDST,@#SAVERE+12 ;RHDST IS UNPREDICTABLE
4643
4644      ;*COMPARE REGISTERS BEFORE WRITE HEADER AND DATA WITH AFTER
4645 021644 004037 034070 JSR        RO,@#COMREG ;COMPARE SAVED REGISTERS WITH
4646 021650 004512      SAVERE     ;GOOD DATA SAVED IN 'SAVERE'
4647 021652 002254      WC        ;TEST DATA STARTING FROM 'RHWC'
4648 021654 000022      18.      ;18. REGISTERS TO BE COMPARED
4649 021656 021662      1$       ;RETURN TO 1$ ON ERROR
4650 021660 021666      2$       ;RETURN TO 2$ ON NO ERROR
4651
4652 021662 104055      1$:      ERROR 55      ;WRITING HEADER AND DATA WITH
4653 021664 000207      RTS      PC    ;EXPECTED ADDRESS OVERFLOW ERROR
4654                          :CAUSED IMPROPER REGISTER
4655                          :CHANGE
4656                          :GOOD DATA GIVES WHAT SHOULD
4657                          :BE THERE
4658                          :RECEIVED DATA GIVES WHAT
4659                          :WAS THERE AFTER WRITE
4660                          :HEADER AND DATA
4661
  
```



```
4662
4663      ;*NOW PREPARE TO DO A READ HEADER AND DATA
4664      ;*(THE FIRST READ OPERATION)
4665
4666      ;*CHECK THE DRIVE TYPE AND FILL
4667      ;*WRITE FROM BUFFER WITH APPROPRIATE EXPECTED HEADER
4668 021666      2$: JSR    PC,@#CLDISK      ;SET R1-RHCS1, R2-RHCS2
4669 021666 004737 033072
4670
4671
4672 021672 005737 004636      TST    @#RP06      ;TEST FOR RP06 DRIVE
4673 021676 001411      BEQ    19$          ;TREAT UNIT AS AN RP04
4674                          ;TREAT AS AN RP06
4675
4676 021700 004037 032716      JSR    R0,@#FLHEAD      ;SAVE HEADER DATA IN WRFROM
4677 021704 002370      WRFROM      ;LOCATION WHERE SAVED
4678 021706 000004      4          ;NUMBER OF WORDS SAVED
4679 021710 011456      11456      ;FIRST DATA WORD
4680 021712 011025      <18.*400>!<21.>      ;SECOND DATA WORD
4681 021714 000000      0          ;THIRD DATA WORD
4682 021716 000000      0          ;FOURTH DATA WORD
4683 021720 000410      BR     20$          ;CONTINUE WITH TEST
4684
4685
4686 021722      19$: JSR    R0,@#FLHEAD      ;SAVE HEADER DATA IN WRFROM
4687 021722 004037 032716      WRFROM      ;LOCATION WHERE SAVED
4688 021726 002370      4          ;NUMBER OF WORDS SAVED
4689 021730 000004      10632      ;FIRST DATA WORD
4690 021732 010632      <18.*400>!<21.>      ;SECOND DATA WORD
4691 021734 011025      0          ;THIRD DATA WORD
4692 021736 000000      0          ;FOURTH DATA WORD
4693 021740 000000      0          ;CONTINUE
4694 021742      20$:
4695
4696
4697      ;*FILL WRITE FROM BUFFER WITH EXPECTED DATA
4698 021742 004037 032742      JSR    R0,@#CLAREA      ;CLEAR 256. WORDS, FROM WRFROM+<4*2>
4699 021746 002400      WRFROM+<4*2>      ;STARTING FROM WRFROM+<4*2>
4700 021750 000400      256.          ;256. WORDS
4701 021752 065125      <26.*2000>!<18.*40>!<21.>      ;FILL WITH <26.*2000>!<18.*40>!<
4702
4703      ;*FILL WRITE FROM BUFFER WITH 377 FROM WORDS 261 TO 266
4704 021754 004037 032742      JSR    R0,@#CLAREA      ;CLEAR 6 WORDS, FROM WRFROM+<260.*2>
4705 021760 003400      WRFROM+<260.*2>      ;STARTING FROM WRFROM+<260.*2>
4706 021762 000006      6          ;6 WORDS
4707 021764 000377      377          ;FILL WITH 377
4708
4709      ;*CLEAR READ INTO BUFFER
4710 021766 004037 032742      JSR    R0,@#CLAREA      ;CLEAR 266. WORDS, FROM REINTO
4711 021772 003434      REINTO      ;STARTING FROM REINTO
4712 021774 000412      266.          ;266. WORDS
4713 021776 000377      377          ;FILL WITH 377
4714
4715 022000 004737 033072      JSR    PC,@#CLDISK      ;SET R1-RHCS1, R2-RHCS2
4716
4717
```



```

4774 022150 004037 032774 JSR RO,@#FILLRE ;MOV -70. INTO SAVED RHWC
4775 022154 002172 RHWC ;SAVED REGISTER TO CHANGE
4776 022156 177672 -70. ;DATA
4777 022160 004037 033762 JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
4778 022164 002200 RHCS1 ;CHANGE RHCS1 REGISTER
4779 022166 000002 2 ;2 BIT/BITS TO BE CHANGED
4780 022170 000001 1 ;NEW VALUE OF SC IS 1
4781 022172 100000 SC ;CHANGE SC BIT
4782 022174 000001 1 ;NEW VALUE OF TRE IS 1
4783 022176 040000 TRE ;CHANGE TRE BIT
4784 022200 004037 033762 JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
4785 022204 002222 RHDS1 ;CHANGE RHDS1 REGISTER
4786 022206 000003 3 ;3 BIT/BITS TO BE CHANGED
4787 022210 000001 1 ;NEW VALUE OF ATA IS 1
4788 022212 100000 ATA ;CHANGE ATA BIT
4789 022214 000001 1 ;NEW VALUE OF ERR IS 1
4790 022216 040000 ERR ;CHANGE ERR BIT
4791 022220 000001 1 ;NEW VALUE OF LBT IS 1
4792 022222 002000 LBT ;CHANGE LBT BIT
4793 022224 004037 032774 JSR RO,@#FILLRE ;MOV AOE INTO SAVED RHER1
4794 022230 002202 RHER1 ;SAVED REGISTER TO CHANGE
4795 022232 001000 AOE ;DATA
4796
4797 ;*CHECK DRIVE TYPE BEFORE SETTING UP 'RHCA'
4798
4799 022234 005737 004636 TST @#RPO6 ;TEST FOR RPO6 DRIVE
4800 022240 001405 BEQ 13$ ;TREAT UNIT AS AN RPO4
4801 ;TREAT UNIT AS AN RPO6
4802 022242 004037 032774 JSR RO,@#FILLRE ;MOV 815. INTO SAVED RHCA
4803 022246 002212 RHCA ;SAVED REGISTER TO CHANGE
4804 022250 001457 815. ;DATA
4805 022252 000404 BR 14$ ;CONTINUE
4806 022254 13$:
4807 022254 004037 032774 JSR RO,@#FILLRE ;MOV 411. INTO SAVED RHCA
4808 022260 002212 RHCA ;SAVED REGISTER TO CHANGE
4809 022262 000633 411. ;DATA
4810 022264 14$: ;CONTINUE WITH TEST
4811
4812
4813 022264 053737 004644 004536 BIS @#ATTENT,@#SAVERE+24 ;SET APPROPRIATE 'ATA' BITS
4814 022272 017737 157706 00452' MOV @#RHDST,@#SAVERE+12 ;RHDST IS UNPREDICTABLE
4815
4816 ;*COMPARE REGISTERS BEFORE READ HEADER AND DATA WITH
4817 ;*REGISTERS AFTER COMMAND
4818 022300 004037 034070 JSR RO,@#COMREG ;COMPARE SAVED REGISTERS WITH
4819 022304 004512 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
4820 022306 002254 WC ;TEST DATA STARTING FROM 'RHWC'
4821 022310 000022 18. ;18. REGISTERS TO BE COMPARED
4822 022312 022316 3$ ;RETURN TO 3$ ON ERROR
4823 022314 022322 4$ ;RETURN TO 4$ ON NO ERROR
4824
4825 022316 104055 3$: ERROR 55 ;READING HEADER AND DATA WITH
4826 022320 000207 RTS PC ;EXPECTED ADDRESS OVERFLOW
4827 ;ERROR CAUSED IMPROPER
4828 ;REGISTER CHANGE
4829 ;GOOD DATA GIVES WHAT SHOULD

```



```
4830                                     ;BE THERE
4831                                     ;RECEIVED DATA GIVES WHAT
4832                                     ;WAS THERE AFTER COMMAND
4833
4834                                     ;*NOW COMPARE THE DATA READ
4835 022322                               4$: JSR      RO,@#COMPAR      ;COMPARE TWO BLOCKS OF MEMORY
4836 022322 004037 035120                WRFROM      ;GOOD DATA STARTS FROM WRFROM
4837 022326 002370                        REINTO      ;TEST DATA STARTS FROM REINTO
4838 022330 003434                        266.        ;266., WORDS TO BE COMPARED
4839 022332 000412                        5$          ;RETURN TO 5$ ON ERROR
4840 022334 022340                        6$          ;RETURN TO 6$ ON NO ERROR
4841 022336 022344
4842
4843 022340 104056                        5$: ERROR 56      ;DATA READ WITH AN EXPECTED
4844 022342 000207                        RTS      PC      ;ADDRESS OVERFLOW ERROR
4845                                     ;IS INCORRECT
4846                                     ;WORD NO 1 TO 260 SHOULD
4847                                     ;BE READ CORRECTLY
4848                                     ;WORD NO 261 TO 266 SHOULD
4849                                     ;NOT CHANGE DUE TO THE READ
4850 022344                               6$: JSR      PC,@#CLDISK    ;SET R1-RHCS1, R2-RHCS2
4851 022344 004737 033072
4852
```



```

4853
4854
4855
4856
4857
4858
4859 022350 004037 032716
4860 022354 002370
4861 022356 000004
4862 022360 010000
4863 022362 000000
4864 022364 000000
4865 022366 000000
4866 022370 004037 032742
4867 022374 002400
4868 022376 000400
4869 022400 000000
4870
4871
4872 022402 004037 032742
4873 022406 003434
4874 022410 000404
4875 022412 000377
4876
4877 022414 004737 033072
4878
4879
4880 022420 004037 035054
4881 022424 000000
4882 022426 000
4883 022427 000
4884 022430 177374
4885 022432 003434
4886 022434 000000
4887 022436 014000
4888 022440 002350
4889
4890
4891 022442 004037 033240
4892 022446 002172
4893 022450 004512
4894 022452 000021
4895 022454 004737 033152
4896 022460 104401 057176
4897 022464 000000
4898
4899 022466 013777 004506 157472
4900
4901 022474 013746 002350
4902 022500 052716 000101
4903 022504 012677 157470
4904
4905 022510 104413
4906 022512 002200
4907 022514 000200
4908 022516 004704

;*NOW PREPARE TO READ CYLINDER 0, SECTOR 0, TRACK 0
;*TO SEE THAT NOTHING GOT WRITTEN ON THERE
;*WITH THE ADDRESS OVER FLOW BIT SET (AOE)

;*FILL WRITE FROM BUFFER WITH EXPECTED HEADER
JSR RO,@#FLHEAD ;SAVE HEADER DATA IN WRFROM
WRFROM ;LOCATION WHERE SAVED
4 ;NUMBER OF WORDS SAVED
10000 ;FIRST DATA WORD
0 ;SECOND DATA WORD
0 ;THIRD DATA WORD
0 ;FOURTH DATA WORD
JSR RO,@#CLAREA ;CLEAR 256. WORDS, FROM WRFROM+<4*2>
WRFROM+<4*2> ;STARTING FROM WRFROM+<4*2>
256. ;256. WORDS
0 ;FILL WITH 0

;*FILL READ INTO BUFFER WITH 377
JSR RO,@#CLAREA ;CLEAR 260. WORDS, FROM REINTO
REINTO ;STARTING FROM REINTO
260. ;260. WORDS
377 ;FILL WITH 377

JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2

;*FILL COMMAND FOR READ HEADER AND DATA
JSR RO,@#RUN ;SETUP TO RUN FOR DATA COMMAND
0 ;CYLINDER 0
.BYTE 0 ;SECTOR 0
.BYTE 0 ;TRACK 0
-256.-4 ;WORD COUNT (DATA) = 256. +
REINTO ;BUS ADDRESS
0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
ECI!FMT22 ;16 BITS PER WORD FORMAT
REFOR ;GET READY TO DO A REFOR

;*SAVE REGISTERS FOR COMPARISON AFTER READ
JSR RO,@#SAVER ;SAVE REGISTERS
RHWC ;RHWC IS THE FIRST REGISTER SAVED
SAVERE ;STARTING ADDRESS OF WHERE
17. ;NUMBER OF REGISTERS
JSR PC,@#CHECKT ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF NOT
HALT ;STOP TEST

MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS

MOV @#REFOR,-(SP) ;GET READY TO MOVE COMMAND
BIS #GO!IE,(SP) ;GET READY TO SET GO AND
MOV (SP)+,@RHCS1 ;GO WITH

WAT ;WAIT FOR RDY BIT TO SET
RHCS1 ;WAIT FOR RHCS1 REGISTER
RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
2500. ;ALLOW 25000 MICRO SECONDS
    
```


4965
4966 022616
4967
4968
4969

3\$:

;'AOE' OVER FLOWED INTO HERE


```

4970
4971 022616 000004          TST25: SCOPE
4972 022620 012706 001000  MOV      #STACK,SP      ;RESET STACK
4973 022624 012737 000025 004504  MOV      #25,@#TSTNM    ;SAVE TEST NUMBER
4974 022632 004737 033072  JSR      PC,@#CLDISK    ;SET R1-RHCS1, R2-RHCS2
4975
4976                                     ;*FIRST WRITE HEADER AND DATA CYLINDER 0, TRACK 0, SECTOR 0
4977                                     ;*FILL WRITE FROM BUFFER WITH HEADER
4978 022636 004037 032716  JSR      RO,@#FLHEAD    ;SAVE HEADER DATA IN WRFROM
4979 022642 002370          WRFROM          ;LOCATION WHERE SAVED
4980 022644 000004          4                ;NUMBER OF WORDS SAVED
4981 022646 010000          10000           ;FIRST DATA WORD
4982 022650 000000          0                ;SECOND DATA WORD
4983 022652 000000          0                ;THIRD DATA WORD
4984 022654 000000          0                ;FOURTH DATA WORD
4985
4986                                     ;*FILL WRITE FROM BUFFER WITH DATA
4987 022656 004037 032742  JSR      RO,@#CLAREA    ;CLEAR 256. WORDS, FROM WRFROM+<4*2>
4988 022662 002400          WRFROM+<4*2>    ;STARTING FROM WRFROM+<4*2>
4989 022664 000400          256.            ;256. WORDS
4990 022666 000000          0                ;FILL WITH 0
4991
4992                                     ;*FILL COMMAND
4993 022670 004037 035054  JSR      RO,@#RIJN      ;SETUP TO RUN FOR DATA COMMAND
4994 022674 000000          0                ;CYLINDER 0
4995 022676          000          .BYTE 0          ;SECTOR 0
4996 022677          000          .BYTE 0          ;TRACK 0
4997 022700 177374          -256.-4         ;WORD COUNT (DATA) = 256. +
4998 022702 002370          WRFROM          ;BUS ADDRESS
4999 022704 000000          0                ;DO NOT INHIBIT BUS ADDRESS INCREMENT
5000 022706 010000          FMT22           ;16 BITS PER WORD FORMAT
5001 022710 002344          WRIFOR         ;GET READY TO DO A WRIFOR
5002
5003 022712 004737 033152  JSR      PC,@#CHECKT    ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
5004 022716 104401 057176  TYPE      ,CPHALT      ;CANNOT CONTINUE TESTING IF NOT
5005 022722 000000          HALT           ;STOP TEST
5006
5007 022724 013777 004506 157234  MOV      @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
5008
5009 022732 013746 002344  MOV      @#WRIFOR,-(SP) ;GET READY TO MOVE COMMAND
5010 022736 052716 000101  BIS      #GO!IE,(SP)    ;GET READY TO SET GO AND
5011 022742 012677 157232  MOV      (SP)+,@RHCS1   ;GO WITH
5012
5013 022746 104413          WAT             ;WAIT FOR RDY BIT TO SET
5014 022750 002200          RHCS1          ;WAIT FOR RHCS1 REGISTER
5015 022752 000200          RDY            ;WAIT FOR RDY BIT IN RHCS1 REGISTER
5016 022754 004704          2500.          ;ALLOW 25000 MICRO SECONDS
5017 022756 004704          2500.          ;RDY MUST SET BETWEEN
5018
5019 022760 004737 033072  JSR      PC,@#CLDISK    ;SET R1-RHCS1, R2-RHCS2
5020
    
```



```

5021
5022          ;*NOW PREPARE TO WRITE WITH WRONG FORMAT
5023
5024          ;*FILL WRITE FROM BUFFER
5025 022764 004037 032742 JSR   R0,@#CLAREA  ;CLEAR 256. WORDS, FROM WRFROM
5026 022770 002370 WRFROM  ;STARTING FROM WRFROM
5027 022772 000400 256.  ;256. WORDS
5028 022774 000377 377  ;FILL WITH 377
5029
5030          ;*FILL WRITE DATA COMMAND
5031 022776 004037 035054 JSR   R0,@#RUN  ;SETUP TO RUN FOR DATA COMMAND
5032 023002 000000 0  ;CYLINDER 0
5033 023004 000 .BYTE 0  ;SECTOR 0
5034 023005 000 .BYTE 0  ;TRACK 0
5035 023006 177400 -256. ;WORD COUNT = 256.
5036 023010 002370 WRFROM  ;BUS ADDRESS
5037 023012 000000 0  ;DO NOT INHIBIT BUS ADDRESS INCREMENT
5038 023014 000000 0  ;18 BITS PER WORD FORMAT
5039 023016 002342 WRIDAT  ;GET READY TO DO A WRIDAT
5040
5041          ;*SAVE REGISTERS FOR COMPARISON AFTER ATTEMPTED WRITE DATA
5042          ;*WITH WRONG FORMAT
5043 023020 004037 033240 JSR   R0,@#SAVER  ;SAVE REGISTERS
5044 023024 002172 RHWC  ;RHWC IS THE FIRST REGISTER SAVED
5045 023026 004512 SAVERE ;STARTING ADDRESS OF WHERE
5046 023030 000022 18.  ;NUMBER OF REGISTERS
5047
5048 023032 004737 033152 JSR   PC,@#CHECKT ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
5049 023036 104401 057176 TYPE  ,CPHALT  ;CANNOT CONTINUE TESTING IF NOT
5050 023042 000000 HALT  ;STOP TEST
5051
5052 023044 013777 004506 157114 MOV   @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
5053
5054 023052 013746 002342 MOV   @#WRIDAT,-(SP) ;GET READY TO MOVE COMMAND
5055 023056 052716 000101 BIS   #GO!IE,(SP)  ;GET READY TO SET GO AND
5056 023062 012677 157112 MOV   (SP)+,@RHCS1 ;GO WITH
5057
5058 023066 104413 WAT  ;WAIT FOR RDY BIT TO SET
5059 023070 002200 RHCS1 ;WAIT FOR RHCS1 REGISTER
5060 023072 000200 RDY  ;WAIT FOR RDY BIT IN RHCS1 REGISTER
5061 023074 001522 850. ;ALLOW 8500 MICRO SECONDS
5062 023076 001510 840. ;RDY MUST SET BETWEEN
5063
5064          ;*CHANGE SAVED REGISTERS TO EXPECTED VALUE
5065 023100 005737 004640 TST   @#RH70  ;RH70 CONTROLLER ?
5066 023104 001411 BEQ   7$  ;IF NOT, SKIP NEXT RH70 CODE
5067
5068 023106 004037 032774 JSR   R0,@#FILLRE ;MOV -248. INTO SAVED RHWC
5069 023112 002172 RHWC  ;SAVED REGISTER TO CHANGE
5070 023114 177410 -248. ;DATA
5071 023116 004037 032774 JSR   R0,@#FILLRE ;MOV WRFROM+<8.*2> INTO SAVED RHBA
5072 023122 002174 RHBA  ;SAVED REGISTER TO CHANGE
5073 023124 002410 WRFROM+<8.*2> ;DATA
5074 023126 000410 BR    8$  ;SKIP NEXT RH11 CODE
5075
5076
    
```



```

5077
5078 023130
5079 023130 004037 032774
5080 023134 002172
5081 023136 177502
5082 023140 004037 032774
5083 023144 002174
5084 023146 002574
5085
5086 023150
5087 023150 004037 033762
5088 023154 002200
5089 023156 000002
5090 023160 000001
5091 023162 100000
5092 023164 000001
5093 023166 040000
5094 023170 004037 033762
5095 023174 002176
5096 023176 000001
5097 023200 000001
5098 023202 000200
5099 023204 004037 033762
5100 023210 002222
5101 023212 000002
5102 023214 000001
5103 023216 100000
5104 023220 000001
5105 023222 040000
5106 023224 004037 032774
5107 023230 002204
5108 023232 000001
5109 023234 053737 004644 004536
5110 023242 004037 033762
5111 023246 002202
5112 023250 000001
5113 023252 000001
5114 023254 000020
5115 023256 017746 156714
5116 023262 042716 177477
5117 023266 042737 000300 004516
5118 023274 052637 004516
5119
5120
5121
5122
5123 023300 004037 034070
5124 023304 004512
5125 023306 002254
5126 023310 000022
5127 023312 023316
5128 023314 023322
5129
5130 023316 104057
5131 023320 000207
5132

7$:
JSR RO,@#FILLRE ;MOV -190. INTO SAVED RHWC
RHWC ;SAVED REGISTER TO CHANGE
-190. ;DATA
JSR RO,@#FILLRE ;MOV WRFROM+<66.*2> INTO SAVED RHBA
RHBA ;SAVED REGISTER TO CHANGE
WRFROM+<66.*2> ;DATA

8$:
JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
RHCS1 ;CHANGE RHCS1 REGISTER
2 ;2 BIT/BITS TO BE CHANGED
1 ;NEW VALUE OF SC IS 1
SC ;CHANGE SC BIT
1 ;NEW VALUE OF TRE IS 1
TRE ;CHANGE TRE BIT
JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
RHCS2 ;CHANGE RHCS2 REGISTER
1 ;1 BIT/BITS TO BE CHANGED
1 ;NEW VALUE OF OR IS 1
OR ;CHANGE OR BIT
JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
RHDS1 ;CHANGE RHDS1 REGISTER
2 ;2 BIT/BITS TO BE CHANGED
1 ;NEW VALUE OF ATA IS 1
ATA ;CHANGE ATA BIT
1 ;NEW VALUE OF ERR IS 1
ERR ;CHANGE ERR BIT
JSR RO,@#FILLRE ;MOV 1 INTO SAVED RHDST
RHDST ;SAVED REGISTER TO CHANGE
1 ;DATA
BIS @#ATTENT,@#SAVERE+24 ;SET APPROPRIATE 'ATA' BITS
JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
RHER1 ;CHANGE RHER1 REGISTER
1 ;1 BIT/BITS TO BE CHANGED
1 ;NEW VALUE OF FER IS 1
FER ;CHANGE FER BIT
MOV @RHCS2,-(SP) ;GET RHCS2
BIC #^C<IR!OR>,(SP) ;KEEP IR AND OR
BIC #IR!OR,@#SAVERE+4 ;CLEAR SAVED IR OR
BIS (SP)+,@#SAVERE+4 ;SET OR IR AS REQUIRED

;*COMPARE REGISTERS BEFORE WRITE DATA WITH AFTER ATTEMPT
JSR RO,@#COMREG ;COMPARE SAVED REGISTERS WITH
SAVERE ;GOOD DATA SAVED IN 'SAVERE'
WC ;TEST DATA STARTING FROM 'RHWC'
18. ;18. REGISTERS TO BE COMPARED
1$ ;RETURN TO 1$ ON ERROR
2$ ;RETURN TO 2$ ON NO ERROR

1$:
ERROR 57 ;ATTEMPTING TO WRITE DATA
RTS PC ;WITH WRONG FORMAT BIT CAUSED
;IMPROPER REGISTER CHANGE
    
```



```

5133                                     ;GOOD DATA GIVES WHAT SHOULD
5134                                     ;BE THERE
5135                                     ;RECEIVED DATA GIVES WHAT WAS
5136                                     ;THERE AFTER ATTEMPTED WRITE
5137
5138                                     ;*NOW PREPARE TO READ WITH CORRECT FORMAT TO CHECK
5139                                     ;*THAT NOTHING GOT WRITTEN
5140 023322                                2$:
5141 023322 004737 033072                JSR    PC,@#CLDISK    ;SET R1-RHCS1, R2-RHCS2
5142                                     ;*FILL WRITE FROM BUFFER WITH EXPECTED DATA
5143 023326 004037 032742                JSR    RO,@#CLAREA   ;CLEAR 256. WORDS, FROM WRFROM
5144 023332 002370                        WRFROM                ;STARTING FROM WRFROM
5145 023334 000400                        256.                  ;256. WORDS
5146 023336 000000                        0                      ;FILL WITH 0
5147
5148                                     ;*FILL READ INTO BUFFER WITH 125252
5149 023340 004037 032742                JSR    RO,@#CLAREA   ;CLEAR 256. WORDS, FROM REINTO
5150 023344 003434                        REINTO                ;STARTING FROM REINTO
5151 023346 000400                        256.                  ;256. WORDS
5152 023350 125252                        125252                ;FILL WITH 125252
5153
5154                                     ;*FILL COMMAND TO READ DATA
5155 023352 004037 035054                JSR    RO,@#RUN      ;SETUP TO RUN FOR DATA COMMAND
5156 023356 000000                        0                      ;CYLINDER 0
5157 023360 000                                .BYTE 0                ;SECTOR 0
5158 023361 000                                .BYTE 0                ;TRACK 0
5159 023362 177400                        -256.                  ;WORD COUNT = 256.
5160 023364 003434                        REINTO                ;BUS ADDRESS
5161 023366 000000                        0                      ;DO NOT INHIBIT BUS ADDRESS INCREMENT
5162 023370 014000                        FMT22!ECI             ;16 BITS PER WORD FORMAT
5163 023372 002346                        READAT                ;GET READY TO DO A READAT
5164
5165                                     ;*SAVE REGISTERS FOR COMPARISON AFTER NORMAL READ
5166 023374 004037 033240                JSR    RO,@#SAVER    ;SAVE REGISTERS
5167 023400 002172                        RHWC                   ;RHWC IS THE FIRST REGISTER SAVED
5168 023402 004512                        SAVERE                ;STARTING ADDRESS OF WHERE
5169 023404 000022                        18.                   ;NUMBER OF REGISTERS
5170
5171 023406 004737 033152                JSR    PC,@#CHECKT   ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
5172 023412 104401 057176                TYPE ,CPHALT          ;CANNOT CONTINUE TESTING IF NOT
5173 023416 000000                        HALT                  ;STOP TEST
5174
5175 023420 013777 004506 156540          MOV    @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
5176
5177 023426 013746 002346                MOV    @#READAT,-(SP) ;GET READY TO MOVE COMMAND
5178 023432 052716 000101                BIS    #GO!IE,(SP)   ;GET READY TO SET GO AND
5179 023436 012677 156536                MOV    (SP)+,@RHCS1  ;GO WITH
5180
5181 023442 104413                        WAT                   ;WAIT FOR RDY BIT TO SET
5182 023444 002200                        RHCS1                 ;WAIT FOR RHCS1 REGISTER
5183 023446 000200                        RDY                   ;WAIT FOR RDY BIT IN RHCS1 REGISTER
5184 023450 001614                        908.                  ;ALLOW 9080 MICRO SECONDS
5185 023452 001507                        839.                  ;RDY MUST SET BETWEEN
5186
5187                                     ;*CHANGE SAVED REGISTERS TO EXPECTED VALUE
5188 023454 004037 032774                JSR    RO,@#FILLRE   ;MOV REINTO+<256.*2> INTO SAVED RHBA
    
```



```

5189 023460 002174          RHBA          ;SAVED REGISTER TO CHANGE
5190 023462 004434          REINTO+<256.*2> ;DATA
5191 023464 004037 032774 JSR    RO,@#FILLRE ;MOV 0 INTO SAVED RHWC
5192 023470 002172          RHWC          ;SAVED REGISTER TO CHANGE
5193 023472 000000          0            ;DATA
5194 023474 004037 032774 JSR    RO,@#FILLRE ;MOV 1 INTO SAVED RHDST
5195 023500 002204          RHDST        ;SAVED REGISTER TO CHANGE
5196 023502 000001          1            ;DATA
5197 023504 017746 156466 MOV    @RHCS2,-(SP) ;GET RHCS2
5198 023510 042716 177477 BIC    #^C<IR!OR>,(SP) ;KEEP IR AND OR
5199 023514 042737 000300 004516 BIC    #IR!OR,@#SAVERE+4 ;CLEAR SAVED IR OR
5200 023522 052637 004516 BIS    (SP)+,@#SAVERE+4 ;SET OR IR AS REQUIRED
5201
5202
5203
5204
5205          ;*COMPARE REGISTERS BEFORE READ WITH AFTER
5206 023526 004037 034070 JSR    RO,@#COMREG ;COMPARE SAVED REGISTERS WITH
5207 023532 004512          SAVERE        ;GOOD DATA SAVED IN 'SAVERE'
5208 023534 002254          WC          ;TEST DATA STARTING FROM 'RHWC'
5209 023536 000022          18.         ;18. REGISTERS TO BE COMPARED
5210 023540 023544          3$          ;RETURN TO 3$ ON ERROR
5211 023542 023550          4$          ;RETURN TO 4$ ON NO ERROR
5212
5213 023544 104033          3$:          ERROR    33          ;READ DATA AFTER AN
5214 023546 000207          RTS      PC          ;ATTEMPTED WRITE WITH WRONG
5215          ;IMPROPER REGISTER CHANGE
5216          ;FORMAT CAUSED
5217          ;GOOD DATA GIVES WHAT SHOULD
5218          ;BE THERE
5219          ;RECEIVED DATA GIVES WHAT
5220          ;WAS THERE AFTER READ
5221
5222          ;*COMPARE DATA READ AFTER ATTEMPTED WRITE WITH
5223          ;*WRONG FORMAT BIT
5224 023550          4$:
5225 023550 004037 035120 JSR    RO,@#COMPAR ;COMPARE TWO BLOCKS OF MEMORY
5226 023554 002370          WRFROM       ;GOOD DATA STARTS FROM WRFROM
5227 023556 003434          REINTO       ;TEST DATA STARTS FROM REINTO
5228 023560 000400          256.        ;256., WORDS TO BE COMPARED
5229 023562 023566          5$          ;RETURN TO 5$ ON ERROR
5230 023564 023572          6$          ;RETURN TO 6$ ON NO ERROR
5231
5232 023566 104034          5$:          ERROR    34          ;DATA READ AFTER AN ATTEMPT
5233 023570 000207          RTS      PC          ;TO WRITE WITH WRONG FORMAT
5234          ;WAS INCORRECT
5235
5236 023572          6$:
5237

```



```

5238 023572 000004          TST26: SCOPE
5239 023574 012706 001000  MOV    #STACK,SP      ;RESET STACK
5240 023600 012737 000026 004504  MOV    #26,@#TSTNM    ;SAVE TEST NUMBER
5241 023606 004737 033072  JSR    PC,@#CLDISK    ;SET R1-RHCS1, R2-RHCS2
5242
5243          ;*FILL WRITE FROM BUFFER WITH 107070
5244 023612 004037 032742  JSR    RO,@#CLAREA    ;CLEAR 256. WORDS, FROM WRFROM
5245 023616 002370          WRFROM                ;STARTING FROM WRFROM
5246 023620 000400          256.                  ;256. WORDS
5247 023622 107070          107070                ;FILL WITH 107070
5248
5249          ;*FILL READ INTO BUFFER WITH 107070
5250 023624 004037 032742  JSR    RO,@#CLAREA    ;CLEAR 256. WORDS, FROM REINTO
5251 023630 003434          REINTO                ;STARTING FROM REINTO
5252 023632 000400          256.                  ;256. WORDS
5253 023634 107070          107070                ;FILL WITH 107070
5254
5255          ;*FILL COMMAND TO READ WITH WRONG FORMAT
5256 023636 004037 035054  JSR    RU,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
5257 023642 000000          0                      ;CYLINDER 0
5258 023644          000                    ;SECTOR 0
5259 023645          000                    ;TRACK 0
5260 023646 177400          -256.                  ;WORD COUNT = 256.
5261 023650 003434          REINTO                ;BUS ADDRESS
5262 023652 000000          0                      ;DO NOT INHIBIT BUS ADDRESS INCREMENT
5263 023654 004000          ECI                    ;18 BITS PER WORD FORMAT
5264 023656 002346          READAT                ;GET READY TO DO A READAT
5265
5266          ;*SAVE REGISTERS FOR COMPARAISON AFTER READ
5267 023660 004037 033240  JSR    RO,@#SAVER     ;SAVE REGISTERS
5268 023664 002172          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
5269 023666 004512          SAVERE                 ;STARTING ADDRES OF WHERE
5270 023670 000022          18.                    ;NUMBER OF REGISTERS
5271
5272 023672 004737 033152  JSR    PC,@#CHECKT    ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
5273 023676 104401 057176  TYPE    ,CPHALT        ;CANNOT CONTINUE TESTING IF NOT
5274 023702 000000          HALT                  ;STOP TEST
5275
5276 023704 013777 004506 156254  MOV    @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
5277
5278 023712 013746 002346  MOV    @#READAT,-(SP) ;GET READY TO MOVE COMMAND
5279 023716 052716 000101  BIS    #GO!IE,(SP)    ;GET READY TO SET GO AND
5280 023722 012677 156252  MOV    (SP)+,@RHCS1   ;GO WITH
5281
5282 023726 104413          WAT                    ;WAIT FOR RDY BIT TO SET
5283 023730 002200          RHCS1                 ;WAIT FOR RHCS1 REGISTER
5284 023732 000200          RDY                    ;WAIT FOR RDY BIT IN RHCS1 REGISTER
5285 023734 001522          850.                  ;ALLOW 8500 MICRO SECONDS
5286 023736 001510          840.                  ;RDY MUST SET BETWEEN
5287
5288          ;*CHANGE SAVED REGISTERS TO EXPECTED VALUE
5289
5290 023740 004037 033762  JSR    RO,@#CHREG     ;CHANGE BITS IN SAVED REGISTER
5291 023744 002200          RHCS1                 ;CHANGE RHCS1 REGISTER
5292 023746 000002          2                      ;2 BIT/BITS TO BE CHANGED
5293 023750 000001          1                      ;NEW VALUE OF SC IS 1
    
```



```

5294 023752 100000 SC ;CHANGE SC BIT
5295 023754 000001 1 ;NEW VALUE OF TRE IS 1
5296 023756 040000 TRE ;CHANGE TRE BIT
5297 023760 004037 033762 JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
5298 023764 002222 RHDS1 ;CHANGE RHDS1 REGISTER
5299 023766 000002 2 ;2 BIT/BITS TO BE CHANGED
5300 023770 000001 1 ;NEW VALUE OF ATA IS 1
5301 023772 100000 ATA ;CHANGE ATA BIT
5302 023774 000001 1 ;NEW VALUE OF ERR IS 1
5303 023776 040000 ERR ;CHANGE ERR BIT
5304 024000 004037 032774 JSR RO,@#FILLRE ;MOV 1 INTO SAVED RHDST
5305 024004 002204 RHDST ;SAVED REGISTER TO CHANGE
5306 024006 000001 1 ;DATA
5307 024010 004037 033762 JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
5308 024014 002202 RHER1 ;CHANGE RHER1 REGISTER
5309 024016 000001 1 ;1 BIT/BITS TO BE CHANGED
5310 024020 000001 1 ;NEW VALUE OF FER IS 1
5311 024022 000020 FER ;CHANGE FER BIT
5312 024024 053737 004644 004536 BIS @#ATTENT,@#SAVERE+24 ;SET APPROPRIATE 'ATA' BITS
5313 024032 017746 156140 MOV @RHCS2,-(SP) ;GET RHCS2
5314 024036 042716 177477 BIC #^C<IR!OR>,(SP) ;KEEP IR AND OR
5315 024042 042737 000300 004516 BIC #IR!OR,@#SAVERE+4;CLEAR SAVED IR OR
5316 024050 052637 004516 BIS (SP)+,@#SAVERE+4;SET OR IR AS REQUIRED
5317
5318
5319
5320 ;*COMPARE REGISTERS BEFORE WRITE DATA WITH AFTER ATTEMPT
5321 024054 004037 034070 JSR RO,@#COMREG ;COMPARE SAVED REGISTERS WITH
5322 024060 004512 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
5323 024062 002254 WC ;TEST DATA STARTING FROM 'RHWC'
5324 024064 000022 18. ;18. REGISTERS TO BE COMPARED
5325 024066 024072 1$ ;RETURN TO 1$ ON ERROR
5326 024070 024076 2$ ;RETURN TO 2$ ON NO ERROR
5327
5328 024072 104057 1$: ERROR 57 ;ATTEMPTING TO READ DATA
5329 024074 000207 RTS PC ;WITH WRONG FORMAT BIT CAUSED
5330 ;IMPROPER REGISTER CHANGE
5331 ;GOOD DATA GIVES WHAT SHOULD BE
5332 ;THERE
5333 ;RECEIVED DATA GIVES WHAT WAS THERE
5334 ;AFTER READ DATA
5335
5336 ;*COMPARE READ INTO BUFFER TO CHECK THAT NOTHING WAS READ
5337 024076 2$:
5338 024076 004037 035120 JSR RO,@#COMPAR ;COMPARE TWO BLOCKS OF MEMORY
5339 024102 002370 WRFROM ;GOOD DATA STARTS FROM WRFROM
5340 024104 003434 REINTO ;TEST DATA STARTS FROM REINTO
5341 024106 000400 256. ;256. WORDS TO BE COMPARED
5342 024110 024114 3$ ;RETURN TO 3$ ON ERROR
5343 024112 024120 4$ ;RETURN TO 4$ ON NO ERROR
5344
5345 024114 104034 3$: ERROR 34 ;ATTEMPT TO READ
5346 024116 000207 RTS PC ;WITH WRONG FORMAT BIT
5347 ;CHANGED READ INTO BUFFER
5348 ;GOOD DATA GIVES WHAT SHOULD
5349 ;BE THERE
    
```


5350
5351
5352
5353 024120
5354

4\$:

:BAD DATA GIVES WHAT WAS
:THERE AFTER READ DATA


```

5355
5356 024120 000004          TST27: SCOPE
5357 024122 012706 001000  MOV    #STACK,SP      ;RESET STACK
5358 024126 012737 000027 004504  MOV    #27,@#TSTNM    ;SAVE TEST NUMBER
5359 024134 004737 033072          JSR    PC,@#CLDISK    ;SET R1-RHCS1, R2-RHCS2
5360 024140 012737 002200 004650  MOV    #RHCS1,@#TMP0  ;FIRST REGISTER TO BE TESTED
5361 024146 012737 000007 004656  MOV    #7,@#TMP5      ;NUMBER OF REGISTERS TO BE TESTED
5362
5363          ;*PREPARE TO WRITE HEADER AND DATA CYLINDER 1, TRACK 0, SECTOR 0
5364          ;*FILL WRITE FROM BUFFER WITH HEADER
5365
5366 024154          ST22:
5367 024154 004737 033072  JSR    PC,@#CLDISK    ;SET R1-RHCS1, R2-RHCS2
5368 024160 004037 032716  JSR    RO,@#FLHEAD    ;SAVE HEADER DATA IN WRFROM
5369 024164 002370          WRFROM ;LOCATION WHERE SAVED
5370 024166 000004          4 ;NUMBER OF WORDS SAVED
5371 024170 010001          10001 ;FIRST DATA WORD
5372 024172 000000          0 ;SECOND DATA WORD
5373 024174 000000          0 ;THIRD DATA WORD
5374 024176 000000          0 ;FOURTH DATA WORD
5375
5376          ;*FILL WRITE FROM BUFFER WITH DATA
5377 024200 004037 032742  JSR    RO,@#CLAREA    ;CLEAR 256. WORDS, FROM WRFROM+<4*2>
5378 024204 002400          WRFROM+<4*2> ;STARTING FROM WRFROM+<4*2>
5379 024206 000400          256. ;256. WORDS
5380 024210 002000          2000 ;FILL WITH 2000
5381
5382 024212 004037 035054  ;*FILL COMMAND
5383 024216 000001          JSR    RO,@#RUN      ;SETUP TO RUN FOR DATA COMMAND
5384 024220 000          1 ;CYLINDER 1
5385 024221 000          .BYTE 0 ;SECTOR 0
5386 024222 177464          .BYTE 0 ;TRACK 0
5387 024224 002370          -200.-4 ;WORD COUNT (DATA) = 200. +
5388 024226 000000          WRFROM ;BUS ADDRESS
5389 024230 010000          0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
5390 024232 002344          FMT22 ;16 BITS PER WORD FORMAT
5391          WRIFOR ;GET READY TO DO A WRIFOR
5392 024234 004737 033152  JSR    PC,@#CHECKT    ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
5393 024240 104401 057176  TYPE   ,CPHALT      ;CANNOT CONTINUE TESTING IF NOT
5394 024244 000000          HALT ;STOP TEST
5395
5396 024246 013777 004506 155712  MOV    @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
5397
5398 024254 013746 002344  MOV    @#WRIFOR,-(SP) ;GET READY TO MOVE COMMAND
5399 024260 052716 000101  BIS    #GO!IE,(SP)   ;GET READY TO SET GO AND
5400 024264 012677 155710  MOV    (SP)+,@RHCS1  ;GO WITH
5401
5402          ;*TIME IS NOT IMPORTANT
5403 024270 104413          WAT ;WAIT FOR RDY BIT TO SET
5404 024272 002200          RHCS1 ;WAIT FOR RHCS1 REGISTER
5405 024274 000200          RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
5406 024276 004704          2500. ;ALLOW 25000 MICRO SECONDS
5407 024300 004704          2500. ;RDY MUST SET BETWEEN
5408
5409          ;*NOW BRING THE HEADS TO CYLINDER 0
5410

```



```

5411 024302 004737 033152 JSR PC,@#CHECKT ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
5412 024306 104401 057176 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF NOT
5413 024312 000000 HALT ;STOP TEST
5414
5415 024314 004037 033042 JSR RO,@#SEEKCY ;SEEK FOR
5416 024320 000000 0 ;CYLINDER 0
5417
5418 024322 013777 004506 155636 MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
5419
5420 024330 013746 002352 MOV @#SEECOM,-(SP) ;GET READY TO MOVE COMMAND
5421 024334 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET GO AND
5422 024340 012677 155634 MOV (SP)+,@RHCS1 ;GO WITH
5423
5424 024344 104413 WAT ;WAIT FOR DRY BIT TO SET
5425 024346 002222 RHDS1 ;WAIT FOR RHDS1 REGISTER
5426 024350 000200 DRY ;WAIT FOR DRY BIT IN RHDS1 REGISTER
5427 024352 002776 1534. ;ALLOW 15340 MICRO SECONDS
5428 024354 001502 834. ;DRY MUST SET BETWEEN
5429
5430 ;*PREPARE FOR A READ DATA
5431
5432 ;*FILL WRITE FROM BUFFER WITH EXPECTED DATA FROM READ
5433 024356 004037 032742 JSR RO,@#CLAREA ;CLEAR 150. WORDS, FROM WRFROM
5434 024362 002370 WRFROM ;STARTING FROM WRFROM
5435 024364 000226 150. ;150. WORDS
5436 024366 002000 2000 ;FILL WITH 2000
5437 024370 004037 032742 JSR RO,@#CLAREA ;CLEAR 106. WORDS, FROM WRFROM+<150.*2>
5438 024374 003044 WRFROM+<150.*2> ;STARTING FROM WRFROM+<150.*2>
5439 024376 000152 106. ;106. WORDS
5440 024400 000077 77 ;FILL WITH 77
5441
5442 ;*FILL READ INTO BUFFER WITH DATA OTHER THAN WHAT IS EXPECTED
5443 024402 004037 032742 JSR RO,@#CLAREA ;CLEAR 256. WORDS, FROM REINTO
5444 024406 003434 REINTO ;STARTING FROM REINTO
5445 024410 000400 256. ;256. WORDS
5446 024412 000077 77 ;FILL WITH 77
5447
5448 024414 004037 035054 ;*FILL READ DATA COMMAND
5449 024420 000001 JSR RO,@#RUN ;SETUP TO RUN FOR DATA COMMAND
5450 024422 000 .BYTE 0 ;CYLINDER 1
5451 024423 000 .BYTE 0 ;SECTOR 0
5452 024424 177552 -150. ;TRACK 0
5453 024426 003434 REINTO ;WORD COUNT = 150.
5454 024430 000000 0 ;BUS ADDRESS
5455 024432 014000 ECI!FMT22 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
5456 024434 002346 READAT ;16 BITS PER WORD FORMAT
5457 ;GET READY TO DO A READAT
5458
5459 ;*SAVE REGISTERS FOR COMPARISON AFTER ATTEMPTED WRITE
5460 024436 004037 033240 ;*INTO A REGISTER WHILE THE READ IS GOING ON
5461 024442 002172 JSR RO,@#SAVER ;SAVE REGISTERS
5462 024444 004512 RHWC ;RHWC IS THE FIRST REGISTER SAVED
5463 024446 000022 SAVERE ;STARTING ADDRESS OF WHERE
5464 18. ;NUMBER OF REGISTERS
5465 024450 004737 033152 JSR PC,@#CHECKT ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
5466 024454 104401 057176 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF NOT

```


REGISTER MODIFICATION REFUSED - RHER1(BIT #2),RMR

```

5467 024460 000000 HALT ;STOP TEST
5468
5469 024462 013777 004506 155476 MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
5470
5471 024470 013746 002346 MOV @#READAT,-(SP) ;GET READY TO MOVE COMMAND
5472 024474 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET GO AND
5473 024500 012677 155474 MOV (SP)+,@RHCS1 ;GO WITH
5474
5475 024504 013700 004650 MOV @#TMP0,R0 ;SET UP R0 FOR WRITE
5476 024510 012730 002006 MOV #BIT1!BIT2!BIT10,@(R0)+ ;ATTEMPT TO WRITE INTO
5477 ;REGISTERS DURING IMPLIED SEEK
5478 024514 010037 004650 MOV R0,@#TMP0 ;SAVE OFF R0
5479
5480 ;*NOW RMR IS SET BUT THE COMPLETION OF READ MUST BE
5481 ;*WAITED ON
5482
5483 024520 104413 WAT ;WAIT FOR RDY BIT TO SET
5484 024522 002200 RHCS1 ;WAIT FOR RHCS1 REGISTER
5485 024524 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
5486 024526 002237 1183. ;ALLOW 11830 MICRO SECONDS
5487 024530 002237 1183. ;RDY MUST SET BETWEEN
5488
5489 ;*CHANGE SAVED REGISTERS TO EXPECTED VALUE
5490 024532 004037 033762 JSR R0,@#CHREG ;CHANGE BITS IN SAVED REGISTER
5491 024536 002234 RHCC ;CHANGE RHCC REGISTER
5492 024540 000001 1 ;1 BIT/BITS TO BE CHANGED
5493 024542 000001 1 ;NEW VALUE OF BIT0 IS 1
5494 024544 000001 BIT0 ;CHANGE BIT0 BIT
5495 024546 004037 033762 JSR R0,@#CHREG ;CHANGE BITS IN SAVED REGISTER
5496 024552 002200 RHCS1 ;CHANGE RHCS1 REGISTER
5497 024554 000001 1 ;1 BIT/BITS TO BE CHANGED
5498 024556 000001 1 ;NEW VALUE OF SC IS 1
5499 024560 100000 SC ;CHANGE SC BIT
5500 024562 004037 032774 JSR R0,@#FILLRE ;MOV 1 INTO SAVED RHDST
5501 024566 002204 RHDST ;SAVED REGISTER TO CHANGE
5502 024570 000001 1 ;DATA
5503 024572 004037 033762 JSR R0,@#CHREG ;CHANGE BITS IN SAVED REGISTER
5504 024576 002222 RHDS1 ;CHANGE RHDS1 REGISTER
5505 024600 000002 2 ;2 BIT/BITS TO BE CHANGED
5506 024602 000001 1 ;NEW VALUE OF ATA IS 1
5507 024604 100000 ATA ;CHANGE ATA BIT
5508 024606 000001 1 ;NEW VALUE OF ERR IS 1
5509 024610 040000 ERR ;CHANGE ERR BIT
5510 024612 053737 004644 004536 BIS @#ATTENT,@#SAVERE+24 ;SET APPROPRIATE 'ATA' BITS
5511 024620 004037 033762 JSR R0,@#CHREG ;CHANGE BITS IN SAVED REGISTER
5512 024624 002202 RHER1 ;CHANGE RHER1 REGISTER
5513 024626 000001 1 ;1 BIT/BITS TO BE CHANGED
5514 024630 000001 1 ;NEW VALUE OF RMR IS 1
5515 024632 000004 RMR ;CHANGE RMR BIT
5516 024634 004037 032774 JSR R0,@#FILLRE ;MOV REINTO+<150.*2> INTO SAVED RHBA
5517 024640 002174 RHBA ;SAVED REGISTER TO CHANGE
5518 024642 004110 REINTO+<150.*2> ;DATA
5519 024644 004037 032774 JSR R0,@#FILLRE ;MOV 0 INTO SAVED RHWC
5520 024650 002172 RHWC ;SAVED REGISTER TO CHANGE
5521 024652 000000 0 ;DATA
5522

```


REGISTER MODIFICATION REFUSED - RHER1(BIT #2),RMR

```

5523 ;*COMPARE REGISTERS BEFORE READ DATA WITH REGISTERS
5524 ;*AFTER READ AND ATTEMPTED MODIFICATION OF REGISTER
5525 024654 004037 034070 JSR RO,@#COMREG ;COMPARE SAVED REGISTERS WITH
5526 024660 004512 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
5527 024662 002254 WC ;TEST DATA STARTING FROM 'RHWC'
5528 024664 000022 18. ;18. REGISTERS TO BE COMPARED
5529 024666 024672 2$ ;RETURN TO 2$ ON ERROR
5530 024670 024712 3$ ;RETURN TO 3$ ON NO ERROR
5531
5532 024672 2$: MOV RO,-(SP) ;:PUSH RO ON STACK
5533 024672 010046 MOV @#TMP0,RO ;:GET REGISTER BEEING MODIFYED + 2 POINTER
5534 024674 013700 004650 MOV -(RO),@#$BDADR ;:GET ADDRESS OF REGISTER BEING MODIFIED
5535 024700 014037 001122 ERROR 60 ;:ATTEMPTING TO MODIFY REGISTER
5536 024704 104060 MOV (SP)+,RO ;:POP STACK INTO RO
5537 024706 012600 RTS PC ;:DURING A READ COMMAND CAUSED
5538 024710 000207 ;:IMPROPER REGISTER CHANGE
5539 ;:GOOD DATA GIVES WHAT SHOULD
5540 ;:BE THERE
5541 ;:RECEIVED DATA GIVES WHAT WAS
5542 ;:THERE AFTER READ
5543
5544 ;*COMPARE DATA READ
5545 024712 3$: JSR RO,@#COMPAR ;:COMPARE TWO BLOCKS OF MEMORY
5546 024712 004037 035120 WRFROM ;:GOOD DATA STARTS FROM WRFROM
5547 024716 002370 REINTO ;:TEST DATA STARTS FROM REINTO
5548 024720 003434 4$ ;:4$, WORDS TO BE COMPARED
5549 024722 024726 ST23 ;:RETURN TO ST23 ON ERROR
5550 024724 024732
5551
5552 024726 104034 4$: ERROR 34 ;:DATA READ WITH AN ATTEMPTED
5553 024730 000207 RTS PC ;:MODIFICATION OF REGISTER
5554 ;:DURING READ CAUSED ERROR
5555 024732 005337 004656 ST23: DEC @#TMP5 ;:COUNT DOWN
5556 024736 001002 BNE 1$ ;:BRANCH IF 7 NOT DONE
5557 024740 000137 024750 JMP TST30 ;:JUMP TO NEXT TEST
5558 024744 000137 024154 1$: JMP @#ST22 ;:JUMP TO BEGINING OF TEST
5559

```



```

5560 024750 000004          TST30: SCOPE
5561 024752 012706 001000  MOV      #STACK,SP          ;RESET STACK
5562 024756 012737 000030 004504  MOV      #30,@#TSTNM        ;SAVE TEST NUMBER
5563 024764 004737 033072          JSR      PC,@#CLDISK        ;SET R1-RHCS1, R2-RHCS2
5564
5565 024770 012737 002200 004650  MOV      #RHCS1,@#TMP0      ;FILL REGISTER TO BE MODIFIED
5566 024776 012737 000007 004656  MOV      #7,@#TMP5          ;NUMBER OF REGISTERS TO BE TESTED
5567
5568          ;*PREPARE TO WRITE HEADER AND DATA
5569
5570 025004          ST24:
5571 025004 004737 033072          JSR      PC,@#CLDISK        ;SET R1-RHCS1, R2-RHCS2
5572
5573          ;*FILL WRITE FROM BUFFER WITH HEADER
5574          JSR      RO,@#FLHEAD    ;SAVE HEADER DATA IN WRFROM
5575 025010 004037 032716  WRFROM    ;LOCATION WHERE SAVED
5576 025014 002370          4          ;NUMBER OF WORDS SAVED
5577 025016 000004          10001      ;FIRST DATA WORD
5578 025020 010001          0          ;SECOND DATA WORD
5579 025022 000000          0          ;THIRD DATA WORD
5580 025024 000000          0          ;FOURTH DATA WORD
5581 025026 000000
5582
5583          ;*FILL WRITE FROM BUFFER WITH DATA
5584 025030 004037 032742  JSR      RO,@#CLAREA    ;CLEAR 256. WORDS, FROM WRFROM+<4*2>
5585 025034 002400          WRFROM+<4*2>          ;STARTING FROM WRFROM+<4*2>
5586 025036 000400          256.          ;256. WORDS
5587 025040 070707          070707        ;FILL WITH 070707
5588
5589          ;*FILL WRITE FROM BUFFER WITH NEXT SECTOR HEADER
5590 025042 004037 032716  JSR      RO,@#FLHEAD    ;SAVE HEADER DATA IN WRFROM+<260.*2>
5591 025046 003400          WRFROM+<260.*2>      ;LOCATION WHERE SAVED
5592 025050 000004          4          ;NUMBER OF WORDS SAVED
5593 025052 010001          10001      ;FIRST DATA WORD
5594 025054 000001          1          ;SECOND DATA WORD
5595 025056 000000          0          ;THIRD DATA WORD
5596 025060 000000          0          ;FOURTH DATA WORD
5597
5598          ;*FILL WRITE FROM BUFFER WITH WITH NEXT SECTOR DATA
5599 025062 004037 032742  JSR      RO,@#CLAREA    ;CLEAR 4 WORDS, FROM WRFROM+<268.*2>
5600 025066 003420          WRFROM+<268.*2>      ;STARTING FROM WRFROM+<268.*2>
5601 025070 000004          4          ;4 WORDS
5602 025072 070707          70707        ;FILL WITH 70707
5603
5604          ;*NOW THE WRITE HEADER AND DATA COMMAND WILL BE FILLED
5605 025074 004037 035054  JSR      RO,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
5606 025100 000001          1          ;CYLINDER 1
5607 025102 000          .BYTE 0          ;SECTOR 0
5608 025103 000          .BYTE 0          ;TRACK 0
5609 025104 177364          -264.-4        ;WORD COUNT (DATA) = 264. +
5610 025106 002370          WRFROM          ;BUS ADDRESS
5611 025110 000000          0          ;DO NOT INHIBIT BUS ADDRESS INCREMENT
5612 025112 010000          FMT22          ;16 BITS PER WORD FORMAT
5613 025114 002344          WRIFOR          ;GET READY TO DO A WRIFOR
5614
5615 025116 004737 033152          JSR      PC,@#CHECKT      ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
    
```



```

5616 025122 104401 057176 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF NOT
5617 025126 000000 HALT ;STOP TEST
5618
5619 025130 013777 004506 155030 MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
5620
5621 025136 013746 002344 MOV @#WRIFOR,-(SP) ;GET READY TO MOVE COMMAND
5622 025142 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET GO AND
5623 025146 012677 155026 MOV (SP)+,@RHCS1 ;GO WITH
5624
5625 ;*ONE REVOLUTION=16670 MICRO SEC, ONE SECTOR=760 MICRO SEC
5626 025152 104413 WAT ;WAIT FOR RDY BIT TO SET
5627 025154 002200 RHCS1 ;WAIT FOR RHCS1 REGISTER
5628 025156 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
5629 025160 001725 981. ;ALLOW 9810 MICRO SECONDS
5630 025162 001502 834. ;RDY MUST SET BETWEEN
5631
5632 ;*NOW PREPARE FOR THE WRITE DATA COMMAND
5633
5634 ;*FILL WRITE FROM BUFFER WITH 256 OF 2000 AND 4 OF 2001
5635 025164 004037 032742 JSR RO,@#CLAREA ;CLEAR 256. WORDS, FROM WRFROM
5636 025170 002370 WRFROM ;STARTING FROM WRFROM
5637 025172 000400 256. ;256. WORDS
5638 025174 002000 2000 ;FILL WITH 2000
5639 025176 004037 032742 JSR RO,@#CLAREA ;CLEAR 4 WORDS, FROM WRFROM+<256.*2>
5640 025202 003370 WRFROM+<256.*2> ;STARTING FROM WRFROM+<256.*2>
5641 025204 000004 4 ;4 WORDS
5642 025206 002001 2001 ;FILL WITH 2001
5643
5644 ;*FILL WRITE DATA COMMAND
5645 025210 004037 035054 JSR RO,@#RUN ;SETUP TO RUN FOR DATA COMMAND
5646 025214 000001 1 ;CYLINDER 1
5647 025216 000 .BYTE 0 ;SECTOR 0
5648 025217 000 .BYTE 0 ;TRACK 0
5649 025220 177400 -256. ;WORD COUNT = 256.
5650 025222 002370 WRFROM ;BUS ADDRESS
5651 025224 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
5652 025226 010000 FMT22 ;16 BITS PER WORD FORMAT
5653 025230 002342 WRIDAT ;GET READY TO DO A WRIDAT
5654
5655 ;*SAVE REGISTERS FOR COMPARISON AFTER ATTEMPTED
5656 ;*REGISTER MODIFICATION DURING A WRITE DATA
5657 025232 004037 033240 JSR RO,@#SAVER ;SAVE REGISTERS
5658 025236 002172 RHWC ;RHWC IS THE FIRST REGISTER SAVED
5659 025240 004512 SAVERE ;STARTING ADDRES OF WHERE
5660 025242 000022 18. ;NUMBER OF REGISTERS
5661
5662 025244 004737 033152 JSR PC,@#CHECKT ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
5663 025250 104401 057176 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF NOT
5664 025254 000000 HALT ;STOP TEST
5665
5666 025256 013777 004506 154702 MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
5667
5668 025264 013746 002342 MOV @#WRIDAT,-(SP) ;GET READY TO MOVE COMMAND
5669 025270 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET GO AND
5670 025274 012677 154700 MOV (SP)+,@RHCS1 ;GO WITH
5671

```


H 10

CZRJJCO, RP04/5/6 FCTNL CTRLR2 MACY11 30A(1052) 27-JUL-78 12:32 PAGE 125 SEQ 0124
CZRJJC.P11 26-JUL-78 10:07 T30 REGISTER MODIFICATION REFUSED - RHER1 (BIT #2), 'RMR'

```

5672 025300 013700 004650 MOV @#TMP0,RO ;SET RO TO REG ADDRESS
5673 025304 012730 002002 MOV #BIT1!BIT10,@(RO)+ ;ATTEMPT TO WRITE INTO A REGISTER
5674 ;DURING WRITE DATA
5675 025310 010037 004650 MOV RO,@#TMP0 ;SAVE OFF NEW REG ADDRESS
5676
5677 ;*NOW RMR MUST BE SET BUT THE COMPLETION OF
5678 ;*WRITE DATA MUST BE WAITED ON
5679
5680 025314 104413 WAT ;WAIT FOR RDY BIT TO SET
5681 025316 002200 RHCS1 ;WAIT FOR RHCS1 REGISTER
5682 025320 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
5683 025322 001725 981. ;ALLOW 9810 MICRO SECONDS
5684 025324 001502 834. ;RDY MUST SET BETWEEN
5685
5686 ;*CHANGE SAVED REGISTERS TO EXPECTED VALUE
5687 025326 004037 033762 JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
5688 025332 002234 RHCC ;CHANGE RHCC REGISTER
5689 025334 000001 1 ;1 BIT/BITS TO BE CHANGED
5690 025336 000001 1 ;NEW VALUE OF BIT0 IS 1
5691 025340 000001 BIT0 ;CHANGE BIT0 BIT
5692 025342 004037 033762 JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
5693 025346 002200 RHCS1 ;CHANGE RHCS1 REGISTER
5694 025350 000001 1 ;1 BIT/BITS TO BE CHANGED
5695 025352 000001 1 ;NEW VALUE OF SC IS 1
5696 025354 100000 SC ;CHANGE SC BIT
5697 025356 004037 033762 JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
5698 025362 002222 RHDS1 ;CHANGE RHDS1 REGISTER
5699 025364 000002 2 ;2 BIT/BITS TO BE CHANGED
5700 025366 000001 1 ;NEW VALUE OF ATA IS 1
5701 025370 100000 ATA ;CHANGE ATA BIT
5702 025372 000001 1 ;NEW VALUE OF ERR IS 1
5703 025374 040000 ERR ;CHANGE ERR BIT
5704 025376 053737 004644 004536 BIS @#ATTENT,@#SAVERE+24 ;SET APPROPRIATE 'ATA' BITS
5705 025404 004037 033762 JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
5706 025410 002202 RHER1 ;CHANGE RHER1 REGISTER
5707 025412 000001 1 ;1 BIT/BITS TO BE CHANGED
5708 025414 000001 1 ;NEW VALUE OF RMR IS 1
5709 025416 000004 RMR ;CHANGE RMR BIT
5710 025420 004037 032774 JSR RO,@#FILLRE ;MOV 1 INTO SAVED RHDST
5711 025424 002204 RHDST ;SAVED REGISTER TO CHANGE
5712 025426 000001 1 ;DATA
5713 025430 004037 032774 JSR RO,@#FILLRE ;MOV WRFROM+<256.*2> INTO SAVED RHBA
5714 025434 002174 RHBA ;SAVED REGISTER TO CHANGE
5715 025436 003370 WRFROM+<256.*2> ;DATA
5716 025440 004037 032774 JSR RO,@#FILLRE ;MOV 0 INTO SAVED RHWC
5717 025444 002172 RHWC ;SAVED REGISTER TO CHANGE
5718 025446 000000 0 ;DATA
5719
5720 ;*COMPARE REGISTERS BEFORE WRITE DATA WITH REGISTERS
5721 ;*AFTER WRITE AND ATTEMPTED MODIFICATION OF REGISTER
5722 025450 004037 034070 JSR RO,@#COMREG ;COMPARE SAVED REGISTERS WITH
5723 025454 004512 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
5724 025456 002254 WC ;TEST DATA STARTING FROM 'RHWC'
5725 025460 000022 18. ;18. REGISTERS TO BE COMPARED
5726 025462 025466 2$ ;RETURN TO 2$ ON ERROR
5727 025464 025506 3$ ;RETURN TO 3$ ON NO ERROR

```



```

5728
5729 025466          2$:
5730 025466 010046      MOV    R0,-(SP)          ;;PUSH R0 ON STACK
5731 025470 013700 004650  MOV    @#TMP0,R0        ;;GET REGISTER BEEING MODIFIED + 2 POINTER
5732 025474 014037 001122  MOV    -(R0),@#$BDADR   ;;GET ADDRESS OF REGISTER BEING MODIFIED
5733 025500 104060      ERROR  60              ;;ATTEMPTING TO MODIFY REGISTER
5734 025502 012600      MOV    (SP)+,R0         ;;POP STACK INTO R0
5735 025504 000207      RTS     PC              ;;DURING A WRITE COMMAND CAUSED
5736                                     ;;IMPROPER REGISTER GIVES WHAT SHOULD
5737                                     ;;GOOD DATA GIVES WHAT SHOULD
5738                                     ;;BE THERE
5739                                     ;;RECEIVED DATA GIVES WHAT WAS
5740                                     ;;THERE AFTER READ
5741                                     ;*CLEAR ALL ERROR FLAGS
5742 025506          3$:
5743 025506 004737 033072  JSR    PC,@#CLDISK     ;;SET R1-RHCS1, R2-RHCS2
5744
5745                                     ;*FILL WRITE FROM BUFFER WITH EXPECTED DATA
5746 025512 004037 032742  JSR    R0,@#CLAREA     ;;CLEAR 256. WORDS, FROM WRFROM
5747 025516 002370      WRFROM                ;;STARTING FROM WRFROM
5748 025520 000400      256.                 ;;256. WORDS
5749 025522 002000      2000                 ;;FILL WITH 2000
5750 025524 004037 032742  JSR    R0,@#CLAREA     ;;CLEAR 4 WORDS, FROM WRFROM+<256.*2>
5751 025530 003370      WRFROM+<256.*2>      ;;STARTING FROM WRFROM+<256.*2>
5752 025532 000004      4                    ;;4 WORDS
5753 025534 002001      2001                 ;;FILL WITH 2001
5754
5755                                     ;*NOW THE READ DATA COMMAND WILL BE FILLED
5756 025536 004037 035054  JSR    R0,@#RUN        ;;SETUP TO RUN FOR DATA COMMAND
5757 025542 000001      1                    ;;CYLINDER 1
5758 025544 000      .BYTE 0                ;;SECTOR 0
5759 025545 000      .BYTE 0                ;;TRACK 0
5760 025546 177374      -260.                ;;WORD COUNT = 260.
5761 025550 003434      REINTO                ;;BUS ADDRESS
5762 025552 000000      0                    ;;DO NOT INHIBIT BUS ADDRESS INCREMENT
5763 025554 014000      ECI!FMT22            ;;16 BITS PER WORD FORMAT
5764 025556 002346      READAT                ;;GET READY TO DO A READAT
5765
5766                                     ;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ DATA COMMAND
5767 025560 004037 033240  JSR    R0,@#SAVER      ;;SAVE REGISTERS
5768 025564 002172      RHWC                  ;;RHWC IS THE FIRST REGISTER SAVED
5769 025566 004512      SAVERE                ;;STARTING ADDRES OF WHERE
5770 025570 000022      18.                  ;;NUMBER OF REGISTERS
5771
5772 025572 004737 033152  JSR    PC,@#CHECKT    ;;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
5773 025576 104401 057176  TYPE    .CPHALT        ;;CANNOT CONTINUE TESTING IF NOT
5774 025602 000000      HALT                 ;;STOP TEST
5775
5776 025604 013777 004506 154354  MOV    @#RP4VEC,@RPVEC ;;SET RP04 VECTOR ADDRESS
5777
5778 025612 013746 002346  MOV    @#READAT,-(SP)  ;;GET READY TO MOVE COMMAND
5779 025616 052716 000101  BIS    #GO!IE,(SP)     ;;GET READY TO SET GO AND
5780 025622 012677 154352  MOV    (SP)+,@RHCS1    ;;GO WITH
5781 025626 011100      MOV    @R1,R0         ;;SAVE RHCS1 DURING ABOVE OPERATION
5782 025630 011305      MOV    @R3,R5         ;;SAVE RHDS1 DURING ABOVE OPERATION
5783
  
```



```

5784 025632 104413          WAT          ;WAIT FOR RDY BIT TO SET
5785 025634 002200          RHCS1         ;WAIT FOR RHCS1 REGISTER
5786 025636 000200          RDY          ;WAIT FOR RDY BIT IN RHCS1 REGISTER
5787 025640 001725          981.         ;ALLOW 9810 MICRO SECONDS
5788 025642 001502          834.         ;RDY MUST SET BETWEEN
5789 025644 013746 002346    MOV @#READAT,-(SP) ;SAVE COMMAND
5790 025650 052716 004101    BIS #IE!DVA!GO,(SP) ;INCLUDE IE!DVA!GO
5791 025654 011637 001124    MOV (SP),@#$GDDAT ;SAVE FOR PRINTOUT
5792 025660 022600          CMP (SP)+,R0  ;DURING ABOVE OPERATION ONLY IE!DVA!GO
5793 025662 001405          BEQ 70$      ;BRANCH IF GOOD
5794 025664 010037 001126    MOV R0,@#$BDDAT ;BAD DATA
5795 025670 010137 004500    MOV R1,@#REGADR ;FAILING REGISTER RHCS1
5796 025674 104021          ERROR 21     ;DURING ABOVE OPERATION ONLY
5797 025676 012746 010500    MOV #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
5798 025702 011637 001124    MOV (SP),@#$GDDAT ;SAVE FOR PRINTOUT
5799 025706 022605          CMP (SP)+,R5  ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
5800 025710 001405          BEQ 72$      ;BRANCH IF GOOD
5801 025712 010537 001126    MOV R5,@#$BDDAT ;BAD DATA
5802 025716 010337 004500    MOV R3,@#REGADR ;FAILING REGISTER RHDS1
5803 025722 104063          ERROR 63     ;DURING ABOVE OPERATION ONLY
5804
5805          ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
5806 025724 004037 032774    JSR R0,@#FILLRE ;MOV 0 INTO SAVED RHWC
5807 025730 002172          RHWC         ;SAVED REGISTER TO CHANGE
5808 025732 000000          0           ;DATA
5809
5810 025734 004037 032774    JSR R0,@#FILLRE ;MOV REINTO+<260.*2> INTO SAVED RHBA
5811 025740 002174          RHBA         ;SAVED REGISTER TO CHANGE
5812 025742 004444          REINTO+<260.*2> ;DATA
5813 025744 004037 032774    JSR R0,@#FILLRE ;MOV 2 INTO SAVED RHDST
5814 025750 002204          RHDST        ;SAVED REGISTER TO CHANGE
5815 025752 000002          2           ;DATA
5816
5817          ;*NOW COMPARE REGISTERS BEFORE READ DATA WITH
5818          ;*AFTER COMMAND
5819 025754 004037 034070    JSR R0,@#COMREG ;COMPARE SAVED REGISTERS WITH
5820 025760 004512          SAVERE       ;GOOD DATA SAVED IN 'SAVERE'
5821 025762 002254          WC          ;TEST DATA STARTING FROM 'RHWC'
5822 025764 000022          18.         ;18. REGISTERS TO BE COMPARED
5823 025766 025772          4$          ;RETURN TO 4$ ON ERROR
5824 025770 025776          5$          ;RETURN TO 5$ ON NO ERROR
5825
5826 025772 104033 4$:      ERROR 33     ;READ DATA CAUSED IMPROPER REGISTER
5827 025774 000207          RTS PC      ;CHANGE
5828          ;GOOD DATA GIVES WHAT SHOULD BE THERE
5829          ;RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND
5830          ;*NOW READ INTO BUFFER WILL BE CHECKED TO SEE THAT READ
5831          ;*WAS GOOD
5832 025776 5$:
5833 025776 004037 035120    JSR R0,@#COMPAR ;COMPARE TWO BLOCKS OF MEMORY
5834 026002 002370          WRFROM      ;GOOD DATA STARTS FROM WRFROM
5835 026004 003434          REINTO      ;TEST DATA STARTS FROM REINTO
5836 026006 000400          256.       ;256. WORDS TO BE COMPARED
5837 026010 026014          6$          ;RETURN TO 6$ ON ERROR
5838 026012 026020          ST28       ;RETURN TO ST28 ON NO ERROR
5839

```


K 10

5840	026014	104034		6\$:	ERROR	34		:READ DATA ERROR AFTER A WRITE DATA
5841	026016	000207			RTS	PC		:WITH REGISTER MODIFICATION
5842								:WITHIN THE WRITE DATA
5843								:*IF ALL 7 REGISTERS NOT COMPLETE THEN REPEAT
5844	026020	005337	004656	ST28:	DEC	@#TMP5		:COUNT DOWN
5845	026024	001002			BNE	1\$:BRANCH IF 7 NOT DONE
5846	026026	000137	026036		JMP	TST31		:JUMP TO NEXT TEST
5847	026032	000137	025004	1\$:	JMP	ST24		:JUMP TO BEGINING OF TEST
5848								


```

5849
5850 026036 000004          TST31: SCOPE
5851 026040 012706 001000  MOV      #STACK,SP      ;RESET STACK
5852 026044 012737 000031 004504  MOV      #31,@#TSTNM    ;SAVE TEST NUMBER
5853 026052 004737 033072  JSR      PC,@#CLDISK    ;SET R1-RHCS1, R2-RHCS2
5854
5855                      ;*FILL WRITE FROM BUFFER WITH EXPECTED DATA
5856 026056 004037 032742  JSR      R0,@#CLAREA    ;CLEAR 256. WORDS, FROM WRFROM
5857 026062 002370          WRFROM                ;STARTING FROM WRFROM
5858 026064 000400          256.                  ;256. WORDS
5859 026066 000000          0                      ;FILL WITH 0
5860                      ;*FILL READ INTM BUFFER WITH ALL ONES
5861 026070 004037 032742  JSR      R0,@#CLAREA    ;CLEAR 256. WORDS, FROM REINTO
5862 026074 003434          REINTO                ;STARTING FROM REINTO
5863 026076 000400          256.                  ;256. WORDS
5864 026100 177777          -1                     ;FILL WITH -1
5865                      ;*NOW THE READ DATA COMMAND WILL BE FILLED
5866 026102 004037 035054  JSR      R0,@#RUN      ;SETUP TO RUN FOR DATA COMMAND
5867 026106 000000          0                      ;CYLINDER 0
5868 026110          000                ;SECTOR 0
5869 026111          000                ;TRACK 0
5870 026112 177400          -256.                 ;WORD COUNT = 256.
5871 026114 003434          REINTO                ;BUS ADDRESS
5872 026116 000000          0                      ;DO NOT INHIBIT BUS ADDRESS INCREMENT
5873 026120 014000          ECI!FMT22             ;16 BITS PER WORD FORMAT
5874 026122 002346          READAT               ;GET READY TO DO A READAT
5875                      ;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ DATA COMMAND
5876 026124 004037 033240  JSR      R0,@#SAVER    ;SAVE REGISTERS
5877 026130 002172          RHWC                  ;RHWC IS THE FIRST REGISTER SAVED
5878 026132 004512          SAVERE               ;STARTING ADDRES OF WHERE
5879 026134 000022          18.                  ;NUMBER OF REGISTERS
5880
5881 026136 004737 033152  JSR      PC,@#CHECKT   ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
5882 026142 104401 057176  TYPE      ,CPHALT      ;CANNOT CONTINUE TESTING IF NOT
5883 026146 000000          HALT                 ;STOP TEST
5884
5885 026150 013777 004506 154010  MOV      @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
5886
5887
5888 026156 013746 002346  MOV      @#READAT,-(SP) ;GET READY TO MOVE COMMAND
5889 026162 052716 000101  BIS      #GO!IE,(SP)    ;GET READY TO SET GO AND
5890 026166 012677 154006  MOV      (SP)+,@RHCS1   ;GO WITH
5891 026172 011100          MOV      @R1,R0        ;SAVE RHCS1 DURING ABOVE OPERATION
5892 026174 011305          MOV      @R3,R5        ;SAVE RHDS1 DURING ABOVE OPERATION
5893
5894 026176 012777 177777 154012  MOV      #-1,@RHAS     ;WRITE INTO RHAS THIS SHOULD
5895                      ;NOT SET RMR
5896
5897                      ;*TIME IS NOT IMPORTANT
5898 026204 104413          WAT                   ;WAIT FOR RDY BIT TO SET
5899 026206 002200          RHCS1                 ;WAIT FOR RHCS1 REGISTER
5900 026210 000200          RDY                    ;WAIT FOR RDY BIT IN RHCS1 REGISTER
5901 026212 003326          1750.                 ;ALLOW 17500 MICRO SECONDS
5902 026214 000175          125.                  ;RDY MUST SET BETWEEN
5903 026216 013746 002346  MOV      @#READAT,-(SP) ;SAVE COMMAND
5904 026222 052716 004101  BIS      #IE!DVA!GO,(SP) ;INCLUDE IE!DVA!GO
  
```



```

5905 026226 011637 001124      MOV      (SP),@#$GDDAT      ;SAVE FOR PRINTOUT
5906 026232 022600              CMP      (SP)+,R0          ;DURING ABOVE OPERATION ONLY IE!DVA!GO
5907 026234 001405              BEQ      64$              ;BRANCH IF GOOD
5908 026236 010037 001126      MOV      R0,@#$BDDAT      ;BAD DATA
5909 026242 010137 004500      MOV      R1,@#REGADR      ;FAILING REGISTER RHCS1
5910 026246 104021              ERROR    21              ;DURING ABOVE OPERATION ONLY
5911 026250 012746 010500      64$:    MOV      #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
5912 026254 011637 001124      MOV      (SP),@#$GDDAT      ;SAVE FOR PRINTOUT
5913 026260 022605              CMP      (SP)+,R5          ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
5914 026262 001405              BEQ      66$              ;BRANCH IF GOOD
5915 026264 010537 001126      MOV      R5,@#$BDDAT      ;BAD DATA
5916 026270 010337 004500      MOV      R3,@#REGADR      ;FAILING REGISTER RHDS1
5917 026274 104063              ERROR    63              ;DURING ABOVE OPERATION ONLY
5918
5919                          ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
5920 026276 004037 032774      JSR      R0,@#FILLRE      ;MOV 0 INTO SAVED RHWC
5921 026302 002172              RHWC
5922 026304 000000              0                          ;SAVED REGISTER TO CHANGE
5923 026306 004037 032774      JSR      R0,@#FILLRE      ;MOV REINTO+<256.*2> INTO SAVED RHBA
5924 026312 002174              RHBA
5925 026314 004434              REINTO+<256.*2>          ;DATA
5926 026316 004037 032774      JSR      R0,@#FILLRE      ;MOV 1 INTO SAVED RHDST
5927 026322 002204              RHDST
5928 026324 000001              1                          ;SAVED REGISTER TO CHANGE
5929 026326 004037 032774      JSR      R0,@#FILLRE      ;MOV 0 INTO SAVED RHCC
5930 026332 002234              RHCC
5931 026334 000000              0                          ;SAVED REGISTER TO CHANGE
5932
5933                          ;*NOW COMPARE REGISTERS BEFORE READ DATA WITH
5934                          ;*AFTER COMMAND
5935 026336 004037 034070      JSR      R0,@#COMREG      ;COMPARE SAVED REGISTERS WITH
5936 026342 004512              SAVERE                      ;GOOD DATA SAVED IN 'SAVERE'
5937 026344 002254              WC                          ;TEST DATA STARTING FROM 'RHWC'
5938 026346 000022              18.                         ;18. REGISTERS TO BE COMPARED
5939 026350 026354              1$                          ;RETURN TO 1$ ON ERROR
5940 026352 026360              2$                          ;RETURN TO 2$ ON NO ERROR
5941
5942 026354 104033              1$:    ERROR    33          ;READ DATA CAUSED IMPROPER REGISTER
5943 026356 000207              RTS      PC                ;CHANGE
5944
5945                          ;GOOD DATA GIVES WHAT SHOULD BE THERE
5946                          ;RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND
5947                          ;*NOW READ INTO BUFFER WILL BE CHECKED TO SEE THAT READ
5948                          ;*WAS GOOD
5949 026360 004037 035120      2$:    JSR      R0,@#COMPAR      ;COMPARE TWO BLOCKS OF MEMORY
5950 026364 002370              WRFROM                      ;GOOD DATA STARTS FROM WRFROM
5951 026366 003434              REINTO                      ;TEST DATA STARTS FROM REINTO
5952 026370 000400              256.                        ;256., WORDS TO BE COMPARED
5953 026372 026376              3$                          ;RETURN TO 3$ ON ERROR
5954 026374 026402              4$                          ;RETURN TO 4$ ON NO ERROR
5955
5956 026376 104034              3$:    ERROR    34          ;READ DATA ERROR AFTER WRITING INTO
5957 026400 000207              RTS      PC                ;RHAS DURING READ
5958
5959 026402              4$:

```



```

5960
5961 026402 000004          TST32: SCOPE
5962 026404 012706 001000    MOV    #STACK,SP      ;RESET STACK
5963 026410 012737 000032 004504    MOV    #32,@#TSTNM   ;SAVE TEST NUMBER
5964 026416 004737 033072    JSR    PC,@#CLDISK   ;SET R1-RHCS1, R2-RHCS2
5965
5966 026422 005737 004640    TST    @#RH70        ;RH70 CONTROLLER ?
5967 026426 001402          BEQ    30$           ;SKIP NEXT IF NOT = 1
5968 026430 000137 027230    JMP    TST33         ;IF SO SKIP THIS TEST -----)
5969 026434          30$:
5970
5971          ;*GENERATE ILLEGAL FUNCTION
5972
5973 026434 005037 001200    CLR    @#$TMP1       ;GET READY TO MAKE ILLEGAL FUNCTION
5974 026440 012700 002322    1$:  MOV    #FUTABL,RO   ;GET POINTER TO BEGINNING OF COMMANDS
5975 026444 012705 000021    MOV    #17,R5       ;COUNTER (17 GOOD FUNCTIONS)
5976 026450 023720 001200    2$:  CMP    @#$TMP1,(RO)+ ;IS THIS A LEGAL FUNCTION
5977 026454 001004          BNE    3$           ;BRANCH IF NOT LEGAL
5978 026456 062737 000002 001200    ADD    #2,@#$TMP1   ;MAKE ANOTHER FUNCTION
5979 026464 000765          BR     1$           ;GET READY TO TEST NEW FUNCTION
5980 026466 005305          3$:  DEC    R5           ;NOT LEGAL SO DECREMENT COUNTER
5981 026470 001367          BNE    2$           ;BRANCH IF 17 NOT DONE
5982 026472 032737 000100 001200    BIT    #100,@#$TMP1 ;ALL BITS UP TO BIT #5 COMPARED?
5983 026500 001001          BNE    20$          ;BRANCH OUT IF DONE
5984 026502 000402          BR     19$          ;BRANCH TO CONTINUE
5985 026504 000137 027230    20$: JMP    @#7$         ; DONE
5986 026510 013737 001200 002364 19$:  MOV    @#$TMP1,@#ILLEGL ;AN ILLEGAL FUNCTION IS FOUND
5987 026516 062737 000002 001200    ADD    #2,@#$TMP1   ;GET READY FOR NEW FUNCTION NEXT TIME
5988
5989          ;*ILLEGAL FUNCTION HAS BEEN FOUND
5990          ;*IT IS IN 'ILLEGL'
5991 026524 012737 026532 001110    MOV    #4$,@#$LPERR ;ERROR RETURN POINT
5992
5993          ;*SAVE REGISTERS FOR COMPARISON AFTER GO
5994          4$:
5995 026532 004737 033072    JSR    PC,@#CLDISK   ;SET R1-RHCS1, R2-RHCS2
5996 026536 005077 153430    CLR    @#RHC        ;CLEAR WORD COUNT
5997 026542 005077 153426    CLR    @#RHBA        ;CLEAR BUS ADDRESS
5998 026546 023727 002364 000050    CMP    @#ILLEGL,#50 ;50 AND HIGHER FUNCTIONS ARE DATA
5999          ;FUNCTIONS WHICH WILL SET MXF AND TRE
6000 026554 103014          BHS    13$          ;BRANCH IF ILLEGL IS HIGHER THAN 50
6001 026556 012737 100000 027132    MOV    #SC,@#11$+12 ;EXPECTED VALUE OF RHCS1 SHOULD HAVE
6002          ;ONLY SC ADDED
6003 026564 005037 027154          CLR    @#12$+12    ;EXPECTED VALUE OF RHCS2 SHOULD HAVE
6004          ;NOTHING ADDED
6005 026570 005037 027160    CLR    @#12$+16     ;NO BITS TO BE CLEARED IN RHCS2
6006 026574 005037 027170    CLR    @#15$+6      ;RHBA SHOULD BE 0
6007 026600 005037 027200    CLR    @#16$+6      ;CLEAR SAVED RHWC
6008 026604 000500          BR     14$          ;BRANCH
6009 026606 022737 000064 002364 13$:  CMP    #64,@#ILLEGL ;IS FUNCTION 64
6010 026614 001020          BNE    17$          ;BRANCH IF NOT
6011 026616 012737 140000 027132    MOV    #SC!TRE,@#11$+12 ;SAVED RHCS1 SHOULD HAVE SC AND TRE
6012 026624 012737 000204 027170    MOV    #204,@#15$+6 ;RHBA SHOULD HAVE 204
6013 026632 012737 000102 027200    MOV    #102,@#16$+6 ;RHWC SHOULD HAVE 102
6014 026640 012737 001200 027154    MOV    #MXF!OR,@#12$+12 ;RHCS2 SHOULD HAVE MXF AND OR
6015 026646 012737 000100 027160    MOV    #IR,@#12$+16 ;RHCS2 SHOULD HAVE IR CLEARED
    
```


6016	026654	000454			BR	14\$:BRANCH
6017	026656	022737	000066	002364	17\$:	CMP	#66,@#:LLEGL ;IS FUNCTION 66
6018	026664	001030			BNE	18\$:BRANCH IF NOT
6019	026666	012777	177672	153276	MOV	#-70,@RHWC	: MOVE 70 INTO RHWC
6020	026674	012777	002370	153272	MOV	#WRFROM,@RHBA	:FILL RHBA WITH WRFROM
6021	026702	012737	140000	027132	MOV	#SC!TRE,@#11\$+12	:SAVED RHCS1
6022	026710	012737	002164	027170	MOV	#WRFROM-<66.*2>,15\$+6	:RHBA
6023	026716	012737	177774	027200	MOV	#-4.,16\$+6	:SAVED RHWC
6024	026724	012737	001200	027154	MOV	#MXF!OR,@#12\$+12	:SAVED RHCS2
6025	026732	005037	027160		CLR	@#12\$+16	:RHCS2
6026	026736	012737	000100	027160	MOV	#IR,@#12\$+16	:RHCS2 SHOULD HAVE IR CLEARED
6027	026744	000420			BR	14\$:BRANCH
6028	026746	005077	153220		18\$:	CLR	@RHWC ;CLEAR RHWC
6029	026752	005077	153216		CLR	@RHBA	:CLEAR RHBA
6030	026756	012737	140000	027132	MOV	#SC!TRE,@#11\$+12	:RHCS1 SHOULD HAVE SC AND TRE
6031	026764	005037	027170		CLR	@#15\$+6	:RHBA
6032	026770	005037	027200		CLR	@#16\$+6	:RHWC
6033	026774	012737	001000	027154	MOV	#MXF,@#12\$+12	:RHCS2
6034	027002	005037	027160		CLR	@#12\$+16	:RHCS2
6035	027006				14\$:		
6036	027006	004037	033240		JSR	RO,@#SAVER	:SAVE REGISTERS
6037	027012	002172			RHWC		:RHWC IS THE FIRST REGISTER SAVED
6038	027014	004512			SAVERE		:STARTING ADDRES OF WHERE
6039	027016	000022			18.		:NUMBER OF REGISTERS
6040							
6041	027020	004737	033152		JSR	PC,@#CHECKT	:CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
6042	027024	104401	057176		TYPE	,CPHALT	:CANNOT CONTINUE TESTING IF NOT
6043	027030	000000			HALT		:STOP TEST
6044	027032	013746	002364		MOV	@#ILLEGL,-(SP)	:GET ILLEGAL FUNCTION
6045	027036	052716	000101		BIS	#GO!IE,(SP)	:INCLUDE IE AND GO
6046	027042	012611			MOV	(SP)+,@R1	:GO TO RHCS1 WITH ILLEGAL FUNCTION
6047	027044	104413			WAT		:WAIT FOR RDY BIT TO SET
6048	027046	002200			RHCS1		:WAIT FOR RHCS1 REGISTER
6049	027050	000200			RDY		:WAIT FOR RDY BIT IN RHCS1 REGISTER
6050	027052	001614			908.		:ALLOW 9080 MICRO SECONDS
6051	027054	001613			907.		:RDY MUST SET BETWEEN
6052							
6053							:*CHANGE SAVED REGISTERS TO EXPECTED VALUE
6054	027056	004037	033762		JSR	RO,@#CHREG	:CHANGE BITS IN SAVED REGISTER
6055	027062	002202			RHER1		:CHANGE RHER1 REGISTER
6056	027064	000001			1		:1 BIT/BITS TO BE CHANGED
6057	027066	000001			1		:NEW VALUE OF ILF IS 1
6058	027070	000001			ILF		:CHANGE ILF BIT
6059	027072	004037	033762		JSR	RO,@#CHREG	:CHANGE BITS IN SAVED REGISTER
6060	027076	002222			RHDS1		:CHANGE RHDS1 REGISTER
6061	027100	000002			2		:2 BIT/BITS TO BE CHANGED
6062	027102	000001			1		:NEW VALUE OF ATA IS 1
6063	027104	100000			ATA		:CHANGE ATA BIT
6064	027106	000001			1		:NEW VALUE OF ERR IS 1
6065	027110	040000			ERR		:CHANGE ERR BIT
6066	027112	053737	004644	004536	BIS	@#ATTENT,@#SAVERE+24	:SET APPROPRIATE 'ATA' BITS
6067							
6068							:*RHCS1 WILL HAVE SC AND TRE ADDED IF FUNCTION IS GREATER THAN 50
6069	027120				11\$:		
6070	027120	004037	033762		JSR	RO,@#CHREG	:CHANGE BITS IN SAVED REGISTER
6071	027124	002200			RHCS1		:CHANGE RHCS1 REGISTER


```

6072 027126 000001          1          ;1 BIT/BITS TO BE CHANGED
6073 027130 000001          1          ;NEW VALUE OF SC IS 1
6074 027132 100000          SC         ;CHANGE SC BIT
6075 027134 053737 002364 004520  BIS      @#ILLEGL,@#SAVERE+6;INCLUDE ILLEGAL FUNCTION
6076                                     ;IN RHCS1
6077                                     ;*RHCS2 WILL HAVE NOTHING ADDED IF FUNCTION IS LESS THAN 50
6078 027142                                     12$:
6079 027142 004037 033762  JSR      RO,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
6080 027146 002176          RHCS2     ;CHANGE RHCS2 REGISTER
6081 027150 000002          2          ;2 BIT/BITS TO BE CHANGED
6082 027152 000001          1          ;NEW VALUE OF MXF IS 1
6083 027154 001000          MXF        ;CHANGE MXF BIT
6084 027156 000000          0          ;NEW VALUE OF IR IS 0
6085 027160 000100          IR         ;CHANGE IR BIT
6086 027162                                     15$:
6087 027162 004037 032774  JSR      RO,@#FILLRE     ;MOV 0 INTO SAVED RHBA
6088 027166 002174          RHBA       ;SAVED REGISTER TO CHANGE
6089 027170 000000          0          ;DATA
6090 027172                                     16$:
6091 027172 004037 032774  JSR      RO,@#FILLRE     ;MOV 0 INTO SAVED RHWC
6092 027176 002172          RHWC       ;SAVED REGISTER TO CHANGE
6093 027200 000000          0          ;DATA
6094
6095                                     ;*NOW COMPARE REGISTERS AFTER GIVING AN ILLEGAL COMMAND
6096 027202 004037 034070  JSR      RO,@#COMREG     ;COMPARE SAVED REGISTERS WITH
6097 027206 004512          SAVERE     ;GOOD DATA SAVED IN 'SAVERE'
6098 027210 002254          WC         ;TEST DATA STARTING FROM 'RHWC'
6099 027212 000022          18.       ;18. REGISTERS TO BE COMPARED
6100 027214 027220          5$        ;RETURN TO 5$ ON ERROR
6101 027216 027224          6$        ;RETURN TO 6$ ON NO ERROR
6102
6103 027220 104051          5$:      ERROR  51      ;GIVING ILLEGAL FUNCTION CAUSED
6104 027222 000207          RTS      PC      ;IMPROPER REGISTER CHANGE
6105                                     ;GOOD DATA GIVES WHAT
6106                                     ;SHOULD BE THERE
6107                                     ;RECEIVED DATA GIVES REGISTER
6108                                     ;CONTENTS AFTER ILLEGAL
6109                                     ;FUNCTION WA GIVEN
6110 027224 000137 026440  6$:      JMP      @#1$      ;BRANCH FOR NEXT FUNCTION
6111 027230          7$:
6112
6113 027230          10$:
6114
6115
6116
6117

```



```

6118
6119 027230 000004          TST33: SCOPE
6120 027232 012706 001000      MOV    #STACK,SP          :RESET STACK
6121 027236 012737 000033 004504  MOV    #33,@#TSTNM       :SAVE TEST NUMBER
6122 027244 004737 033072      JSR    PC,@#CLDISK       :SET R1-RHCS1, R2-RHCS2
6123
6124          ;*THESE ARE REGULAR SETUPS
6125 027250 012777 177374 152714  MOV    #-260.,@RHWC      :256 DATA WORDS 4 HEADER WORDS
6126 027256 012700 002370      MOV    #WRFROM,R0        :THESE TWO INSTRUCTIONS GETS
6127 027262 010077 152706      MOV    R0,@RHBA         :ADDR. OF WRFROM BUFFER INTO R0 AND
6128          :BUS ADDRESS REGISTER
6129 027266 012710 010000      MOV    #FMT22,(R0);     :FORMAT=16 BIT WORDS
6130          :CYLINDER=0
6131 027272 012720 000001      MOV    #1,(R0)+         :TRACK=0, SECTOR=1, KEYS=0
6132 027276 005020          CLR    (R0)+            :KEY1=0
6133 027300 005020          CLR    (R0)+            :KEY2=0
6134 027302 012705 000400      MOV    #256.,R5         :COUNTER
6135
6136          ;*SETUP DATA, WRITE HEADER & DATA, AND FORMAT OF THE WRITE
6137 027306 012720 177777          1$:  MOV    #-1,(R0)+       :MOVE ALL ONES FOR DATA
6138 027312 005305          DEC    R5
6139 027314 001374          BNE    1$              :BRANCH IF DATA NOT COMPLETE
6140 027316 012777 000001 152660  MOV    #1,@RH DST       :TRACK=0 SECTOR=1
6141 027324 004737 033152      JSR    PC,@#CHECKT     :CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
6142 027330 104401 057176      TYPE   ,CPHALT        :CANNOT CONTINUE TESTING IF NOT
6143 027334 000000          HALT                   :STOP TEST
6144 027336 013711 002344      MOV    @#WRIFOR,@R1    :GET READY FOR WRITE HEADER AND
6145          :DATA WITH 62 IN RHCS1
6146 027342 005037 004632      CLR    @#ERFLG$       :CLEAR ERROR FLAG
6147 027346 012777 010000 152634  MOV    #FMT22,@RHOF    :FORMAT BIT=1 (16 BIT WORDS)
6148 027354 005077 152632      CLR    @RHCA          :CYLINDER =0
6149
6150          ;*SAVE REGISTERS FOR COMPARISON AFTER READ
6151 027360 004037 033240      JSR    R0,@#SAVER      :SAVE REGISTERS
6152 027364 002172          RHWC                   :RHWC IS THE FIRST REGISTER SAVED
6153 027366 004512          SAVERE                 :STARTING ADDRES OF WHERE
6154 027370 000023          19.                   :NUMBER OF REGISTERS

```



```

6155
6156
6157
6158
6159
6160
6161 027372 013700 002220      MOV    @#RHMR,RO      ;NOW RO HAS MAINTENANCE REG. ADDR.
6162 027376 012710 000001      MOV    #DMD,@RO      ;SET DIAGNOSTIC MODE
6163 027402 052710 000004      BIS    #MINX,@RO     ;SET INDEX
6164 027406 042710 000004      BIC    #MINX,@RO     ;CLEAR INDEX THIS GIVES
6165
6166
6167 027412 052777 000001 152560  BIS    #GO,@RHCS1    ;ISSUE THE 'GO' BIT TO THE RH11
6168 027420 012737 000113 004642  RUNWAT: MOV    #75.,@#RUNCTR ;LOAD 'RUN' LINE DELAY COUNTER
6169
6170
6171 027426 005337 004642      1$:   DEC    @#RUNCTR ;= APPROX 450 US ON 11/50 CPU WITH CORE
6172 027432 001375
6173
6174
6175 027434 052710 000004      BIS    #MINX,@RO     ;ISSUE THE FIRST DIAGNOSTIC INDEX PULSE
6176 027440 042710 000004      BIC    #MINX,@RO     ;SET INDEX PULSE
6177
6178
6179 027444 052710 000004      BIS    #MINX,@RO     ;RESET INDEX
6180 027450 042710 000004      BIC    #MINX,@RO     ;SECOND INDEX PULSE
6181
6182
6183 027454 052710 000004      BIS    #MINX,@RO     ;SET INDEX
6184 027460 042710 000004      BIC    #MINX,@RO     ;CLEAR INDEX
6185
6186
6187
6188 027464 004037 033762      JSR    RO,@#CHREG    ;*CHANGE SAVED REGISTERS TO EXPECTED VALUE
6189 027470 002200      RHCS1                ;CHANGE BITS IN SAVED REGISTER
6190 027472 000002      2                    ;CHANGE RHCS1 REGISTER
6191 027474 000001      1                    ;2 BIT/BITS TO BE CHANGED
6192 027476 100000      SC                   ;NEW VALUE OF SC IS 1
6193 027500 000001      1                    ;CHANGE SC BIT
6194 027502 040000      TRE                  ;NEW VALUE OF TRE IS 1
6195 027504 004037 033762      JSR    RO,@#CHREG    ;CHANGE TRE BIT
6196 027510 002222      RHDS1                ;CHANGE BITS IN SAVED REGISTER
6197 027512 000002      2                    ;CHANGE RHDS1 REGISTER
6198 027514 000001      1                    ;2 BIT/BITS TO BE CHANGED
6199 027516 100000      ATA                  ;NEW VALUE OF ATA IS 1
6200 027520 000001      1                    ;CHANGE ATA BIT
6201 027522 040000      ERR                  ;NEW VALUE OF ERR IS 1
6202 027524 004037 032774      JSR    RO,@#FILLRE   ;CHANGE ERR BIT
6203 027530 002204      RHDST                ;MOV 2 INTO SAVED RHDST
6204 027532 000002      2                    ;SAVED REGISTER TO CHANGE
6205 027534 004037 033762      JSR    RO,@#CHREG    ;DATA
6206 027540 002202      RHER1                ;CHANGE BITS IN SAVED REGISTER
6207 027542 000001      1                    ;CHANGE RHER1 REGISTER
6208 027544 000001      1                    ;1 BIT/BITS TO BE CHANGED
6209 027546 020000      OPI                  ;NEW VALUE OF OPI IS 1
6210 027550 053737 004644 004536  BIS    @#ATTENT,@#SAVERE+24 ;CHANGE OPI BIT
        ;SET APPROPRIATE 'ATA' BITS
    
```



```

6211 027556 004037 033762 JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
6212 027562 002220 RHMR ;CHANGE RHMR REGISTER
6213 027564 000001 1 ;1 BIT/BITS TO BE CHANGED
6214 027566 000001 1 ;NEW VALUE OF DMD IS 1
6215 027570 000001 DMD ;CHANGE DMD BIT
6216
6217
6218 ;*RHWC,RHBA AND OR AND IR BITS OF RHCS2 WILL NOT BE CHECKED
6219 027572 017737 152374 004512 MOV @RHWC,@#SAVERE ;SAVED RHWC
6220 027600 017737 152370 004514 MOV @RHBA,@#SAVERE+2;SAVED RHBA
6221 027606 017746 152364 MOV @RHCS2,-(SP) ;GET RHCS2
6222 027612 042716 177477 BIC #^C<IR!OR>,(SP) ;GET 'IR' & 'OR' STATES
6223 027616 042737 000300 004516 BIC #IR!OR,@#SAVERE+4;CLEAR 'IR' & 'OR' BITS
6224 027624 052637 004516 BIS (SP)+,@#SAVERE+4;SET 'OR' & 'IR' AS REQUIRED
6225
6226
6227 ;*COMPARE REGISTERS BEFORE WRITE WITH RESULTS AFTER WRITE
6228 027630 004037 034070 JSR RO,@#COMREG ;COMPARE SAVED REGISTERS WITH
6229 027634 004512 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
6230 027636 002254 WC ;TEST DATA STARTING FROM 'RHWC'
6231 027640 000021 17. ;17. REGISTERS TO BE COMPARED
6232 027642 027646 2$ ;RETURN TO 2$ ON ERROR
6233 027644 027652 3$ ;RETURN TO 3$ ON NO ERROR
6234
6235 027646 104071 2$: ERROR 71 ;FORCING OPI CAUSED
6236 027650 000207 RTS PC ;IMPROPER REGISTER CHANGE
6237 ;GOOD DATA GIVES WHAT SHOULD BE THERE
6238 ;RECEIVED DATA GIVES WHAT WAS THERE
6239 ;AFTER 3 INDEX PULSES WERE ISSUED
6240
6241
6242 027652 004737 033072 3$: JSR PC,@#CLDISK ;CLEAR THE 'GO' BIT
    
```



```

6243
6244 027656 000004          TST34: SCOPE
6245 027660 012706 001000  MOV      #STACK,SP          ;RESET STACK
6246 027664 012737 000034 004504  MOV      #34,@#TSTNM        ;SAVE TEST NUMBER
6247 027672 004737 033072      JSR      PC,@#CLDISK        ;SET R1-RHCS1, R2-RHCS2
6248 027676 012737 000025 027732  MOV      #21,@#1$+12       ;SET UP TO START FROM
6249 027704 012737 000025 027746  MOV      #21,@#2$+6         ;SECTOR 21.
6250 027712 012737 000056 004652  MOV      #46,@#TMP1         ;46 SECTORS TO COVER 3 TRACKS
6251
6252          ;*FILL WRITE FROM BUFFER WITH THE HEADER
6253 027720          1$:      JSR      R0,@#FLHEAD        ;SAVE HEADER DATA IN WRFROM
6254 027720 004037 032716  WRFROM          ;LOCATION WHERE SAVED
6255 027724 002370          4          ;NUMBER OF WORDS SAVED
6256 027726 000004          10000        ;FIRST DATA WORD
6257 027730 010000          21.          ;SECOND DATA WORD
6258 027732 000025          0          ;THIRD DATA WORD
6259 027734 000000          0          ;FOURTH DATA WORD
6260 027736 000000
6261
6262          ;*NOW THE WRITE HEADER AND DATA COMMAND WILL BE SETUP
6263 027740          2$:      JSR      R0,@#RUN          ;SETUP TO RUN FOR DATA COMMAND
6264 027740 004037 035054  0          ;CYLINDER 0
6265 027744 000000          .BYTE 21.      ;SECTOR 21.
6266 027746 025          .BYTE 0        ;TRACK 0
6267 027747 000          -0-4        ;WORD COUNT (DATA) = 0 +
6268 027750 177774          WRFROM        ;BUS ADDRESS
6269 027752 002370          0          ;DO NOT INHIBIT BUS ADDRESS INCREMENT
6270 027754 000000          FMT22       ;16 BITS PER WORD FORMAT
6271 027756 010000          WRIFOR      ;GET READY TO DO A WRIFOR
6272 027760 002344
6273
6274 027762 004737 033152      JSR      PC,@#CHECKT        ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
6275 027766 104401 057176      TYPE      ,CPHALT          ;CANNOT CONTINUE TESTING IF NOT
6276 027772 000000          HALT          ;STOP TEST
6277
6278 027774 013777 004506 152164  MOV      @#RP4VEC,@RPVEC    ;SET RP04 VECTOR ADDRESS
6279
6280 030002 013746 002344      MOV      @#WRIFOR,-(SP)     ;GET READY TO MOVE COMMAND
6281 030006 052716 000101      BIS      #GO!IE,(SP)        ;GET READY TO SET GO AND
6282 030012 012677 152162      MOV      (SP)+,@RHCS1       ;GO WITH
6283 030016 011100          MOV      @R1,R0             ;SAVE RHCS1 DURING ABOVE OPERATION
6284 030020 011305          MOV      @R3,R5             ;SAVE RHDS1 DURING ABOVE OPERATION
6285
6286          ;*ONE REVOLUTION=16670 MICRO SEC, ONE SECTOR=760 MICRO SEC
6287 030022 104413          WAT          ;WAIT FOR RDY BIT TO SET
6288 030024 002200          RHCS1        ;WAIT FOR RHCS1 REGISTER
6289 030026 000200          RDY          ;WAIT FOR RDY BIT IN RHCS1 REGISTER
6290 030030 003237          1695.        ;ALLOW 16950 MICRO SECONDS
6291 030032 001515          845.        ;RDY MUST SET BETWEEN
6292
6293          ;*NOW ONE MORE SECTOR HAS BEEN WRITTEN
6294          ;*'SC' WILL BE CHECKED TO MAKE SURE
6295          ;*NO ERRORS OCCURED
6296
6297 030034 017737 152140 002262  MOV      @RHCS1,@#CS1       ;GET RHCS1
6298 030042 032737 100000 002262  BIT      #SC,@#CS1          ;IS 'SC' SET ?
    
```



```
6299 030050 001403          BEQ      3$          ;BRANCH IF "SPECIAL CONDITION" NOT SET
6300 030052 004737 035014   JSR      PC,@#PUTREG ;READ & SAVE ALL RH11 & RP04 REGISTERS
6301 030056 104072          ERROR    72          ;THERE WAS AN UNDEFINED ERROR AFTER
6302                                     ;A WRITE HEADER AND DATA
6303
6304                                     ;*A SECTOR HAS BEEN FORMATTED NOW,
6305                                     ;*THE HARDWARE WILL BE CLEARED AND
6306                                     ;*CHANGES WILL BE MADE TO FORMAT NEXT SECTOR.
6307
6308 030060          3$:
6309 030060 004737 033072   JSR      PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
6310 030064 013705 027732   MOV      @#1$+12,R5  ;GET SECTOR TRACK WORD
6311 030070 005205          INC      R5           ;+ 1
6312 030072 122705 000026   CMPB    #22.,R5      ;IS IT 22 SECTORS (WHOLE TRACK DONE) ?
6313 030076 001405          BEQ      4$          ;YES...DO NEXT TRACK
6314 030100 010537 027732   MOV      R5,@#1$+12 ;NO...RESTORE SECTOR TRACK FOR DATA
6315 030104 010537 027746   MOV      R5,@#2$+6   ;RESTORE SECTOR TRACK FOR "RUN" ROUTINE
6316 030110 000410          BR       5$          ;CHECK FOR 46 SECTORS COMPLETED
6317
6318 030112 105037 027732   4$:  CLRB   @#1$+12    ;SET SECTOR = 0 FOR DATA WRITTEN
6319 030116 105237 027733   INCB   @#1$+13    ;INCR TRACK FOR DATA WRITTEN
6320 030122 105037 027746   CLRB   @#2$+6     ;SET SECTOR = 0 FOR "RUN" ROUTINE
6321 030126 105237 027747   INCB   @#2$+7     ;INCR TRACK FOR THE "RUN" ROUTINE
6322
6323 030132 005337 004652   5$:  DEC     @#TMP1     ;ARE 46 SECTORS DONE ?
6324 030136 001270          BNE     1$          ;CONTINUE FORMATTING IF NOT
6325
6326 030140          6$:          ;GO ON TO NEXT TEST IF SO
```



```

6327
6328
6329 030140 000004          TST35: SCOPE
6330 030142 012706 001000  MOV      #STACK,SP      ;RESET STACK
6331 030146 012737 000035 004504  MOV      #35,@#TSTNM    ;SAVE TEST NUMBER
6332 030154 004737 033072          JSR      PC,@#CLDISK    ;SET R1-RHCS1, R2-RHCS2
6333 030160 012737 000025 030214  MOV      #21.,@#1$+12  ;SET UP TO START FROM
6334 030166 012737 000025 030230  MOV      #21.,@#2$+6   ;SECTOR 21.
6335 030174 012737 000056 004652  MOV      #46.,@#TMP1   ;46 SECTORS TO COVER 3 TRACKS
6336
6337          ;*FILL WRITE FROM BUFFER WITH HEADER
6338 030202          1$:      JSR      R0,@#FLHEAD    ;SAVE HEADER DATA IN WRFROM
6339 030202 004037 032716  WRFROM          ;LOCATION WHERE SAVED
6340 030206 002370          4          ;NUMBER OF WORDS SAVED
6341 030210 000004          10000         ;FIRST DATA WORD
6342 030212 010000          21.           ;SECOND DATA WORD
6343 030214 000025          0              ;THIRD DATA WORD
6344 030216 000000          0              ;FOURTH DATA WORD
6345 030220 000000
6346
6347          ;*NOW THE WRITE HEADER AND DATA COMMAND WILL BE FILLED
6348 030222          2$:      JSR      R0,@#RUN      ;SETUP TO RUN FOR DATA COMMAND
6349 030222 004037 035054  0              ;CYLINDER 0
6350 030226 000000          .BYTE 21.       ;SECTOR 21.
6351 030230 025           .BYTE 0           ;TRACK 0
6352 030231 000           -0-4         ;WORD COUNT (DATA) = 0 +
6353 030232 177774          WRFROM        ;BUS ADDRESS
6354 030234 002370          0              ;DO NOT INHIBIT BUS ADDRESS INCREMENT
6355 030236 000000          FMT22        ;16 BITS PER WORD FORMAT
6356 030240 010000          WRIFOR       ;GET READY TO DO A WRIFOR
6357 030242 002344
6358
6359 030244 004737 033152          JSR      PC,@#CHECKT   ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
6360 030250 104401 057176          TYPE      ,CPHALT    ;CANNOT CONTINUE TESTING IF NOT
6361 030254 000000          HALT          ;STOP TEST
6362
6363 030256 013777 004506 151702  MOV      @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
6364
6365 030264 013746 002344          MOV      @#WRIFOR,-(SP) ;GET READY TO MOVE COMMAND
6366 030270 052716 000101          BIS      #GO!IE,(SP)   ;GET READY TO SET GO AND
6367 030274 012677 151700          MOV      (SP)+,@RHCS1  ;GO WITH
6368 030300 011100          MOV      @R1,R0        ;SAVE RHCS1 DURING ABOVE OPERATION
6369 030302 011305          MOV      @R3,R5        ;SAVE RHDS1 DURING ABOVE OPERATION
6370
6371          ;*ONE REVOLUTION=16670 MICRO SEC, ONE SECTOR=760 MICRO SEC
6372 030304 104413          WAT          ;WAIT FOR RDY BIT TO SET
6373 030306 002200          RHCS1       ;WAIT FOR RHCS1 REGISTER
6374 030310 000200          RDY         ;WAIT FOR RDY BIT IN RHCS1 REGISTER
6375 030312 003237          1695.      ;ALLOW 16950 MICRO SECONDS
6376 030314 001515          845.      ;RDY MUST SET BETWEEN
6377
6378          ;*NOW ONE MORE SECTOR HAS BEEN WRITTEN
6379          ;*'SC' WILL BE CHECKED TO MAKE SURE
6380          ;*NO ERRORS OCCURED
6381
6382 030316 017737 151656 002262  MOV      @RHCS1,@#CS1  ;GET RHCS1
  
```



```

6383 030324 032737 100000 002262 BIT #SC,@#CS1 ;IS 'SC' SET ?
6384 030332 001405 BEQ 3$ ;BRANCH IF 'SPECIAL CONDITION' NOT SET
6385 030334 004737 035014 JSR PC,@#PUTREG ;READ & SAVE ALL RH11 & RP04 REGISTERS
6386 030340 104072 ERROR 72 ;THERE WAS AN UNDEFINED ERROR AFTER
6387 ;A WRITE HEADER AND DATA
6388
6389 030342 000137 030576 JMP TST36 ; THIS IS A SETUP ERROR AND 'OPI' TEST CAN'T CONTI
6390 ;GO ON TO NEXT TEST
6391
6392 ;*ONE SECTOR HAS BEEN FORMATTED NOW,
6393 ;*THE HARDWARE WILL BE CLEARED AND
6394 ;*CHANGES WILL BE MADE TO FORMAT NEXT SECTOR.
6395
6396 030346 3$: JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
6397 030346 004737 033072 MOV @#1$+12,R5 ;GET SECTOR TRACK WORD
6398 030352 013705 030214 INC R5 ;+ 1
6399 030356 005205 CMPB #22.,R5 ;IS IT 22 (WHOLE TRACK) ?
6400 030360 122705 000026 BEQ 4$ ;YES...DO NEXT TRACK
6401 030364 001405 MOV R5,@#1$+12 ;NO...RESTORE SECTOR TRACK FOR DATA WRITTEN
6402 030366 010537 030214 MOV R5,@#2$+6 ;RESTORE SECTOR TRACK FOR 'RUN' ROUTINE
6403 030372 010537 030230 BR 5$ ;CHECK FOR 46 SECTORS COMPLETED
6404 030376 000410
6405
6406 030400 105037 030214 4$: CLRB @#1$+12 ;SET SECTOR = 0 FOR DATA WRITTEN
6407 030404 105237 030215 INCB @#1$+13 ;INCR TRACK FOR THE 'RUN' ROUTINE
6408 030410 105037 030230 CLRB @#2$+6 ;SET SECTOR = 0 FOR DATA WRITTEN
6409 030414 105237 030231 INCB @#2$+7 ;INCR TRACK FOR THE 'RUN' ROUTINE
6410
6411 030420 005337 004652 5$: DEC @#TMP1 ;ARE 46 SECTORS DONE ?
6412 030424 001266 BNE 1$ ;CONTINUE IF NOT
6413
6414 ;*NOW 46 SECTORS HAVE BEEN FORMATTED
6415
6416 ;*READ HEADER AND DATA FOR 46 SECTORS=11960 WORDS
6417 ;*WITH BUS ADDRESS INHIBITED
6418
6419 JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
6420 030426 004737 033072
6421
6422 ;*FILL READ HEADER AND DATA COMMAND
6423
6424 030432 004037 035054 JSR RO,@#RUN ;SETUP TO RUN FOR DATA COMMAND
6425 030436 000000 0 ;CYLINDER 0
6426 030440 025 ;SECTOR 21.
6427 030441 000 ;TRACK 0
6428 030442 150510 -11956.-4 ;WORD COUNT (DATA) = 11956. +
6429 030444 003434 REINTO ;BUS ADDRESS
6430 030446 000010 BAI ;INHIBIT BUS ADDRESS INCREMENT
6431 030450 014000 FMT22!ECI ;16 BITS PER WORD FORMAT
6432 030452 002350 REFOR ;GET READY TO DO A REFOR
6433 030454 004737 033152 JSR PC,@#CHECKT ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
6434 030460 104401 057176 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF NOT
6435 030464 000000 HALT ;STOP TEST
6436 030466 013777 004506 151472 MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
6437 030474 013746 002350 MOV @#REFOR,-(SP) ;GET READY TO MOVE COMMAND
6438 030500 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET GO AND
    
```



```

6439 030504 012677 151470      MOV      (SP)+,@RHCS1      ;GO WITH
6440 030510 011100      MOV      @R1,R0           ;SAVE RHCS1 DURING ABOVE OPERATION
6441 030512 011305      MOV      @R3,R5           ;SAVE RHDS1 DURING ABOVE OPERATION
6442
6443      ;*TIME IS NOT IMPORTANT
6444
6445 030514 104413      WAT                    ;WAIT FOR RDY BIT TO SET
6446 030516 002200      RHCS1                 ;WAIT FOR RHCS1 REGISTER
6447 030520 000200      RDY                    ;WAIT FOR RDY BIT IN RHCS1 REGISTER
6448 030522 121320      41680.                ;ALLOW 416800 MICRO SECONDS
6449 030524 121320      41680.                ;RDY MUST SET BETWEEN
6450
6451      ;*NOW THAT ALL 11960 WORDS HAVE BEEN READ
6452      ;*'OPI' WILL BE CHECKED TO BE NOT SET
6453
6454 030526 017737 151450 002264      MOV      @RHER1,@#ER1     ;GET RHER1
6455 030534 032737 020000 002264      BIT      #OPI,@#ER1      ;IS 'OPI' SET ?
6456 030542 001403      BEQ      6$              ;CHECK 'SC' IF NOT
6457 030544 004737 035014      JSR      PC,@#PUTREG     ;READ & SAVE ALL RH11 & RP04 REGISTERS
6458 030550 104074      ERROR    74             ;READ HEADER AND DATA
6459                                     ;OVER 3 INDEX PULSES
6460                                     ;CAUSED 'OPI' TO SET
6461
6462      ;*'SC' WILL BE CHECKED
6463
6464 030552 017737 151422 002262 6$:      MOV      @RHCS1,@#CS1    ;GET RHCS1
6465 030560 032737 100000 002262      BIT      #SC,@#CS1      ;IS 'SC' SET ?
6466 030566 001403      BEQ      7$              ;CONTINUE TESTING IF NOT
6467 030570 004737 035014      JSR      PC,@#PUTREG     ;READ & SAVE ALL RH11 & RP04 REGISTERS
6468 030574 104072      ERROR    72             ;READ HEADER AND DATA
6469                                     ;FOR 11960 WORDS, THAT IS OVER THREE
6470                                     ;INDEX PULSES, CAUSED AN UNDEFINED ERROR
6471
6472 030576      7$:                    ;CONTINUE WITH THE NEXT TEST
    
```



```

6473
6474
6475 030576 000004          TST36: SCOPE
6476 030600 012706 001000  MOV      #STACK,SP      ;RESET STACK
6477 030604 012737 000036 004504  MOV      #36,@#TSTNM    ;SAVE TEST NUMBER
6478 030612 004737 033072  JSR      PC,@#CLDISK    ;SET R1-RHCS1, R2-RHCS2
6479
6480          ;*THE FOLLOWING CLEARS ARE TO INITIALIZE TEST FROM CYLINDER 0
6481 030616 005037 030724  CLR      @#1$+12        ;START WITH SECTOR/TRACK = 0
6482 030622 005037 030742  CLR      @#2$+10        ;START WITH DATA = 0
6483 030626 005037 030752  CLR      @#3$+6         ;START WITH 0 FOR COMMAND
6484
6485 030632 012737 000023 001200  MOV      #19.,@#$TMP1    ;19 TRACKS TO BE WRITTEN
6486
6487          ;*THIS GETS THE HEADS TO CYLINDER 0
6488
6489 030640 004737 033152  JSR      PC,@#CHECKT    ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
6490 030644 104401 057176  TYPE     ,CPHALT        ;CANNOT CONTINUE TESTING IF NOT
6491 030650 000000          HALT                    ;STOP TEST
6492
6493 030652 013777 004506 151306  MOV      @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
6494
6495 030660 013746 002326  MOV      @#RECALI,-(SP) ;GET READY TO MOVE COMMAND
6496 030664 052716 000101  BIS      #GO!IE,(SP)    ;GET READY TO SET GO AND
6497 030670 012677 151304  MOV      (SP)+,@RHCS1   ;GO WITH
6498
6499 030674 104413          WAT                    ;WAIT FOR DRY BIT TO SET
6500 030676 002222          RHDS1                 ;WAIT FOR RHDS1 REGISTER
6501 030700 000200          DRY                    ;WAIT FOR DRY BIT IN RHDS1 REGISTER
6502 030702 060650          25000.                ;ALLOW 250000 MICRO SECONDS
6503 030704 060650          25000.                ;DRY MUST SET BETWEEN
6504
6505 030706 004737 033072  JSR      PC,@#CLDISK    ;SET R1-RHCS1, R2-RHCS2
6506
6507
6508          ;*FILL WRITE FROM BUFFER WITH HEADER
6509 030712          1$:
6510 030712 004037 032716  JSR      RO,@#FLHEAD    ;SAVE HEADER DATA IN WRFROM
6511 030716 002370          WRFROM                 ;LOCATION WHERE SAVED
6512 030720 000004          4                       ;NUMBER OF WORDS SAVED
6513 030722 010000          10000                  ;FIRST DATA WORD
6514 030724 000000          <0*400>!0              ;SECOND DATA WORD
6515 030726 000000          0                       ;THIRD DATA WORD
6516 030730 000000          0                       ;FOURTH DATA WORD
6517
6518          ;*FILL WRITE FROM BUFFER WITH DATA
6519 030732          2$:
6520 030732 004037 032742  JSR      RO,@#CLAREA    ;CLEAR 256. WORDS, FROM WRFROM+10
6521 030736 002400          WRFROM+10              ;STARTING FROM WRFROM+10
6522 030740 000400          256.                   ;256. WORDS
6523 030742 000000          <0.*2000>!<0.*40>!0    ;FILL WITH <0.*2000>!<0.*40>!0
6524
6525          ;*THE WRITE HEADER AND DATA COMMAND WILL BE FILLED
6526 030744          3$:
6527 030744 004037 035054  JSR      RO,@#RUN       ;SETUP TO RUN FOR DATA COMMAND
6528 030750 000000          0                       ;CYLINDER 0
  
```



```

6529 030752 000 .BYTE 0 ;SECTOR 0
6530 030753 000 .BYTE 0 ;TRACK 0
6531 030754 177374 -256.-4 ;WORD COUNT (DATA) = 256. +
6532 030756 002370 WRFROM ;BUS ADDRESS
6533 030760 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
6534 030762 010000 FMT22 ;16 BITS PER WORD FORMAT
6535 030764 002344 WRIFOR ;GET READY TO DO A WRIFOR
6536
6537 030766 004737 033152 JSR PC,@#CHECKT ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
6538 030772 104401 057176 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF NOT
6539 030776 000000 HALT ;STOP TEST
6540
6541 031000 013777 004506 151160 MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
6542
6543 031006 013746 002344 MOV @#WRIFOR,-(SP) ;GET READY TO MOVE COMMAND
6544 031012 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET GO AND
6545 031016 012677 151156 MOV (SP)+,@RHCS1 ;GO WITH
6546
6547 ;*ONE REVOLUTION = 16670 MICRO SEC., ONE SECTOR = 760
6548 ;*MICRO SEC. MAX TIME ALLOWED = ONE REVOLUTION + HEAD
6549 ;*SWITCH + 2 SECTORS, MIN TIME ALLOWED = SECTOR (FIRST CASE)
6550 ;*IF THERE IS A FAILURE HERE HALT PROGRAM AFTER ERROR WITH
6551 ;*SWITCH 15 AND SEE CURRENT CYLINDER REGISTER TO DETERMINE
6552 ;*WHAT CYLINDER IS FAILING
6553
6554 031022 104413 WAT ;WAIT FOR RDY BIT TO SET
6555 031024 002200 RHCS1 ;WAIT FOR RHCS1 REGISTER
6556 031026 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
6557 031030 003162 1650. ;ALLOW 16500 MICRO SECONDS
6558 031032 001572 890. ;RDY MUST SET BETWEEN
6559
6560 ;*NOW SECTOR 0 OF ONE TRACK HAS BEEN WRITTEN CHECK COMPOSIT
6561 ;*ERROR BIT TO BE SURE NO ERRORS HAPPENED
6562
6563 ;*SAVE REGISTERS IN SAVE TABLE
6564 031034 004737 035014 JSR PC,@#PUTREG
6565
6566 031040 032737 040000 002304 BIT #ERR,@#DS1 ;ANY DISK ERRORS
6567 031046 001004 BNE 9$ ;BRANCH IF YES
6568 031050 032737 040000 002262 BIT #TRE,@#CS1 ;ANY RH ERRORS
6569 031056 001401 BEQ 4$ ;BRANCH IF NO
6570
6571 031060 104066 9$: ERROR 66 ;SOME ERRORS OCCURRED
6572 ;WHILE DOING WRITE HEADER
6573 ;AND DATA
6574
6575 ;*THE FOLLOWING 3 ADDS SETS UP FOR NEXT TRACK WRITING
6576
6577 031062 062737 000400 030724 4$: ADD #400,@#1$+12 ;NEXT TRACK FOR HEADER
6578 031070 062737 000040 030742 ADD #40,@#2$+10 ;NEXT TRACK FOR DATA
6579 031076 062737 000400 030752 ADD #400,@#3$+6 ;NEXT TRACK FOR COMMAND
6580
6581 031104 005337 001200 DEC @#$TMP1 ;COUNT 19 TRACKS
6582 031110 001300 BNE 1$
6583
6584 ;*THE FOLLOWING CLEARS SETS UP FOR READ HEADER AND DATA

```



```

6585 031112 005037 031172 CLR @#SST3+12 ;START WITH SECTOR/TRACK = 0
6586 031116 005037 031210 CLR @#SST4 10 ;START WITH DATA = 0
6587 031122 005037 031220 CLR @#SST5+6 ;START WITH 0 FOR COMMAND
6588
6589 031126 004737 033072 JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
6590
6591 ;*SET UP FOR READ HEADER AND DATA
6592 031132 012737 000023 001200 SST1: MOV #19.,@#$TMP1 ;19 TRACKS TO BE READ
6593
6594 ;*FILL READ INTO BUFFER WITH ALL ONES
6595 031140 SST2:
6596 031140 004037 032742 JSR RO,@#CLAREA ;CLEAR 260. WORDS, FROM REINTO
6597 031144 003434 REINTO ;STARTING FROM REINTO
6598 031146 000404 260. ;260. WORDS
6599 031150 177777 -1 ;FILL WITH -1
6600 031152 013737 031140 001110 MOV @#SST2,@#$LPERR ;SET LOOP POINT
6601
6602 ;*FILL WRITE FROM BUFFER WITH EXPECTED HEADER
6603 031160 SST3:
6604 031160 004037 032716 JSR RO,@#FLHEAD ;SAVE HEADER DATA IN WRFROM
6605 031164 002370 WRFROM ;LOCATION WHERE SAVED
6606 031166 000004 4 ;NUMBER OF WORDS SAVED
6607 031170 010000 10000 ;FIRST DATA WORD
6608 031172 000000 0 ;SECOND DATA WORD
6609 031174 000000 0 ;THIRD DATA WORD
6610 031176 000000 0 ;FOURTH DATA WORD
6611
6612 ;*FILL WRITE FROM BUFFER WITH EXPECTED DATA
6613 031200 SST4:
6614 031200 004037 032742 JSR RO,@#CLAREA ;CLEAR 256. WORDS, FROM WRFROM+<4*2>
6615 031204 002400 WRFROM+<4*2> ;STARTING FROM WRFROM+<4*2>
6616 031206 000400 256. ;256. WORDS
6617 031210 000000 <0.*2000>!<0*40>!0 ;FILL WITH <0.*2000>!<0*40>!0
6618
6619 ;*FILL COMMAND FOR READ HEADER AND DATA
6620 031212 SST5:
6621 031212 004037 035054 JSR RO,@#RUN ;SETUP TO RUN FOR DATA COMMAND
6622 031216 000000 0 ;CYLINDER 0
6623 031220 000 .BYTE 0 ;SECTOR 0
6624 031221 000 .BYTE 0 ;TRACK 0
6625 031222 177374 -256.-4 ;WORD COUNT (DATA) = 256. +
6626 031224 003434 REINTO ;BUS ADDRESS
6627 031226 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
6628 031230 014000 ECI!FMT22 ;16 BITS PER WORD FORMAT
6629 031232 002350 REFOR ;GET READY TO DO A REFOR
6630
6631 031234 004737 033152 JSR PC,@#CHECKT ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
6632 031240 104401 057176 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF NOT
6633 031244 000000 HALT ;STOP TEST
6634
6635 031246 013777 004506 150712 MOV @#RP4VEC,@#RPVEC ;SET RP04 VECTOR ADDRESS
6636
6637 031254 013746 002350 MOV @#REFOR,-(SP) ;GET READY TO MOVE COMMAND
6638 031260 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET GO AND
6639 031264 012677 150710 MOV (SP)+,@#RHCS1 ;GO WITH
6640
    
```



```

6641 031270 104413          WAT          ;WAIT FOR RDY BIT TO SET
6642 031272 002200          RHCS1        ;WAIT FOR RHCS1 REGISTER
6643 031274 000200          RDY          ;WAIT FOR RDY BIT IN RHCS1 REGISTER
6644 031276 003162          1650.       ;ALLOW 16500 MICRO SECONDS
6645 031300 001572          890.        ;RDY MUST SET BETWEEN
6646
6647                          ;*NOW SECTOR 0 OF ONE TRACK HAS BEEN READ CHECK COMPOSIT
6648                          ;*ERROR BIT TO BE SURE NO ERROR HAPPENED
6649
6650                          ;*SAVE REGISTERS IN SAVE TABLE
6651 031302 004737 035014    JSR          PC,@#PUTREG
6652
6653 031306 032737 040000 002304    BIT          #ERR,@#DS1 ;ANY DISK ERRORS
6654 031314 001004          BNE          10$       ;BRANCH IF YES
6655 031316 032737 040000 002262    BIT          #TRE,@#CS1 ;ANY RH ERRORS
6656 031324 001401          BEQ          11$       ;BRANCH IF NO
6657
6658 031326 104066          10$:        ERROR    66          ;SOME ERRORS OCCURRED
6659                          ;WHILE DOING READ
6660                          ;HEADER AND DATA
6661
6662                          ;*NOW THE READ DATA WILL BE COMPARED DATA IN EACH SECTOR
6663                          ;*IS UNIQUE IF PROGRAM IS HALTED ON ERROR THEN LOOK AT
6664                          ;*RHDST TO GET WHAT TRACK IS IN ERROR. LOOKING AT THE DATA
6665                          ;*BITS NO 4,5,6,7,8 IN GOOD DATA ALSO GIVES TRACK NUMBER
6666                          ;*IN GOOD DATA ALSO GIVES TRACK NUMBER
6667
6668 031330          11$:
6669 031330 004037 035120    JSR          RO,@#COMPAR ;COMPARE TWO BLOCKS OF MEMORY
6670 031334 002370          WRFROM        ;GOOD DATA STARTS FROM WRFROM
6671 031336 003434          REINTO        ;TEST DATA STARTS FROM REINTO
6672 031340 000404          260.         ;260., WORDS TO BE COMPARED
6673 031342 031346          12$         ;RETURN TO 12$ ON ERROR
6674 031344 031352          13$         ;RETURN TO 13$ ON NO ERROR
6675
6676                          ;BITS 4,5,6,7,8
6677 031346 104067          12$:        ERROR    67          ;READ HEADER AND DATA
6678 031350 000207          RTS          PC        ;ERROR
6679                          ;HEAD SELECTION ERROR
6680                          ;DATA READ GIVES NATURE
6681                          ;OF ERROR
6682                          ;EXCEPT FOR THE
6683                          ;FOUR HEADER WORDS
6684                          ;THE BITS 4,5,6,7,8
6685                          ;GIVE THE TRACK NUMBER
6686
6687                          ;*NOW INCREMENT TO READ NEXT TRACK
6688
6689 031352 062737 000400 031172 13$:    ADD          #400,@#SST3+12 ;NEXT TRACK FOR HEADER
6690 031360 062737 000040 031210          ADD          #40,@#SST4+10 ;NEXT TRACK FOR DATA
6691 031366 062737 000400 031220          ADD          #400,@#SST5+6 ;NEXT TRACK FOR COMMAND
6692
6693 031374 005337 001200          DEC          @#$TMP1    ;COUNT 19 TRACKS
6694 031400 001001          BNE          5$
6695 031402 000402          BR          TST37      ;TO NEXT TEST
6696 031404 000137 031140          5$:        JMP          @#$SST2    ;JUMP BACK
    
```



```

6697
6698 031410 000004
6699 031412 012706 001000
6700 031416 012737 000037 004504
6701 031424 004737 033072
6702 031430 004737 033152
6703 031434 104401 057176
6704 031440 000000
6705
6706
6707
6708
6709 031442 012737 010000 031554
6710 031450 005037 031574
6711 031454 005037 031602
6712
6713
6714 031460 013777 004506 150500
6715
6716 031466 013746 002326
6717 031472 052716 000101
6718 031476 012677 150476
6719
6720 031502 104413
6721 031504 002222
6722 031506 000200
6723 031510 060650
6724 031512 060650
6725
6726
6727
6728
6729 031514 005737 004636
6730 031520 001404
6731
6732
6733 031522 012737 001001 001200
6734 031530 000403
6735
6736 031532 012737 000401 001200 14$:
6737 031540 15$:
6738
6739 031540 004737 033072
6740
6741
6742 031544
6743 031544 004037 032716
6744 031550 002370
6745 031552 000004
6746 031554 010000
6747 031556 000000
6748 031560 000000
6749 031562 000000
6750
6751
6752 031564
    
```

TST37: SCOPE

```

MOV #STACK,SP ;RESET STACK
MOV #37,@#TSTNM ;SAVE TEST NUMBER
JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
JSR PC,@#CHECKT ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF NOT
HALT ;STOP TEST

;*SET UP TO INITIALIZE TEST FROM CYLINDER 0, TRACK 0,
;*SECTOR 0
MOV #10000,@#1$+10 ;CYLINDER HEADER DATA
CLR @#2$+10 ;DATA
CLR @#3$+4 ;CYLINDER COMMAND RHCA

;*THIS IS TO GET THE HEADS TO CYLINDER ZERO
MOV @#RP4VEC,@#RPVEC ;SET RP04 VECTOR ADDRESS

MOV @#RECALI,-(SP) ;GET READY TO MOVE COMMAND
BIS #GO!IE,(SP) ;GET READY TO SET GO AND
MOV (SP)+,@#RHCS1 ;GO WITH

WAT ;WAIT FOR DRY BIT TO SET
RHDS1 ;WAIT FOR RHDS1 REGISTER
DRY ;WAIT FOR DRY BIT IN RHDS1 REGISTER
25000. ;ALLOW 250000 MICRO SECONDS
25000. ;DRY MUST SET BETWEEN

;*THE DRIVE TYPE IS CHECKED AND THE APPROPRIATE MAX.
;*CYLINDER DIFFERENCE IS SET UP
TST @#RP06 ;TEST FOR RP06 DRIVE
BEQ 14$ ;TREAT UNIT AS AN RP04
;TREAT AS AN RP06

MOV #513.,@#TMP1 ;513 CYLINDERS
BR 15$ ;CONTINUE

MOV #257.,@#TMP1 ;257 CYLINDERS
;CONTINUE WITH TEST

JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2

;*FILL WRITE FROM BUFFER WITH HEADER
1$:
JSR RO,@#FLHEAD ;SAVE HEADER DATA IN WRFROM
WRFROM ;LOCATION WHERE SAVED
4 ;NUMBER OF WORDS SAVED
10000 ;FIRST DATA WORD
0 ;SECOND DATA WORD
0 ;THIRD DATA WORD
0 ;FOURTH DATA WORD

;*FILL WRITE FROM BUFFER WITH DATA
2$:
    
```



```

6753 031564 004037 032742 JSR R0,@#CLAREA ;CLEAR 256. WORDS, FROM WRFROM+10
6754 031570 002400 WRFROM+10 ;STARTING FROM WRFROM+10
6755 031572 000400 256. ;256. WORDS
6756 031574 000000 0 ;FILL WITH 0
6757
6758 ;*THE WRITE HEADER AND DATA COMMAND WILL BE LOADED
6759 031576 3$: JSR R0,@#RUN ;SETUP TO RUN FOR DATA COMMAND
6760 031576 004037 035054 0 ;CYLINDER 0
6761 031602 000000 .BYTE 0 ;SECTOR 0
6762 031604 000 .BYTE 0 ;TRACK 0
6763 031605 000 -256.-4 ;WORD COUNT (DATA) = 256. +
6764 031606 177374 WRFROM ;BUS ADDRESS
6765 031610 002370 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
6766 031612 000000 FMT22 ;16 BITS PER WORD FORMAT
6767 031614 010000 WRIFOR ;GET READY TO DO A WRIFOR
6768 031616 002344
6769
6770 031620 004737 033152 JSR PC,@#CHECKT ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
6771 031624 104401 057176 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF NOT
6772 031630 000000 HALT ;STOP TEST
6773
6774 031632 013777 004506 150326 MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
6775
6776 031640 013746 002344 MOV @#WRIFOR,-(SP) ;GET READY TO MOVE COMMAND
6777 031644 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET GO AND
6778 031650 012677 150324 MOV (SP)+,@RHCS1 ;GO WITH
6779
6780 ;*ONE REVOLUTION = 16670 MICRO SECONDS, ONE SECTOR = 760
6781 ;*MICRO SECONDS, ONE SEEK = 7000 MICRO SECONDS.
6782 ;*MAX TIME = 1 REVOLUTION + 1 SEEK + 2 SECTORS
6783 ;*MIN TIME = 1 SECTOR
6784
6785 031654 104413 WAT ;WAIT FOR RDY BIT TO SET
6786 031656 002200 RHCS1 ;WAIT FOR RHCS1 REGISTER
6787 031660 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
6788 031662 002354 1260. ;ALLOW 12600 MICRO SECONDS
6789 031664 002354 1260. ;RDY MUST SET BETWEEN
6790
6791 ;*NOW ONE SECTOR WRITE IS COMPLETE. CHANGES WILL BE MADE
6792 ;*FOR THE NEXT SECTOR, THEN THE ABOVE WILL BE REPEATED
6793 ;*UNTIL CYLINDER 256./512. IS REACHED
6794 031666 005237 031554 INC @#1$+10 ;CYLINDER HEADER DATA
6795 031672 005237 031574 INC @#2$+10 ;DATA
6796 031676 005237 031602 INC @#3$+4 ;CYLINDER COMMAND (RHCA)
6797 031702 005337 001200 DEC @#$TMP1 ;COUNT DOWN FOR 256./512. CYLINDERS
6798 031706 001316 BNE 1$ ;DO NEXT WRITE IF 256./512. NOT DONE
6799
6800 ;*NOW ALL 256./512. CYLINDERS HAVE CYLINDER NUMBER WRITTEN
6801 ;*AS DATA ON SECTOR 0, TRACK 0. NOW A RECALIBRATE, FOLLOWED
6802 ;*BY READ HEADER AND DATA, THEN A CHECK WILL BE DONE ON
6803 ;*CYLINDER 0,1,2,4,8,16,32,64,128,256,512, AND 0
6804
6805 031710 013737 031740 001110 MOV @#4$,@#$LPERR ;LOOP ON ERROR
6806 031716 005037 001200 CLR @#$TMP1 ;CYLINDER COUNTER
    
```



```

6807
6808
6809
6810
6811 031722 012737 010000 032022
6812 031730 005037 032042
6813 031734 005037 032050
6814 031740
6815 031740 004737 033072
6816
6817 031744 013777 004506 150214
6818
6819 031752 013746 002326
6820 031756 052716 000101
6821 031762 012677 150212
6822
6823 031766 104413
6824 031770 002222
6825 031772 000200
6826 031774 060650
6827 031776 060650
6828
6829
6830 032000 004037 032742
6831 032004 003434
6832 032006 000404
6833 032010 177777
6834
6835
6836 032012
6837 032012 004037 032716
6838 032016 002370
6839 032020 000004
6840 032022 010000
6841 032024 000000
6842 032026 000000
6843 032030 000000
6844 032032
6845 032032 004037 032742
6846 032036 002400
6847 032040 000400
6848 032042 000000
6849
6850
6851 032044
6852 032044 004037 035054
6853 032050 000000
6854 032052 000
6855 032054 000
6856 032054 177374
6857 032056 003434
6858 032060 000000
6859 032062 014000
6860 032064 002350
6861
6862 032066 004737 033152

```

```

;*INITIALIZE, RECALIBRATE, AND READ CYLINDERS
;*SETUP FOR CYLINDER 0
MOV #10000,@#5$+10 ;CYLINDER HEADER (DATA)
CLR @#6$+10 ;DATA
CLR @#7$+4 ;CYLINDER COMMAND (RHCA)
4$: JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
MOV @#RECALI,-(SP) ;GET READY TO MOVE COMMAND
BIS #GO!IE,(SP) ;GET READY TO SET GO AND
MOV (SP)+,@RHCS1 ;GO WITH
WAT ;WAIT FOR DRY BIT TO SET
RHDS1 ;WAIT FOR RHDS1 REGISTER
DRY ;WAIT FOR DRY BIT IN RHDS1 REGISTER
25000. ;ALLOW 250000 MICRO SECONDS
25000. ;DRY MUST SET BETWEEN
;*CLEAR READ INTO BUFFER WITH ALL ONES
JSR R0,@#CLAREA ;CLEAR 260. WORDS, FROM REINTO
REINTO ;STARTING FROM REINTO
260. ;260. WORDS
-1 ;FILL WITH -1
5$: ;*FILL WRITE FROM BUFFER WITH EXPECTED HEADER
JSR R0,@#FLHEAD ;SAVE HEADER DATA IN WRFROM
WRFROM ;LOCATION WHERE SAVED
4 ;NUMBER OF WORDS SAVED
10000 ;FIRST DATA WORD
0 ;SECOND DATA WORD
0 ;THIRD DATA WORD
0 ;FOURTH DATA WORD
6$: JSR R0,@#CLAREA ;CLEAR 256. WORDS, FROM WRFROM+10
WRFROM+10 ;STARTING FROM WRFROM+10
256. ;256. WORDS
0 ;FILL WITH 0
7$: ;*FILL READ HEADER AND DATA COMMAND
JSR R0,@#RUN ;SETUP TO RUN FOR DATA COMMAND
0 ;CYLINDER 0
.BYTE 0 ;SECTOR 0
.BYTE 0 ;TRACK 0
-256.-4 ;WORD COUNT (DATA) = 256. +
REINTO ;BUS ADDRESS
0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
ECI!FMT22 ;16 BITS PER WORD FORMAT
REFOR ;GET READY TO DO A REFOR
JSR PC,@#CHECKT ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1

```



```

6863 032072 104401 057176 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF NOT
6864 032076 000000 HALT ;STOP TEST
6865
6866 032100 013777 004506 150060 MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
6867
6868 ;*ONE SECTOR = 760 MICRO SECONDS, ONE REVOLUTION =
6869 ;*16670 MICRO SECONDS, MAX SEEK = 52000 MICRO SECONDS
6870 ;*MAX TIME = ONE REV + 1 SEEK + 1 SECTOR
6871 ;*MIN TIME = 1 SECTOR
6872
6873 032106 013746 002350 MOV @#REFOR,-(SP) ;GET READY TO MOVE COMMAND
6874 032112 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET GO AND
6875 032116 012677 150056 MOV (SP)+,@RHCS1 ;GO WITH
6876
6877 032122 104413 WAT ;WAIT FOR RDY BIT TO SET
6878 032124 002200 RHCS1 ;WAIT FOR RHCS1 REGISTER
6879 032126 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
6880 032130 006620 3472. ;ALLOW 34720 MICRO SECONDS
6881 032132 006620 3472. ;RDY MUST SET BETWEEN
6882
6883 ;*CHECK READ WORDS AS ALL READ COMMANDS HAVE BEEN CHECKED
6884
6885 ;*(DATA ERRORS MAY IMPLY "IMPLIED SEEK" ERRORS)
6886
6887
6888 032134 004037 035120 JSR RO,@#COMPAR ;COMPARE TWO BLOCKS OF MEMORY
6889 032140 002370 WRFROM ;GOOD DATA STARTS FROM WRFROM
6890 032142 003434 REINTO ;TEST DATA STARTS FROM REINTO
6891 032144 000404 260. ;260. WORDS TO BE COMPARED
6892 032146 032152 8$ ;RETURN TO 8$ ON ERROR
6893 032150 032156 9$ ;RETURN TO 9$ ON NO ERROR
6894
6895 032152 104070 8$: ERROR 70 ;READ HEADER AND DATA ERROR
6896 032154 000207 RTS PC ;DATA GIVES EXPECTED CYLINDER
6897
6898 ;*NOW ONE CYLINDER HAS BEEN CHECKED. CHANGES WILL BE MADE
6899 ;*TO READ THE NEXT CYLINDER AND THE ABOVE SECTOR READ WILL BE
6900 ;*REPEATED
6901
6902 032156 005737 001200 9$: TST @#$TMP1 ;IS IT ZERO ?
6903 032162 001003 BNE 10$ ;BRANCH IF NOT ZERO
6904 032164 005237 001200 INC @#$TMP1 ;ADD ONE IF = 0
6905 032170 000416 BR 11$ ;PUT ONE IN CYLINDER
6906
6907 032172 005737 004636 10$: TST @#RP06 ;TEST FOR RP06 DRIVE
6908 032176 001404 BEQ 16$ ;TREAT UNIT AS AN RP04
6909 ;TREAT AS AN RP06
6910
6911 032200 022737 001000 001200 CMP #512.,@#$TMP1 ;IS IT PASSED 512 CYLINDERS YET ?
6912 032206 000403 BR 17$ ;CONTINUE
6913 032210 022737 000400 001200 16$: CMP #256.,@#$TMP1 ;IS IT PASSED 256 CYLINDERS YET ?
6914 032216 17$: ;CONTINUE
6915
6916 032216 101421 BLOS 12$ ;YES, SO GO TO ZERO
6917 032220 063737 001200 001200 ADD @#$TMP1,@#$TMP1 ;DOUBLE THE CYLINDER
6918 032226 013737 001200 032042 11$: MOV @#$TMP1,@#6$+10 ;MAKE CYLINDER ADDRESS THE DATA
    
```



```

6919 032234 013746 001200      MOV    @#$TMP1,-(SP)    ;GET CYLINDER NUMBER
6920 032240 052716 010000      BIS    #FMT22,(SP)    ;INCLUDE FORMAT BIT
6921 032244 012637 032022      MOV    (SP)+,@#5$+10  ;HEADER DATA (CYLINDER)
6922 032250 013737 001200 032050  MOV    @#$TMP1,@#7$+4 ;CYLINDER COMMAND (RHCA)
6923 032256 000137 031740      JMP    @#4$           ;RETURN TO RECALIBRATE
6924
6925 032262 005737 004636      12$:   TST    @#RP06    ;TEST FOR RP06 DRIVE
6926 032266 001405              BEQ    18$           ;TREAT UNIT AS AN RP04
6927                               ;TREAT AS AN RP06
6928
6929 032270 022737 002000 001200  CMP    #1024.,@#$TMP1 ;512 DONE YET ?
6930 032276 001421              BEQ    13$           ;OUT ----->
6931 032300 000404              BR     19$           ;CONTINUE
6932 032302 022737 001000 001200 18$:   CMP    #512.,@#$TMP1 ;256 DONE YET ?
6933 032310 001414              BEQ    13$           ;OUT ----->
6934 032312              19$:   ;CONTINUE
6935
6936 032312 063737 001200 001200  ADD    @#$TMP1,@#$TMP1 ;DOUBLE THE CYLINDER
6937 032320 012737 010000 032022  MOV    #10000,@#5$+10 ;CYLINDER HEADER DATA
6938 032326 005037 032042      CLR    @#6$+10        ;DATA
6939 032332 005037 032050      CLR    @#7$+4        ;CYLINDER COMMAND (RHCA)
6940 032336 000137 031740      JMP    @#4$           ;RETURN TO THE RECALIBRATE
6941
6942 032342              13$:   ;END OF TEST
  
```



```

6943
6944 032342 000004          TST40: SCOPE
6945 032344 012737 000001 001212 MOV #1,$TIMES ;:DO 1 ITERATION
6946 032352 012737 000000 177776 MOV #0,PS ;:REINSTATE PS TO 0
6947 032360 104401 032366 TYPE ,65$ ;:TYPE ASCIZ STRING
6948 032364 000425 BR 64$ ;:GET OVER THE ASCIZ
6949 032440 013746 004616 MOV @#UNIT,-(SP) ;:GET READY TO TYPE UNIT NUMBER
6950 032444 104405 TYPDS
6951 032446 104401 032454 TYPE ,67$ ;:TYPE ASCIZ STRING
6952 032452 000402 BR 66$ ;:GET OVER THE ASCIZ
6953 032460 013746 001112 MOV @#$ERTTL,-(SP) ;:GET READY TO TYPE NUMBER OF ERRORS
6954 032464 104405 TYPDS
6955 032466 005037 001112 CLR @#$ERTTL ;:CLEAR TOTAL NUMBER OF ERRORS
6956 032472 005037 001102 CLR @#$TSTNM ;:CLEAR TEST NUMBER
6957 032476 005737 004626 TST @#$SELECT ;:STARTING FROM 200 ?
6958 032502 001413 BEQ 3$ ;:TEST NEXT DRIVE IF SO
6959 ;:CONTINUE TESTING THIS ONE IF NOT
6960
6961 032504 005237 001100 INC @#$PASS ;:INCREASE PASS COUNT
6962 032510 104401 032701 TYPE ,SENDMG ;:TYPE END PASS #
6963 032514 013746 001100 MOV @#$PASS,-(SP) ;:GET PASS NO.
6964 032520 104405 TYPDS ;:TYPE IT OUT
6965 032522 104401 032676 TYPE ,SENULL
6966 032526 000137 007720 JMP @#TST4 ;:JUMP TEST 4 ----->
6967
6968 032532 012737 177777 040746 3$: MOV #-1,@#PRITEM ;:CLEAR PREVIOUS ITEM NUMBER
6969 032540 005337 004620 DEC @#NOUNITS ;:NO. OF UNITS PRESENT DECREMENTED
6970 032544 001413 BEQ $EOP ;:BRANCH IF ALL DRIVES COMPLETE
6971 032546 013700 004616 MOV @#UNIT,R0 ;:UNIT UNDER TEST
6972 032552 012701 004576 MOV #UNITS,R1 ;:TABLE
6973 032556 022100 1$: CMP (R1)+,R0 ;:IS THIS UNIT JUST TESTED ?
6974 032560 001401 BEQ 2$ ;:CONTINUE IF YES
6975 032562 000775 BR 1$ ;:INCREMENT IF NO
6976 032564 011137 004616 2$: MOV (R1),@#UNIT ;:THIS IS NEXT UNIT
6977 032570 000137 007720 JMP @#TST4 ;:TEST THE NEXT DRIVE ----->
6978
    
```



```

6979
6980
6981
6982
6983
6984
6985
6986
6987 032574 000004          SCOPE
6988 032576 005037 001102  CLR      $TSTNM      ;;ZERO THE TEST NUMBER
6989 032602 005037 001212  CLR      $TIMES      ;;ZERO THE NUMBER OF ITERATIONS
6990 032606 005237 001100  INC      $PASS      ;;INCREMENT THE PASS NUMBER
6991 032612 042737 100000 001100  BIC      #100000,$PASS ;;DON'T ALLOW A NEG. NUMBER
6992 032620 005327          DEC      (PC)+      ;;LOOP?
6993 032622 000001          $EOPCT: .WORD      1
6994 032624 003022          BGT      $DOAGN      ;;YES
6995 032626 012737          MOV      (PC)+,@(PC)+ ;;RESTORE COUNTER
6996 032630 000001          $ENDCT: .WORD      1
6997 032632 032622          $EOPCT
6998 032634 104401 032701  TYPE     , $ENDMG      ;;TYPE 'END PASS #'
6999 032640 013746 001100  MOV      $PASS,-(SP)  ;;SAVE $PASS FOR TYPEOUT
7000 032644 104405          TYPDS    ;;GO TYPE--DECIMAL ASCII WITH SIGN
7001 032646 104401 032676  TYPE     , $ENULL      ;;TYPE A NULL CHARACTER
7002 032652 013700 000042  $GET42: MOV      @#42,R0 ;;GET MONITOR ADDRESS
7003 032656 001405          BEQ      $DOAGN      ;;BRANCH IF NO MONITOR
7004 032660 000005          RESET   ;;CLEAR THE WORLD
7005 032662 004710          $ENDAD: JSR     PC,(R0) ;;GO TO MONITOR
7006 032664 000240          NOP     ;;SAVE ROOM
7007 032666 000240          NOP     ;;FOR
7008 032670 000240          NOP     ;;ACT11
7009 032672 000137          JMP     @(PC)+      ;;RETURN
7010 032674 006176          $RTNAD: .WORD     TST1
7011 032676 377 377 000  $ENULL: .BYTE    -1,-1,0 ;;NULL CHARACTER STRING
7012 032701 015 042412 042116  $ENDMG: .ASCIIZ  <15><12>/END PASS #/
7013 032706 050040 051501 020123
7014 032714 000043
7015
  
```



```

7016
7017
7018      ;THIS FILLS MEMORY WITH GIVEN DATA
7019      ;USED CHIEFLY FOR HEADER INFORMATION
7020      ;CALL IS
7021      ;      JSR      RO,@#FLHEAD      ;FILL HEADER
7022      ;      LOC      ;LOCATION WHERE SAVED
7023      ;      XN       ;NUMBER OF WORDS
7024      ;      XD1     ;DATA REPEATED XN TIMES
7025      ;      XD2     ;DATA REPEATED XN TIMES
7026
7027
7028
7029
7030      FLHEAD:
7031      032716 010146      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
7032      032720 010246      MOV      R2,-(SP)      ;;PUSH R2 ON STACK
7033      032722 012001      MOV      (R0)+,R1      ;R1 HAS ADDRESS OF WHERE TO SAVE
7034      032724 012002      MOV      (R0)+,R2      ;R2 HAS NUMBER OF WORDS
7035
7036      ;*NOW FILL DATA
7037
7038      032726 012021      1$:      MOV      (R0)+,(R1)+      ;SAVE DATA
7039      032730 005302      DEC      R2              ;DECREMENT COUNT
7040      032732 001375      BNE     1$              ;BRANCH IF INCOMPLETE
7041      032734 012602      MOV      (SP)+,R2      ;;POP STACK INTO R2
7042      032736 012601      MOV      (SP)+,R1      ;;POP STACK INTO R1
7043      032740 000200      RTS      R0
7044
7045
7046
7047      ;THIS CLEARS ANY BLOCK OF MEMORY.
7048      ;FILLING IT WITH ANY DATA
7049      ;CALL IS
7050      ;      JSR      RO,@#CLAREA
7051      ;      F        ;FROM
7052      ;      N        ;NUMBER OF WORDS
7053      ;      D        ;DATA TO BE FILLED
7054
7055      ;R1 WILL HAVE STARTING ADDRESS OF BLOCK TO BE FILLED
7056      ;R2 WILL HAVE NUMBER OF WORDS
7057      ;R3 WILL HAVE DATA
7058
7059      CLAREA:
7060      032742 010146      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
7061      032744 010246      MOV      R2,-(SP)      ;;PUSH R2 ON STACK
7062      032746 010346      MOV      R3,-(SP)      ;;PUSH R3 ON STACK
7063      032750 012001      MOV      (R0)+,R1      ;FROM
7064      032752 012002      MOV      (R0)+,R2      ;NUMBER
7065      032754 012003      MOV      (R0)+,R3      ;DATA
7066      032756 010321      1$:      MOV      R3,(R1)+      ;MOVE DATA
7067      032760 005302      DEC      R2              ;COUNT
7068      032762 001375      BNE     1$              ;BRANCH IF NOT COMPLETE
7069      032764 012603      MOV      (SP)+,R3      ;;POP STACK INTO R3
7070      032766 012602      MOV      (SP)+,R2      ;;POP STACK INTO R2
7071      032770 012601      MOV      (SP)+,R1      ;;POP STACK INTO R1
    
```


7072 032772 000200 RTS R0 ;RETURN TO MAIN PROGRAM

7073
7074
7075
7076
7077
7078
7079
7080
7081
7082
7083
7084
7085
7086
7087
7088
7089
7090
7091
7092
7093
7094
7095
7096
7097
7098
7099

;THIS IS A SUBROUTINE TO FILL SAVED REGISTER LOCATION
;WITH GIVEN VALUE
;CALL IS
: JSR R0,@#FILLRE
: RHXX ;REGISTER NAME
: D ;DATA

FILLRE:
MOV R1,-(SP) ;:PUSH R1 ON STACK
MOV R2,-(SP) ;:PUSH R2 ON STACK
MOV (R0)+,R1 ;:ADDRESS OF ADDRESS OF REGISTER
MOV (R0)+,R2 ;:DATA
SUB #RHC,R1 ;:OFFSET
MOV R2,SAVERE(R1) ;:DATA IS MOVED IN
MOV (SP)+,R2 ;:POP STACK INTO R2
MOV (SP)+,R1 ;:POP STACK INTO R1
RTS R0 ;:RETURN TO MAIN PROGRAM

002172
004512


```

7148
7149
7150
7151
7152      ;THIS CHECKS DEVICE AVAILABLE (DVA) AND READY (RDY) IN RHCS1
7153      ;AND CHECKS MEDIUM ON LINE (MOL), DEVICE PRESENT (DPR), DEVICE READY (DRY) IN RHDS1
7154
7155      ;IT MAY CHECK VOLUME VALID (VV) IN RHDS1, DEPENDING ON ENTRY POINT
7156
7157
7158      033126  000000      PCJSR:  0      ;PC OF JSR
7159
7160      033130  011637  033126      CHECK:  MOV    (SP),@#PCJSR      ;SAVE PC OF JSR+4
7161      033134  162737  000004  033126      SUB     #4,@#PCJSR      ;GET PC OF JSR
7162      033142  011346      MOV     @R3,-(SP)      ;GET RHDS1
7163      033144  052716  000100      BIS     #VV,(SP)      ;DONT CHECK VV BIT
7164      033150  000406      BR     CHECKC      ;GOTO COMMON CHECK ROUTINE
7165
7166      033152  011637  033126      CHECKT: MOV    (SP),@#PCJSR      ;SAVE PC OF JSR+4
7167      033156  162737  000004  033126      SUB     #4,@#PCJSR      ;GET PC OF JSR
7168      033164  011346      MOV     @R3,-(SP)      ;GET RHDS1 & DO VV CHECK AT 3$
7169
7170      033166  011146      CHECKC: MOV    @R1,-(SP)      ;GET CS1
7171      033170  042716  173577      BIC     #173577,(SP)      ;CLEAR UNWANTED BITS
7172      033174  022726  004200      CMP     #DVA!RDY,(SP)+      ;RHCS1 SHOULD HAVE DEVICE AVAILABLE
7173      ;AND BE READY
7174      033200  001403      BEQ     3$      ;BRANCH IF GOOD
7175      033202  011137  001122      MOV     @R1,@#$BDADR      ;BAD DATA REGISTER (RHCS1)
7176      033206  104062      ERROR   62      ;RHCS1 DID NOT HAVE DEVICE
7177      ;AVAILABLE RIGHT AT THE START
7178      ;ALL OTHER BITS SHOULD BE 0
7179      033210  042716  102000      3$:    BIC     #ATA!LBT,(SP)      ;CLEAR UNWANTED BITS
7180      033214  022726  010700      CMP     #MOL!DPR!DRY!VV,(SP)+ ;RHDS1 SHOULD HAVE THESE SET
7181      033220  001404      BEQ     7$      ;BRANCH IF GOOD
7182      033222  011337  001122      MOV     @R3,@#$BDADR      ;BAD DATA IN REGISTER (RHDS1)
7183      033226  104061      ERROR   61      ;RHDS1 HAS SOME BITS OTHER
7184      ;THAN MOL, DRY, DPR, VV SET
7185      ;ALL OTHER BITS SHOULD BE 0
7186      033230  000207      RTS     PC      ;RETURN TO TEST AND HALT
7187
7188      033232  062716  000006      7$:    ADD     #6,(SP)      ;ADJUST STACK TO JUMP OVER HALT IN TEST
7189      033236  000207      RTS     PC      ;RETURN TO TEST AND CONTINUE
  
```

CZR.
CZR.

7
7
7
7
7
7
7
7

7190
7191
7192
7193
7194
7195
7196
7197
7198
7199
7200
7201
7202
7203
7204
7205
7206
7207
7208
7209
7210
7211
7212
7213
7214
7215
7216
7217
7218
7219
7220
7221
7222
7223
7224
7225
7226
7227
7228
7229
7230
7231
7232
7233
7234
7235
7236
7237
7238
7239
7240
7241
7242
7243
7244
7245

```

; *THIS IS A SUBROUTINE TO SAVE REGISTERS
; *IN THE REGISTER TABLE TO ANY LOCATION
; *THE CALL IS
; *JSR R0,@#SAVER
; *      F      :FROM
; *      T      :TO
; *      N      :NUMBER OF WORDS SAVED
; *F MUST ALWAYS BE RHCS1
; *T MUST ALWAYS BE SAVRE
  
```

SAVER:

```

MOV R1,-(SP)      ;;PUSH R1 ON STACK
MOV R2,-(SP)      ;;PUSH R2 ON STACK
MOV R3,-(SP)      ;;PUSH R3 ON STACK
MOV (R0)+,R1      :FROM
MOV (R0)+,R2      :TO
MOV (R0)+,R3      :NUMBER
1$: MOV @(R1)+,(R2)+ :SAVE REGISTER CONTENTS
DEC R3            :COUNT
BNE 1$           :BRANCH IF NOT DONE
MOV (SP)+,R3     ;;POP STACK INTO R3
MOV (SP)+,R2     ;;POP STACK INTO R2
MOV (SP)+,R1     ;;POP STACK INTO R1
RTS R0
  
```

```

;WHEN AN EVENT IS TO BE TIMED THE RP04 VECTORS TO 'TIME 1'
;PRIORITY OF PROCESS OR IS 4
;PRIORITY OF TRAPS MUST BE 6
;PRIORITY OF RP04 INTERRUPTS IS 7
;
  
```

```

TIME1: CLR @PCLCSR      ;STOP THE CLOCK
        MOV @PCLCTR,@#WAITTM ;GET TIME ON CLOCK
TIME2: MOV @RHCC,@#FINACC ;GET CURRENT CYLINDER
        MOV @RHLLA,@#FINALA ;GET LOOK AHEAD
        RTI              ;RETURN TO WAIT P OR WAIT.T
  
```

```

;THIS IS A WAIT LOOP WHEN AN EVENT IS TO BE TIMED
;THE CALL IS
  
```

```

033240 010146
033240 010246
033242 010346
033244 012001
033246 012002
033250 012003
033252 013122
033254 005303
033256 001375
033260 012603
033262 012602
033264 012601
033266 000200
033270
005077 146746
017737 146746 033330
017737 146724 004564
017737 146720 004562
000002
  
```



```

7246      :      WAT
7247      :      A      ;ABSOLUTE REGISTER ADDRESS
7248      :      B      ;BIT WAITED FOR
7249      :      TA     ;TIME ALLOWED GIVEN IN 10 MICROSEC
7250      :      TO     ;TOLERANCE PLUS/MINUS IN 10 MICROSEC
7251      :
7252      :R1-WILL HAVE TIME ALLOWED IN 10 MICRO SECONDS
7253      :R2-WILL HAVE TOLERANCE PLUS/MINUS IN 10 MICRO SECONDS
7254      :MINIMUM TIME THAT CAN BE MEASURED IS ABOUT 12 MICRO SECONDS
7255      :FOR THE SLOWEST PROCESSOR
7256
7257 033322 000000      WAITPC: 0      ;WAT PC
7258 033324 000000      WAITRE: 0      ;WAIT ON REGISTER ADDRESS
7259 033326 000000      WAITBT: 0      ;WAIT ON BIT
7260 033330 000000      WAITTM: 0      ;WAITED TIME
7261 033332 005037 033330      WAIT.P: CLR @#WAITTM ;CLEAR WAITED TIME
7262 033336 005077 146704      CLR @PCLBUF ;CLEAR COUNT SET BUFFER
7263 033342 012777 000021 146674      MOV #GO!BIT4,@PCLCSR ;COUNT UP, 100 KHZ, START CLOCK
7264 033350 010046      MOV R0,-(SP) ;:PUSH R0 ON STACK
7265 033352 010146      MOV R1,-(SP) ;:PUSH R1 ON STACK
7266 033354 010246      MOV R2,-(SP) ;:PUSH R2 ON STACK
7267 033356 010346      MOV R3,-(SP) ;:PUSH R3 ON STACK
7268 033360 016600 000010      MOV 10(SP),R0 ;R0 HAS ADDRESS OF NEXT LOCATION
7269 033364 010037 033322      MOV R0,@#WAITPC ;NOW WAITPC HAS WAT PC + 2
7270 033370 162737 000002 033322      SUB #2,@#WAITPC ;WAT PC IS IN WAITPC
7271 033376 013037 033324      MOV @ (R0)+,@#WAITRE ;WAIT ON REGISTER ADDRESS
7272 033402 012037 033326      MOV (R0)+,@#WAITBT ;WAIT ON BIT
7273 033406 012001      MOV (R0)+,R1 ;R1 HAS TIME IN 10 MSEC
7274 033410 012002      MOV (R0)+,R2 ;R2 HAS TOLERANCE IN 10 MSEC
7275 033412 010066 000010      MOV R0,10(SP) ;RESTORE RETURN ON STACK
7276
7277      ;*THIS SECTION WAITS FOR BIT, THROUGH TWO COUNT DOWNS
7278
7279 033416 013703 033570      MOV @#TIMCNT,R3 ;R3 IS A TEMPORARY COUNTER
7280 033422 033777 033326 177674 1$: BIT @#WAITBT,@WAITRE ;IS REQUIRED BIT THERE
7281 033430 001025      BNE 4$ ;BRANCH IF YES
7282 033432 005303      DEC R3 ;COUNT IF REQUIRED BIT NOT THERE
7283 033434 001372      BNE 1$
7284 033436 013703 033570      MOV @#TIMCNT,R3 ;TEMPORARY COUNTER
7285 033442 033777 033326 177654 2$: BIT @#WAITBT,@WAITRE ;IS REQUIRED BIT THERE
7286 033450 001015      BNE 4$ ;BRANCH IF YES
7287 033452 005303      DEC R3 ;COUNT IF REQUIRED BIT NOT THERE
7288 033454 001372      BNE 2$
7289 033456 017737 177642 001126      MOV @WAITRE,@#$BDDAT ;REGISTER CONTENTS FOR TYPEOUT
7290 033464 032777 000100 146506      BIT #IE,@RHCS1 ;DID ANY INTERRUPT OCCUR
7291 033472 001402      BEQ 3$ ;BRANCH IF YES
7292 033474 104001      ERROR 1 ;RP04 DID NOT INTERRUPT
7293 033476 000427      BR 7$ ;OUT
7294 033500 104002      3$: ERROR 2 ;RP04 INTERRUPTED BUT WAITED
7295      ;ON BIT DID NOT OCCUR
7296      ;EVEN AFTER TWO COUNT DOWNS
7297      ;FROM 177777 TO 0
7298 033502 000425      BR 7$ ;OUT
7299
7300      ;*NOW TIME AND TOLERANCE WILL BE CHECKED
7301 033504 017737 177614 001126 4$: MOV @WAITRE,@#$BDDAT ;REGISTER CONTENTS FOR TYPEOUT
    
```



```

7302 033512 032777 000100 146460      BIT      #IE,@RHCS1      ;DID ANY INTERRUPT OCCUR
7303 033520 001402      BEQ      5$            ;BRANCH IF YES
7304 033522 104003      ERROR   3            ;INTERRUPT DID NOT OCCUR EVEN
7305                                     ;AFTER ONE BNE AND ONE MOV
7306                                     ;OF THE WAITED ON BIT SETTING
7307 033524 000414      BR       7$            ;OUT
7308 033526 160201      SUB     R2,R1          ;R1 NOW HAS LOWER LIMIT OF TIME
7309 033530 023701 033330      CMP     @#WAITTM,R1   ;FOR GOOD RESULTS, WAITTM
7310                                     ;MUST BE GREATER OR EQUAL
7311                                     ;TORI
7312 033534 103002      BHIS   6$            ;BRANCH IF GOOD
7313 033536 104004      ERROR   4            ;BIT DID OCCUR BUT TIME
7314                                     ;TAKEN IS BELOW LOWER LIMIT
7315 033540 000406      BR       7$            ;OUT
7316
7317 033542 060202      6$:    ADD     R2,R2          ;DOUBLE TOLERANCE
7318 033544 060201      ADD     R2,R1          ;R1 NOW HAS UPPER LIMIT OF TIME
7319 033546 020137 033330      CMP     R1,@#WAITTM   ;FOR GOOD RESULTS, WAITTM
7320                                     ;MUST BE LESS OR EQUAL TO R1
7321 033552 103001      BHIS   7$            ;BRANCH IF GOOD
7322 033554 104004      ERROR   4            ;BIT DID OCCUR BUT TIME TAKEN
7323                                     ;IS ABOVE UPPER LIMIT
7324 033556      7$:
7325 033556 012603      MOV     (SP)+,R3      ;;POP STACK INTO R3
7326 033560 012602      MOV     (SP)+,R2      ;;POP STACK INTO R2
7327 033562 012601      MOV     (SP)+,R1      ;;POP STACK INTO R1
7328 033564 012600      MOV     (SP)+,R0      ;;POP STACK INTO R0
7329 033566 000002      RTI
7330                                     ;RETURN TO MAIN TEST
7331
7332
7333
7334
7335
7336                                     ;THIS IS A WAIT LOOP WHEN NO P-CLOCK IS AVAILABLE
7337                                     ;NO TIMING IS DONE
7338                                     ;CALL IS
7339                                     ;      WAT
7340                                     ;      A      ;ABSOLUTE REGISTER ADDRESS
7341                                     ;      B      ;BIT WAITED FOR
7342                                     ;      TA     ;TIME-NOT USED HERE
7343                                     ;      TO     ;TIME-NOT USED HERE
7344                                     ;R3-IS A TEMPORARY COUNTER
7345
7346 033570 177777      TIMCNT: 177777      ;COUNT FOR WAIT LOOP
7347 033572 000025      RPTCTR: 25         ;COUNT FOR INTERRUPT WAIT (11/70 CPU)
7348
7349
7350                                     WAIT.T:
7351 033574 010046      MOV     R0,-(SP)      ;;PUSH R0 ON STACK
7352 033576 010346      MOV     R3,-(SP)      ;;PUSH R3 ON STACK
7353
7354 033600 016600 000004      MOV     4(SP),R0     ;R0 HAS ADDRESS OF NEXT LOCATION
7355 033604 010037 033322      MOV     R0,@#WAITPC  ;WAT PC +2 IS IN WAITPC
7356 033610 162737 000002 033322      SUB     #2,@#WAITPC  ;WAT PC IS IN WAITPC
7357 033616 013037 033324      MOV     @(R0)+,@#WAITRE ;WAIT ON REGISTER ADDRESS
  
```



```

7358 033622 012037 033326      MOV      (R0)+,@#WAITBT  ;WAIT ON BIT
7359 033626 022020              CMP      (R0)+,(R0)+    ;DUMP NEXT TWO WORDS-TA, TO
7360 033630 010066 000004      MOV      R0,4(SP)       ;RESTORE RETURN ON STACK
7361
7362                               ;*THIS HAS THE TWO COUNT DOWNS FROM 177777
7363
7364 033634 013703 033570      MOV      @#TIMCNT,R3    ;R3 HAS TEMPORARY COUNT
7365 033640 033777 033326 177456 1$:  BIT      @#WAITBT,@WAITRE ;IS REQUIRED BIT THERE ?
7366 033646 001025              BNE      4$             ;CHECK FOR THE INTERRUPT
7367 033650 005303              DEC      R3             ;COUNT IF REQUIRED BIT NOT THERE
7368 033652 001372              BNE      1$
7369 033654 013703 033570      MOV      @#TIMCNT,R3    ;SECOND COUNT DOWN FROM 177777
7370 033660 033777 033326 177436 2$:  BIT      @#WAITBT,@WAITRE ;IS REQUIRED BIT THERE ?
7371 033666 001015              BNE      4$             ;CHECK FOR INTERRUPT
7372 033670 005303              DEC      R3             ;COUNT IF REQUIRED BIT NOT THERE
7373 033672 001372              BNE      2$
7374 033674 017737 177424 001126  MOV      @WAITRE,@#$BDDAT ;REGISTER CONTENTS FOR TYPEOUT
7375 033702 032777 000100 146270  BIT      #IE,@RHCS1     ;DID ANY INTERRUPT OCCUR ?
7376 033710 001402              BEQ      3$             ;BRANCH IF YES
7377
7378 033712 104001              ERROR    1              ;RP04 DID NOT INTERRUPT
7379                               ;BIT DID NOT OCCUR
7380 033714 000417              BR       5$             ;OUT ----->
7381
7382 033716 104002              3$:  ERROR    2              ;RP04 INTERRUPTED BUT
7383                               ;WAITED ON BIT DID NOT OCCUR
7384                               ;EVEN AFTER TWO COUNT DOWNS
7385                               ;FROM 177777 TO 0
7386 033720 000415              BR       5$             ;OUT ----->
7387
7388                               ;*BIT DID SET SO CHECK IF INTERRUPT OCCURRED
7389
7390                               ;*THE AMOUNT OF TIME ALLOWED CAN BE CHANGED BY ALTERING LOCATION
7391                               ;*'RPTCTR' ABOVE
7392
7393 033722 013703 033572              4$:  MOV      @#RPTCTR,R3    ;LOAD COUNTER WITH COUNT
7394 033726 005303              6$:  DEC      R3             ;COUNT DOWN ONE
7395 033730 001376              BNE      6$             ;DO AGAIN IF NOT ZERO YET
7396
7397
7398 033732 032777 000100 146240  BIT      #IE,@RHCS1     ;DID ANY INTERRUPT OCCUR ?
7399 033740 001405              BEQ      5$             ;BRANCH IF YES
7400 033742 017737 177356 001126  MOV      @WAITRE,@#$BDDAT ;REGISTER CONTENTS FOR TYPEOUT
7401 033750 104003              ERROR    3              ;INTERRUPT DID NOT OCCUR
7402                               ;EVEN AFTER ONE BNE OF
7403                               ;THE WAITED ON BIT OCCURING
7404 033752 000400              BR       5$             ;OUT ----->
7405
7406                               5$:
7407 033754 012603              MOV      (SP)+,R3       ;;POP STACK INTO R3
7408 033756 012600              MOV      (SP)+,R0       ;;POP STACK INTO R0
7409 033760 000002              RTI                      ;RETURN TO MAIN TEST
7410

```



```

7411
7412      ;THIS CHANGES REGISTER SAVED VALUE
7413      ;CALL IS
7414      ;      JSR      RO,@#CHREG
7415      ;      R          ;REGISTER TO BE CHANGED
7416      ;      N          ;NUMBER OF BITS TO BE CHANGED
7417      ;      NEW       ;NEW VALUE OF BIT MUST BE 0 OR 1
7418      ;      P          ;POSITION OF BIT TO BE CHANGED
7419      ;NEW AND P WILL BE REPEATED N NUMBER OF TIMES
7420
7421      CHREG:
7422      033762 010146      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
7423      033764 010246      MOV      R2,-(SP)      ;;PUSH R2 ON STACK
7424      033766 012001      MOV      (R0)+,R1      ;R1 HAS ADDRESS OF ADDRESS OF REGISTER
7425      033770 012002      MOV      (R0)+,R2      ;R2 HAS NUMBER OF CHANGES
7426      033772 162701 002172      SUB      #RHWC,R1      ;R1 HAS OFFSET OF REQUIRED REGISTER
7427      033776 005720      1$:     TST      (R0)+      ;IS A BIC OR A BIS TO BE DONE
7428      034000 001403      BEQ      2$           ;BRANCH IF A BIC IS REQUIRED
7429      034002 052061 004512      BIS      (R0)+,SAVERE(R1) ;SET REQUIRED BIT
7430      034006 000402      BR       3$           ;BRANCH TO DECREMENT COUNT
7431      034010 042061 004512      2$:     BIC      (R0)+,SAVERE(R1) ;CLEAR REQUIRED BIT
7432      034014 005302      3$:     DEC      R2          ;DECREMENT NUMBER OF CHANGES
7433      034016 001367      BNE      1$           ;BRANCH IF NOT COMPLETE
7434      034020 012602      MOV      (SP)+,R2      ;;POP STACK INTO R2
7435      034022 012601      MOV      (SP)+,R1      ;;POP STACK INTO R1
7436      034024 000200      RTS      RO          ;RETURN TO MAIN PROGRAM
7437
7438
7439
7440
7441
7442
7443      ;THIS FILLS A BLOCK WITH INCREMENTAL DATA
7444      ;CALL IS
7445      ;      JSR      RO,@#FILL
7446      ;      F          ;FROM
7447      ;      N          ;NUMBER OF WORDS
7448      ;      S          ;STARTING VALUE OF DATA
7449      ;      I          ;INCREMENT DATA BY
7450
7451      FILL:
7452      034026 010146      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
7453      034030 010246      MOV      R2,-(SP)      ;;PUSH R2 ON STACK
7454      034032 010346      MOV      R3,-(SP)      ;;PUSH R3 ON STACK
7455      034034 010446      MOV      R4,-(SP)      ;;PUSH R4 ON STACK
7456      034036 012001      MOV      (R0)+,R1      ;R1 HAS ADDRESS WHERE DATA IS TO GO
7457      034040 012002      MOV      (R0)+,R2      ;R2 HAS NUMBER OF WORDS TO BE FILLED
7458      034042 012003      MOV      (R0)+,R3      ;STARTING VALUE OF DATA
7459      034044 012004      MOV      (R0)+,R4      ;R4 HAS INCREMENT
7460
7461      ;*NOW DATA WILL BE FILLED
7462      034046 010321      1$:     MOV      R3,(R1)+      ;FILL DATA
7463      034050 060403      ADD      R4,R3          ;GET NEXT VALUE OF DATA
7464      034052 005302      DEC      R2          ;DECREMENT COUNT
7465      034054 001374      BNE      1$           ;BRANCH IF ALL NOT DONE
7466      034056 012604      MOV      (SP)+,R4      ;;POP STACK INTO R4
    
```



```

7467 034060 012603      MOV      (SP)+,R3      ;;POP STACK INTO R3
7468 034062 012602      MOV      (SP)+,R2      ;;POP STACK INTO R2
7469 034064 012601      MOV      (SP)+,R1      ;;POP STACK INTO R1
7470 034066 000200      RTS       R0           ;RETURN TO MAIN PROGRAM
7471
7472
7473
7474
7475
7476
7477
7478      ;THIS IS A SUBROUTINE TO COMPARE REGISTERS
7479      ;GOOD DATA IS ALREADY SAVED IN 'SAVERE'
7480      ;TEST DATA IS IN THE REGISTERS
7481      ;CALL IS
7482      ;      JSR      R0,@#COMREG
7483      ;      SAVERE      ;GOOD DATA
7484      ;      RHCS1      ;ADDRESS OF ADDRESS TEST DATA
7485      ;      N.         ;RETURN FOR ERROR
7486      ;      RG         ;RETURN FOR GOOD COMPARISON
7487      ;ON RETURN WITH ERROR '$GDDAT' HAS GOOD DATA, '$BDDAT' HAS BAD DATA
7488      ;'REGADR' HAS REGISTER ADDRESS
7489      COMREG:
7490      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
7491      MOV      R2,-(SP)      ;;PUSH R2 ON STACK
7492      MOV      R3,-(SP)      ;;PUSH R3 ON STACK
7493      MOV      R4,-(SP)      ;;PUSH R4 ON STACK
7494      MOV      R5,-(SP)      ;;PUSH R5 ON STACK
7495      MOV      (R0)+,R1      ;R1 HAS ADDRESS OF GOOD DATA
7496      MOV      (R0)+,R2      ;R2 HAS ADDRESS OF ADDRESS OF TEST DATA
7497      MOV      (R0)+,R3      ;R3 HAS NUMBER OF WORDS
7498      MOV      (R0)+,R4      ;R4 HAS RETURN FOR ERROR
7499      MOV      (R0),R0       ;R0 HAS RETURN ON NO ERROR
7500      ;*NOW SAVE REGISTERS
7501      JSR      PC,@#PUTREG      ;SAVE REGISTERS
7502      MOV      @#SAVERE+25,@#AS+1;MAKE UPPER BYTE OF R HAS SAME
7503      MOV      #-2,R5         ;PRESET R5 TO -2
7504      ;*NOW COMPARES WILL MADE
7505      1$:      ADD      #2,R5      ;INCREMENT TO INDEX
7506      CMP      (R1)+,(R2)+      ;COMPARE REGISTER CONTENTS
7507      BEQ      2$              ;BRANCH IF GOOD
7508      MOV      -(R1),@#$GDDAT    ;SAVE GOOD DATA
7509      MOV      -(R2),@#$BDDAT    ;SAVE BAD DATA
7510      MOV      RHC(R5),@#REGADR  ;SAVE ADDRESS OF FAILING REGISTER
7511      JSR      PC,@R4          ;RETURN TO MAIN PROGRAM
7512      ;TO PRINT ERROR
7513      CMP      (R1)+,(R2)+      ;UNDO -(R1) AND -(R2) FOR ERRORS
7514      MOV      @SWR,-(SP)      ;GET SWITCH SETTING
7515      BIC      #^C600,(SP)      ;KEEP ONLY SWITCH 7 AND 8
7516      CMP      #SW07,(SP)+      ;IS 7 SET AND 8 DOWN
7517      BEQ      3$              ;BRANCH OUT IF YES
7518      2$:      DEC      R3         ;ARE ALL COMPARES DONE
7519      BNE      1$              ;BRANCH IF NOT COMPLETE
7520
7521      3$:      MOV      (SP)+,R5      ;;POP STACK INTO R5
7522
    
```


7523	034210	012604	MOV	(SP)+,R4	::POP STACK INTO R4
7524	034212	012603	MOV	(SP)+,R3	::POP STACK INTO R3
7525	034214	012602	MOV	(SP)+,R2	::POP STACK INTO R2
7526	034216	012601	MOV	(SP)+,R1	::POP STACK INTO R1
7527	034220	000200	RTS	R0	:RETURN TO MAIN PROGRAM
7528	034222	000000	4\$: .WORD	0	:TEMP STORAGE

7529
 7530
 7531
 7532
 7533
 7534
 7535
 7536
 7537
 7538
 7539
 7540
 7541
 7542
 7543
 7544
 7545
 7546
 7547
 7548
 7549
 7550 034224 000000
 7551 034226
 7552 034226 005037 177776
 7553 034232 012737 177777 040746
 7554 034240 104401 034246
 7555 034244 000421
 7556 034310 013746 004504
 7557 034314 104402
 7558 034316 104401 034324
 7559 034322 000414
 7560 034354 013746 001110
 7561 034360 104402
 7562 034362 104401 001223
 7563 034366 104401 034374
 7564 034372 000430
 7565 034454 104401 034462
 7566 034460 000420
 7567 034522 104401 034530
 7568 034526 000432
 7569 034614 104412
 7570 034616 062716 000002
 7571 034622 012637 001106
 7572 034626 104401 034634
 7573 034632 000417
 7574 034672 104401 034700
 7575 034676 000440
 7576 035000 104412
 7577 035002 012637 001110
 7578 035006 013746 001106
 7579 035012 000002
 7580
 7581
 7582
 7583
 7584

:HERE IS A DETAILED EXPLANATION OF HOW THE LOOP ON ERROR WORKS.
 :ON HITTING AN ERROR IF THE LOOP ON ERROR SWITCH IS SET, THE
 :PROGRAM GOES BACK - USUALLY BACK TO THE BEGINNING OF THE TEST.

 :WHEN THIS OPERATOR SELECTABLE SCOPE LOOP IS USED THEN THE POINT
 :THE PROGRAM GOES BACK TO CAN BE CHANGED.
 :THE RESTRICTIONS TO THE POINT WHERE THE PROGRAM CAN GO ARE: -
 :1. IT MUST BE WITHIN THE TEST UNDER CONSIDERATION
 :2. LOOP ON ERROR SWITCH MUST BE SET
 :3. THE ERROR MUST OCCUR WITHIN THE TEST UNDER CONSIDERATION
 :IF THE ERROR DOES NOT OCCUR WITHIN THE TEST UNDER CONSIDERATION
 :THE PROGRAM WILL REVERT TO NORMAL OPERATION. HOWEVER, IF LOOP ON
 :TEST SWITCH IS SET AND THIS OPERATOR SELECTABLE SCOPE LOOP IS USED
 :THEN THE PROGRAM WILL LOOP BACK TO THE SELECTED POINT WHEN IT
 :COMES TO THE END OF THE TEST UNDER CONSIDERATION.
 :
 :AFTER LOOPING FOR SOME TIME IF THE LOOP SWITCH IS PUT DOWN THEN
 :NORMAL OPERATION WILL CONTINUE.

TESTAD: 0 ;FIRST ADDRESS OF TEST
 OPERSEL:
 CLR PS ;MAKE PROCESSOR STATUS ZERO
 MOV #-1,@#PRITEM ;CLEAR PREVIOUS ITEM NUMBER
 TYPE ,65\$;:TYPE ASCIZ STRING
 BR 64\$;:GET OVER THE ASCIZ
 MOV @#TSTNM,-(SP) ;GET READY TO TYPE TEST
 TYPOC ;NUMBER
 TYPE ,67\$;:TYPE ASCIZ STRING
 BR 66\$;:GET OVER THE ASCIZ
 MOV @#\$LPERR,-(SP) ;GET READY TO TYPE LOOP BACK PC
 TYPOC
 TYPE , \$CRLF
 TYPE ,69\$;:TYPE ASCIZ STRING
 BR 68\$;:GET OVER THE ASCIZ
 TYPE ,71\$;:TYPE ASCIZ STRING
 BR 70\$;:GET OVER THE ASCIZ
 TYPE ,73\$;:TYPE ASCIZ STRING
 BR 72\$;:GET OVER THE ASCIZ
 RDOCT
 ADD #2,(SP) ;GET LPADR
 MOV (SP)+,@#\$LPADR
 TYPE ,75\$;:TYPE ASCIZ STRING
 BR 74\$;:GET OVER THE ASCIZ
 TYPE ,77\$;:TYPE ASCIZ STRING
 BR 76\$;:GET OVER THE ASCIZ
 RDOCT
 MOV (SP)+,@#\$LPERR ;GET LPERR
 MOV @#\$LPADR,-(SP)
 RTI


```

7585
7586
7587
7588
7589
7590
7591
7592
7593
7594
7595 035014
7596 035014 010046
7597 035016 010146
7598 035020 010246
7599 035022 012700 002172
7600 035026 012701 002254
7601 035032 012702 000022
7602 035036 013021
7603 035040 005302
7604 035042 001375
7605 035044 012602
7606 035046 012601
7607 035050 012600
7608 035052 000207

                PUTREG:
                MOV     R0,-(SP)          ;;PUSH R0 ON STACK
                MOV     R1,-(SP)          ;;PUSH R1 ON STACK
                MOV     R2,-(SP)          ;;PUSH R2 ON STACK
                MOV     #RHC,R0           ;STARTING ADDRESS OF REGISTERS
                MOV     #WC,R1            ;STARTING ADDRESS OF SAVING LOCATIONS
                MOV     #RHC-RHC+2/2,R2  ;NUMBER OF REG. INTO R2
10$:            MOV     @(R0)+,(R1)+     ;SAVE HARDWARE REG.
                DEC     R2
                BNE    10$
                MOV     (SP)+,R2         ;;POP STACK INTO R2
                MOV     (SP)+,R1         ;;POP STACK INTO R1
                MOV     (SP)+,R0         ;;POP STACK INTO R0
                RTS     PC
  
```



```

7609      :THIS IS A DATA COMMAND SETUP SUBROUTINE
7610      :THE CALL IS
7611      :      JSR      R0,@#RUN
7612      :      C          :CYLINDER
7613      :      S          :SECTOR
7614      :      T          :TRACK
7615      :      -W        :WORD COUNT
7616      :      B          :BUS ADDRESS
7617      :      BAI        :BUS ADDRESS INHIBIT
7618      :      FMT22!ECI!HCI :FMT22=1 =16 BIT WORDS
7619      :                  :ECI = ECC CORRECTION INHIBIT
7620      :                  :HCI = HEADER COMPARE INHIBIT
7621      :                  :COMMAND ADDRESS
7622 035054 012077 145132  RUN:  MOV      (R0)+,@RHCA  :CYLINDER
7623 035060 012077 145120      MOV      (R0)+,@RHDST :DESIRED SECTOR/TRACK
7624 035064 012077 145102      MOV      (R0)+,@RHWC  :WORD COUNT
7625 035070 012077 145100      MOV      (R0)+,@RHBA  :BUS ADDRESS
7626 035074 013746 004616      MOV      @#UNIT,-(SP) :GET UNIT NO
7627 035100 052016          BIS      (R0)+,(SP)  :SET BUS ADDRESS INHIBIT
7628 035102 012677 145070      MOV      (SP)+,@RHCS2 :UNIT NO AND BAI TO RHCS2
7629 035106 012077 145076      MOV      (R0)+,@RHOF  :FORMAT, ECC INHIBIT, HEADER
7630          :COMPARE, IF THERE
7631 035112 013077 145062      MOV      @(R0)+,@RHCS1 :COMMAND IN RHCS1
7632 035116 000200          RTS      R0          :RETURN TO MAIN PROGRAM
7633
7634
7635
7636

```

```

7637      :THIS IS A SUBROUTINE TO COMPARE TWO BLOCKS IN MEMORY
7638      :R1 HAS GOOD DATA BUFFER ADDRESS
7639      :R2 HAS TEST DATA BUFFER ADDRESS
7640      :R5 HAS ADDRESS OF RETURN ON ERROR
7641      :R3 HAS NUMBER OF WORDS TO BE COMPARED
7642      :R4 HAS ONE MORE THAN NUMBER OF WORDS TO BE COMPARED
7643      :CALL IS
7644      :      JSR      R0,@#COMPAR
7645      :      G          :ADDRESS OF GOOD DATA
7646      :      T          :ADDRESS OF TEST DATA
7647      :      N          :NUMBER OF WORDS TO BE COMPARED
7648      :      RE         :RETURN ON ERROR
7649      :      RG         :RETURN ON NO ERROR
7650
7651
7652

```

```

7653 035120          COMPAR:
7654 035120 010146      MOV      R1,-(SP)  ;;PUSH R1 ON STACK
7655 035122 010246      MOV      R2,-(SP)  ;;PUSH R2 ON STACK
7656 035124 010346      MOV      R3,-(SP)  ;;PUSH R3 ON STACK
7657 035126 010446      MOV      R4,-(SP)  ;;PUSH R4 ON STACK
7658 035130 010546      MOV      R5,-(SP)  ;;PUSH R5 ON STACK
7659 035132 012001      MOV      (R0)+,R1  :ADDRESS OF GOOD DATA BUFFER
7660 035134 012002      MOV      (R0)+,R2  :ADDRESS OF TEST DATA BUFFER
7661 035136 012003      MOV      (R0)+,R3  :NO OF WORDS TO BE COMPARED
7662 035140 012005      MOV      (R0)+,R5  :RETURN ON ERROR
7663 035142 011000      MOV      (R0),R0   :RETURN ON NO ERROR
7664 035144 010304      MOV      R3,R4   :NO OF WORDS TO BE COMPARED

```



```

7665 035146 005204          INC      R4
7666 035150 010437 004502  1$:     MOV      R4,@#ERWORD      ;FOR ERROR WORD NO
7667 035154 022122          CMP      (R1)+,(R2)+      ;COMPARE GOOD WITH TEST DATA
7668 035156 001417          BEQ      2$               ;BRANCH IF GOOD
7669
7670 035160 014137 001124          MOV      -(R1),@#$GDDAT   ;GOOD DATA
7671 035164 014237 001126          MOV      -(R2),@#$BDDAT   ;BAD DATA
7672 035170 160337 004502          SUB      R3,@#ERWORD      ;ERROR WORD NO.
7673 035174 004715          JSR      PC,@R5           ;RETURN TO PRINT ERROR
7674 035176 022122          CMP      (R1)+,(R2)+      ;UNDO -(R1) AND -(R2) FOR ERRORS
7675 035200 017746 143734          MOV      @SWR,-(SP)       ;GET SWITCH SETTING
7676 035204 042716 177177          BIC      #^C600,(SP)      ;KEEP ONLY SWITCH 7 AND 8
7677 035210 022726 000200          CMP      #SW07,(SP)+      ;IS 7 SET AND 8 RESET
7678 035214 001402          BEQ      3$               ;BRANCH OUT IF YES
7679 035216 005303          2$:     DEC      R3
7680 035220 001353          BNE      1$               ;BRANCH IF ALL NOT DEVICE
7681 035222
7682 035222 012605          3$:     MOV      (SP)+,R5          ;;POP STACK INTO R5
7683 035224 012604          MOV      (SP)+,R4          ;;POP STACK INTO R4
7684 035226 012603          MOV      (SP)+,R3          ;;POP STACK INTO R3
7685 035230 012602          MOV      (SP)+,R2          ;;POP STACK INTO R2
7686 035232 012601          MOV      (SP)+,R1          ;;POP STACK INTO R1
7687 035234 000200          RTS      R0               ;RETURN TO MAIN PROGRAM
7688 ;*      THIS ROUTINE WILL ALLOW THE CHANGE OF THE BASE
7689 ;*      ADDRESS FROM 176700 TO ANY TYPED VALUE
7690
7691 035236          BASECH:
7692 035236 104401 035244          TYPE     ,65$             ;;TYPE ASCIZ STRING
7693 035242 000425          BR       64$             ;;GET OVER THE ASCIZ
7694 035316 013746 002200          MOV      @#RHCS1,-(SP)   ;GET READY TO TYPE OLD BASE
7695 035322 104402
7696 035324 104401 035332          TYPOC   TYPE     ,67$             ;;TYPE ASCIZ STRING
7697 035330 000425          BR       66$             ;;GET OVER THE ASCIZ
7698 035404 004737 037340          JSR      PC,@#$TKINT     ;INITIALIZE THE TTY KEYBOARD
7699 035410 104412
7700 035412 012700 002170          RDOCT   MOV      #RHDB,R0     ;GET STARTING ADDRESS OF REGISTERS
7701 035416 012701 000026          MOV      #22,R1          ;NUMBER OF REGISTERS
7702 035422 012737 036226 000004          MOV      #ADTIMO,@#4     ;SET UP TRAP CATCHER FOR TEST
7703 035430 021637 002200          CMP      @SP,@#RHCS1     ;NEW ADDRESS
7704 035434 001407          BEQ      1$               ;NO, JUST OLD ONE RETYPED
7705 035436 005776 000000          TST     @0(SP)           ;DO THE ADDRESS ACCESS
7706 035442 163716 002200          SUB      @#RHCS1,@SP     ;GET THE ADDRESS OFFSET
7707 035446 061620          2$:     ADD      @SP,(R0)+      ;AND PLUG IT IN
7708 035450 005301          DEC      R1               ;ONE LESS REGISTER TO DO
7709 035452 001375          BNE      2$               ;BUT WE'RE NOT DONE YET!
7710 035454
7711 035454 104401 035462          1$:     TYPE     ,69$             ;;TYPE ASCIZ STRING
7712 035460 000417          BR       68$             ;;GET OVER THE ASCIZ
7713 035520 013746 002166          MOV      @#RPVEC,-(SP)   ;GET READY TO TYPE OLD VECTOR ADDRESS
7714 035524 104402
7715 035526 104401 035534          TYPOC   TYPE     ,71$             ;;TYPE ASCIZ STRING
7716 035532 000437          BR       70$             ;;GET OVER THE ASCIZ
7717 035632 104412
7718 035634 012637 002166          RDOCT   MOV      (SP)+,@#RPVEC ;SETUP VECTOR ADDRESS
7719 035640 104401 035646          TYPE     ,73$             ;;TYPE ASCIZ STRING
7720 035644 000417          BR       72$             ;;GET OVER THE ASCIZ

```


END OF PASS ROUTINE

7721	035704	013746	002200	MOV	@#RHCS1,-(SP)	
7722	035710	104402		TYPOC		
7723	035712	104401	035720	TYPE	,75\$::TYPE ASCIZ STRING
7724	035716	000417		BR	,74\$::GET OVER THE ASCIZ
7725	035756	013746	002166	MOV	@#RPVEC,-(SP)	
7726	035762	104402		TYPOC		
7727	035764	104401	035772	TYPE	,77\$::TYPE ASCIZ STRING
7728	035770	000417		BR	,76\$::GET OVER THE ASCIZ
7729	036030	104401	036036	TYPE	,79\$::TYPE ASCIZ STRING
7730	036034	000402		BR	,78\$::GET OVER THE ASCIZ
7731	036042	104401	036050	TYPE	,81\$::TYPE ASCIZ STRING
7732	036046	000424		BR	,80\$::GET OVER THE ASCIZ
7733	036120	104401	036126	TYPE	,83\$::TYPE ASCIZ STRING
7734	036124	000426		BR	,82\$::GET OVER THE ASCIZ
7735	036202	012746	000200	MOV	#RA,-(SP)	
7736	036206	104402		TYPOC		
7737	036210	104401	036216	TYPE	,85\$::TYPE ASCIZ STRING
7738	036214	000402		BR	,84\$::GET OVER THE ASCIZ
7739	036222	000137	004710	JMP	@#BEGIN	:DO IT OVER AGAIN
7740	036226					
7741	036226	104401	036234	TYPE	,65\$::TYPE ASCIZ STRING
7742	036232	000424		BR	,64\$::GET OVER THE ASCIZ
7743	036304	022626		CMP	(SP)+,(SP)+	:RESTORE STACK
7744	036306	000137	035236	JMP	@#BASECH	:AND DO THE QUERY AGAIN!
7745						

ADTIMO:


```

7754                                     .SBTTL SYSMAC LIBRARY ROUTINES
7755
7756
7757 036402 104407                      CKSWR          ::TEST FOR CHANGE IN SOFT-SWR
7758 036404 032777 040000 142526 1$:  BIT          #BIT14,@SWR  ::LOOP ON PRESENT TEST?
7759 036412 001111                      BNE          $OVER   ::YES IF SW14=1
7760 036414 000416                      $XTSTR: BR      6$    ::IF RUNNING ON THE 'XOR' TESTER CHANGE
7761 036416 013746 000004              MOV          @#ERRVEC,-(SP) ::SAVE THE CONTENTS OF THE ERROR VECTOR
7762 036422 012737 036442 000004      MOV          #5$,@#ERRVEC  ::SET FOR TIMEOUT
7763 036430 005737 177060              TST          @#177060     ::TIME OUT ON XOR?
7764 036434 012637 000004              MOV          (SP)+,@#ERRVEC ::RESTORE THE ERROR VECTOR
7765 036440 000463                      BR           $SVLAD      ::GO TO THE NEXT TEST
7766 036442 022626                      5$:  CMP      (SP)+,(SP)+  ::CLEAR THE STACK AFTER A TIME OUT
7767 036444 012637 000004              MOV          (SP)+,@#ERRVEC ::RESTORE THE ERROR VECTOR
7768 036450 000423                      BR           7$         ::LOOP ON THE PRESENT TEST
7769 036452 032777 000400 142460      BIT          #BIT08,@SWR  ::LOOP ON SPEC. TEST?
7770 036460 001404                      BEQ          2$         ::BR IF NO
7771 036462 127737 142452 001102      CMPB        @SWR,$TSTNM   ::ON THE RIGHT TEST? SWR<7:0>
7772 036470 001462                      BEQ          $OVER      ::BR IF YES
7773 036472 105737 001103 2$:  TSTB        $ERFLG      ::HAS AN ERROR OCCURRED?
7774 036476 001421                      BEQ          3$         ::BR IF NO
7775 036500 123737 001115 001103      CMPB        $ERMAX,$ERFLG ::MAX. ERRORS FOR THIS TEST OCCURRED?
7776 036506 101015                      BHI          3$         ::BR IF NO
7777 036510 032777 001000 142422      BIT          #BIT09,@SWR  ::LOOP ON ERROR?
7778 036516 001404                      BEQ          4$         ::BR IF NO
7779 036520 013737 001110 001106 7$:  MOV          $LPERR,$LPADR ::SET LOOP ADDRESS TO LAST SCOPE
7780 036526 000443                      BR           $OVER
7781 036530 105037 001103 4$:  CLRB        $ERFLG      ::ZERO THE ERROR FLAG
7782 036534 005037 001212              CLR         $TIMES      ::CLEAR THE NUMBER OF ITERATIONS TO MAKE
7783 036540 000415                      BR           1$         ::ESCAPE TO THE NEXT TEST
7784 036542 032777 004000 142370 3$:  BIT          #BIT11,@SWR  ::INHIBIT ITERATIONS?
7785 036550 001011                      BNE          1$         ::BR IF YES
7786 036552 005737 001100              TST         $PASS       ::IF FIRST PASS OF PROGRAM
7787 036556 001406                      BEQ          1$         ::INHIBIT ITERATIONS
7788 036560 005237 001104              INC         $ICNT       ::INCREMENT ITERATION COUNT
7789 036564 023737 001212 001104      CMP         $TIMES,$ICNT ::CHECK THE NUMBER OF ITERATIONS MADE
7790 036572 002021                      BGE         $OVER       ::BR IF MORE ITERATION REQUIRED
7791 036574 012737 000001 001104 1$:  MOV          #1,$ICNT    ::REINITIALIZE THE ITERATION COUNTER
7792 036602 013737 036652 001212      MOV          $MXCNT,$TIMES ::SET NUMBER OF ITERATIONS TO DO
7793 036610 105237 001102 3$:  $SVLAD: INCB        $TSTNM  ::COUNT TEST NUMBERS
7794 036614 011637 001106              MOV          (SP),$LPADR  ::SAVE SCOPE LOOP ADDRESS
7795 036620 011637 001110              MOV          (SP),$LPERR  ::SAVE ERROR LOOP ADDRESS
7796 036624 005037 001214              CLR         $ESCAPE     ::CLEAR THE ESCAPE FROM ERROR ADDRESS
7797 036630 112737 000001 001115      MOVB        #1,$ERMAX    ::ONLY ALLOW ONE(1) ERROR ON NEXT TEST
7798 036636 013777 001102 142276 $OVER: MOV          $TSTNM,@DISPLAY ::DISPLAY TEST NUMBER
7799 036644 013716 001106              MOV          $LPADR,(SP)  ::FUDGE RETURN ADDRESS
7800 036650 000002                      RTI                ::FIXES PS
7801 036652 000004                      $MXCNT: 4          ::MAX. NUMBER OF ITERATIONS
    
```


7802	036654	010046				MOV	R0,-(SP)	::PUSH R0 ON STACK
7803	036656	010146				MOV	R1,-(SP)	::PUSH R1 ON STACK
7804	036660	010246				MOV	R2,-(SP)	::PUSH R2 ON STACK
7805	036662	010346				MOV	R3,-(SP)	::PUSH R3 ON STACK
7806	036664	010546				MOV	R5,-(SP)	::PUSH R5 ON STACK
7807	036666	012746	020200			MOV	#20200,-(SP)	::SET BLANK SWITCH AND SIGN
7808	036672	016605	000020			MOV	20(SP),R5	::GET THE INPUT NUMBER
7809	036676	100004				BPL	1\$::BR IF INPUT IS POS.
7810	036700	005405				NEG	R5	::MAKE THE BINARY NUMBER POS.
7811	036702	112766	000055	000001		MOVB	#'-,1(SP)	::MAKE THE ASCII NUMBER NEG.
7812	036710	005000			1\$:	CLR	R0	::ZERO THE CONSTANTS INDEX
7813	036712	012703	037070			MOV	#\$DBLK,R3	::SETUP THE OUTPUT POINTER
7814	036716	112723	000040			MOVB	#' ,(R3)+	::SET THE FIRST CHARACTER TO A BLANK
7815	036722	005002			2\$:	CLR	R2	::CLEAR THE BCD NUMBER
7816	036724	016001	037060			MOV	\$DTBL(R0),R1	::GET THE CONSTANT
7817	036730	160105			3\$:	SUB	R1,R5	::FORM THIS BCD DIGIT
7818	036732	002402				BLT	4\$::BR IF DONE
7819	036734	005202				INC	R2	::INCREASE THE BCD DIGIT BY 1
7820	036736	000774				BR	3\$	
7821	036740	060105			4\$:	ADD	R1,R5	::ADD BACK THE CONSTANT
7822	036742	005702				TST	R2	::CHECK IF BCD DIGIT=0
7823	036744	001002				BNE	5\$::FALL THROUGH IF 0
7824	036746	105716				TSTB	(SP)	::STILL DOING LEADING 0'S?
7825	036750	100407				BMI	7\$::BR IF YES
7826	036752	106316			5\$:	ASLB	(SP)	::MSD?
7827	036754	103003				BCC	6\$::BR IF NO
7828	036756	116663	000001	177777		MOVB	1(SP),-1(R3)	::YES--SET THE SIGN
7829	036764	052702	000060		6\$:	BIS	#'0,R2	::MAKE THE BCD DIGIT ASCII
7830	036770	052702	000040		7\$:	BIS	#' ,R2	::MAKE IT A SPACE IF NOT ALREADY A DIGIT
7831	036774	110223				MOVB	R2,(R3)+	::PUT THIS CHARACTER IN THE OUTPUT BUFFER
7832	036776	005720				TST	(R0)+	::JUST INCREMENTING
7833	037000	020027	000010			CMP	R0,#10	::CHECK THE TABLE INDEX
7834	037004	002746				BLT	2\$::GO DO THE NEXT DIGIT
7835	037006	003002				BGT	8\$::GO TO EXIT
7836	037010	010502				MOV	R5,R2	::GET THE LSD
7837	037012	000764				BR	6\$::GO CHANGE TO ASCII
7838	037014	105726			8\$:	TSTB	(SP)+	::WAS THE LSD THE FIRST NON-ZERO?
7839	037016	100003				BPL	9\$::BR IF NO
7840	037020	116663	177777	177776		MOVB	-1(SP),-2(R3)	::YES--SET THE SIGN FOR TYPING
7841	037026	105013			9\$:	CLRB	(R3)	::SET THE TERMINATOR
7842	037030	012605				MOV	(SP)+,R5	::POP STACK INTO R5
7843	037032	012603				MOV	(SP)+,R3	::POP STACK INTO R3
7844	037034	012602				MOV	(SP)+,R2	::POP STACK INTO R2
7845	037036	012601				MOV	(SP)+,R1	::POP STACK INTO R1
7846	037040	012600				MOV	(SP)+,R0	::POP STACK INTO R0
7847	037042	104401	037070			TYPE	,\$DBLK	::NOW TYPE THE NUMBER
7848	037046	016666	000002	000004		MOV	2(SP),4(SP)	::ADJUST THE STACK
7849	037054	012616				MOV	(SP)+,(SP)	
7850	037056	000002				RTI		::RETURN TO USER
7851	037060	023420				\$DTBL:	10000.	
7852	037062	001750					1000.	
7853	037064	000144					100.	
7854	037066	000012					10.	

7855	037100	105737	001157		\$TYPE:	TSTB	\$TPFLG	::IS THERE A TERMINAL?
7856	037104	100002				BPL	1\$::BR IF YES
7857	037106	000000				HALT		::HALT HERE IF NO TERMINAL
7858	037110	000407				BR		::LEAVE
7859	037112	010046			1\$:	MOV	R0,-(SP)	::SAVE R0
7860	037114	017600	000002			MOV	@2(SP),R0	::GET ADDRESS OF ASCIZ STRING
7861	037120	112046			2\$:	MOVB	(R0)+,-(SP)	::PUSH CHARACTER TO BE TYPED ONTO STACK
7862	037122	001005				BNE	4\$::BR IF IT ISN'T THE TERMINATOR
7863	037124	005726				TST	(SP)+	::IF TERMINATOR POP IT OFF THE STACK
7864	037126	012600			60\$:	MOV	(SP)+,R0	::RESTORE R0
7865	037130	062716	000002		3\$:	ADD	#2,(SP)	::ADJUST RETURN PC
7866	037134	000002				RTI		::RETURN
7867	037136	122716	000011		4\$:	CMPB	#HT,(SP)	::BRANCH IF <HT>
7868	037142	001430				BEQ	8\$	
7869	037144	122716	000200			CMPB	#CRLF,(SP)	::BRANCH IF NOT <CRLF>
7870	037150	001006				BNE	5\$	
7871	037152	005726				TST	(SP)+	::POP <CR><LF> EQUIV
7872	037154	104401				TYPE		::TYPE A CR AND LF
7873	037156	001223				\$CRLF		
7874	037160	105037	037314			CLRB	\$CHARCNT	::CLEAR CHARACTER COUNT
7875	037164	000755				BR	2\$::GET NEXT CHARACTER
7876	037166	004737	037250		5\$:	JSR	PC,\$TYPEC	::GO TYPE THIS CHARACTER
7877	037172	123726	001156		6\$:	CMPB	\$FILLC,(SP)+	::IS IT TIME FOR FILLER CHARS.?
7878	037176	001350				BNE	2\$::IF NO GO GET NEXT CHAR.
7879	037200	013746	001154			MOV	\$NULL,-(SP)	::GET # OF FILLER CHARS. NEEDED
7880	037204	105366	000001		7\$:	DECB	1(SP)	::DOES A NULL NEED TO BE TYPED?
7881	037210	002770				BLT	6\$::BR IF NO--GO POP THE NULL OFF OF STACK
7882	037212	004737	037250			JSR	PC,\$TYPEC	::GO TYPE A NULL
7883	037216	105337	037314			DECB	\$CHARCNT	::DO NOT COUNT AS A COUNT
7884	037222	000770				BR	7\$::LOOP
7885	037224	112716	000040		8\$:	MOVB	#' ,(SP)	::REPLACE TAB WITH SPACE
7886	037230	004737	037250		9\$:	JSR	PC,\$TYPEC	::TYPE A SPACE
7887	037234	132737	000007	037314		BITB	#7,\$CHARCNT	::BRANCH IF NOT AT
7888	037242	001372				BNE	9\$::TAB STOP
7889	037244	005726				TST	(SP)+	::POP SPACE OFF STACK
7890	037246	000724				BR	2\$::GET NEXT CHARACTER
7891	037250	105777	141674		\$TYPEC:	TSTB	@\$TPS	::WAIT UNTIL PRINTER IS READY
7892	037254	100375				BPL	\$TYPEC	
7893	037256	116677	000002	141666		MOVB	2(SP),@\$TPB	::LOAD CHAR TO BE TYPED INTO DATA REG.
7894	037264	122766	000015	000002		CMPB	#CR,2(SP)	::IS CHARACTER A CARRIAGE RETURN?
7895	037272	001003				BNE	1\$::BRANCH IF NO
7896	037274	105037	037314			CLRB	\$CHARCNT	::YES--CLEAR CHARACTER COUNT
7897	037300	000406				BR	\$TYPEX	::EXIT
7898	037302	122766	000012	000002	1\$:	CMPB	#LF,2(SP)	::IS CHARACTER A LINE FEED?
7899	037310	001402				BEQ	\$TYPEX	::BRANCH IF YES
7900	037312	105227				INCB	(PC)+	::COUNT THE CHARACTER
7901	037314	000000			\$CHARCNT:	.WORD	0	::CHARACTER COUNT STORAGE
7902	037316	000207			\$TYPEX:	RTS	PC	

7903	037320	000000			\$TKCNT: .WORD	0	::NUMBER OF ITEMS IN QUEUE
7904	037322	000000			\$TKQIN: .WORD	0	::INPUT POINTER
7905	037324	000000			\$TKQOUT: .WORD	0	::OUTPUT POINTER
7906	037340	005037	037320		\$TKINT: CLR	\$TKCNT	::CLEAR COUNT OF ITEMS IN QUEUE
7907	037344	012737	037326	037322	MOV	#\$TKQSRT,\$TKQIN	::MOVE THE STARTING ADDRESS OF THE
7908	037352	013737	037322	037324	MOV	\$TKQIN,\$TKQOUT	::QUEUE INTO THE INPUT & OUTPUT POINTERS.
7909	037360	012737	037410	000060	MOV	#\$TKSRV,@#TKVEC	::INITIALIZE THE KEYBOARD VECTOR
7910	037366	012737	000200	000062	MOV	#200,@#TKVEC+2	::'BR' LEVEL 4
7911	037374	005777	141546		TST	@\$TKB	::CLEAR DONE FLAG
7912	037400	012777	000100	141536	MOV	#100,@\$TKS	::ENABLE TTY KEYBOARD INTERRUPT
7913	037406	000207			RTS	PC	::RETURN TO CALLER
7914	037410	117746	141532		\$TKSRV: MOV	@\$TKB,-(SP)	::PICKUP THE CHARACTER
7915	037414	042716	177600		BIC	#^C177,(SP)	::STRIP THE JUNK
7916	037420	021627	000003		CMP	(SP),#3	::IS IT A CONTROL C?
7917	037424	001007			BNE	1\$::BRANCH IF NO
7918	037426	104401	040377		TYPE	,\$CNTLC	::TYPE A CONTROL-C (^C)
7919	037432	004737	037340		JSR	PC,\$TKINT	::INIT THE KEYBOARD
7920	037436	005726			TST	(SP)+	::CLEAN UP STACK
7921	037440	000137	034226		JMP	OPERSEL	::CONTROL C RESTART
7922	037444	021627	000007		1\$: CMP	(SP),#7	::IS IT A CONTROL G?
7923	037450	001004			BNE	2\$::BRANCH IF NO
7924	037452	022737	000176	001140	CMP	#SWREG,SWR	::IS SOFT-SWR SELECTED?
7925	037460	001500			BEQ	6\$::GO TO SWR CHANGE
7926	037462	022737	000011	037320	CMP	#9,\$TKCNT	::IS THE QUEUE FULL?
7927	037470	001004			BNE	3\$::BRANCH IF NO
7928	037472	104401	001216		TYPE	,\$BELL	::RING THE TTY BELL
7929	037476	005726			TST	(SP)+	::CLEAN CHARACTER OFF OF STACK
7930	037500	000451			BR	5\$::EXIT
7931	037502	021627	000023		3\$: CMP	(SP),#23	::IS IT A CONTROL-S?
7932	037506	001021			BNE	32\$::BRANCH IF NO
7933	037510	005077	141430		CLR	@\$TKS	::DISABLE TTY KEYBOARD INTERRUPTS
7934	037514	005726			TST	(SP)+	::CLEAN CHAR OFF STACK
7935	037516	105777	141422		31\$: TST	@\$TKS	::WAIT FOR A CHAR
7936	037522	100375			BPL	31\$::LOOP UNTIL ITS THERE
7937	037524	117746	141416		MOV	@\$TKB,-(SP)	::GET THE CHARACTER
7938	037530	042716	177600		BIC	#^C177,(SP)	::MAKE IT 7-BIT ASCII
7939	037534	022627	000021		CMP	(SP)+,#21	::IS IT A CONTROL-Q?
7940	037540	001366			BNE	31\$::BRANCH IF NO
7941	037542	012777	000100	141374	MOV	#100,@\$TKS	::REENABLE TTY KEYBOARD INTERRUPTS
7942	037550	000002			RTI		::RETURN
7943	037552	005237	037320		32\$: INC	\$TKCNT	::COUNT THIS CHARACTER
7944	037556	021627	000140		CMP	(SP),#140	::IS IT UPPER CASE?
7945	037562	002405			BLT	4\$::BRANCH IF YES
7946	037564	021627	000175		CMP	(SP),#175	::IS IT A SPECIAL CHAR?
7947	037570	003002			BGT	4\$::BRANCH IF YES
7948	037572	042716	000040		BIC	#40,(SP)	::MAKE IT UPPER CASE
7949	037576	112677	177520		4\$: MOV	(SP)+,@\$TKQIN	::AND PUT IT IN QUEUE
7950	037602	005237	037322		INC	\$TKQIN	::UPDATE THE POINTER
7951	037606	023727	037322	037337	CMP	\$TKQIN,\$\$TKQEND	::GO OFF THE END?
7952	037614	001003			BNE	5\$::BRANCH IF NO
7953	037616	012737	037326	037322	MOV	#\$TKQSRT,\$TKQIN	::RESET THE POINTER
7954	037624	000002			5\$: RTI		::RETURN
7955	037626	022737	000176	001140	\$CKSWR: CMP	#SWREG,SWR	::IS THE SOFT-SWR SELECTED
7956	037634	001124			BNE	15\$::EXIT IF NOT
7957	037636	105777	141302		TST	@\$TKS	::IS A CHAR WAITING?
7958	037642	100121			BPL	15\$::IF NOT, EXIT

7959	037644	117746	141276			MOVB	@\$TKB,-(SP)	::YES
7960	037650	042716	177600			BIC	#^C177,(SP)	::MAKE IT 7-BIT ASCII
7961	037654	021627	000007			CMP	(SP),#7	::IS IT A CONTROL-G?
7962	037660	001300				BNE	2\$::IF NOT, PUT IT IN THE TTY QUEUE
7963	037662	123727	001134	000001	6\$:	CMPB	\$AUTOB,#1	::ARE WE RUNNING IN AUTO-MODE?
7964	037670	001674				BEQ	2\$::BRANCH IF YES
7965	037672	005726				TST	(SP)+	::CLEAR CONTROL-G OFF STACK
7966	037674	004737	037340			JSR	PC,\$TKINT	::FLUSH THE TTY INPUT QUEUE
7967	037700	005077	141240			CLR	@\$TKS	::DISABLE TTY KEYBOARD INTERRUPTS
7968	037704	112737	000001	001135		MOVB	#1,\$INTAG	::SET INTERRUPT MODE INDICATOR
7969	037712	104401	040411			TYPE	,\$CNTLG	::ECHO THE CONTROL-G (^G)
7970	037716	104401	040416			\$GTSWR: TYPE	,\$MSWR	::TYPE CURRENT CONTENTS
7971	037722	013746	000176			MOV	SWREG,-(SP)	::SAVE SWREG FOR TYPEOUT
7972	037726	104402				TYPOC		::GO TYPE--OCTAL ASCII(ALL DIGITS)
7973	037730	104401	040427			TYPE	,\$MNEW	::PROMPT FOR NEW SWR
7974	037734	005046			19\$:	CLR	-(SP)	::CLEAR COUNTER
7975	037736	005046				CLR	-(SP)	::THE NEW SWR
7976	037740	105777	141200		7\$:	TSTB	@\$TKS	::CHAR THERE?
7977	037744	100375				BPL	7\$::IF NOT TRY AGAIN
7978	037746	117746	141174			MOVB	@\$TKB,-(SP)	::PICK UP CHAR
7979	037752	042716	177600			BIC	#^C177,(SP)	::MAKE IT 7-BIT ASCII
7980	037756	021627	000003			CMP	(SP),#3	::IS IT A CONTROL-C?
7981	037762	001015				BNE	9\$::BRANCH IF NOT
7982	037764	104401	040377			TYPE	,\$CNTLC	::YES, ECHO CONTROL-C (^C)
7983	037770	062706	000006			ADD	#6,SP	::CLEAN UP STACK
7984	037774	123727	001135	000001		CMPB	\$INTAG,#1	::REENABLE TTY KEYBOARD INTERRUPTS?
7985	040002	001003				BNE	8\$::BRANCH IF NO
7986	040004	012777	000100	141132		MOV	#100,@\$TKS	::ALLOW TTY KEYBOARD INTERRUPTS
7987	040012	000137	034226		8\$:	JMP	OPERSEL	::CONTROL-C RESTART
7988	040016	021627	000025		9\$:	CMP	(SP),#25	::IS IT A CONTROL-U?
7989	040022	001005				BNE	10\$::BRANCH IF NOT
7990	040024	104401	040404			TYPE	,\$CNTLU	::YES, ECHO CONTROL-U (^U)
7991	040030	062706	000006		20\$:	ADD	#6,SP	::IGNORE PREVIOUS INPUT
7992	040034	000737				BR	19\$::LET'S TRY IT AGAIN
7993	040036	021627	000015		10\$:	CMP	(SP),#15	::IS IT A <CR>?
7994	040042	001022				BNE	16\$::BRANCH IF NO
7995	040044	005766	000004			TST	4(SP)	::YES, IS IT THE FIRST CHAR?
7996	040050	001403				BEQ	11\$::BRANCH IF YES
7997	040052	016677	000002	141060		MOV	2(SP),@SWR	::SAVE NEW SWR
7998	040060	062706	000006		11\$:	ADD	#6,SP	::CLEAR UP STACK
7999	040064	104401	001223		14\$:	TYPE	,\$CRLF	::ECHO <CR> AND <LF>
8000	040070	123727	001135	000001		CMPB	\$INTAG,#1	::RE-ENABLE TTY KBD INTERRUPTS?
8001	040076	001003				BNE	15\$::BRANCH IF NOT
8002	040100	012777	000100	141036		MOV	#100,@\$TKS	::RE-ENABLE TTY KBD INTERRUPTS
8003	040106	000002			15\$:	RTI		::RETURN
8004	040110	004737	037250		16\$:	JSR	PC,\$TYPEC	::ECHO CHAR
8005	040114	021627	000060			CMP	(SP),#60	::CHAR < 0?
8006	040120	002420				BLT	18\$::BRANCH IF YES
8007	040122	021627	000067			CMP	(SP),#67	::CHAR > 7?
8008	040126	003015				BGT	18\$::BRANCH IF YES
8009	040130	042726	000060			BIC	#60,(SP)+	::STRIP-OFF ASCII
8010	040134	005766	000002			TST	2(SP)	::IS THIS THE FIRST CHAR
8011	040140	001403				BEQ	17\$::BRANCH IF YES
8012	040142	006316				ASL	(SP)	::NO, SHIFT PRESENT
8013	040144	006316				ASL	(SP)	::CHAR OVER TO MAKE
8014	040146	006316				ASL	(SP)	::ROOM FOR NEW ONE.


```

8015 040150 005266 000002      17$: INC 2(SP)      ;;KEEP COUNT OF CHAR
8016 040154 056616 177776      BIS -2(SP),(SP)   ;;SET IN NEW CHAR
8017 040160 000667          BR 7$           ;;GET THE NEXT ONE
8018 040162 104401 001222      18$: TYPE , $QUES ;;TYPE ?<CR><LF>
8019 040166 000720          BR 20$         ;;SIMULATE CONTROL-U
8020 040170 011646          $RDCHR: MOV (SP),-(SP) ;;PUSH DOWN THE PC AND
8021 040172 016666 000004 000002 MOV 4(SP),2(SP) ;;THE PS
8022 040200 005066 000004      CLR 4(SP)      ;;GET READY FOR A CHARACTER
8023 040204 005046          CLR -(SP)     ;;PUT NEW PS ON STACK
8024 040206 012746 040214      MOV #64$,-(SP) ;;PUT NEW PC ON STACK
8025 040212 000002          RTI          ;;POP NEW PC AND PS
8026 040214 005737 037320      1$: TST $TKCNT  ;;WAIT ON A CHARACTER
8027 040220 001775          BEQ 1$        ;;
8028 040222 005337 037320      DEC $TKCNT     ;;DECREMENT THE COUNTER
8029 040226 117766 177072 000004 MOVB @ $TKQOUT,4(SP) ;;GET ONE CHARACTER
8030 040234 005237 037324      INC $TKQOUT    ;;UPDATE THE POINTER
8031 040240 023727 037324 037337 CMP $TKQOUT,#$TKQEND ;;DID IT GO OFF OF THE END?
8032 040246 001003          BNE 2$        ;;BRANCH IF NO
8033 040250 012737 037326 037324 MOV #$TKQSRT,$TKQOUT ;;RESET THE POINTER
8034 040256 000002          RTI          ;;RETURN
8035 040260 010346          $RDLIN: MOV R3,-(SP) ;;SAVE R3
8036 040262 012703 040366      1$: MOV #$TTYIN,R3 ;;GET ADDRESS
8037 040266 022703 040377      2$: CMP #$TTYIN+9.,R3 ;;BUFFER FULL?
8038 040272 101405          BLOS 4$       ;;BR IF YES
8039 040274 104410          RDCHR       ;;GO READ ONE CHARACTER FROM THE TTY
8040 040276 112613          MOVB (SP)+,(R3) ;;GET CHARACTER
8041 040300 122713 000177      10$: CMPB #177,(R3) ;;IS IT A RUBOUT
8042 040304 001003          BNE 3$       ;;SKIP IF NOT
8043 040306 104401 001222      4$: TYPE , $QUES ;;TYPE A '?'
8044 040312 000763          BR 1$        ;;CLEAR THE BUFFER AND LOOP
8045 040314 111337 040364      3$: MOVB (R3),9$ ;;ECHO THE CHARACTER
8046 040320 104401 040364      TYPE ,9$
8047 040324 122723 000015      CMPB #15,(R3)+ ;;CHECK FOR RETURN
8048 040330 001356          BNE 2$       ;;LOOP IF NOT RETURN
8049 040332 105063 177777      CLRB -1(R3)   ;;CLEAR RETURN (THE 15)
8050 040336 104401 001224      TYPE , $LF   ;;TYPE A LINE FEED
8051 040342 012603          MOV (SP)+,R3 ;;RESTORE R3
8052 040344 011646          MOV (SP),-(SP) ;;ADJUST THE STACK AND PUT ADDRESS OF THE
8053 040346 016666 000004 000002 MOV 4(SP),2(SP) ;; FIRST ASCII CHARACTER ON IT
8054 040354 012766 040366 000004 MOV #$TTYIN,4(SP)
8055 040362 000002          RTI          ;;RETURN
8056 040364 000          9$: .BYTE 0 ;;STORAGE FOR ASCII CHAR. TO TYPE
8057 040365 000          .BYTE 0 ;;TERMINATOR
8058 040377 136 006503 000012 $CNTLC: .ASCIZ /^C/<15><12> ;;CONTROL 'C'
8059 040404 052536 005015 000 $CNTLU: .ASCIZ /^U/<15><12> ;;CONTROL 'U'
8060 040411 136 006507 000012 $CNTLG: .ASCIZ /^G/<15><12> ;;CONTROL 'G'
8061 040416 005015 053523 020122 $MSWR: .ASCIZ <15><12>/SWR = /
8062 040424 020075 000
8063 040427 040 047040 053505 $MNEW: .ASCIZ / NEW = /
8064 040434 036440 000040
8065
    ;FROM THE TTY
    
```


8066	040440	011646			\$RDOCT: MOV	(SP),-(SP)	::PROVIDE SPACE FOR THE
8067	040442	016666	000004	000002	MOV	4(SP),2(SP)	::INPUT NUMBER
8068	040450	010046			MOV	R0,-(SP)	::PUSH R0 ON STACK
8069	040452	010146			MOV	R1,-(SP)	::PUSH R1 ON STACK
8070	040454	010246			MOV	R2,-(SP)	::PUSH R2 ON STACK
8071	040456	104411			1\$: RDLIN		::READ AN ASCII LINE
8072	040460	012600			MOV	(SP)+,R0	::GET ADDRESS OF 1ST CHARACTER
8073	040462	010037	040566		MOV	R0,5\$::AND SAVE IT
8074	040466	005001			CLR	R1	::CLEAR DATA WORD
8075	040470	005002			CLR	R2	
8076	040472	112046			2\$: MOVB	(R0)+,-(SP)	::PICKUP THIS CHARACTER
8077	040474	001420			BEQ	3\$::IF ZERO GET OUT
8078	040476	122716	000060		CMPB	#'0,(SP)	::MAKE SURE THIS CHARACTER
8079	040502	003026			BGT	4\$::IS AN OCTAL DIGIT
8080	040504	122716	000067		CMPB	#'7,(SP)	
8081	040510	002423			BLT	4\$	
8082	040512	006301			ASL	R1	::*2
8083	040514	006102			ROL	R2	
8084	040516	006301			ASL	R1	::*4
8085	040520	006102			ROL	R2	
8086	040522	006301			ASL	R1	::*8
8087	040524	006102			ROL	R2	
8088	040526	042716	177770		BIC	#^C7,(SP)	::STRIP THE ASCII JUNK
8089	040532	062601			ADD	(SP)+,R1	::ADD IN THIS DIGIT
8090	040534	000756			BR	2\$::LOOP
8091	040536	005726			3\$: TST	(SP)+	::CLEAN TERMINATOR FROM STACK
8092	040540	010166	000012		MOV	R1,12(SP)	::SAVE THE RESULT
8093	040544	010237	040576		MOV	R2,\$HIOCT	
8094	040550	012602			MOV	(SP)+,R2	::POP STACK INTO R2
8095	040552	012601			MOV	(SP)+,R1	::POP STACK INTO R1
8096	040554	012600			MOV	(SP)+,R0	::POP STACK INTO R0
8097	040556	000002			RTI		::RETURN
8098	040560	005726			4\$: TST	(SP)+	::CLEAN PARTIAL FROM STACK
8099	040562	105010			CLRB	(R0)	::SET A TERMINATOR
8100	040564	104401			TYPE		::TYPE UP THRU THE BAD CHAR.
8101	040566	000000			5\$: .WORD	0	
8102	040570	104401	001222		TYPE	,\$QUES	::''?' 'CR' & 'LF''
8103	040574	000730			BR	1\$::TRY AGAIN
8104	040576	000000			\$HIOCT: .WORD	0	::HIGH ORDER BITS GO HERE


```

8105 040600 104407          CKSWR          ;;TEST FOR CHANGE IN SOFT-SWR
8106 040602 012737 177777 004632      MOV      #-1,@#ERFLG$  ;;SET ERROR FLAG
8107 040610 105237 001103          INCB     $ERFLG        ;;SET THE ERROR FLAG
8108 040614 001775          BEQ      7$           ;;DON'T LET THE FLAG GO TO ZERO
8109 040616 013777 001102 140316      MOV      $STNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
8110 040624 032777 002000 140306      BIT      #BIT10,@SWR   ;;BELL ON ERROR?
8111 040632 001402          BEQ      1$           ;;NO - SKIP
8112 040634 104401 001216          TYPE    $BELL         ;;RING BELL
8113 040640 005237 001112          INC      $ERTTL       ;;COUNT THE NUMBER OF ERRORS
8114 040644 011637 001116          MOV      (SP),$ERRPC  ;;GET ADDRESS OF ERROR INSTRUCTION
8115 040650 162737 000002 001116      SUB      #2,$ERRPC
8116 040656 117737 140234 001114      MOVB    @$ERRPC,$ITEMB ;;STRIP AND SAVE THE ERROR ITEM CODE
8117 040664 032777 020000 140246      BIT      #BIT13,@SWR  ;;SKIP TYPEOUT IF SET
8118 040672 001004          BNE     20$          ;;SKIP TYPEOUTS
8119 040674 004737 040750          JSR     PC,$ERRRYP   ;;GO TO USER ERROR ROUTINE
8120 040700 104401 001223          TYPE    $CRLF
8121 040704 005777 140230          2$:     TST      @SWR      ;;HALT ON ERROR
8122 040710 100002          BPL     3$           ;;SKIP IF CONTINUE
8123 040712 000000          HALT
8124 040714 104407          CKSWR          ;;TEST FOR CHANGE IN SOFT-SWR
8125 040716 032777 001000 140214      3$:     BIT      #BIT09,@SWR ;;LOOP ON ERROR SWITCH SET?
8126 040724 001402          BEQ     4$           ;;BR IF NO
8127 040726 013716 001110          MOV     $LPERR,(SP)  ;;FUDGE RETURN FOR LOOPING
8128 040732 005737 001214          4$:     TST      $ESCAPE ;;CHECK FOR AN ESCAPE ADDRESS
8129 040736 001402          BEQ     5$           ;;BR IF NONE
8130 040740 013716 001214          MOV     $ESCAPE,(SP) ;;FUDGE RETURN ADDRESS FOR ESCAPE
8131 040744 000002          RTI
  
```


8132
 8133
 8134
 8135
 8136
 8137
 8138
 8139
 8140
 8141
 8142
 8143
 8144
 8145
 8146
 8147
 8148 040746 000000
 8149
 8150 040750 017746 140164
 8151 040754 042716 177277
 8152 040760 022726 000100
 8153 040764 001001
 8154 040766 000402
 8155 040770 000137 041710
 8156
 8157 040774
 8158 040774 104401 041002
 8159 041000 000406
 8160 041016 013746 002254
 8161 041022 104402
 8162 041024 104401 041032
 8163 041030 000406
 8164 041046 013746 002256
 8165 041052 104402
 8166 041054 104401 041062
 8167 041060 000406
 8168 041076 013746 002260
 8169 041102 104402
 8170 041104 104401 041112
 8171 041110 000406
 8172 041126 013746 002262
 8173 041132 104402
 8174 041134 104401 041142
 8175 041140 000406
 8176 041156 013746 002304
 8177 041162 104402
 8178 041164 104401 041172
 8179 041170 000406
 8180 041206 013746 002264
 8181 041212 104402
 8182 041214 104401 041222
 8183 041220 000406
 8184 041236 013746 002270
 8185 041242 104402
 8186 041244 104401 041252
 8187 041250 000406

::*****

.SBTTL ERROR MESSAGE TYPEOUT ROUTINE

;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" (\$ITEMB) TO DETERMINE WHICH
 ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" (\$ERRTB),
 ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
 ;*IT IS A COPY OF THE \$ERRTYP SUBROUTINE FROM SYSMAC.
 ;*WITH ONLY MINOR CHANGES
 ;*FIRST IF SWITCH 6 IS SET AND SWITCH 8 RESET THEN
 ;*ALL REGISTER CONTENTS WILL BE TYPED BEFOR REPORTING THE ERROR
 ;*SECOND IF THE CURRENT ERROR HAS THE SAME ITEM NUMBER
 ;*AS THE PREVIOUS ERROR THEN ONLY THE DATA WILL BE TYPED
 ;*AND NOT THE ERROR MESSAGE AND HEADER.

PRITEM: 0 ;PREVIOUS ITEM NO. LOCATION

```

$ERRTYP: MOV @SWR,-(SP) ;GET SWITCH SETTING
          BIC #^C500,(SP) ;KEEP ONLY SWITCH 8 AND 6
          CMP #SW06,(SP)+ ;IS 6 SET AND 8 RESET
          BNE 1$ ;IF NOT BRANCH
          BR 2$ ;BRANCH IF SW 6 IS SET AND 8 RESET
1$: JMP @#TYPERR ;JUMP IF SW 8 IS SET
          ;OR IF SW 8 IS RESET AND SW 6 IS RESET
2$:
          TYPE ,65$ ;:TYPE ASCIZ STRING
          BR 64$ ;:GET OVER THE ASCIZ
          MOV @#WC,-(SP) ;GET READY TO TYPE RHWC CONTENTS
          TYPOC
          TYPE ,67$ ;:TYPE ASCIZ STRING
          BR 66$ ;:GET OVER THE ASCIZ
          MOV @#BA,-(SP) ;GET READY TO TYPE RHBA CONTENTS
          TYPOC
          TYPE ,69$ ;:TYPE ASCIZ STRING
          BR 68$ ;:GET OVER THE ASCIZ
          MOV @#CS2,-(SP) ;GET READY TO TYPE RHCS2 CONTENTS
          TYPOC
          TYPE ,71$ ;:TYPE ASCIZ STRING
          BR 70$ ;:GET OVER THE ASCIZ
          MOV @#CS1,-(SP) ;GET READY TO TYPE RHCS1 CONTENTS
          TYPOC
          TYPE ,73$ ;:TYPE ASCIZ STRING
          BR 72$ ;:GET OVER THE ASCIZ
          MOV @#DS1,-(SP) ;GET READY TO TYPE RHDS1 CONTENTS
          TYPOC
          TYPE ,75$ ;:TYPE ASCIZ STRING
          BR 74$ ;:GET OVER THE ASCIZ
          MOV @#ER1,-(SP) ;GET READY TO TYPE RHER1 CONTENTS
          TYPOC
          TYPE ,77$ ;:TYPE ASCIZ STRING
          BR 76$ ;:GET OVER THE ASCIZ
          MOV @#ER2,-(SP) ;GET READY TO TYPE RHER2 CONTENTS
          TYPOC
          TYPE ,79$ ;:TYPE ASCIZ STRING
          BR 78$ ;:GET OVER THE ASCIZ
    
```


8188	041266	013746	002276	MOV	@#ER3,-(SP)	:GET READY TO TYPE RHER3 CONTENTS
8189	041272	104402		TYPOC		
8190	041274	104401	041302	TYPE	,81\$::TYPE ASCIZ STRING
8191	041300	000406		BR	80\$::GET OVER THE ASCIZ
8192	041316	013746	002266	MOV	@#DST,-(SP)	:GET READY TO TYPE RHDST CONTENTS
8193	041322	104402		TYPOC		
8194	041324	104401	041332	TYPE	,83\$::TYPE ASCIZ STRING
8195	041330	000406		BR	82\$::GET OVER THE ASCIZ
8196	041346	013746	002274	MOV	@#CA,-(SP)	:GET READY TO TYPE RHCA CONTENTS
8197	041352	104402		TYPOC		
8198	041354	104401	041362	TYPE	,85\$::TYPE ASCIZ STRING
8199	041360	000406		BR	84\$::GET OVER THE ASCIZ
8200	041376	013746	002300	MOV	@#AS,-(SP)	:GET READY TO TYPE RHAS CONTENTS
8201	041402	104402		TYPOC		
8202	041404	104401	041412	TYPE	,87\$::TYPE ASCIZ STRING
8203	041410	000406		BR	86\$::GET OVER THE ASCIZ
8204	041426	013746	002272	MOV	@#OF,-(SP)	:GET READY TO TYPE RHOF CONTENTS
8205	041432	104402		TYPOC		
8206	041434	104401	041442	TYPE	,89\$::TYPE ASCIZ STRING
8207	041440	000406		BR	88\$::GET OVER THE ASCIZ
8208	041456	013746	002302	MOV	@#MR,-(SP)	:GET READY TO TYPE RHMR CONTENTS
8209	041462	104402		TYPOC		
8210	041464	104401	041472	TYPE	,91\$::TYPE ASCIZ STRING
8211	041470	000406		BR	90\$::GET OVER THE ASCIZ
8212	041506	013746	002320	MOV	@#LA,-(SP)	:GET READY TO TYPE RHLA CONTENTS
8213	041512	104402		TYPOC		
8214	041514	104401	041522	TYPE	,93\$::TYPE ASCIZ STRING
8215	041520	000406		BR	92\$::GET OVER THE ASCIZ
8216	041536	013746	002316	MOV	@#CC,-(SP)	:GET READY TO TYPE RHCC CONTENTS
8217	041542	104402		TYPOC		
8218	041544	104401	041552	TYPE	,95\$::TYPE ASCIZ STRING
8219	041550	000406		BR	94\$::GET OVER THE ASCIZ
8220	041566	013746	002312	MOV	@#EC1,-(SP)	:GET READY TO TYPE RHEC1 CONTENTS
8221	041572	104402		TYPOC		
8222	041574	104401	041602	TYPE	,97\$::TYPE ASCIZ STRING
8223	041600	000406		BR	96\$::GET OVER THE ASCIZ
8224	041616	013746	002314	MOV	@#EC2,-(SP)	:GET READY TO TYPE RHEC2 CONTENTS
8225	041622	104402		TYPOC		
8226	041624	104401	041632	TYPE	,99\$::TYPE ASCIZ STRING
8227	041630	000406		BR	98\$::GET OVER THE ASCIZ
8228	041646	013746	002306	MOV	@#DT,-(SP)	:GET READY TO TYPE RHDT CONTENTS
8229	041652	104402		TYPOC		
8230	041654	104401	041662	TYPE	,101\$::TYPE ASCIZ STRING
8231	041660	000406		BR	100\$::GET OVER THE ASCIZ
8232	041676	013746	002310	MOV	@#SN,-(SP)	:GET READY TO TYPE RHSN CONTENTS
8233	041702	104402		TYPOC		
8234	041704	005037	040746	CLR	@#PRITEM	:CLEAR PREVIOUS ERROR ITEM
8235						
8236	041710			TYPERR:		
8237	041710	104401	001223	TYPE	,\$CRLF	::'CARRIAGE RETURN' & 'LINE FEED'
8238	041714	010046		MOV	RO,-(SP)	:SAVE RO
8239	041716	005000		CLR	RO	:PICKUP THE ITEM INDEX
8240	041720	153700	001114	BISB	@#\$ITEMB,RO	
8241	041724	001004		BNE	1\$:IF ITEM NUMBER IS ZERO, JUST
8242						:TYPE THE PC OF THE ERROR
8243	041726	013746	001116	MOV	\$ERRPC,-(SP)	:SAVE \$ERRPC FOR TYPEOUT


```

8289
8290 042076 017646 000000          $TYPOS: MOV    @ (SP), -(SP)      :: PICKUP THE MODE
8291 042102 116637 000001 042321  MOVB   1 (SP), $OFILL    :: LOAD ZERO FILL SWITCH
8292 042110 112637 042323          MOVB   (SP)+, $OMODE+1  :: NUMBER OF DIGITS TO TYPE
8293 042114 062716 000002          ADD    #2, (SP)        :: ADJUST RETURN ADDRESS
8294 042120 000406                    BR     $TYPON
8295 042122 112737 000001 042321  $TYPOC: MOVB   #1, $OFILL    :: SET THE ZERO FILL SWITCH
8296 042130 112737 000006 042323  MOVB   #6, $OMODE+1    :: SET FOR SIX(6) DIGITS
8297 042136 112737 000005 042320  $TYPON: MOVB   #5, $OCNT    :: SET THE ITERATION COUNT
8298 042144 010346                    MOV    R3, -(SP)       :: SAVE R3
8299 042146 010446                    MOV    R4, -(SP)       :: SAVE R4
8300 042150 010546                    MOV    R5, -(SP)       :: SAVE R5
8301 042152 113704 042323          MOVB   $OMODE+1, R4    :: GET THE NUMBER OF DIGITS TO TYPE
8302 042156 005404                    NEG    R4
8303 042160 062704 000006          ADD    #6, R4          :: SUBTRACT IT FOR MAX. ALLOWED
8304 042164 110437 042322          MOVB   R4, $OMODE      :: SAVE IT FOR USE
8305 042170 113704 042321          MOVB   $OFILL, R4     :: GET THE ZERO FILL SWITCH
8306 042174 016605 000012          MOV    12(SP), R5     :: PICKUP THE INPUT NUMBER
8307 042200 005003                    CLR    R3
8308 042202 006105                    1$:   ROL    R5        :: CLEAR THE OUTPUT WORD
8309 042204 000404                    BR     3$             :: ROTATE MSB INTO 'C'
8310 042206 006105                    2$:   ROL    R5        :: GO DO MSB
8311 042210 006105                    ROL    R5             :: FORM THIS DIGIT
8312 042212 006105                    ROL    R5
8313 042214 010503                    MOV    R5, R3
8314 042216 006103                    3$:   ROL    R3        :: GET LSB OF THIS DIGIT
8315 042220 105337 042322          DECB   $OMODE         :: TYPE THIS DIGIT?
8316 042224 100016                    BPL    7$             :: BR IF NO
8317 042226 042703 177770          BIC    #177770, R3    :: GET RID OF JUNK
8318 042232 001002                    BNE    4$             :: TEST FOR 0
8319 042234 005704                    TST    R4             :: SUPPRESS THIS 0?
8320 042236 001403                    BEQ    5$             :: BR IF YES
8321 042240 005204                    4$:   INC    R4        :: DON'T SUPPRESS ANYMORE 0'S
8322 042242 052703 000060          BIS    #'0, R3        :: MAKE THIS DIGIT ASCII
8323 042246 052703 000040          5$:   BIS    #' , R3   :: MAKE ASCII IF NOT ALREADY
8324 042252 110337 042316          MOVB   R3, 8$        :: SAVE FOR TYPING
8325 042256 104401 042316          TYPE   , 8$         :: GO TYPE THIS DIGIT
8326 042262 105337 042320          7$:   DECB   $OCNT    :: COUNT BY 1
8327 042266 003347                    BGT    2$             :: BR IF MORE TO DO
8328 042270 002402                    BLT    6$             :: BR IF DONE
8329 042272 005204                    INC    R4             :: INSURE LAST DIGIT ISN'T A BLANK
8330 042274 000744                    BR     2$             :: GO DO THE LAST DIGIT
8331 042276 012605                    6$:   MOV    (SP)+, R5  :: RESTORE R5
8332 042300 012604                    MOV    (SP)+, R4     :: RESTORE R4
8333 042302 012603                    MOV    (SP)+, R3     :: RESTORE R3
8334 042304 016666 000002 000004  MOV    2(SP), 4(SP)  :: SET THE STACK FOR RETURNING
8335 042312 012616                    MOV    (SP)+, (SP)
8336 042314 000002                    RTI
8337 042316 000          8$:   .BYTE  0          :: RETURN
8338 042317 000          .BYTE  0          :: STORAGE FOR ASCII DIGIT
8339 042320 000          $OCNT: .BYTE  0          :: TERMINATOR FOR TYPE ROUTINE
8340 042321 000          $OFILL: .BYTE  0          :: OCTAL DIGIT COUNTER
8341 042322 000000          $OMODE: .WORD  0          :: ZERO FILL SWITCH
          :: NUMBER OF DIGITS TO TYPE
  
```


8342	042324	010046			\$TRAP: MOV R0, -(SP)	::SAVE R0	
8343	042326	016600	000002		MOV 2(SP), R0	::GET TRAP ADDRESS	
8344	042332	005740			TST -(R0)	::BACKUP BY 2	
8345	042334	111000			MOVB (R0), R0	::GET RIGHT BYTE OF TRAP	
8346	042336	006300			ASL R0	::POSITION FOR INDEXING	
8347	042340	016000	042360		MOV \$TRPAD(R0), R0	::INDEX TO TABLE	
8348	042344	000200			RTS R0	::GO TO ROUTINE	
8349	042346	011646			\$TRAP2: MOV (SP), -(SP)	::MOVE THE PC DOWN	
8350	042350	016666	000004	000002	MOV 4(SP), 2(SP)	::MOVE THE PSW DOWN	
8351	042356	000002			RTI	::RESTORE THE PSW	
8352	042360	042346			\$TRPAD: .WORD \$TRAP2		
8353	042362	037100			\$TYPE ::CALL=TYPE	TRAP+1(104401)	TTY TYPEOUT ROUTINE
8354	042364	042122			\$TYPOC ::CALL=TYPOC	TRAP+2(104402)	TYPE OCTAL NUMBER (WITH LEADING ZEROS)
8355	042366	042076			\$TYPOS ::CALL=TYPOS	TRAP+3(104403)	TYPE OCTAL NUMBER (NO LEADING ZEROS)
8356	042370	042136			\$TYPON ::CALL=TYPON	TRAP+4(104404)	TYPE OCTAL NUMBER (AS PER LAST CALL)
8357	042372	036654			\$TYPDS ::CALL=TYPDS	TRAP+5(104405)	TYPE DECIMAL NUMBER (WITH SIGN)
8358	042374	037716			\$GTSWR ::CALL=GTSWR	TRAP+6(104406)	GET SOFT-SWR SETTING
8359	042376	037626			\$CKSWR ::CALL=CKSWR	TRAP+7(104407)	TEST FOR CHANGE IN SOFT-SWR
8360	042400	040170			\$RDCHR ::CALL=RDCHR	TRAP+10(104410)	TTY TYPEIN CHARACTER ROUTINE
8361	042402	040260			\$RDLIN ::CALL=RDLIN	TRAP+11(104411)	TTY TYPEIN STRING ROUTINE
8362	042404	040440			\$RDOCT ::CALL=RDOCT	TRAP+12(104412)	READ AN OCTAL NUMBER FROM TTY
8363	042406	033574			WAIT.T ::CALL=WAT	TRAP+13(104413)	DONT ADD ABOVE THIS TRAP


```

8364 042410 012737 042554 000024 $PWRDN: MOV    #$ILLUP,@#PWRVEC  ;;SET FOR FAST UP
8365 042416 012737 000340 000026      MOV    #340,@#PWRVEC+2  ;;PRIO:7
8366 042424 010046      MOV    R0,-(SP)        ;;PUSH R0 ON STACK
8367 042426 010146      MOV    R1,-(SP)        ;;PUSH R1 ON STACK
8368 042430 010246      MOV    R2,-(SP)        ;;PUSH R2 ON STACK
8369 042432 010346      MOV    R3,-(SP)        ;;PUSH R3 ON STACK
8370 042434 010446      MOV    R4,-(SP)        ;;PUSH R4 ON STACK
8371 042436 010546      MOV    R5,-(SP)        ;;PUSH R5 ON STACK
8372 042440 017746 136474      MOV    @SWR,-(SP)     ;;PUSH @SWR ON STACK
8373 042444 010637 042560      MOV    SP,$SAVR6     ;;SAVE SP
8374 042450 012737 042462 000024      MOV    #$PWRUP,@#PWRVEC  ;;SET UP VECTOR
8375 042456 000000      HALT
8376 042460 000776      BR    .-2            ;;HANG UP
8377 042462 012737 042554 000024 $PWRUP: MOV    #$ILLUP,@#PWRVEC  ;;SET FOR FAST DOWN
8378 042470 013706 042560      MOV    $SAVR6,SP     ;;GET SP
8379 042474 005037 042560      CLR    $SAVR6        ;;WAIT LOOP FOR THE TTY
8380 042500 005237 042560      1$:  INC    $SAVR6     ;;WAIT FOR THE INC
8381 042504 001375      BNE    1$            ;;OF WORD
8382 042506 012677 136426      MOV    (SP)+,@SWR    ;;POP STACK INTO @SWR
8383 042512 012605      MOV    (SP)+,R5      ;;POP STACK INTO R5
8384 042514 012604      MOV    (SP)+,R4      ;;POP STACK INTO R4
8385 042516 012603      MOV    (SP)+,R3      ;;POP STACK INTO R3
8386 042520 012602      MOV    (SP)+,R2      ;;POP STACK INTO R2
8387 042522 012601      MOV    (SP)+,R1      ;;POP STACK INTO R1
8388 042524 012600      MOV    (SP)+,R0      ;;POP STACK INTO R0
8389 042526 012737 042410 000024      MOV    #$PWRDN,@#PWRVEC  ;;SET UP THE POWER DOWN VECTOR
8390 042534 012737 000340 000026      MOV    #340,@#PWRVEC+2  ;;PRIO:7
8391 042542 104401      TYPE    $POWER        ;;REPORT THE POWER FAILURE
8392 042544 042562      $PWRMG: .WORD $POWER  ;;POWER FAIL MESSAGE POINTER
8393 042546 012716      MOV    (PC)+,(SP)    ;;RESTART AT BEGIN
8394 042550 004710      $PWRAD: .WORD BEGIN  ;;RESTART ADDRESS
8395 042552 000002      RTI
8396 042554 000000      $ILLUP: HALT        ;;THE POWER UP SEQUENCE WAS STARTED
8397 042556 000776      BR    .-2            ;; BEFORE THE POWER DOWN WAS COMPLETE
8398 042560 000000      $SAVR6: 0           ;;PUT THE SP HERE
8399 042562 005015 047520 042527 $POWER: .ASCIZ <15><12>'POWER''
8400 042570 000122
8401
    
```


8402
8403
8404
8405
8406
8407
8408
8409
8410
8411
8412
8413
8414
8415
8416
8417
8418
8419
8420
8421
8422
8423
8424
8425
8426
8427
8428
8429
8430
8431
8432
8433
8434
8435
8436
8437
8438
8439
8440
8441
8442
8443
8444
8445
8446
8447
8448
8449
8450
8451
8452
8453
8454
8455
8456
8457

042572 050122 032060 042040
042600 042111 047040 052117
042606 044440 052116 051105
042614 052522 052120 000
042621 111 052116 051105
042626 052522 052120 042440
042634 040516 046102 020105
042642 044502 020124 047504
042650 047127 041040 052125
042656 042440 050130 041505
042664 042524 020104 044502
042672 020124 044504 020104
042700 047516 020124 042523
042706 000124
042710 050122 032060 042040
042716 042111 047040 052117
042724 044440 052116 051105
042732 052522 052120 053440
042740 042510 020116 054105
042746 042520 052103 042105
042754 041040 052111 042040
042762 042111 051440 052105
042770 000
042771 105 050130 041505
042776 042524 020104 044502
043004 020124 044504 020104
043012 042523 020124 052502
043020 020124 044524 042515
043026 044440 020123 047111
043034 042440 051122 051117
043042 026440 052040 046511
043050 020105 047111 030440
043056 020060 044515 051103
043064 051517 041505 020056
043072 042504 044503 040515
043100 000114
043102 044122 051501 042040
043110 042517 020123 047516
043116 020124 046103 040505
043124 020122 054502 046440
043132 053117 047111 020107
043140 047111 040440 046114
043146 047440 042516 000123
043154 047514 042101 047111
043162 020107 044122 051105
043170 020061 047506 020122

: ERROR AND MESSAGE TABLE CONDIMENTS
: *****

EM1: .ASCIZ /RP04 DID NOT INTERRUPT/

EM2: .ASCIZ /INTERRUPT ENABLE BIT DOWN BUT EXPECTED BIT DID NOT SET/

EM3: .ASCIZ /RP04 DID NOT INTERRUPT WHEN EXPECTED BIT DID SET/

EM4: .ASCIZ /EXPECTED BIT DID SET BUT TIME IS IN ERROR - TIME IN 10 MICROSEC. DECIMA

EM5: .ASCIZ /RHAS DOES NOT CLEAR BY MOVING IN ALL ONES/

EM6: .ASCIZ /LOADING RHER1 FOR ALL UNITS DID NOT SET ANY RHAS BITS/

8458	043176	046101	020114	047125	
8459	043204	052111	020123	044504	
8460	043212	020104	047516	020124	
8461	043220	042523	020124	047101	
8462	043226	020131	044122	051501	
8463	043234	041040	052111	000123	
8464	043242	047516	020116	054105	EM7: .ASCIZ /NON EXISTENT REGISTER, PROGRAM ABORTED./
8465	043250	051511	042524	052116	
8466	043256	051040	043505	051511	
8467	043264	042524	026122	050040	
8468	043272	047522	051107	046501	
8469	043300	040440	047502	052122	
8470	043306	042105	000056		
8471	043312	052123	050117	042520	EM10: .ASCIZ /STOPPED DRIVE HAS MOL BIT IN RHDS1 SET/
8472	043320	020104	051104	053111	
8473	043326	020105	040510	020123	
8474	043334	047515	020114	044502	
8475	043342	020124	047111	051040	
8476	043350	042110	030523	051440	
8477	043356	052105	000		
8478					
8479	043361	127	052111	020110	EM11: .ASCIZ /WITH SPINDLE POWERED DOWN RHCS2 SHOULD ONLY HAVE UNIT NO: AND IR SET/
8480	043366	050123	047111	046104	
8481	043374	020105	047520	042527	
8482	043402	042522	020104	047504	
8483	043410	047127	051040	041510	
8484	043416	031123	051440	047510	
8485	043424	046125	020104	047117	
8486	043432	054514	044040	053101	
8487	043440	020105	047125	052111	
8488	043446	047040	035117	040440	
8489	043454	042116	044440	020122	
8490	043462	042523	000124		
8491	043466	043101	042524	020122	EM12: .ASCIZ /AFTER SPINDLE POWERED UP, NO PACK ACKN. RHDS1 SHOULD HAVE MOL=1, VV=0/
8492	043474	050123	047111	046104	
8493	043502	020105	047520	042527	
8494	043510	042522	020104	050125	
8495	043516	020054	047516	050040	
8496	043524	041501	020113	041501	
8497	043532	047113	020056	044122	
8498	043540	051504	020061	044123	
8499	043546	052517	042114	044040	
8500	043554	053101	020105	047515	
8501	043562	036514	026061	053040	
8502	043570	036526	000060		
8503	043574	044527	044124	051440	EM13: .ASCIZ /WITH SPINDLE POWERED, NO INTIALIZE, RHCS1 SHOULD HAVE GO=0, DVA=1, RDY=
8504	043602	044520	042116	042514	
8505	043610	050040	053517	051105	
8506	043616	042105	020054	047516	
8507	043624	044440	052116	040511	
8508	043632	044514	042532	020054	
8509	043640	044122	051503	020061	
8510	043646	044123	052517	042114	
8511	043654	044040	053101	020105	
8512	043662	047507	030075	020054	
8513	043670	053104	036501	026061	

8514	043676	051040	054504	030475	
8515	043704	020054	042511	030075	
8516	043712	000			
8517	043713	101	052106	051105	EM14: .ASCIZ /AFTER SPINDLE POWERED UP RHCC SHOULD BE=0/
8518	043720	051440	044520	042116	
8519	043726	042514	050040	053517	
8520	043734	051105	042105	052440	
8521	043742	020120	044122	041503	
8522	043750	051440	047510	046125	
8523	043756	020104	042502	030075	
8524	043764	000			
8525	043765	120	041501	020113	EM15: .ASCII /PACK ACKNOWLEDGE COMMAND CAUSED AN ERROR/<15><12>
8526	043772	041501	047113	053517	
8527	044000	042514	043504	020105	
8528	044006	047503	046515	047101	
8529	044014	020104	040503	051525	
8530	044022	042105	040440	020116	
8531	044030	051105	047522	006522	
8532	044036	012			
8533	044037	107	047517	020104	.ASCIZ /GOOD DATA IS BEFORE COMMAND, REC DATA IS AFTER COMMAND/
8534	044044	040504	040524	044440	
8535	044052	020123	042502	047506	
8536	044060	042522	041440	046517	
8537	044066	040515	042116	020054	
8538	044074	042522	020103	040504	
8539	044102	040524	044440	020123	
8540	044110	043101	042524	020122	
8541	044116	047503	046515	047101	
8542	044124	000104			
8543	044126	047516	047455	020120	EM16: .ASCII /NO-OP COMMAND CAUSED AN ERROR/<15><12>
8544	044134	047503	046515	047101	
8545	044142	020104	040503	051525	
8546	044150	042105	040440	020116	
8547	044156	051105	047522	006522	
8548	044164	012			
8549	044165	107	047517	020104	.ASCIZ /GOOD DATA IS BEFORE COMMAND, REC DATA IS AFTER COMMAND/
8550	044172	040504	040524	044440	
8551	044200	020123	042502	047506	
8552	044206	042522	041440	046517	
8553	044214	040515	042116	020054	
8554	044222	042522	020103	040504	
8555	044230	040524	044440	020123	
8556	044236	043101	042524	020122	
8557	044244	047503	046515	047101	
8558	044252	000104			
8559	044254	051104	053111	020105	EM17: .ASCII /DRIVE CLEAR COMMAND CAUSED AN ERROR/<15><12>
8560	044262	046103	040505	020122	
8561	044270	047503	046515	047101	
8562	044276	020104	040503	051525	
8563	044304	042105	040440	020116	
8564	044312	051105	047522	006522	
8565	044320	012			
8566	044321	107	047517	020104	.ASCIZ /GOOD DATA GIVES SHOULD BE, REC DATA GIVES AFTER COMMAND/
8567	044326	040504	040524	043440	
8568	044334	053111	051505	051440	
8569	044342	047510	046125	020104	

8570	044350	042502	020054	042522	
8571	044356	020103	040504	040524	
8572	044364	043440	053111	051505	
8573	044372	040440	052106	051105	
8574	044400	041440	046517	040515	
8575	044406	042116	000		
8576	044411	122	040505	026504	EM20: .ASCII /READ-IN COMMAND CAUSED AN ERROR/<15><12>
8577	044416	047111	041440	046517	
8578	044424	040515	042116	041440	
8579	044432	052501	042523	020104	
8580	044440	047101	042440	051122	
8581	044446	051117	005015		
8582	044452	047507	042117	042040	.ASCIZ /GOOD DATA GIVES SHOULD BE, REC DATA GIVES REG. CONTENTS AFTER COMMAND/
8583	044460	052101	020101	044507	
8584	044466	042526	020123	044123	
8585	044474	052517	042114	041040	
8586	044502	026105	051040	041505	
8587	044510	042040	052101	020101	
8588	044516	044507	042526	020123	
8589	044524	042522	027107	041440	
8590	044532	047117	042524	052116	
8591	044540	020123	043101	042524	
8592	044546	020122	047503	046515	
8593	044554	047101	000104		
8594					
8595	044560	044122	051503	020061	EM21: .ASCIZ /RHCS1 CONTENTS DURING COMMAND WAS IN ERROR/
8596	044566	047503	052116	047105	
8597	044574	051524	042040	051125	
8598	044602	047111	020107	047503	
8599	044610	046515	047101	020104	
8600	044616	040527	020123	047111	
8601	044624	042440	051122	051117	
8602	044632	000			
8603	044633	122	042110	030523	EM22: .ASCIZ /RHDS1 CONTENTS DURING COMMAND WAS IN ERROR/
8604	044640	041440	047117	042524	
8605	044646	052116	020123	052504	
8606	044654	044522	043516	041440	
8607	044662	046517	040515	042116	
8608	044670	053440	051501	044440	
8609	044676	020116	051105	047522	
8610	044704	000122			
8611	044706	047125	047514	042101	EM23: .ASCII /UNLOAD COMMAND CAUSED AN ERROR/<15><12>
8612	044714	041440	046517	040515	
8613	044722	042116	041440	052501	
8614	044730	042523	020104	047101	
8615	044736	042440	051122	051117	
8616	044744	005015			
8617	044746	047507	042117	042040	.ASCIZ /GOOD DATA GIVES SHOULD BE, REC DATA GIVES REGISTER CONT. AFTER COMMAND/
8618	044754	052101	020101	044507	
8619	044762	042526	020123	044123	
8620	044770	052517	042114	041040	
8621	044776	026105	051040	041505	
8622	045004	042040	052101	020101	
8623	045012	044507	042526	020123	
8624	045020	042522	044507	052123	
8625	045026	051105	041440	047117	

8626	045034	027124	040440	052106	
8627	045042	051105	041440	046517	
8628	045050	040515	042116	000	
8629	045055	117	043106	042523	EM24: .ASCII /OFFSET COMMAND CAUSED AN ERROR/<15><12>
8630	045062	020124	047503	046515	
8631	045070	047101	020104	040503	
8632	045076	051525	042105	040440	
8633	045104	020116	051105	047522	
8634	045112	006522	012		
8635	045115	107	047517	020104	.ASCIZ /GOOD DATA GIVES SHOULD BE, REC DATA GIVES REG. CONT. AFTER COMMAND/
8636	045122	040504	040524	043440	
8637	045130	053111	051505	051440	
8638	045136	047510	046125	020104	
8639	045144	042502	020054	042522	
8640	045152	020103	040504	040524	
8641	045160	043440	053111	051505	
8642	045166	051040	043505	020056	
8643	045174	047503	052116	020056	
8644	045202	043101	042524	020122	
8645	045210	047503	046515	047101	
8646	045216	000104			
8647	045220	042522	052524	047122	EM25: .ASCII /RETURN TO CENTER LINE COMMAND CAUSED AN ERROR/<15><12>
8648	045226	052040	020117	042503	
8649	045234	052116	051105	046040	
8650	045242	047111	020105	047503	
8651	045250	046515	047101	020104	
8652	045256	040503	051525	042105	
8653	045264	040440	020116	051105	
8654	045272	047522	006522	012	
8655	045277	107	047517	020104	.ASCIZ /GOOD DATA GIVES SHOULD BE, REC DATA GIVES REG. CONT. AFTER COMMAND/
8656	045304	040504	040524	043440	
8657	045312	053111	051505	051440	
8658	045320	047510	046125	020104	
8659	045326	042502	020054	042522	
8660	045334	020103	040504	040524	
8661	045342	043440	053111	051505	
8662	045350	051040	043505	020056	
8663	045356	047503	052116	020056	
8664	045364	043101	042524	020122	
8665	045372	047503	046515	047101	
8666	045400	000104			
8667	045402	030065	020060	043117	EM26: .ASCIZ /500 OFFSET COMMANDS ONE AFTER THE OTHER CAUSED AN ERROR/
8668	045410	051506	052105	041440	
8669	045416	046517	040515	042116	
8670	045424	020123	047117	020105	
8671	045432	043101	042524	020122	
8672	045440	044124	020105	052117	
8673	045446	042510	020122	040503	
8674	045454	051525	042105	040440	
8675	045462	020116	051105	047522	
8676	045470	000122			
8677	045472	051127	052111	020105	EM27: .ASCII /WRITE HEADER AND DATA CAUSED IMPROPER REGISTER CHANGE/<15><12>
8678	045500	042510	042101	051105	
8679	045506	040440	042116	042040	
8680	045514	052101	020101	040503	
8681	045522	051525	042105	044440	

8682	045530	050115	047522	042520	
8683	045536	020122	042522	044507	
8684	045544	052123	051105	041440	
8685	045552	040510	043516	006505	
8686	045560	012			
8687	045561	107	047517	020104	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
8688	045566	040504	040524	043440	
8689	045574	053111	051505	053440	
8690	045602	040510	020124	044123	
8691	045610	052517	042114	041040	
8692	045616	020105	044124	051105	
8693	045624	006505	012		
8694	045627	122	041505	044505	.ASCIZ /RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND/
8695	045634	042526	020104	040504	
8696	045642	040524	043440	053111	
8697	045650	051505	053440	040510	
8698	045656	020124	040527	020123	
8699	045664	044124	051105	020105	
8700	045672	043101	042524	020122	
8701	045700	047503	046515	047101	
8702	045706	000104			
8703	045710	051127	052111	020105	EM30: .ASCIZ /WRITE HEADER AND DATA CHANGED WRITE FROM BUFFER/
8704	045716	042510	042101	051105	
8705	045724	040440	042116	042040	
8706	045732	052101	020101	044103	
8707	045740	047101	042507	020104	
8708	045746	051127	052111	020105	
8709	045754	051106	046517	041040	
8710	045762	043125	042506	000122	
8711					
8712	045770	042522	042101	044040	EM31: .ASCII /READ HEADER AND DATA CAUSED IMPROPER REGISTER CHANGE/<15><12>
8713	045776	040505	042504	020122	
8714	046004	047101	020104	040504	
8715	046012	040524	041440	052501	
8716	046020	042523	020104	046511	
8717	046026	051120	050117	051105	
8718	046034	051040	043505	051511	
8719	046042	042524	020122	044103	
8720	046050	047101	042507	005015	
8721	046056	047507	042117	042040	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
8722	046064	052101	020101	044507	
8723	046072	042526	020123	044127	
8724	046100	052101	051440	047510	
8725	046106	046125	020104	042502	
8726	046114	052040	042510	042522	
8727	046122	005015			
8728	046124	042522	042503	053111	.ASCIZ /RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND/
8729	046132	042105	042040	052101	
8730	046140	020101	044507	042526	
8731	046146	020123	044127	052101	
8732	046154	053440	051501	052040	
8733	046162	042510	042522	040440	
8734	046170	052106	051105	041440	
8735	046176	046517	040515	042116	
8736	046204	000			
8737	046205	127	044522	042524	EM32: .ASCIZ /WRITE HEADER DATA FOLLOWED BY READ HEADER AND DATA CAUSED DATA ERROR/

8738	046212	044040	040505	042504	
8739	046220	020122	040504	040524	
8740	046226	043040	046117	047514	
8741	046234	042527	020104	054502	
8742	046242	051040	040505	020104	
8743	046250	042510	042101	051105	
8744	046256	040440	042116	042040	
8745	046264	052101	020101	040503	
8746	046272	051525	042105	042040	
8747	046300	052101	020101	051105	
8748	046306	047522	000122		
8749	046312	042522	042101	042040	EM33: .ASCII /READ DATA CAUSED IMPROPER REGISTER CHANGE/<15><12>
8750	046320	052101	020101	040503	
8751	046326	051525	042105	044440	
8752	046334	050115	047522	042520	
8753	046342	020122	042522	044507	
8754	046350	052123	051105	041440	
8755	046356	040510	043516	006505	
8756	046364	012			
8757	046365	107	047517	020104	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
8758	046372	040504	040524	043440	
8759	046400	053111	051505	053440	
8760	046406	040510	020124	044123	
8761	046414	052517	042114	041040	
8762	046422	020105	044124	051105	
8763	046430	006505	012		
8764	046433	122	041505	044505	.ASCIZ /RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND/
8765	046440	042526	020104	040504	
8766	046446	040524	043440	053111	
8767	046454	051505	053440	040510	
8768	046462	020124	040527	020123	
8769	046470	044124	051105	020105	
8770	046476	043101	042524	020122	
8771	046504	047503	046515	047101	
8772	046512	000104			
8773	046514	042522	042101	042040	EM34: .ASCIZ /READ DATA INCORRECT/
8774	046522	052101	020101	047111	
8775	046530	047503	051122	041505	
8776	046536	000124			
8777	046540	051127	052111	020105	EM35: .ASCII /WRITE DATA COMMAND CAUSED IMPROPER REGISTER CHANGE/<15><12>
8778	046546	040504	040524	041440	
8779	046554	046517	040515	042116	
8780	046562	041440	052501	042523	
8781	046570	020104	046511	051120	
8782	046576	050117	051105	051040	
8783	046604	043505	051511	042524	
8784	046612	020122	044103	047101	
8785	046620	042507	005015		
8786	046624	047507	042117	042040	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
8787	046632	052101	020101	044507	
8788	046640	042526	020123	044127	
8789	046646	052101	051440	047510	
8790	046654	046125	020104	042502	
8791	046662	052040	042510	042522	
8792	046670	005015			
8793	046672	042522	042503	053111	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER COMMAND/

8794	046700	042105	042040	052101	
8795	046706	020101	044507	042526	
8796	046714	020123	042522	044507	
8797	046722	052123	051105	041440	
8798	046730	047117	042524	052116	
8799	046736	020123	043101	042524	
8800	046744	020122	047503	046515	
8801	046752	047101	000104		
8802	046756	051127	052111	020105	EM36: .ASCIZ /WRITE DATA COMMAND CHANGED WRITE FROM BUFFER/
8803	046764	040504	040524	041440	
8804	046772	046517	040515	042116	
8805	047000	041440	040510	043516	
8806	047006	042105	053440	044522	
8807	047014	042524	043040	047522	
8808	047022	020115	052502	043106	
8809	047030	051105	000		
8810	047033	123	042505	020113	EM37: .ASCII /SEEK COMMAND CAUSED IMPROPER REGISTER CHANGE/<15><12>
8811	047040	047503	046515	047101	
8812	047046	020104	040503	051525	
8813	047054	042105	044440	050115	
8814	047062	047522	042520	020122	
8815	047070	042522	044507	052123	
8816	047076	051105	041440	040510	
8817	047104	043516	006505	012	
8818	047111	107	047517	020104	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
8819	047116	040504	040524	043440	
8820	047124	053111	051505	053440	
8821	047132	040510	020124	044123	
8822	047140	052517	042114	041040	
8823	047146	020105	044124	051105	
8824	047154	006505	012		
8825	047157	122	041505	044505	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER SEEK COMMAND/
8826	047164	042526	020104	040504	
8827	047172	040524	043440	053111	
8828	047200	051505	051040	043505	
8829	047206	051511	042524	020122	
8830	047214	047503	052116	047105	
8831	047222	051524	040440	052106	
8832	047230	051105	051440	042505	
8833	047236	020113	047503	046515	
8834	047244	047101	000104		
8835	047250	051127	052111	020105	EM40: .ASCII /WRITE CHECK CAUSED IMPROPER REGISTER CHANGE/<15><12>
8836	047256	044103	041505	020113	
8837	047264	040503	051525	042105	
8838	047272	044440	050115	047522	
8839	047300	042520	020122	042522	
8840	047306	044507	052123	051105	
8841	047314	041440	040510	043516	
8842	047322	006505	012		
8843	047325	107	047517	020104	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
8844	047332	040504	040524	043440	
8845	047340	053111	051505	053440	
8846	047346	040510	020124	044123	
8847	047354	052517	042114	041040	
8848	047362	020105	044124	051105	
8849	047370	006505	012		

8850	047373	122	041505	044505	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER COMMAND/
8851	047400	042526	020104	040504	
8852	047406	040524	043440	053111	
8853	047414	051505	051040	043505	
8854	047422	051511	042524	020122	
8855	047430	047503	052116	047105	
8856	047436	051524	040440	052106	
8857	047444	051105	041440	046517	
8858	047452	040515	042116	000	
8859					
8860	047457	114	041517	044513	EM41: .ASCII /LOCKING OUT WRITE BY WRITE LOCK BUTTON CAUSED IMPROPER REGISTER CHANGE/
8861	047464	043516	047440	052125	
8862	047472	053440	044522	042524	
8863	047500	041040	020131	051127	
8864	047506	052111	020105	047514	
8865	047514	045503	041040	052125	
8866	047522	047524	020116	040503	
8867	047530	051525	042105	044440	
8868	047536	050115	047522	042520	
8869	047544	020122	042522	044507	
8870	047552	052123	051105	041440	
8871	047560	040510	043516	006505	
8872	047566	012			
8873	047567	107	047517	020104	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
8874	047574	040504	040524	043440	
8875	047602	053111	051505	053440	
8876	047610	040510	020124	044123	
8877	047616	052517	042114	041040	
8878	047624	020105	044124	051105	
8879	047632	006505	012		
8880	047635	122	041505	044505	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER WRITES WERE LOCKED OUT/
8881	047642	042526	020104	040504	
8882	047650	040524	043440	053111	
8883	047656	051505	051040	043505	
8884	047664	051511	042524	020122	
8885	047672	047503	052116	047105	
8886	047700	051524	040440	052106	
8887	047706	051105	053440	044522	
8888	047714	042524	020123	042527	
8889	047722	042522	046040	041517	
8890	047730	042513	020104	052517	
8891	047736	000124			
8892	047740	052101	042524	050115	EM42: .ASCII /ATTEMPTING TO WRITE WITH WRITES LOCKED OUT CAUSED IMPROPER REGISTER CHA
8893	047746	044524	043516	052040	
8894	047754	020117	051127	052111	
8895	047762	020105	044527	044124	
8896	047770	053440	044522	042524	
8897	047776	020123	047514	045503	
8898	050004	042105	047440	052125	
8899	050012	041440	052501	042523	
8900	050020	020104	046511	051120	
8901	050026	050117	051105	051040	
8902	050034	043505	051511	042524	
8903	050042	020122	044103	047101	
8904	050050	042507	005015		
8905	050054	047507	042117	042040	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>

8906	050062	052101	020101	044507	
8907	050070	042526	020123	044127	
8908	050076	052101	051440	047510	
8909	050104	046125	020104	042502	
8910	050112	052040	042510	042522	
8911	050120	005015			
8912	050122	042522	042503	053111	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ATTEMPTED WRITE/
8913	050130	042105	042040	052101	
8914	050136	020101	044507	042526	
8915	050144	020123	042522	044507	
8916	050152	052123	051105	041440	
8917	050160	047117	042524	052116	
8918	050166	020123	043101	042524	
8919	050174	020122	052101	042524	
8920	050202	050115	042524	020104	
8921	050210	051127	052111	000105	
8922	050216	051127	052111	047111	EM43: .ASCII /WRITING WITH WRITES LOCKED OUT CHANGED DISK DATA/<15><12>
8923	050224	020107	044527	044124	
8924	050232	053440	044522	042524	
8925	050240	020123	047514	045503	
8926	050246	042105	047440	052125	
8927	050254	041440	040510	043516	
8928	050262	042105	042040	051511	
8929	050270	020113	040504	040524	
8930	050276	005015			
8931	050300	047507	042117	042040	.ASCII /GOOD DATA GIVES WHAT WAS ON DISK BEFORE WRITE WITH WRITE LOCKED OUT/<15
8932	050306	052101	020101	044507	
8933	050314	042526	020123	044127	
8934	050322	052101	053440	051501	
8935	050330	047440	020116	044504	
8936	050336	045523	041040	043105	
8937	050344	051117	020105	051127	
8938	050352	052111	020105	044527	
8939	050360	044124	053440	044522	
8940	050366	042524	046040	041517	
8941	050374	042513	020104	052517	
8942	050402	006524	012		
8943	050405	127	051501	040440	.ASCII /WAS ATTEMPTED/<15><12>
8944	050412	052124	046505	052120	
8945	050420	042105	005015		
8946	050424	042522	042503	053111	.ASCII /RECEIVED DATA GIVES WHAT WAS READ BACK AFTER WRITE/<15><12>
8947	050432	042105	042040	052101	
8948	050440	020101	044507	042526	
8949	050446	020123	044127	052101	
8950	050454	053440	051501	051040	
8951	050462	040505	020104	040502	
8952	050470	045503	040440	052106	
8953	050476	051105	053440	044522	
8954	050504	042524	005015		
8955	050510	044527	044124	053440	.ASCIZ /WITH WRITE LOCKED OUT WAS ATTEMPTED/
8956	050516	044522	042524	046040	
8957	050524	041517	042513	020104	
8958	050532	052517	020124	040527	
8959	050540	020123	052101	042524	
8960	050546	050115	042524	000104	
8961	050554	047105	041101	044514	EM44: .ASCII /ENABLING WRITES BY WRITE LOCK BUTTON CAUSED IMPROPER REGISTER CHANGE/<1

8962	050562	043516	053440	044522
8963	050570	042524	020123	054502
8964	050576	053440	044522	042524
8965	050604	046040	041517	020113
8966	050612	052502	052124	047117
8967	050620	041440	052501	042523
8968	050626	020104	046511	051120
8969	050634	050117	051105	051040
8970	050642	043505	051511	042524
8971	050650	020122	044103	047101
8972	050656	042507	005015	
8973	050662	047507	042117	042040
8974	050670	052101	020101	044507
8975	050676	042526	020123	044127
8976	050704	052101	051440	047510
8977	050712	046125	020104	042502
8978	050720	052040	042510	042522
8979	050726	005015		
8980	050730	042522	042503	053111
8981	050736	042105	042040	052101
8982	050744	020101	044507	042526
8983	050752	020123	042522	044507
8984	050760	052123	051105	041440
8985	050766	047117	042524	052116
8986	050774	020123	043101	042524
8987	051002	020122	051127	052111
8988	051010	020105	047514	045503
8989	051016	041040	052125	047524
8990	051024	006516	012	
8991	051027	105	040516	046102
8992	051034	042105	053440	044522
8993	051042	042524	000123	
8994	051046	051124	047101	043123
8995	051054	051105	044522	043516
8996	051062	047440	020116	040514
8997	051070	052123	041040	047514
8998	051076	045503	026440	041440
8999	051104	046131	047111	042504
9000	051112	020122	030464	027060
9001	051120	026440	034040	032061
9002	051126	026056	051440	041505
9003	051134	047524	020122	030462
9004	051142	020054	005015	
9005	051146	051124	041501	020113
9006	051154	034061	020054	040503
9007	051162	051525	042105	044440
9008	051170	050115	047522	042520
9009	051176	020122	042522	044507
9010	051204	052123	051105	041440
9011	051212	040510	043516	006505
9012	051220	012		
9013	051221	107	047517	020104
9014	051226	040504	040524	043440
9015	051234	053111	051505	053440
9016	051242	040510	020124	044123
9017	051250	052517	042114	041040

.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>

.ASCII /RECEIVED DATA GIVES REGISTER CONTENTS AFTER WRITE LOCK BUTTON/<15><12>

.ASCIIZ /ENABLED WRITES/

EM45: .ASCII /TRANSFERRING ON LAST BLOCK - CYLINDER 410. - 814., SECTOR 21, /<15><12>

.ASCII /TRACK 18, CAUSED IMPROPER REGISTER CHANGE/<15><12>

.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>

9018	051256	020105	044124	051105	
9019	051264	006505	012		
9020	051267	122	041505	044505	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER TRANSFER/
9021	051274	042526	020104	040504	
9022	051302	040524	043440	053111	
9023	051310	051505	051040	043505	
9024	051316	051511	042524	020122	
9025	051324	047503	052116	047105	
9026	051332	051524	040440	052106	
9027	051340	051105	052040	040522	
9028	051346	051516	042506	000122	
9029	051354	040504	040524	051040	EM46: .ASCII /DATA READ FROM LAST BLOCK - CYLINDER 410. - 814., SECTOR 21./<15><12>
9030	051362	040505	020104	051106	
9031	051370	046517	046040	051501	
9032	051376	020124	046102	041517	
9033	051404	020113	020055	054503	
9034	051412	044514	042116	051105	
9035	051420	032040	030061	020056	
9036	051426	020055	030470	027064	
9037	051434	020054	042523	052103	
9038	051442	051117	031040	026061	
9039	051450	005015			
9040	051452	051124	041501	020113	.ASCIZ /TRACK 18, IS IN ERROR/
9041	051460	034061	020054	051511	
9042	051466	044440	020116	051105	
9043	051474	047522	000122		
9044	051500	051124	047101	043123	EM47: .ASCII /TRANSFERRING DATA FROM NONEXISTANT SECTOR CAUSED IMPROPER /<15><12>
9045	051506	051105	044522	043516	
9046	051514	042040	052101	020101	
9047	051522	051106	046517	047040	
9048	051530	047117	054105	051511	
9049	051536	040524	052116	051440	
9050	051544	041505	047524	020122	
9051	051552	040503	051525	042105	
9052	051560	044440	050115	047522	
9053	051566	042520	020122	005015	
9054	051574	042522	044507	052123	.ASCII /REGISTER CHANGE, GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
9055	051602	051105	041440	040510	
9056	051610	043516	026105	043440	
9057	051616	047517	020104	040504	
9058	051624	040524	043440	053111	
9059	051632	051505	053440	040510	
9060	051640	020124	044123	052517	
9061	051646	042114	041040	020105	
9062	051654	044124	051105	006505	
9063	051662	012			
9064	051663	122	041505	044505	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ATTEMPTED TRANSFER/
9065	051670	042526	020104	040504	
9066	051676	040524	043440	053111	
9067	051704	051505	051040	043505	
9068	051712	051511	042524	020122	
9069	051720	047503	052116	047105	
9070	051726	051524	040440	052106	
9071	051734	051105	040440	052124	
9072	051742	046505	052120	042105	
9073	051750	052040	040522	051516	

B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z

9074	051756	042506	000122		
9075	051762	051124	047101	043123	EM50: .ASCII /TRANSFERRING FROM NONEXISTANT SECTOR CAUSED DATA ERROR/<15><12>
9076	051770	051105	044522	043516	
9077	051776	043040	047522	020115	
9078	052004	047516	042516	044530	
9079	052012	052123	047101	020124	
9080	052020	042523	052103	051117	
9081	052026	041440	052501	042523	
9082	052034	020104	040504	040524	
9083	052042	042440	051122	051117	
9084	052050	005015			
9085	052052	047507	042117	042040	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
9086	052060	052101	020101	044507	
9087	052066	042526	020123	044127	
9088	052074	052101	051440	047510	
9089	052102	046125	020104	042502	
9090	052110	052040	042510	042522	
9091	052116	005015			
9092	052120	040502	020104	040504	.ASCIIZ /BAD DATA GIVES WHAT WAS IN BUFFER AFTER TRANSFER/
9093	052126	040524	043440	053111	
9094	052134	051505	053440	040510	
9095	052142	020124	040527	020123	
9096	052150	047111	041040	043125	
9097	052156	042506	020122	043101	
9098	052164	042524	020122	051124	
9099	052172	047101	043123	051105	
9100	052200	000			
9101					

9102	052201	107	053111	047111	EM51: .ASCII /GIVING ILLEGAL FUNCTION CAUSED IMPROPER REGISTER CHANGE/<15><12>
9103	052206	020107	046111	042514	
9104	052214	040507	020114	052506	
9105	052222	041516	044524	047117	
9106	052230	041440	052501	042523	
9107	052236	020104	046511	051120	
9108	052244	050117	051105	051040	
9109	052252	043505	051511	042524	
9110	052260	020122	044103	047101	
9111	052266	042507	005015		
9112	052272	047507	042117	042040	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
9113	052300	052101	020101	044507	
9114	052306	042526	020123	044127	
9115	052314	052101	051440	047510	
9116	052322	046125	020104	042502	
9117	052330	052040	042510	042522	
9118	052336	005015			
9119	052340	042522	042503	053111	.ASCIIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ILLEGAL FUNCTION IS GIVEN/
9120	052346	042105	042040	052101	
9121	052354	020101	044507	042526	
9122	052362	020123	042522	044507	
9123	052370	052123	051105	041440	
9124	052376	047117	042524	052116	
9125	052404	020123	043101	042524	
9126	052412	020122	046111	042514	
9127	052420	040507	020114	052506	
9128	052426	041516	044524	047117	
9129	052434	044440	020123	044507	

9130	052442	042526	000116		
9131	052446	051127	052111	020105	EM52: .ASCII /WRITE DATA ON NONEXISTANT SECTOR CAUSED IMPROPER REGISTER CHANGE/<15><1
9132	052454	040504	040524	047440	
9133	052462	020116	047516	042516	
9134	052470	044530	052123	047101	
9135	052476	020124	042523	052103	
9136	052504	051117	041440	052501	
9137	052512	042523	020104	046511	
9138	052520	051120	050117	051105	
9139	052526	051040	043505	051511	
9140	052534	042524	020122	044103	
9141	052542	047101	042507	005015	
9142	052550	047507	042117	042040	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
9143	052556	052101	020101	044507	
9144	052564	042526	020123	044127	
9145	052572	052101	051440	047510	
9146	052600	046125	020104	042502	
9147	052606	052040	042510	042522	
9148	052614	005015			
9149	052616	042522	042503	053111	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ATTEMPTED WRITE DATA/
9150	052624	042105	042040	052101	
9151	052632	020101	044507	042526	
9152	052640	020123	042522	044507	
9153	052646	052123	051105	041440	
9154	052654	047117	042524	052116	
9155	052662	020123	043101	042524	
9156	052670	020122	052101	042524	
9157	052676	050115	042524	020104	
9158	052704	051127	052111	020105	
9159	052712	040504	040524	000	
9160	052717	122	040505	020104	EM53: .ASCIZ /READ HEADER AND DATA AFTER A SEARCH CAUSED DATA ERROR/
9161	052724	042510	042101	051105	
9162	052732	040440	042116	042040	
9163	052740	052101	020101	043101	
9164	052746	042524	020122	020101	
9165	052754	042523	051101	044103	
9166	052762	041440	052501	042523	
9167	052770	020104	040504	040524	
9168	052776	042440	051122	051117	
9169	053004	000			
9170	053005	101	052124	046505	EM54: .ASCII /ATTEMPTING COMMAND WITH INVALID ADDRESS CAUSED IMPROPER REGISTER CHANGE
9171	053012	052120	047111	020107	
9172	053020	047503	046515	047101	
9173	053026	020104	044527	044124	
9174	053034	044440	053116	046101	
9175	053042	042111	040440	042104	
9176	053050	042522	051523	041440	
9177	053056	052501	042523	020104	
9178	053064	046511	051120	050117	
9179	053072	051105	051040	043505	
9180	053100	051511	042524	020122	
9181	053106	044103	047101	042507	
9182	053114	005015			
9183	053116	047507	042117	042040	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
9184	053124	052101	020101	044507	
9185	053132	042526	020123	044127	

9186	053140	052101	051440	047510
9187	053146	046125	020104	042502
9188	053154	052040	042510	042522
9189	053162	005015		
9190	053164	042522	042503	053111
9191	053172	042105	042040	052101
9192	053200	020101	044507	042526
9193	053206	020123	042522	044507
9194	053214	052123	051105	041440
9195	053222	047117	042524	052116
9196	053230	020123	043101	042524
9197	053236	020122	050117	051105
9198	053244	052101	047511	000116
9199	053252	051127	052111	047111
9200	053260	020107	051117	051040
9201	053266	040505	044504	043516
9202	053274	053440	052111	020110
9203	053302	054105	042520	052103
9204	053310	042105	040440	042104
9205	053316	042522	051523	047440
9206	053324	042526	043122	047514
9207	053332	020127	051105	047522
9208	053340	006522	012	
9209	053343	103	052501	042523
9210	053350	020104	046511	051120
9211	053356	050117	051105	051040
9212	053364	043505	051511	042524
9213	053372	020122	044103	047101
9214	053400	042507	005015	
9215	053404	047507	042117	042040
9216	053412	052101	020101	044507
9217	053420	042526	020123	044127
9218	053426	052101	051440	047510
9219	053434	046125	020104	042502
9220	053442	052040	042510	042522
9221	053450	005015		
9222	053452	042522	042503	053111
9223	053460	042105	042040	052101
9224	053466	020101	044507	042526
9225	053474	020123	042522	044507
9226	053502	052123	051105	041440
9227	053510	047117	042524	052116
9228	053516	020123	043101	042524
9229	053524	020122	050117	051105
9230	053532	052101	047511	000116
9231	053540	040504	040524	051040
9232	053546	040505	020104	044527
9233	053554	044124	040440	020116
9234	053562	054105	042520	052103
9235	053570	042105	040440	042104
9236	053576	042522	051523	047440
9237	053604	042526	043122	047514
9238	053612	020127	051105	047522
9239	053620	020122	051511	044440
9240	053626	041516	051117	042522
9241	053634	052103	005015	

.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER OPERATION/

EM55: .ASCII /WRITING OR READING WITH EXPECTED ADDRESS OVERFLOW ERROR/<15><12>

.ASCII /CAUSED IMPROPER REGISTER CHANGE/<15><12>

.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>

.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER OPERATION/

EM56: .ASCII /DATA READ WITH AN EXPECTED ADDRESS OVERFLOW ERROR IS INCORRECT/<15><12>

9242	053640	047527	042122	047040		.ASCII /WORD NO. 1 TO 260 SHOULD BE READ, WORD NO 261 TO 266 SHOULD/<15><12>
9243	053646	027117	030440	052040		
9244	053654	020117	033062	020060		
9245	053662	044123	052517	042114		
9246	053670	041040	020105	042522		
9247	053676	042101	020054	047527		
9248	053704	042122	047040	020117		
9249	053712	033062	020061	047524		
9250	053720	031040	033066	051440		
9251	053726	047510	046125	006504		
9252	053734	012				
9253	053735	102	020105	044103		.ASCIZ /BE CHANGED/
9254	053742	047101	042507	000104		
9255	053750	052101	042524	050115	EM57:	.ASCII /ATTEMPTING DATA COMMAND WITH WRONG FORMAT BIT CAUSED/<15><12>
9256	053756	044524	043516	042040		
9257	053764	052101	020101	047503		
9258	053772	046515	047101	020104		
9259	054000	044527	044124	053440		
9260	054006	047522	043516	043040		
9261	054014	051117	040515	020124		
9262	054022	044502	020124	040503		
9263	054030	051525	042105	005015		
9264	054036	046511	051120	050117		.ASCII /IMPROPER REGISTER CHANGE/<15><12>
9265	054044	051105	051040	043505		
9266	054052	051511	042524	020122		
9267	054060	044103	047101	042507		
9268	054066	005015				
9269	054070	047507	042117	042040		.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
9270	054076	052101	020101	044507		
9271	054104	042526	020123	044127		
9272	054112	052101	051440	047510		
9273	054120	046125	020104	042502		
9274	054126	052040	042510	042522		
9275	054134	005015				
9276	054136	042522	042503	053111		.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ATTEMPTED DATA TRANSFER/
9277	054144	042105	042040	052101		
9278	054152	020101	044507	042526		
9279	054160	020123	042522	044507		
9280	054166	052123	051105	041440		
9281	054174	047117	042524	052116		
9282	054202	020123	043101	042524		
9283	054210	020122	052101	042524		
9284	054216	050115	042524	020104		
9285	054224	040504	040524	052040		
9286	054232	040522	051516	042506		
9287	054240	000122				
9288	054242	052101	042524	050115	EM60:	.ASCII /ATTEMPTING TO MODIFY REGISTER DURING AN OPERATION CAUSED IMPROPER/<15><
9289	054250	044524	043516	052040		
9290	054256	020117	047515	044504		
9291	054264	054506	051040	043505		
9292	054272	051511	042524	020122		
9293	054300	052504	044522	043516		
9294	054306	040440	020116	050117		
9295	054314	051105	052101	047511		
9296	054322	020116	040503	051525		
9297	054330	042105	044440	050115		

9298	054336	047522	042520	006522	
9299	054344	012			
9300	054345	122	043505	051511	.ASCII /REGISTER CHANGE. GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
9301	054352	042524	020122	044103	
9302	054360	047101	042507	020056	
9303	054366	047507	042117	042040	
9304	054374	052101	020101	044507	
9305	054402	042526	020123	044127	
9306	054410	052101	051440	047510	
9307	054416	046125	020104	042502	
9308	054424	052040	042510	042522	
9309	054432	005015			
9310	054434	042522	042503	053111	.ASCII /RECEIVED DATA GIVES REGISTER CONTENTS AFTER OPERATION WAS ATTEMPTED/<15
9311	054442	042105	042040	052101	
9312	054450	020101	044507	042526	
9313	054456	020123	042522	044507	
9314	054464	052123	051105	041440	
9315	054472	047117	042524	052116	
9316	054500	020123	043101	042524	
9317	054506	020122	050117	051105	
9318	054514	052101	047511	020116	
9319	054522	040527	020123	052101	
9320	054530	042524	050115	042524	
9321	054536	006504	012		
9322	054541	115	042117	044506	.ASCIZ /MODFING REG GIVES ADDRESS OF REGISTER BEING MODIFIED WHICH CAUSED ERROR
9323	054546	043516	051040	043505	
9324	054554	043440	053111	051505	
9325	054562	040440	042104	042522	
9326	054570	051523	047440	020106	
9327	054576	042522	044507	052123	
9328	054604	051105	041040	044505	
9329	054612	043516	046440	042117	
9330	054620	043111	042511	020104	
9331	054626	044127	041511	020110	
9332	054634	040503	051525	042105	
9333	054642	042440	051122	051117	
9334	054650	000			
9335					
9336	054651	104	053105	041511	EM61: .ASCIZ /DEVICE NOT AVAILABLE BEFORE COMMAND WAS TO BE GIVEN/
9337	054656	020105	047516	020124	
9338	054664	053101	044501	040514	
9339	054672	046102	020105	042502	
9340	054700	047506	042522	041440	
9341	054706	046517	040515	042116	
9342	054714	053440	051501	052040	
9343	054722	020117	042502	043440	
9344	054730	053111	047105	000	
9345	054735	122	042110	030523	EM63: .ASCIZ /RHDS1 CONTENTS DURING COMMAND WAS IN ERROR/
9346	054742	041440	047117	042524	
9347	054750	052116	020123	052504	
9348	054756	044522	043516	041440	
9349	054764	046517	040515	042116	
9350	054772	053440	051501	044440	
9351	055000	020116	051105	047522	
9352	055006	000122			
9353	055010	042522	040503	044514	EM64: .ASCII /RECALIBRATE COMMAND CAUSED IMPROPER REGISTER CHANGE/<15><12>

9354	055016	051102	052101	020105	
9355	055024	047503	046515	047101	
9356	055032	020104	040503	051525	
9357	055040	042105	044440	050115	
9358	055046	047522	042520	020122	
9359	055054	042522	044507	052123	
9360	055062	051105	041440	040510	
9361	055070	043516	006505	012	
9362	055075	107	047517	020104	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
9363	055102	040504	040524	043440	
9364	055110	053111	051505	053440	
9365	055116	040510	020124	044123	
9366	055124	052517	042114	041040	
9367	055132	020105	044124	051105	
9368	055140	006505	012		
9369	055143	122	041505	044505	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER COMMAND/
9370	055150	042526	020104	040504	
9371	055156	040524	043440	053111	
9372	055164	051505	051040	043505	
9373	055172	051511	042524	020122	
9374	055200	047503	052116	047105	
9375	055206	051524	040440	052106	
9376	055214	051105	041440	046517	
9377	055222	040515	042116	000	
9378	055227	111	052116	051105	EM65: .ASCIZ /INTERRUPT FAILING/
9379	055234	052522	052120	043040	
9380	055242	044501	044514	043516	
9381	055250	000			
9382	055251	110	040505	042504	EM66: .ASCII /HEADER AND DATA COMMAND FOR HEAD SELECTION TEST/<15><12>
9383	055256	020122	047101	020104	
9384	055264	040504	040524	041440	
9385	055272	046517	040515	042116	
9386	055300	043040	051117	044040	
9387	055306	040505	020104	042523	
9388	055314	042514	052103	047511	
9389	055322	020116	042524	052123	
9390	055330	005015			
9391	055332	040503	051525	042105	.ASCII /CAUSED ERROR/<15><12>
9392	055340	042440	051122	051117	
9393	055346	005015			
9394	055350	044122	051504	020124	.ASCII /RHDST GIVES WHAT TRACK WAS BEING WRITTEN OR READ/<15><12>
9395	055356	044507	042526	020123	
9396	055364	044127	052101	052040	
9397	055372	040522	045503	053440	
9398	055400	051501	041040	044505	
9399	055406	043516	053440	044522	
9400	055414	052124	047105	047440	
9401	055422	020122	042522	042101	
9402	055430	005015			
9403	055432	047117	041440	046131	.ASCIZ /ON CYLINDER 0, SECTOR 0/
9404	055440	047111	042504	020122	
9405	055446	026060	051440	041505	
9406	055454	047524	020122	000060	
9407	055462	042522	042101	044040	EM67: .ASCII /READ HEADER AND DATA ERROR IN HEAD SELECTION TEST/<12><15>
9408	055470	040505	042504	020122	
9409	055476	047101	020104	040504	

9410	055504	040524	042440	051122	
9411	055512	051117	044440	020116	
9412	055520	042510	042101	051440	
9413	055526	046105	041505	044524	
9414	055534	047117	052040	051505	
9415	055542	005124	015		
9416	055545	106	051111	052123	.ASCII /FIRST FOUR WORD NUMBERS ARE HEADER/<12><15>
9417	055552	043040	052517	020122	
9418	055560	047527	042122	047040	
9419	055566	046525	042502	051522	
9420	055574	040440	042522	044040	
9421	055602	040505	042504	005122	
9422	055610	015			
9423	055611	127	051117	020104	.ASCII /WORD NUMBERS 5 TO 260 ARE DATA WORDS/<12><15>
9424	055616	052516	041115	051105	
9425	055624	020123	020065	047524	
9426	055632	031040	030066	040440	
9427	055640	042522	042040	052101	
9428	055646	020101	047527	042122	
9429	055654	005123	015		
9430	055657	111	020116	040504	.ASCIZ /IN DATA WORDS BITS 4,5,6,7,8 GIVE TRACK NUMBER/
9431	055664	040524	053440	051117	
9432	055672	051504	041040	052111	
9433	055700	020123	026064	026065	
9434	055706	026066	026067	020070	
9435	055714	044507	042526	052040	
9436	055722	040522	045503	047040	
9437	055730	046525	042502	000122	
9438					
9439	055736	042522	042101	044040	EM70: .ASCII /READ HEADER AND DATA ERROR IN/<15><12>
9440	055744	040505	042504	020122	
9441	055752	047101	020104	040504	
9442	055760	040524	042440	051122	
9443	055766	051117	044440	006516	
9444	055774	012			
9445	055775	104	043111	042506	.ASCII /DIFFERENCE LINE TEST/<15><12>
9446	056002	042522	041516	020105	
9447	056010	044514	042516	052040	
9448	056016	051505	006524	012	
9449	056023	127	051117	020104	.ASCII /WORD NOS 1-4 GIVE HEADER/<15><12>
9450	056030	047516	020123	026461	
9451	056036	020064	044507	042526	
9452	056044	044040	040505	042504	
9453	056052	006522	012		
9454	056055	127	051117	020104	.ASCIZ /WORD NOS 5-260 GIVE DATA WHICH IS THE CYLINDER ADDRESS/
9455	056062	047516	020123	026465	
9456	056070	033062	020060	044507	
9457	056076	042526	042040	052101	
9458	056104	020101	044127	041511	
9459	056112	020110	051511	052040	
9460	056120	042510	041440	046131	
9461	056126	047111	042504	020122	
9462	056134	042101	051104	051505	
9463	056142	000123			
9464	056144	047506	041522	047111	EM71: .ASCII /FORCING OPI BY 3 INDEX PULSES/<15><12>
9465	056152	020107	050117	020111	

9466	056160	054502	031440	044440	
9467	056166	042116	054105	050040	
9468	056174	046125	042523	006523	
9469	056202	012			
9470	056203	103	052501	042523	.ASCII /CAUSED IMPROPER REGISTER CHANGE/<15><12>
9471	056210	020104	046511	051120	
9472	056216	050117	051105	051040	
9473	056224	043505	051511	042524	
9474	056232	020122	044103	047101	
9475	056240	042507	005015		
9476	056244	047507	042117	042040	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
9477	056252	052101	020101	044507	
9478	056260	042526	020123	044127	
9479	056266	052101	051440	047510	
9480	056274	046125	020104	042502	
9481	056302	052040	042510	042522	
9482	056310	005015			
9483	056312	042522	042503	053111	.ASCIIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER 3 INDEX PULSES/
9484	056320	042105	042040	052101	
9485	056326	020101	044507	042526	
9486	056334	020123	042522	044507	
9487	056342	052123	051105	041440	
9488	056350	047117	042524	052116	
9489	056356	020123	043101	042524	
9490	056364	020122	020063	047111	
9491	056372	042504	020130	052520	
9492	056400	051514	051505	000	
9493	056405	124	042510	042522	EM72: .ASCII /THERE WAS A SETUP ERROR DURING MULTIPLE WRITE/<15><12>
9494	056412	053440	051501	040440	
9495	056420	051440	052105	050125	
9496	056426	042440	051122	051117	
9497	056434	042040	051125	047111	
9498	056442	020107	052515	052114	
9499	056450	050111	042514	053440	
9500	056456	044522	042524	005015	
9501	056464	042510	042101	051105	.ASCII /HEADER AND DATA COMMANDS RESULTING IN AN ABORT/<15><12>
9502	056472	040440	042116	042040	
9503	056500	052101	020101	047503	
9504	056506	046515	047101	051504	
9505	056514	051040	051505	046125	
9506	056522	044524	043516	044440	
9507	056530	020116	047101	040440	
9508	056536	047502	052122	005015	
9509	056544	043117	052040	044510	.ASCII /OF THIS 'OPI' TEST./<15><12><15><12>
9510	056552	020123	047447	044520	
9511	056560	020047	042524	052123	
9512	056566	006456	006412	012	
9513	056573	124	020117	051124	.ASCIIZ /TO TROUBLE SHOOT SETUP ERROR, LOOP ON THIS TEST/
9514	056600	052517	046102	020105	
9515	056606	044123	047517	020124	
9516	056614	042523	052524	020120	
9517	056622	051105	047522	026122	
9518	056630	046040	047517	020120	
9519	056636	047117	052040	044510	
9520	056644	020123	042524	052123	
9521	056652	000			

9522	056653	122	040505	020104	EM73: .ASCII /READ HEADER AND DATA FOR 11960 WORDS /<15><12>
9523	056660	042510	042101	051105	
9524	056666	040440	042116	042040	
9525	056674	052101	020101	047506	
9526	056702	020122	030461	033071	
9527	056710	020060	047527	042122	
9528	056716	020123	005015		
9529	056722	044124	052101	044440	.ASCII /THAT IS 46 SECTORS /<15><12>
9530	056730	020123	033064	051440	
9531	056736	041505	047524	051522	
9532	056744	006440	012		
9533	056747	124	040510	020124	.ASCIZ /THAT IS OVER 3 INDEX PULSES CAUSED AN ERROR/
9534	056754	051511	047440	042526	
9535	056762	020122	020063	047111	
9536	056770	042504	020130	052520	
9537	056776	051514	051505	041440	
9538	057004	052501	042523	020104	
9539	057012	047101	042440	051122	
9540	057020	051117	000		
9541	057023	122	040505	020104	EM74: .ASCII /READ HEADER AND DATA FOR 11960 WORDS /<15><12>
9542	057030	042510	042101	051105	
9543	057036	040440	042116	042040	
9544	057044	052101	020101	047506	
9545	057052	020122	030461	033071	
9546	057060	020060	047527	042122	
9547	057066	020123	005015		
9548	057072	044124	052101	044440	.ASCII /THAT IS 46 SECTORS, THAT IS OVER 3 INDEX /<15><12>
9549	057100	020123	033064	051440	
9550	057106	041505	047524	051522	
9551	057114	020054	044124	052101	
9552	057122	044440	020123	053117	
9553	057130	051105	031440	044440	
9554	057136	042116	054105	006440	
9555	057144	012			
9556	057145	120	046125	042523	.ASCIZ /PULSES CAUSED OPI TO SET/
9557	057152	020123	040503	051525	
9558	057160	042105	047440	044520	
9559	057166	052040	020117	042523	
9560	057174	000124			

9561										
9562	057176	040506	040524	020114	CPHALT: .ASCII	/FATAL ERROR - SEE DOCUMENT LISTING/	<15>	<12>		
9563	057204	051105	047522	020122						
9564	057212	020055	042523	020105						
9565	057220	047504	052503	042515						
9566	057226	052116	046040	051511						
9567	057234	044524	043516	005015						
9568	057242	006440	103412	177777	.ASCII	/	<15>	<12>	<207>	<377>
9569	057250	177607	103777	177777						
9570	057256	044124	020105	047503	.ASCII	/THE CONTROLLER OR DEVICE HAS GONE OFFLINE, LOST/	<15>	<12>		
9571	057264	052116	047522	046114						
9572	057272	051105	047440	020122						
9573	057300	042504	044526	042503						
9574	057306	044040	051501	043440						
9575	057314	047117	020105	043117						
9576	057322	046106	047111	026105						
9577	057330	046040	051517	006524						
9578	057336	012								
9579	057337	047	042522	042101	.ASCII	/'READY', BECOME UNAVAILABLE, OR HAS STATUS BITS/	<15>	<12>		
9580	057344	023531	020054	042502						
9581	057352	047503	042515	052440						
9582	057360	040516	040526	046111						
9583	057366	041101	042514	020054						
9584	057374	051117	044040	051501						
9585	057402	051440	040524	052524						
9586	057410	020123	044502	051524						
9587	057416	005015								
9588	057420	044127	041511	020110	.ASCIIZ	/WHICH CANNOT BE CLEARED/				
9589	057426	040503	047116	052117						
9590	057434	041040	020105	046103						
9591	057442	040505	042522	000104						
9592										
9593										
9594										
9595	057450	041520	020040	020040	DH1: .ASCII	/PC	TEST	WAIT	BIT	REG
9596	057456	020040	042524	052123						REG
9597	057464	020040	020040	040527						RHCS1/
9598	057472	052111	020040	020040						<15>
9599	057500	044502	020124	020040						<12>
9600	057506	020040	042522	020107						
9601	057514	020040	020040	042522						
9602	057522	020107	020040	020040						
9603	057530	044122	051503	006461						
9604	057536	012								
9605	057537	040	020040	020040	.ASCIIZ	/	NO	PC	EXPCTD	ADDRESS
9606	057544	020040	047040	020117						CONTENT
9607	057552	020040	020040	050040						CONTENT
9608	057560	020103	020040	020040						/
9609	057566	042440	050130	052103						
9610	057574	020104	040440	042104						
9611	057602	042522	051523	041440						
9612	057610	047117	042524	052116						
9613	057616	041440	047117	042524						
9614	057624	052116	000011							
9615	057630	041520	020040	020040	DH4: .ASCII	/PC	TEST	WAIT	BIT	REG
9616	057636	020040	042524	052123						TIME IN/

9617	057644	020040	020040	040527					
9618	057652	052111	020040	020040					
9619	057660	044502	020124	020040					
9620	057666	020040	042522	020107					
9621	057674	020040	020040	044524					
9622	057702	042515	044440	006516					
9623	057710	012							
9624	057711	040	020040	020040	.ASCIZ /	NO	PC	EXPCTD	ADDRESS 10 MSEC/
9625	057716	020040	047040	020117					
9626	057724	020040	020040	050040					
9627	057732	020103	020040	020040					
9628	057740	042440	050130	052103					
9629	057746	020104	040440	042104					
9630	057754	042522	051523	030440					
9631	057762	020060	051515	041505					
9632	057770	000							
9633	057771	120	020103	020040	DH5: .ASCII /PC	TEST	REG	GOOD	RECEIVED/<15><12>
9634	057776	020040	052040	051505					
9635	060004	020124	020040	051040					
9636	060012	043505	020040	020040					
9637	060020	043440	047517	020104					
9638	060026	020040	051040	041505					
9639	060034	044505	042526	006504					
9640	060042	012							
9641	060043	040	020040	020040	.ASCIZ /	NO	ADDRESS DATA	DATA/	
9642	060050	020040	047040	020117					
9643	060056	020040	020040	040440					
9644	060064	042104	042522	051523					
9645	060072	042040	052101	020101					
9646	060100	020040	042040	052101					
9647	060106	000101							
9648	060110	041520	020040	020040	DH6: .ASCII /PC	TEST	REG	RECEIVED/<15><12>	
9649	060116	020040	042524	052123					
9650	060124	020040	020040	042522					
9651	060132	020107	020040	020040					
9652	060140	042522	042503	053111					
9653	060146	042105	005015						
9654	060152	020040	020040	020040	.ASCIZ /	NO	ADDRESS DATA/		
9655	060160	020040	047516	020040					
9656	060166	020040	020040	042101					
9657	060174	051104	051505	020123					
9658	060202	040504	040524	000					
9659	060207	120	020103	020040	DH7: .ASCIZ /PC	TEST	REG	ADDRESS/	
9660	060214	020040	052040	051505					
9661	060222	020124	020040	051040					
9662	060230	043505	020040	020040					
9663	060236	040440	042104	042522					
9664	060244	051523	000						
9665									
9666	060247	120	020103	020040	DH10: .ASCII /PC	TEST	FAILING CONTENT	CONTENT CONTENT CONTENT	CONTENT/<15><12>
9667	060254	020040	052040	051505					
9668	060262	020124	020040	043040					
9669	060270	044501	044514	043516					
9670	060276	041440	047117	042524					
9671	060304	052116	041440	047117					
9672	060312	042524	052116	041440					

Address	OpCode	OpCode	OpCode	OpCode	Label	OpCode	OpCode	OpCode	OpCode	OpCode
9729	060776	020040	042522	053103						
9730	061004	020104	020040	046111						
9731	061012	042514	046107	005015						
9732	061020	020040	020040	020040		.ASCIZ /	NO	ADDRESS DATA	DATA	FUNCTN/
9733	061026	020040	047516	020040						
9734	061034	020040	020040	042101						
9735	061042	051104	051505	020123						
9736	061050	040504	040524	020040						
9737	061056	020040	040504	040524						
9738	061064	020040	020040	052506						
9739	061072	041516	047124	000						
9740										
9741	061077	120	020103	020040	DH60:	.ASCII /PC	TEST	REG	GOOD	RECVD MODFING/<15><12>
9742	061104	020040	052040	051505						
9743	061112	020124	020040	051040						
9744	061120	043505	020040	020040						
9745	061126	043440	047517	020104						
9746	061134	020040	051040	041505						
9747	061142	042126	020040	046440						
9748	061150	042117	044506	043516						
9749	061156	005015								
9750	061160	020040	020040	020040		.ASCIZ /	NO	ADDRESS DATA	DATA	REG/
9751	061166	020040	047516	020040						
9752	061174	020040	020040	042101						
9753	061202	051104	051505	020123						
9754	061210	040504	040524	020040						
9755	061216	020040	040504	040524						
9756	061224	020040	020040	042522						
9757	061232	000107								
9758	061234	041520	020040	020040	DH61:	.ASCII /PC	TEST	PC OF	RHDS1/<15><12>	
9759	061242	020040	042524	052123						
9760	061250	020040	020040	041520						
9761	061256	047440	004506	051040						
9762	061264	042110	030523	005015						
9763	061272	020040	020040	020040		.ASCIZ /	NO	JSR	WAS/	
9764	061300	020040	047516	020040						
9765	061306	020040	020040	051512						
9766	061314	020122	020040	020040						
9767	061322	040527	000123							
9768	061326	041520	020040	020040	DH62:	.ASCII /PC	PC OF	RHCS1/<15><12>		
9769	061334	020040	041520	047440						
9770	061342	020106	020040	044122						
9771	061350	051503	006461	012						
9772	061355	040	020040	020040		.ASCIZ /	JSR	WAS/		
9773	061362	020040	045040	051123						
9774	061370	020040	020040	053440						
9775	061376	051501	000							
9776	061401	120	020103	020040	DH65:	.ASCII /PC	TEST	CONT	CONT	CONT/<15><12>
9777	061406	020040	052040	051505						
9778	061414	020124	020040	041440						
9779	061422	047117	020124	020040						
9780	061430	041440	047117	020124						
9781	061436	020040	041440	047117						
9782	061444	006524	012							
9783	061447	040	020040	020040		.ASCIZ /	NO	RHCS1	RHAS	RHDS1/
9784	061454	020040	047040	020117						

9841	062114	001116	004504	004500	DT60:	.WORD	\$ERRPC,TSTNM,REGADR,\$GDDAT,\$BDDAT,\$BDADR,0
9842	062122	001124	001126	001122			
9843	062130	000000					
9844	062132	001116	004504	033126	DT61:	.WORD	\$ERRPC,TSTNM,PCJSR,\$BDADR,0
9845	062140	001122	000000				
9846	062144	001116	004504	033126	DT62:	.WORD	\$ERRPC,TSTNM,PCJSR,\$BDADR,0
9847	062152	001122	000000				
9848	062156	001116	004504	002262	DT65:	.WORD	\$ERRPC,TSTNM,CS1,AS,DS1,0
9849	062164	002300	002304	000000			
9850	062172	001116	004504	002266	DT66:	.WORD	\$ERRPC,TSTNM,DST,ER1,ER2,ER3,CS1,CS2,0
9851	062200	002264	002270	002276			
9852	062206	002262	002260	000000			
9853	062214	001116	004504	002262	DT72:	.WORD	\$ERRPC,TSTNM,CS1,CS2,DS1,DST,CA,ER1,WC,0
9854	062222	002260	002304	002266			
9855	062230	002274	002264	002254			
9856	062236	000000					
9857							
9858	062240	000	000	000	DF1:	.BYTE	0,0,0,0,0,0,0
9859	062243	000	000	000			
9860	062246	000					
9861	062247	000	000	000	DF4:	.BYTE	0,0,0,0,0,1,0
9862	062252	000	000	001			
9863	062255	000					
9864	062256	000	000	000	DF5:	.BYTE	0,0,0,0,0
9865	062261	000	000				
9866	062263	000	000	000	DF6:	.BYTE	0,0,0,0
9867	062266	000					
9868	062267	000	000	000	DF7:	.BYTE	0,0,0
9869	062272	000	000	000	DF10:	.BYTE	0,0,0,0,0,0,0
9870	062275	000	000	000			
9871	062300	000					
9872							
9873	062301	000	000	000	DF26:	.BYTE	0,0,0,0,0,0,0,0
9874	062304	000	000	000			
9875	062307	000	000				
9876							
9877	062311	000	000	000	DF30:	.BYTE	0,0,0,0,0
9878	062314	000	000				
9879							
9880	062316	000	000	000	DF51:	.BYTE	0,0,0,0,0,0
9881	062321	000	000	000			
9882							
9883	062324	000	000	000	DF60:	.BYTE	0,0,0,0,0,0
9884	062327	000	000	000			
9885	062332	000	000	000	DF61:	.BYTE	0,0,0,0
9886	062335	000					
9887	062336	000	000	000	DF62:	.BYTE	0,0,0,0
9888	062341	000					
9889	062342	000	000	000	DF65:	.BYTE	0,0,0,0,0
9890	062345	000	000				
9891	062347	000	000	000	DF66:	.BYTE	0,0,0,0,0,0,0,0,0
9892	062352	000	000	000			
9893	062355	000	000	000			
9894							
9895	062360	000	000	000	DF72:	.BYTE	0,0,0,0,0,0,0,0,0
9896	062363	000	000	000			

9897	062366	000	000	000	
9898					
9899	062372				.EVEN
9900					
9901	000001				.END

EM60	054242	1235	9288#											
EM61	054651	1252	1260	9336#										
EM63	054735	1269	9345#											
EM64	055010	1277	9353#											
EM65	055227	1290	9378#											
EM66	055251	1301	9382#											
EM67	055462	1317	9407#											
EM7	043242	774	8464#											
EM70	055736	1328	9439#											
EM71	056144	1340	9464#											
EM72	056405	1354	9493#											
EM73	056653	1374	9522#											
EM74	057023	1381	9541#											
ERFLG\$	004632	1770#	6146*	8106*										
ERR =	040000	1465#	3229	3459	3559	4071	4168	4285	4376	4613	4790	5105	5303	5509
		5703	6065	6201	6566	6653								
ERRVEC=	000004	628#	1828	1829*	1839*	1931*	1939*	1967*	7761	7762*	7764*	7767*		
ERWORD	004502	1731#	7666*	7672*	9836									
ER1	002264	1674#	6454*	6455	8180	9830	9833	9850	9853					
ER2	002270	1676#	8184	9833	9850									
ER3	002276	1679#	8188	9833	9850									
EXT1 =	000001	1528#												
EXT10 =	000010	1531#												
EXT2 =	000002	1529#												
EXT20 =	000020	1532#												
EXT4 =	000004	1530#												
EXT40 =	000040	1533#												
FEN =	000200	1554#												
FER =	000020	1473#	5114	5311										
FILL	034026	7451#												
FILLRE	032774	2464	2467	2480	2570	2573	2586	2734	2737	2740	2851	2854	2857	2860
		3021	3024	3027	3030	3033	3150	3153	3156	3242	3334	3337	3340	3407
		3410	3423	3426	3444	3542	3545	3560	3807	3810	3813	3908	3911	3914
		4584	4587	4604	4624	4627	4632	4635	4771	4774	4793	4802	4807	4912
		4915	4918	5068	5071	5079	5082	5106	5188	5191	5194	5304	5500	5516
		5519	5710	5713	5716	5806	5810	5813	5920	5923	5926	5929	6087	6091
		6202	7087#											
FINACC	004564	1743#	7238*											
FINALA	004562	1742#	7239*											
FIRST	004634	1772#	1850	1873*										
FLHEAD	032716	2382	2393	2510	2647	2662	2838	3008	3098	3281	3483	3725	3744	3968
		3978	4412	4467	4478	4505	4516	4676	4687	4859	4978	5368	5575	5590
		6254	6339	6510	6604	6743	6837	7030#						
FMT22 =	010000	1575#	2424	2425	2436	2437	2532	2533	2544	2545	2690	2691	2781	2782
		2926	2927	3122	3123	3196	3197	3305	3306	3377	3378	3514	3515	3764
		3765	3865	3866	4009	4010	4020	4021	4121	4122	4239	4240	4336	4436
		4437	4546	4547	4557	4558	4731	4732	4742	4743	4887	4888	5000	5001
		5038	5162	5163	5263	5389	5390	5455	5456	5612	5613	5652	5653	5763
		5764	5873	5874	6129	6147	6271	6272	6356	6357	6431	6432	6534	6535
		6628	6629	6767	6768	6859	6860	6920						
FUTABL	002322	1698#	5974											
GNS =	***** U	629	1858	1860	1863	1865	1867	1870	1872	1917	1948	1961	1965	1996
		2020	2022	2024	2026	2065	2069	2073	2126	2128	2154	2158	2162	2186
		2190	2194	2213	2215	2217	2221	6949	6953	7556	7560	7565	7567	7569
		7574	7576	7694	7698	7713	7717	7721	7725	7729	7731	7733	7735	7739
		7743	7749	8160	8164	8168	8172	8176	8180	8184	8188	8192	8196	8200

PRITEM	040746	1911*	6968*	7553*	8148#	8234*	8252	8256*										
PROG	= 001000	1460#																
PRO	= 000000	628#																
PR1	= 000040	628#																
PR2	= 000100	628#																
PR3	= 000140	628#																
PR4	= 000200	628#																
PR5	= 000240	628#																
PR6	= 000300	628#																
PR7	= 000340	628#																
PS	= 177776	628#	1842*	2235*	2264*	2289*	2817*	2990*	6946*	7552*								
PSEL	= 002000	1443#																
PSU	= 000001	1597#																
PSW	= 177776	628#																
PUTREG	035014	6300	6385	6457	6467	6564	6651	7501	7595#									
PWRVEC	= 000024	628#	1821*	1822*	8364*	8365*	8374*	8377*	8389*	8390*								
RA	000200	638#	7735															
RDCHR	= 104410	8039	8360#															
RDLIN	= 104411	8071	8361#															
RDOCT	= 104412	1917	7569	7576	7699	7717	8362#											
RDY	= 000200	1440#	2236	2247	2265	2320	2564	2640	2714	2833	2947	2985	3003	3145				
		3218	3329	3399	3537	3672	3787	3888	4451	4579	4766	4907	5015	5060				
		5183	5284	5405	5485	5628	5682	5786	5900	6049	6289	6374	6447	6556				
		6643	6787	6879	7172													
READAT	002346	1709#	2533	2545	2557	3197	3211	4337	4351	5163	5177	5264	5278	5456				
		5471	5764	5778	5789	5874	5888	5903										
READIN	002362	1715#																
RECALI	002326	1701#	2635	3627	6495	6716	6819											
REFOR	002350	1710#	2794	2978	3515	3530	3866	3880	3891	4240	4253	4732	4743	4760				
		4888	4901	6432	6437	6629	6637	6860	6873									
REGADR	004500	1730#	1980*	2325*	2332*	2723*	2730*	2958*	2972*	3677*	3684*	3796*	3803*	3897*				
		3904*	5795*	5802*	5909*	5916*	7510*	9824	9826	9838	9841							
REGSAV	040602	8106#																
REGSA1	040610	1929	8107#															
REINTO	003434	1724#	2406	2530	2542	2575	2611	2678	2779	2856	2885	2924	3026	3059				
		3110	3194	3267	3293	3503	3512	3547	3594	3745	3752	3839	3853	3863				
		3913	3941	4227	4237	4324	4334	4711	4729	4740	4773	4838	4873	4885				
		4917	4953	5150	5160	5190	5227	5251	5261	5340	5444	5453	5518	5548				
		5761	5812	5835	5862	5871	5925	5951	6429	6597	6626	6671	6831	6857				
		6890																
RELEAS	002332	1703#																
RESVEC	= 000010	628#																
RETCL	002356	1713#																
RHAS	002216	1638#	1974	1996	5894*													
RHBA	002174	1626#	2468	2574	2738	2855	3025	3154	3338	3411	3427	3546	3811	3912				
		4059	4156	4273	4585	4772	4916	5072	5083	5189	5517	5714	5811	5924				
		5997*	6020*	6029*	6088	6127*	6220	7625*										
RHBAE	002240	1650#	1957															
RHCA	002212	1636#	2476	2582	4625	4633	4803	4808	6148*	7107*	7125*	7622*						
RHCC	002234	1645#	3034	4628	4636	5491	5688	5930	7238	7601								
RHCS1	002200	1631#	2297*	2310*	2454*	2559*	2563	2637*	2707*	2713	2808*	2825*	2832	2858				
		2910*	2953*	3002	3028	3140*	3144	3213*	3217	3231	3324*	3328	3394*	3398				
		3438	3532*	3536	3564	3629*	3662*	3689	3781*	3786	3882*	3887	4041*	4063				
		4138*	4160	4215*	4255*	4277	4353*	4363	4447*	4450	4575*	4578	4591	4762*				
		4765	4778	4903*	4906	5011*	5014	5056*	5059	5088	5179*	5182	5280*	5283				
		5291	5360	5400*	5404	5422*	5473*	5484	5496	5565	5623*	5627	5670*	5681				

UNS = 040000	1483#													
UPE = 020000	1419#													
US1 = 003001	1406#													
US2 = 000002	1407#													
US4 = 000004	1408#													
UWR = 000010	1599#													
VUF = 000002	1598#													
VU30 = 010000	1559#													
VV = 000100	1457#	2316	2327	2338	2725	2948	3679	3798	3899	5797	5911	7163	7180	
WAITBT 033326	7259#	7272*	7280	7285	7358*	7365	7370	9818	9821					
WAITPC 033322	7257#	7269*	7270*	7355*	7356*	9818	9821							
WAITRE 033324	7258#	7271*	7280	7285	7289	7301	7357*	7365	7370	7374	7400	9818	9821	
WAITTM 033330	7237*	7260#	7261*	7309	7319	9821								
WAIT.P 033332	7261#													
WAIT.T 033574	7350#	8363												
WAT = 104413	2314	2457	2562	2638	2712	2811	2831	2911	2983	3001	3143	3216	3327	
	3397	3535	3633	3666	3785	3886	4043	4140	4217	4257	4355	4449	4577	
	4764	4905	5013	5058	5181	5282	5403	5424	5483	5626	5680	5784	5898	
	6047	6287	6372	6445	6499	6554	6641	6720	6785	6823	6877	8363#		
WC 002254	1668#	2346	2487	2593	2748	2869	3042	3164	3250	3347	3464	3574	3703	
	3822	3923	4087	4182	4299	4388	4647	4820	4925	5125	5208	5323	5527	
	5724	5821	5937	6098	6230	7600	8160	9853						
WCE = 040000	1420#													
WCF = 000040	1474#													
WCU = 000001	1547#													
WLE = 004000	1480#													
WRCHDT 002340	1706#													
WRCHEK 002336	1705#													
WRFROM 002370	1723#	2383	2394	2422	2434	2469	2511	2515	2610	2648	2657	2663	2672	
	2688	2739	2839	2846	2884	3009	3016	3058	3099	3120	3155	3184	3266	
	3282	3303	3339	3365	3375	3412	3428	3484	3493	3497	3593	3726	3735	
	3762	3812	3840	3940	3969	3979	3988	4007	4018	4109	4119	4413	4422	
	4434	4468	4479	4489	4506	4517	4526	4544	4555	4586	4677	4688	4699	
	4705	4837	4860	4867	4952	4979	4988	4998	5026	5036	5073	5084	5144	
	5226	5245	5339	5369	5378	5387	5434	5438	5547	5576	5585	5591	5600	
	5610	5636	5640	5650	5715	5747	5751	5834	5857	5950	6020	6022	6126	
	6255	6269	6340	6354	6511	6521	6532	6605	6615	6670	6744	6754	6765	
	6838	6846	6889											
WRIDAT 002342	1707#	3378	3392	4122	4136	5039	5054	5653	5668					
WRIFOR 002344	1708#	2425	2437	2452	2691	2705	2717	3123	3138	3306	3322	3765	3779	
	3790	4010	4021	4039	4437	4445	4547	4558	4573	5001	5009	5390	5398	
	5613	5621	6144	6272	6280	6357	6365	6535	6543	6768	6776			
WRL = 004000	1462#													
WRU = 000400	1555#													
WSU = 000004	1549#													
XE2 007346	2010	2032#												
\$AUTOB 001134	670#	1881*	7963	8065										
\$BDADR 001122	665#	5535*	5732*	7175*	7182*	9830	9841	9844	9846					
\$BDDAT 001126	667#	1978*	2324*	2331*	2722*	2729*	2956*	2970*	3676*	3683*	3795*	3802*	3896*	
	3903*	5794*	5801*	5908*	5915*	7289*	7301*	7374*	7400*	7509*	7671*	9818	9821	
	9824	9826	9836	9838	9841									
\$BELL 001216	698#	7928	8058	8112	8132									
\$CHARC 037314	7874*	7883*	7887	7896*	7901#									
\$CKSWR 037626	7955#	8359												
\$CMTAG 001100	654#	1810	1817	1823	1824									
\$CM1 = 000006	684#	685#	686#	687#	688#	689#	690#							

	5850	5851#	5961	5962#	5968	6119	6120#	6244	6245#	6329	6330#	6389	6475
\$TPB 001152	6476#	6695	6698	6699#	6944	6945#							
\$TPFLG 001157	678#	7893*	7903										
\$TPS 001150	682#	7855	7903										
\$STRAP 042324	677#	7891	7903										
\$STRAP2 042346	1819	8342#											
\$TRP = 000014	8349#	8352											
\$TRPAD 042360	8352#	8354#	8355#	8356#	8357#	8358#	8359#	8360#	8361#	8362#	8363#	8364#	
\$TSTNM 001102	8347	8352#											
\$TTYIN 040366	655#	6956*	6988*	7757	7771	7793*	7798	7802	8109	8132			
\$TYPBN= ***** U	8036	8037	8054	8058#									
\$TYPDS 036654	8358												
\$TYPE 037100	7802#	8357											
\$TYPEC 037250	7855#	8352	8353										
\$TYPEX 037316	7876	7882	7886	7891#	7892	8004							
\$TYPOC 042122	7897	7899	7902#										
\$TYPON 042136	8295#	8354											
\$TYPOS 042076	8294	8297#	8356										
\$XTSTR 036414	8290#	8355											
\$\$GET4= 000000	7760#												
\$OFILL 042321	7004#												
\$4OCAT= ***** U	8291*	8295*	8305	8340#									
. = 062372	7758	8119											
	629#	632#	633#	634#	637#	641#	653#	654#	702	1722#	1723#	1724#	1741#
	1757#	1813	1826	1827	1860#	1865#	1867#	1961#	1965#	1996#	2020#	2022#	2026#
	2065#	2069#	2073#	2128#	2154#	2186#	2190#	2194#	2215#	2217#	2221#	6949#	6953#
	7011	7015	7565#	7569#	7574#	7713#	7721#	7725#	7729#	7735#	7749#	7801	7802
	7855#	7903	7906#	8058#	8065	8105	8132	8160#	8164#	8196#	8200#	8204#	8208#
	8212#	8216#	8228#	8232#	8288#	8376	8397	9899#					

. ABS. 062372 000

ERRORS DETECTED: 0

CZRJJC,DSKW:CZRJJC.SEQ/SOL/NL:TOC:MC:ME:CND=SYSMAC.SML[400,1066],CZRJJC.P11[400,1424]
 RUN-TIME: 42 47 2 SECONDS
 RUN-TIME RATIO: 256/93=2.7
 CORE USED: 37K (73 PAGES)

DOCUMENT PAGES: 227