

RP04/5/6

RP04/5/6 CTRLR 1
CZRJID0

AH 9220D MC

NOV 1979

COPYRIGHT 76 79

digital

FICHE 1 OF 2

MADE IN USA

This microfiche card contains a grid of frames, each containing a small table of data. The data is organized into columns and rows, with some frames containing headers or specific labels. The text is very small and difficult to read, but the overall structure is consistent across the grid.

Frame	Header	Row 1	Row 2	Row 3	Row 4
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

.REM 2

IDENTIFICATION

PRODUCT CODE: AC-9218D-MC
PRODUCT NAME: CZRJID0 RP04/5/6 FUNCTIONAL CONTROLLER TEST PART I
DATE CREATED: MAY, 1979
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) 1976, 1979 DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION

DIGITAL	PDP	UNIBUS	MASSBUS
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 IS USED TO TEST RP04/5/6 DEVICE CONTROL LOGIC CONNECTED TO 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 PROGRAMS DZRJGA AND DZRJHA HAVE 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.
AND IT ASSUMES THAT MAINDEC-11-DZRJH-(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 204---TO SELECT NON-DEFAULT ADDRESSES
START AT ADDRESS 210---FOR UNIT SELECTION
START AT ADDRESS 220---FOR NO MANUAL INTERVENTION

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 START 200 WITH THE FOLLOWING EXCEPTION. THE PROGRAM WILL INTERROGATE THE OPERATOR FOR A NON-STANDARD C.S.R AND VECTOR ADDRESS BEFORE STARTING. ONCE THE QUESTIONS HAVE BEEN CORRECTLY ANSWERED, AND IT IS ALSO NECESSARY TO SELECT A PARTICULAR UNIT FOR TEST (TYPICAL PROGRAM EXECUTION FROM ADDRESS 210), OR IT IS NECESSARY TO RUN THE PROGRAM WITHOUT MANUAL INTERVENTION (TYPICAL PROGRAM EXECUTION FROM ADDRESS 220), THE PROCESSOR MAY BE HALTED AND RESTARTED FROM THE DESIRED RESTART ADDRESS. IF ALL UNITS ARE TO BE CHECKED, THE PROCESSOR NEED NOT BE TOUCHED. THE PROGRAM WILL AUTOMATICALLY RESTART AT ADDRESS 200 AFTER RECEIVING THE NEW DEVICE PARAMETERS.

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.

220 START

ALL SWITCHES MUST BE DOWN FOR WORST CASE RUN. WITH THIS STARTING ADDRESS THE PROGRAM WILL NOT RUN THOSE TESTS THAT NEED

MANUAL INTERVENTION. THIS IS RECOMMENDED ONLY FOR
DEBUGGING WHERE THE ERROR IS NOT IN A TEST THAT REQUIRES MANUAL
INTERVENTION

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 BEIDNG RUN ON A SWITCHLES PROCESSOR (I. E.
AN 11/34) IT WILL DETERMINE THAT A HARWARE SWITCH REGISTER IS
NOT PRESENT, AND WILL USE "SOFTWARE" SWITCH REGISTER. THE
SETTINGS OF THE SWITCHES ARE CONTROLLED THROUGH A KEYBOARD
ROUTINE WHICH IS CALED BY TYPING A 'COBNTROL G'. THE PROGRAM
WILL RECOGNIZE A 'CONTROL G' AT ANY TIME EXCEPT WHEN IT IS AR
A HIGHER PRIORITY PROCESSING AN RPO4/5/6 INTERRUPT. THE
"SOFTWARE" SWITCH VALUE S 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 REGISTER, 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 BE 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 SUBSEQUENT 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 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 OPEN TO 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 DZRPS, 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 SWITCH REGISTER FEATURE WHILE 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 1 MINUTES PROVIDED AN OPERATOR IS PRESENT TO CARRY OUT THE TYPED INSTRUCTIONS IMMEDIATELY. SUBSEQUENT PASSES WILL TAKE 30 SECONDS WHETHER AN OPERATOR IS THERE OR NOT.

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) - SFEK 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

IT = 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.
a

CZRJ1D0, RP04/5/6 FCTNL CTRL1
CZRJ1D.P11 28-MAR-79 09:03

MACY11 30A(1052) 25-MAY-79 10:30 N 1
OPERATIONAL SWITCH SETTINGS PAGE 14

SEQ 0013

660	:*	7	STOP FURTHER COMPARES IF SW08 IS LOW
661	:*	6	TYPE ALL REG. WITH ERROR IF SW8 LOW
662	:*	5	MULT ADDR PLUG TEST IF SW08 IS LOW
663			

```
664 .SBTTL BASIC DEFINITIONS
665
666 ;*INITIAL ADDRESS OF THE STACK POINTER *** 1000 ***
667 001000 STACK= 1000
668 .EQUIV EMT,ERROR ;;BASIC DEFINITION OF ERROR CALL
669 .EQUIV IOT,SCOPE ;;BASIC DEFINITION OF SCOPE CALL
670
671 ;*MISCELLANEOUS DEFINITIONS
672 000011 HT= 11 ;;CODE FOR HORIZONTAL TAB
673 000012 LF= 12 ;;CODE FOR LINE FEED
674 000015 CR= 15 ;;CODE FOR CARRIAGE RETURN
675 000200 CRLF= 200 ;;CODE FOR CARRIAGE RETURN-LINE FEED
676 177776 PS= 177776 ;;PROCESSOR STATUS WORD
677 .EQUIV PS,PSW
678 177774 STKLMT= 177774 ;;STACK LIMIT REGISTER
679 177772 PIRQ= 177772 ;;PROGRAM INTERRUPT REQUEST REGISTER
680 177570 DSWR= 177570 ;;HARDWARE SWITCH REGISTER
681 177570 DDISP= 177570 ;;HARDWARE DISPLAY REGISTER
682
683 ;*GENERAL PURPOSE REGISTER DEFINITIONS
684 000000 R0= %0 ;;GENERAL REGISTER
685 000001 R1= %1 ;;GENERAL REGISTER
686 000002 R2= %2 ;;GENERAL REGISTER
687 000003 R3= %3 ;;GENERAL REGISTER
688 000004 R4= %4 ;;GENERAL REGISTER
689 000005 R5= %5 ;;GENERAL REGISTER
690 000006 R6= %6 ;;GENERAL REGISTER
691 000007 R7= %7 ;;GENERAL REGISTER
692 000006 SP= %6 ;;STACK POINTER
693 000007 PC= %7 ;;PROGRAM COUNTER
694
695 ;*PRIORITY LEVEL DEFINITIONS
696 000000 PR0= 0 ;;PRIORITY LEVEL 0
697 000040 PR1= 40 ;;PRIORITY LEVEL 1
698 000100 PR2= 100 ;;PRIORITY LEVEL 2
699 000140 PR3= 140 ;;PRIORITY LEVEL 3
700 000200 PR4= 200 ;;PRIORITY LEVEL 4
701 000240 PR5= 240 ;;PRIORITY LEVEL 5
702 000300 PR6= 300 ;;PRIORITY LEVEL 6
703 000340 PR7= 340 ;;PRIORITY LEVEL 7
704
705 ;*'SWITCH REGISTER' SWITCH DEFINITIONS
706 100000 SW15= 100000
707 040000 SW14= 40000
708 020000 SW13= 20000
709 010000 SW12= 10000
710 004000 SW11= 4000
711 002000 SW10= 2000
712 001000 SW09= 1000
713 000400 SW08= 400
714 000200 SW07= 200
715 000100 SW06= 100
716 000040 SW05= 40
717 000020 SW04= 20
718 000010 SW03= 10
719 000004 SW02= 4
```



```

720      000002      SW01= 2
721      000001      SW00= 1
722      .EQUIV SW09,SW9
723      .EQUIV SW08,SW8
724      .EQUIV SW07,SW7
725      .EQUIV SW06,SW6
726      .EQUIV SW05,SW5
727      .EQUIV SW04,SW4
728      .EQUIV SW03,SW3
729      .EQUIV SW02,SW2
730      .EQUIV SW01,SW1
731      .EQUIV SW00,SW0
732
733      ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
734      100000      BIT15= 100000
735      040000      BIT14= 40000
736      020000      BIT13= 20000
737      010000      BIT12= 10000
738      004000      BIT11= 4000
739      002000      BIT10= 2000
740      001000      BIT09= 1000
741      000400      BIT08= 400
742      000200      BIT07= 200
743      000100      BIT06= 100
744      000040      BIT05= 40
745      000020      BIT04= 20
746      000010      BIT03= 10
747      000004      BIT02= 4
748      000002      BIT01= 2
749      000001      BIT00= 1
750      .EQUIV BIT09,BIT9
751      .EQUIV BIT08,BIT8
752      .EQUIV BIT07,BIT7
753      .EQUIV BIT06,BIT6
754      .EQUIV BIT05,BIT5
755      .EQUIV BIT04,BIT4
756      .EQUIV BIT03,BIT3
757      .EQUIV BIT02,BIT2
758      .EQUIV BIT01,BIT1
759      .EQUIV BIT00,BIT0
760
761      ;*BASIC "CPU" TRAP VECTOR ADDRESSES
762      000004      ERRVEC= 4          ;;TIME OUT AND OTHER ERRORS
763      000010      RESVEC= 10         ;;RESERVED AND ILLEGAL INSTRUCTIONS
764      000014      TBITVEC=14         ;;"T" BIT
765      000014      TRTVEC= 14         ;;TRACE TRAP
766      000014      BPTVEC= 14         ;;BREAKPOINT TRAP (BPT)
767      000020      IOTVEC= 20         ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
768      000024      PWRVEC= 24         ;;POWER FAIL
769      000030      EMTVEC= 30         ;;EMULATOR TRAP (EMT) **ERROR**
770      000034      TRAPVEC=34         ;;"TRAP" TRAP
771      000060      TKVEC= 60          ;;TTY KEYBOARD VECTOR
772      000064      TPVEC= 64          ;;TTY PRINTER VECTOR
773      000240      PIRQVEC=240        ;;PROGRAM INTERRUPT REQUEST VECTOR
774

```

```

775 .SBTTL TRAP CATCHER
776
777      000000      .=0
778      ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
779      ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
780      ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
781      000174      000174      .=174
782 000174 000000  DISPREG: .WORD 0      ;;SOFTWARE DISPLAY REGISTER
783 000176 000000  SWREG:   .WORD 0      ;;SOFTWARE SWITCH REGISTER
784
785 .SBTTL ACT11 HOOKS
786
787      ;*****
788      ;HOOKS REQUIRED BY ACT11
789      000200      $SVPC=.      ;SAVE PC
790      000046      .=46
791 000046 041140  $ENDAD      ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOP
792      000052      .=52
793 000052 002000  .WORD 2000      ;;2)SET LOC.52 TO 2000
794      000200      .= $SVPC      ;; RESTORE PC
795
796 .SBTTL STARTING ADDRESSES
797
798      000200      .=200
799 000200 000137 005012 RA:      JMP @#BEGIN      ;NORMAL START
800 000204 000137 043660 ADDMOD: JMP @#BASECH      ;START FOR ADDRESS-MODIFICATION
801 000210 000137 004776      JMP @#BEGIN2      ;JUMP TO SELECT DRIVE START
802      000220      .=220
803 000220 000137 004762      JMP @#BEGIN1      ;JUMP TO NO OPERATOR TESTS START
804
805
806
807      ;*STARTING ADDRESS 200 FOR NORMAL STARTS
808      ;*THIS WILL TEST ALL DRIVES ON THE SYSTEM A SINGLE DRIVE AT A TIME
809
810      ;*STARTING ADDRESS 204 FOR NON-DEFAULT ADDRESS PARAMETERS
811      ;*AUTO RESTART AT ADDRESS 200 AFTER LOADING PARAMETERS
812
813      ;*STARTING ADDRESS 210 WILL TEST ONLY ONE SPECIFIED DRIVE
814
815      ;*STARTING ADDRESS 220 WILL JUMP OVER THE TESTS REQUIRING AN OPERATOR
816      ;*AT THE DRIVE
817

```

```
818          .SBTTL MEMORY MANAGEMENT DEFINITIONS
819
820          ;*KT11 VECTOR ADDRESS
821
822          000250 MMVEC= 250
823
824          ;*KT11 STATUS REGISTER ADDRESSES
825
826          177572 SR0= 177572
827          177574 SR1= 177574
828          177576 SR2= 177576
829          172516 SR3= 172516
830
831          ;*KERNEL 'I' PAGE DESCRIPTOR REGISTERS
832
833          172300 KIPDR0= 172300
834          172302 KIPDR1= 172302
835          172304 KIPDR2= 172304
836          172306 KIPDR3= 172306
837          172310 KIPDR4= 172310
838          172312 KIPDR5= 172312
839          172314 KIPDR6= 172314
840          172316 KIPDR7= 172316
841
842          ;*KERNEL 'I' PAGE ADDRESS REGISTERS
843
844          172340 KIPAR0= 172340
845          172342 KIPAR1= 172342
846          172344 KIPAR2= 172344
847          172346 KIPAR3= 172346
848          172350 KIPAR4= 172350
849          172352 KIPAR5= 172352
850          172354 KIPAR6= 172354
851          172356 KIPAR7= 172356
852
853          ;*****
854          001110          .=1110
```

```
855          .SBTTL COMMON TAGS
856
857          ;*****
858          ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
859          ;*USED IN THE PROGRAM.
860
861          001100          .=1100
862 001100          $CMTAG:          ;; START OF COMMON TAGS
863 001100 000000          $PASS: .WORD 0          ;; CONTAINS PASS COUNT
864 001102 000          $TSTNM: .BYTE 0          ;; CONTAINS THE TEST NUMBER
865 001103 000          $ERFLG: .BYTE 0          ;; CONTAINS ERROR FLAG
866 001104 000000          $ICNT: .WORD 0          ;; CONTAINS SUBTEST ITERATION COUNT
867 001106 000000          $LPADR: .WORD 0          ;; CONTAINS SCOPE LOOP ADDRESS
868 001110 000000          $LPERR: .WORD 0          ;; CONTAINS SCOPE RETURN FOR ERRORS
869 001112 000000          $ERTTL: .WORD 0          ;; CONTAINS TOTAL ERRORS DETECTED
870 001114 000          $ITEMB: .BYTE 0          ;; CONTAINS ITEM CONTROL BYTE
871 001115 001          $ERMAX: .BYTE 1          ;; CONTAINS MAX. ERRORS PER TEST
872 001116 000000          $ERRPC: .WORD 0          ;; CONTAINS PC OF LAST ERROR INSTRUCTION
873 001120 000000          $GDADR: .WORD 0          ;; CONTAINS ADDRESS OF 'GOOD' DATA
874 001122 000000          $BDADR: .WORD 0          ;; CONTAINS ADDRESS OF 'BAD' DATA
875 001124 000000          $GDDAT: .WORD 0          ;; CONTAINS 'GOOD' DATA
876 001126 000000          $BDDAT: .WORD 0          ;; CONTAINS 'BAD' DATA
877 001130 000000          .WORD 0          ;; RESERVED--NOT TO BE USED
878 001132 000000          .WORD 0
879 001134 000          $AUTOB: .BYTE 0          ;; AUTOMATIC MODE INDICATOR
880 001135 000          $INTAG: .BYTE 0          ;; INTERRUPT MODE INDICATOR
881 001136 000000          .WORD 0
882 001140 177570          SWR: .WORD DSWR          ;; ADDRESS OF SWITCH REGISTER
883 001142 177570          DISPLAY: .WORD DDISP          ;; ADDRESS OF DISPLAY REGISTER
884 001144 177560          $TKS: 177560          ;; TTY KBD STATUS
885 001146 177562          $TKB: 177562          ;; TTY KBD BUFFER
886 001150 177564          $TPS: 177564          ;; TTY PRINTER STATUS REG. ADDRESS
887 001152 177566          $TPB: 177566          ;; TTY PRINTER BUFFER REG. ADDRESS
888 001154 000          $NULL: .BYTE 0          ;; CONTAINS NULL CHARACTER FOR FILLS
889 001155 002          $FILLS: .BYTE 2          ;; CONTAINS # OF FILLER CHARACTERS REQUIRED
890 001156 012          $FILLC: .BYTE 12          ;; INSERT FILL CHARS. AFTER A 'LINE FEED'
891 001157 000          $TPFLG: .BYTE 0          ;; 'TERMINAL AVAILABLE' FLAG (BIT<07>=0-YES)
892 001160 000000          $REGAD: .WORD 0          ;; CONTAINS THE ADDRESS FROM
893          ;; WHICH ($REGO) WAS OBTAINED
894 001162 000000          $REG0: .WORD 0          ;; CONTAINS (($REGAD)+0)
895 001164 000000          $REG1: .WORD 0          ;; CONTAINS (($REGAD)+2)
896 001166 000000          $REG2: .WORD 0          ;; CONTAINS (($REGAD)+4)
897 001170 000000          $REG3: .WORD 0          ;; CONTAINS (($REGAD)+6)
898 001172 000000          $REG4: .WORD 0          ;; CONTAINS (($REGAD)+10)
899 001174 000000          $REG5: .WORD 0          ;; CONTAINS (($REGAD)+12)
900 001176 000000          $TMP0: .WORD 0          ;; USER DEFINED
901 001200 000000          $TMP1: .WORD 0          ;; USER DEFINED
902 001202 000000          $TMP2: .WORD 0          ;; USER DEFINED
903 001204 000000          $TMP3: .WORD 0          ;; USER DEFINED
904 001206 000000          $TMP4: .WORD 0          ;; USER DEFINED
905 001210 000000          $TMP5: .WORD 0          ;; USER DEFINED
906 001212 000000          $TIMES: 0          ;; MAX. NUMBER OF ITERATIONS
907 001214 000000          $ESCAPE: 0          ;; ESCAPE ON ERROR ADDRESS
908 001216 177607 000377          $BELL: .ASCII <207><377><377>          ;; CODE FOR BELL
909 001222 077          $QUES: .ASCII /?/          ;; QUESTION MARK
910 001223 015          $CRLF: .ASCII <15>          ;; CARRIAGE RETURN
```

CZRJID, RP04/5/6 FCTNL CTLR1 MACY11 30A(1052) 25-MAY-79 10:30 ^{6 2} PAGE 20
CZRJID.P11 28-MAR-79 09:03 COMMON TAGS

SEQ 0019

911 001224 000012
912

\$LF: .ASCIZ <12> ; ;LINE FEED
;

1

913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968

.SBTTL ERROR POINTER TABLE

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

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

\$ERRTB:

;*ITEM1

EM1

;RP04 DID NOT INTERRUPT
;WAITED ON BIT DID NOT OCCUR

DH1

;PC
;WAT PC
;BIT WAITED

DT1

;REG ADDRESS
;REG CONTENTS
;RHCS1 CONTENTS
;\$ERRPC,WAITPC,WAITBT,WAITRE,\$BDDAT,CS1

DF1

;0,0,0,0,0,0

;*ITEM2

EM2

;INTERRUPT ENABLE BIT DOWN BUT
;WAITED ON BIT DID NOT OCCUR

DH1

;PC
;WAT PC
;BIT WAITED

DT1

;REG ADDRESS
;REG CONTENTS
;RHCS1 CONTENTS
;\$ERRPC,WAITPC,WAITBT,WAITRE,\$BDDAT,CS1

DF1

;0,0,0,0,0,0

;*ITEM3

EM3

;RP04 DID NOT INTERRUPT WHEN
;WAITED ON BIT DID SET

DH1

;PC
;WAT PC
;BIT WAITED

DT1

;REG ADDRESS
;RHCS1 CONTENTS
;\$ERRPC,WAITPC,WAITBT,WAITRE,\$BDDAT,CS1

DF1

;0,0,0,0,0,0

;*ITEM4

Line No.	Code	Address	Register	Description
1025				:UNIT NUMBER AND IR HIGH
1026	001330	070071	DH10	:PC
1027				:TEST NO
1028				:FAILING REG. ADR
1029				:CONTENTS OF RHCS1
1030				:CONTENTS OF RHCS2
1031				:CONTENTS OF RHDS1
1032				:CONTENTS OF RHER1
1033	001332	071716	DT10	:\$ERRPC,\$TSTNM,\$BDADR,CS1,CS2,DS1,ER1
1034	001334	072160	DF10	:0,0,0,0,0,0,0
1035				
1036				:*ITEM12
1037	001336	052062	EM12	:AFTER A POWER UP WITH
1038				:NO PACK ACKNOWLEDGE COMMAND
1039				:RHDS1 SHOULD HAVE MOL=1, VV=0
1040	001340	070071	DH10	:PC
1041				:TEST NO
1042				:FAILING REGISTER ADDR.
1043				:CONTENTS OF RHCS1
1044				:CONTENTS OF RHCS2
1045				:CONTENTS OF RHDS1
1046				:CONTENTS OF RHER1
1047	001342	071716	DT10	:\$ERRPC,\$TSTNM,\$BDADR,CS1,CS2,DS1,ER1
1048	001344	072160	DF10	:0,0,0,0,0,0,0
1049				
1050				:*ITEM13
1051	001346	052170	EM13	:AFTER A POWER UP WITHOUT
1052				:ANY INIT RHCS1 SHOULD
1053				:HAVE GO=0, DVA=1, RDY=1
1054				:IE=0, DISREGARD
1055				:ALL OTHER BITS
1056	001350	070071	DH10	:PC
1057				:TEST NO
1058				:FAILING REGISTER ADDR.
1059				:CONTENTS OF RHCS1
1060				:CONTENTS OF RHCS2
1061				:CONTENTS OF RHDS1
1062				:CONTENTS OF RHER1
1063	001352	071716	DT10	:\$ERRPC,\$TSTNM,\$BDADR,CS1,CS2,DS1,ER1
1064	001354	072160	DF10	:0,0,0,0,0,0,0
1065				
1066				:*ITEM14
1067	001356	052312	EM14	:AFTER POWER UP RHCC
1068				:SHOULD BE=0
1069	001360	067613	DH5	:PC
1070				:REG. ADDR.
1071				:GOOD DATA
1072				:RECEIVED DATA
1073	001362	071660	DT5	:\$ERRPC,REGADR,\$GDDAT,\$BDDAT
1074	001364	072144	DF5	:0,0,0,0
1075				
1076				:*ITEM15
1077	001366	052365	EM15	:PACK ACKNOWLEDGE CAUSED
1078				:AN ERROR
1079				:GOOD DATA IS BEFORE COMMAND
1080				:RECEIVED DATA IS AFTER COMMAND

1081	001370	067613	DH5	:PC
1082				:REG. ADDR.
1083				:GOOD DATA
1084				:RECEIVED DATA
1085	001372	071660	DT5	:\$ERRPC,REGADR,\$GDDAT,\$BDDAT
1086	001374	072144	DF5	:0,0,0,0
1087				
1088			:*ITEM16	
1089	001376	052526	EM16	:GIVING A NO-OP COMMAND CAUSED
1090				:AN ERROR
1091				:GOOD DATA GIVES REGISTER
1092				:CONTENTS BEFORE COMMAND
1093				:RECEIVED DATA GIVES REGISTER
1094				:CONTENTS AFTER COMMAND
1095	001400	067613	DH5	:PC
1096				:REG. ADDR.
1097				:GOOD DATA
1098				:RECEIVED DATA
1099	001402	071660	DT5	:\$ERRPC,REGADR,\$GDDAT,\$BDDAT
1100	001404	072144	DF5	:0,0,0,0
1101				
1102			:*ITEM17	
1103	001406	052654	EM17	:DRIVE CLEAR COMMAND
1104				:CAUSED AN ERROR
1105				:GOOD DATA GIVES WHAT SHOULD
1106				:BE THERE
1107				:RECEIVED DATA GIVES WHAT WAS
1108				:THERE AFTER COMMAND
1109	001410	067613	DH5	:PC
1110				:REG. ADDR.
1111				:GOOD DATA
1112				:RECEIVED DATA
1113	001412	071660	DT5	:\$ERRPC,REGADR,\$GDDAT,\$BDDAT
1114	001414	072144	DF5	:0,0,0,0
1115				
1116			:*ITEM20	
1117	001416	053011	EM20	:READ-IN COMMAND GAVE AN ERROR
1118				:GOOD DATA HAS WHAT SHOULD BE THERE
1119				:RECEIVED DATA HAS WHAT WAS
1120				:AFTER COMMAND
1121	001420	067613	DH5	:PC
1122				:REG. ADDR.
1123				:GOOD DATA
1124				:RECEIVED DATA
1125	001422	071660	DT5	:\$ERRPC,REGADR,\$GDDAT,\$BDDAT
1126	001424	072144	DF5	:0,0,0,0
1127				
1128				
1129			:*ITEM 21	
1130	001426	053155	EM21	:RHCSI CONTENTS DURING
1131				:COMMAND WAS IN ERROR
1132	001430	067613	DH5	
1133	001432	071660	DT5	
1134	001434	072144	DF5	
1135				
1136			:*ITEM 22	

1137	001436	053230	EM22	:RHDS1 CONTENTS DURING
1138				:COMM ANS WAS IN ERROR
1139	001440	067613	DH5	
1140	001442	071660	DT5	
1141	001444	072144	DF5	
1142				
1143			:*ITEM 23	
1144	001446	053303	EM23	:UNLOAD COMMAND GAVE AN ERROR
1145				:GOOD DATA GIVES WHAT SHOULD
1146				:BE THERE
1147				:RECEIVED DATA GIVES WHAT WAS
1148				:THERE AFTER COMMAND
1149	001450	067613	DH5	
1150	001452	071660	DT5	
1151	001454	072144	DF5	
1152				
1153			:*ITEM 24	
1154	001456	053446	EM24	:OFFSET COMMAND CAUSED AN ERROR
1155				:GOOD DATA IS WHAT SHOULD BE THERE
1156				:RECEIVED DATA GIVES WHAT WAS THERE
1157				:AFTER AN OFFSET COMMAND
1158	001460	067613	DH5	
1159	001462	071660	DT5	
1160	001464	072144	DF5	
1161				
1162			:*ITEM 25	
1163	001466	053611	EM25	:RETURN TO CENTER LINE COMMAND
1164				:CAUSED AN ERROR
1165				:GOOD DATA GIVES WHAT SHOULD BE
1166				:THERE
1167				:RECEIVED DATA GIVES WHAT WAS
1168				:THERE AFTER COMMAND
1169	001470	067613	DH5	
1170	001472	071660	DT5	
1171	001474	072144	DF5	
1172				
1173			:*ITEM 26	
1174	001476	053773	EM26	:500 OFFSETS CAUSED AN ERROR
1175	001500	070250	DH26	:PC
1176				:CONT. OF RHCS1
1177				:CONT. OF RHCS2
1178				:CONT. OF RHDS1
1179				:CONT. OF RHER1
1180				:CONT. OF RHER2
1181				:CONT. OF RHER3
1182	001502	071736	DT26	:\$ERRPC,CS1,CS2,DS1,ER1,ER2,ER3
1183	001504	072167	DF26	:0,0,0,0,0,0
1184				
1185			:*ITEM 27	
1186	001506	054063	EM27	:WRITE HEADER AND DATA
1187				:CAUSED IMPROPER REGISTER CHANGE
1188				:GOOD DATA GIVES WHAT
1189				:SHOULD BE THERE
1190				:RECEIVED DATA GIVES WHAT
1191				:WAS THERE AFTER COMMAND
1192	001510	067613	DH5	

1193	001512	071660		DT5	
1194	001514	072144		DF5	
1195					
1196			;*ITEM 30		
1197	001516	054301		EM30	:WRITE HEADER AND DATA
1198					:CHANGED WRITE FROM BUFFER
1199	001520	070447		DH30	:PC
1200					:WORD NO
1201					:GOOD DATA
1202					:BAD DATA
1203	001522	071760		DT30	:\$ERRPC,ERWORD,\$GDDAT,\$BDDAT
1204	001524	072177		DF30	:0,0,0,0
1205					
1206			;*ITEM 31		
1207	001526	054361		EM31	:READ HEADER AND DATA CAUSED
1208					:IMPROPER REGISTER CHANGE
1209					:GOOD DATA HAS WHAT SHOULD
1210					:BE THERE
1211					:RECEIVED DATA GIVES WHAT
1212					:WAS THERE AFTER COMMAND
1213	001530	067613		DH5	
1214	001532	071660		DT5	
1215	001534	072144		DF5	
1216					
1217			;*ITEM 32		
1218	001536	054576		EM32	:WRITE HEADER AND DATA FOLLOWED
1219					:BY A READ HEADER AND DATA
1220					:CAUSED A READ/WRITE ERROR
1221	001540	070447		DH30	
1222	001542	071760		DT30	
1223	001544	072177		DF30	
1224					
1225			;*ITEM 33		
1226	001546	054705		EM33	:READ DATA CAUSED IMPROPER REGISTER
1227					:CHANGE
1228					:GOOD DATA GIVES WHAT SHOULD BE THERE
1229					:RECEIVED DATA GIVES WHAT WAS THERE AFTER
1230					:COMMAND
1231	001550	067613		DH5	
1232	001552	071660		DT5	
1233	001554	072144		DF5	
1234					
1235			;*ITEM 34		
1236	001556	055107		EM34	:READ DATA INCORRECT
1237	001560	070447		DH30	
1238	001562	071760		DT30	
1239	001564	072177		DF30	
1240					
1241			;*ITEM 35		
1242	001566	055133		EM35	:WRITE DATA COMMAND CAUSED
1243					:IMPROPER REGISTER CHANGE
1244					:GOOD DATA GIVES WHAT SHOULD BE THERE
1245					:RECEIVED DATA GIVES REGISTER
1246					:CONTENTS AFTER WRITE DATA
1247	001570	067613		DH5	
1248	001572	071660		DT5	

1249	001574	072144	DF5	
1250				
1251				; *ITEM 36
1252	001576	055351	EM36	;WRITE DATA COMMAND CHANGED
1253				;WRITE FROM BUFFER
1254	001600	070447	DH30	
1255	001602	071760	DT30	
1256	001604	072177	DF30	
1257				
1258				; *ITEM 37
1259	001606	055426	EM37	;SEEK COMMAND CAUSED AN
1260				;ERROR
1261				;GOOD DATA GIVES WHAT SHOULD
1262				;BE THERE
1263				;RECEIVED DATA GIVES WHAT
1264				;WAS THERE AFTER SEEK COMMAND
1265	001610	067613	DH5	
1266	001612	071660	DT5	
1267	001614	072144	DF5	
1268				
1269				; *ITEM 40
1270	001616	055643	EM40	;WRITE CHECK CAUSED AN
1271				;IMPROPER REGISTER CHANGE
1272				;GOOD DATA GIVES WHAT SHOULD
1273				;BE THERE
1274				;RECEIVED DATA GIVES WHAT WAS
1275				;THERE AFTER COMMAND
1276	001620	067613	DH5	
1277	001622	071660	DT5	
1278	001624	072144	DF5	
1279				
1280				; *ITEM 41
1281	001626	056052	EM41	;LOCKING OUT WRITES BY WRITE
1282				;LOCK BUTTON CAUSED IMPROPER
1283				;REGISTER CHANGE
1284				;GOOD DATA GIVES WHAT SHOULD
1285				;BE THERE
1286				;RECEIVED DATA GIVES WHAT
1287				;WAS THERE AFTER WRITES
1288				;WERE LOCKED OUT BY
1289				;BUTTON
1290	001630	067613	DH5	
1291	001632	071660	DT5	
1292	001634	072144	DF5	
1293				
1294				; *ITEM 42
1295	001636	056333	EM42	;ATTEMPTING TO WRITE WITH WRITE
1296				;LOCKED OUT CAUSED IMPROPER
1297				;REGISTER CHANGE
1298				;GOOD DATA GIVES WHAT SHOULD
1299				;BE THERE
1300				;RECEIVED DATA GIVES WHAT WAS
1301				;THERE AFTER ATTEMPT
1302	001640	067613	DH5	
1303	001642	071660	DT5	
1304	001644	072144	DF5	

1305				
1306			;*ITEM 43	
1307	001646	056611	EM43	; WRITING WITH WRITE LOCKED
1308				; OUT CHANGED DISK DATA
1309				; GOOD DATA GIVES WHAT WAS
1310				; ON DISK BEFORE WRITE WITH
1311				; WRITE LOCK WAS ATTEMPTED
1312				; RECEIVED DATA GIVES WHAT WAS
1313				; READ BACK AFTER WRITE WITH
1314				; WRITE LOCK WAS ATTEMPTED
1315	001650	070447	DH30	
1316	001652	071760	DT30	
1317	001654	072177	DF30	
1318				
1319			;*ITEM 44	
1320	001656	057153	EM44	; ENABLING WRITES BY WRITE LOCK
1321				; BUTTON CAUSED AN ERROR
1322				; GOOD DATA GIVES WHAT SHOULD
1323				; BE THERE
1324				; RECEIVED DATA GIVES WHAT WAS
1325				; THERE AFTER WRITE LOCK
1326				; BUTTON ENABLED WRITES
1327	001660	067613	DH5	
1328	001662	071660	DT5	
1329	001664	072144	DF5	
1330				
1331			;*ITEM 45	
1332	001666	057445	EM45	; TRANSFERRING ON LAST BLOCK IE. CYLINDER
1333				; 410, SECTOR 21, TRACK 18
1334				; CAUSED IMPROPER REGISTER
1335				; CHANGE
1336				; GOOD DATA GIVES WHAT SHOULD
1337				; BE THERE
1338				; RECEIVED DATA GIVES WHAT WAS
1339				; THERE AFTER TRANSFER
1340	001670	067613	DH5	
1341	001672	071660	DT5	
1342	001674	072144	DF5	
1343				
1344			;*ITEM 46	
1345	001676	057741	EM46	; DATA READ FROM LAST
1346				; BLOCK IE. CYLINDER 410
1347				; SECTOR 21, TRACK 18 IS IN
1348				; ERROR
1349	001700	070447	DH30	
1350	001702	071760	DT30	
1351	001704	072177	DF30	
1352				
1353			;*ITEM 47	
1354	001706	060054	EM47	; TRANSFERRING FROM NONEXISTANT
1355				; SECTOR CAUSED IMPROPER
1356				; REGISTER CHANGE
1357				; GOOD DATA GIVES WHAT SHOULD
1358				; BE THERE
1359				; RECEIVED DATA GIVES WHAT WAS
1360				; THERE AFTER ATTEMPTED

1361					:TRANSFER
1362	001710	067613		DH5	
1363	001712	071660		DT5	
1364	001714	072144		DF5	
1365					
1366			:*ITEM 50		
1367	001716	060336		EM50	:TRANSFERRING FROM NONEXISTANT
1368					:SECTOR CAUSED DATA ERROR
1369					:GOOD DATA GIVES WHAT
1370					:SHOULD BE IN BUFFER
1371					:RECEIVED DATA GIVES WHAT WAS
1372					:IN BUFFER AFTER TRANSFER
1373	001720	070447		DH30	
1374	001722	071760		DT30	
1375	001724	072177		DF30	
1376					
1377			:*ITEM 51		
1378	001726	060555		EM51	:GIVING ILLEGAL FUNCTION CAUSED
1379					:IMPROPER REGISTER CHANGE
1380					:GOOD DATA GIVES WHAT SHOULD BE
1381					:THERE
1382					:RECEIVED DATA GIVES REGISTER
1383					:CONTENTS AFTER ILLEGAL FUNCTION
1384	001730	070563		DH51	:PC
1385					:REG. ADDR.
1386					:GOOD DATA
1387					:RECEIVED DATA
1388					:ILLEGAL FUNCTION
1389	001732	071774		DT51	:SERRPC,REGADR,\$GDDAT,\$BDDAT,ILLEGL
1390	001734	072204		DF51	:0.0.0.0.0
1391					
1392					
1393			:*ITEM 52		
1394	001736	061022		EM52	:WRITE DATA ON NONEXISTANT
1395					:SECTOR CAUSED IMPROPER
1396					:REGISTER CHANGE
1397					:GOOD DATA GIVES WHAT SHOULD
1398					:BE THERE
1399					:RECEIVED DATA GIVES WHAT
1400					:WAS THERE AFTER ATTEMPTED
1401					:WRITE DATA
1402	001740	067613		DH5	
1403	001742	071660		DT5	
1404	001744	072144		DF5	
1405					
1406			:*ITEM 53		
1407	001746	061277		EM53	:READ HEADER AND DATA AFTER
1408					:A SEARCH CAUSED AN ERROR
1409	001750	070447		DH30	
1410	001752	071760		DT30	
1411	001754	072177		DF30	
1412					
1413			:*ITEM 54		
1414	001756	061365		EM54	:ATTEMPTED OPERATION WITH
1415					:INVALID ADDRESS CAUSED
1416					:IMPROPER REGISTER CHANGE

1417					:GOOD DATA GIVES WHAT SHOULD
1418					:BE THERE
1419					:RECEIVED DATA GIVES WHAT WAS
1420					:THERE AFTER OPERATION
1421	001760	067613		DH5	
1422	001762	071660		DT5	
1423	001764	072144		DF5	
1424					
1425					
1426	001766	061632		:*ITEM 55 EM55	
1427					:WRITING/READING WITH EXPECTED
1428					:ADDRESS OVERFLOW ERROR CAUSED
1429					:IMPROPER REGISTER CHANGE
1430					:GOOD DATA GIVES WHAT SHOULD
1431					:BE THERE
1432					:RECEIVED DATA GIVES WHAT
1433	001770	067613		DH5	:WAS THERE AFTER OPERATION
1434	001772	071660		DT5	
1435	001774	072144		DF5	
1436					
1437					
1438	001776	062120		:*ITEM 56 EM56	
1439					:DATA READ WITH AN EXPECTED
1440					:ADDRESS OVERFLOW ERROR IS
1441					:INCORRECT
1442					:WORD NO 1 TO 260 SHOULD
1443					:BE READ
1444					:WORD NOS 261 TO 266 SHOULD
1445	002000	070447		DH30	:NOT CHANGE DUE TO READ
1446	002002	071760		DT30	
1447	002004	072177		DF30	
1448					
1449					
1450	002006	062330		:*ITEM 57 EM57	
1451					:ATTEMPTING DATA COMMAND
1452					:WITH WRONG FORMAT BIT CAUSED
1453					:IMPROPER REGISTER CHANGE
1454					:GOOD DATA GIVES WHAT SHOULD BE
1455					:THERE
1456					:RECEIVED DATA GIVES WHAT WAS
1457					:THERE AFTER ATTEMPTED DATA
1458	002010	067613		DH5	:TRANSFER
1459	002012	071660		DT5	
1460	002014	072144		DF5	
1461					
1462					
1463	002016	062622		:*ITEM 60 EM60	
1464					:ATTEMPTING TO MODIFY REGISTER
1465					:DURING AN OPERATION CAUSED
1466					:IMPROPER REGISTER CHANGE
1467					:GOOD DATA GIVES WHAT SHOULD
1468					:BE THERE
1469					:RECEIVED DATA GIVES WHAT WAS
1470					:THERE AFTER OPERATION
1471	002020	070722		DH60	:WAS COMPLETE
1472					:PC
					:REG. ADDR.

1473				:GOOD DATA
1474				:RECEIVED DATA
1475				:MODFING REGISTER
1476	002022	072012	DT60	:\$ERRPC,REGADR,\$GDDAT,\$BDDAT,\$BDADR
1477	002024	072212	DF60	:0,0,0,0,0
1478				
1479				
1480	002026	063233	:*ITEM 61	:DEVICE NOT AVAILBLE BEFORE COMMAND WAS TO BE GIVEN
1481	002030	071057	EM61	:PC
1482			DH61	:TEST NO.
1483				:PC OF JSR
1484				:RHCS1 CONTENTS
1485	002032	072030	DT61	:\$ERRPC,\$TSTNM,PCJSR,\$BDADR
1486	002034	072220	DF61	:0,0,0,0
1487				
1488				
1489	002036	063314	:*ITEM 62	:RHDS1 HAS STATUS BITS STUCK AT ONE
1490	002040	071152	EM62	:PC
1491			DH62	:TEST NO.
1492				:PC OF JSR
1493				:RHDS1 CONTENTS
1494	002042	072042	DT62	:\$ERRPC,\$TSTNM,PCJSR,\$BDADR
1495	002044	072224	DF62	:0,0,0,0
1496				
1497				
1498				
1499	002046	063375	:*ITEM 63	:RHDS1 CONTENTS DURING
1500			EM63	:COMMAND WAS IN ERROR
1501	002050	067613	DH5	
1502	002052	071660	DT5	
1503	002054	072144	DF5	
1504				
1505				
1506				
1507	002056	063451	:*ITEM 64	:RECALIBRATE COMMAND CAUSED
1508			EM64	:IMPROPER REGISTER CHANGE.
1509				:GOOD DATA GIVES WHAT SHOULD BE
1510				:THERE.
1511				:RECEIVED DATA GIVES WHAT WAS THERE
1512				:AFTER COMMAND
1513	002060	067613	DH5	
1514	002062	071660	DT5	
1515	002064	072144	DF5	
1516				
1517				
1518				
1519				
1520	002066	063670	:*ITEM65	:INTERRUPT FAILING
1521	002070	071245	EM65	:PC
1522			DH65	:TEST NO
1523				:CONTENTS OF RHCS1
1524				:CONTENTS OF RHAS
1525				:CONTENTS OF RHDS1
1526	002072	072054	DT65	:\$ERRPC,TSTNM,CS1,AS,DS1
1527	002074	072230	DF65	:0,0,0,0,0
1528				

1529					
1530			:*ITEM66		
1531	002076	063712	EM66		:HEADER AND DATA COMMAND
1532					:FOR HEAD SELECTION TEST
1533					:CAUSED AN ERROR
1534					:RHDST GIVES WHAT TRACK
1535					:WAS BEING WRITTEN ON CYLINDER 0
1536					:SECTOR 0
1537	002100	071370	DH66		:PC
1538					:RHDST
1539					:RHER1
1540					:RHER2
1541					:RHER3
1542					:RHCS1
1543					:RHCS2
1544	002102	072070	DT66		:\$ERRPC,DST,ER1,ER2,ER3,CS1,CS2
1545	002104	072235	DF66		:0,0,0,0,0,0,0
1546			:*ITEM67		
1547	002106	064104	EM67		:READ HEADER AND DATA ERROR
1548					:IN HEAD SELECTION TEST
1549					:FIRST FOUR WORDS GIVE HEADER
1550					:NEXT WORDS ARE DATA
1551					:GOOD DATA WORDS GIVE
1552					:THE TRACK NUMBER IN
1553					:BITS 4,5,6,7,8
1554	002110	070447	DH30		
1555	002112	071760	DT30		
1556	002114	072177	DF30		
1557			:*ITEM70		
1558	002116	064374	EM70		:READ HEADER AND DATA ERROR
1559					:IN DIFFERENCE LINE TEST
1560					:WORD NOS. 1-4 GIVE
1561					:HEADER
1562					:WORD NOS. 5-260 GIVE DATA
1563					:WHICH IS THE CYLINDER
1564					:ADDRESS
1565	002120	070447	DH30		
1566	002122	071760	DT30		
1567	002124	072177	DF30		
1568					
1569			:*ITEM 71		
1570	002126	064602	EM71		:FORCING OPI CAUSED IMPROPER REGISTER
1571					:CHANGE
1572					:GOOD DATA GIVES WHAT SHOULD
1573					:BE THERE
1574					:RECEIVED DATA GIVES WHAT WAS
1575					:THERE AFTER 3 INDEX PULSES
1576	002130	067613	DH5		:PC
1577					:REG. ADDR.
1578					:GOOD DATA
1579					:RECEIVED DATA
1580	002132	071660	DT5		:\$ERRPC,REGADR,\$GDDAT,\$BDDAT
1581	002134	072144	DF5		:0,0,0,0
1582			:*ITEM 74		
1583	002136	065345	EM74		:WHILE USING UNIBUS B
1584					:READ DATA CAUSED IMPROPER REGISTER

1585				:CHANGE
1586				:GOOD DATA GIVES WHAT SHOULD BE THERE
1587				:RECEIVED DATA GIVES WHAT WAS THERE AFTER
1588				:COMMAND
1589	002140	067613	DH5	
1590	002142	071660	DT5	
1591	002144	072144	DF5	
1592				
1593				
1594	002146	065273	:*ITEM 73 EM73	:WHILE USING UNIBUS B
1595				:READ DATA INCORRECT
1596	002150	070447	DH30	
1597	002152	071760	DT30	
1598	002154	072177	DF30	
1599				
1600				
1601	002156	065345	:*ITEM 74 EM74	:WHILE USING UNIBUS B
1602				:WRITE DATA COMMAND CAUSED
1603				:IMPROPER REGISTER CHANGE
1604				:GOOD DATA GIVES WHAT SHOULD BE THERE
1605				:RECEIVED DATA GIVES REGISTER
1606				:CONTENTS AFTER WRITE DATA
1607	002160	067613	DH5	
1608	002162	071660	DT5	
1609	002164	072144	DF5	
1610				
1611				
1612	002166	065611	:*ITEM 75 EM75	:WHILE USING UNIBUS B
1613				:WRITE DATA COMMAND CHANGED
1614				:WRITE FROM BUFFER
1615	002170	070447	DH30	
1616	002172	071760	DT30	
1617	002174	072177	DF30	
1618				
1619				
1620	002176	065714	:*ITEM 76 EM76	:WHILE USING UNIBUS B
1621				:WRITE CHECK CAUSED AN
1622				:IMPROPER REGISTER CHANGE
1623				:GOOD DATA GIVES WHAT SHOULD
1624				:BE THERE
1625				:RECEIVED DATA GIVES WHAT WAS
1626				:THERE AFTER COMMAND
1627	002200	067613	DH5	
1628	002202	071660	DT5	
1629	002204	072144	DF5	
1630				
1631				
1632	002206	066151	:*ITEM 77 EM77	:CURRENT CYLINDER DOES NOT REFLECT DESIRED 'RHCC'
1633	002210	071466	DH77	:PC
1634				:PC OF JSR
1635				:REGISTER ADDRESS
1636				:GOOD DATA
1637				:BAD DATA
1638	002212	072110	DT77	:\$ERRPC,PCJSR,REGADR,\$GDDAT,\$BDDAT
1639	002214	072245	DF77	:0,0,0,0,0
1640				

1641			:*ITEM 100		
1642	002216	066374	EM100		:ERROR AFTER ADDRESS PLUG CHANGE
1643	002220	067613	DH5		:PC
1644					:REGISTER ADDRESS
1645					:GOOD DATA
1646					:RECEIVED DATA
1647	002222	071660	DT5		:\$ERRPC,REGADR,\$GDDAT,\$BDDAT
1648	002224	072144	DF5		:0,0,0,0
1649					
1650			:*ITEM 101		
1651	002226	066456	EM101		:UNIT DID NOT GO OFFLINE WHEN ADDR
1652					:PLUG WAS REMOVED
1653	002230	070250	DH26		:PC
1654					:CONT OF RHCS1
1655					:CONT OF RHCS2
1656					:CONT OF RHDS1
1657					:CONT OF RHER2
1658					:CONT OF RHER3
1659	002232	071736	DT26		:\$ERRPC,CS1,CS2,DS1,ER2,ER3
1660	002234	072167	DF26		:0,0,0,0,0,0,0
1661					
1662			:*ITEM 102		
1663	002236	066540	EM102		:UNIT DID NOT COME BACK ONLINE WHEN
1664					:ADDR PLUG WAS REPLACED
1665	002240	070250	DH26		:PC
1666					:CONT OF RHCS1
1667					:CONT OF RHCS2
1668					:CONT OF RHDS1
1669					:CONT OF RHER2
1670					:CONT OF RHER3
1671	002242	071736	DT26		:\$ERRPC,CS1,CS2,DS1,ER2,ER3
1672	002244	072167	DF26		:0,0,0,0,0,0,0
1673					
1674			:*ITEM 103		
1675	002246	066617	EM103		:REGISTER CONTENTS INCORRECT BEFORE A
1676					:DIAGNOSTIC SEEK
1677	002250	070250	DH26		:PC
1678					:CONT OF RHCS1
1679					:CONT OF RHCS2
1680					:CONT OF RHDS1
1681					:CONT OF RHER2
1682					:CONT OF RHER3
1683	002252	071736	DT26		:\$ERRPC,CS1,CS2,DS1,ER2,ER3
1684	002254	072167	DF26		:0,0,0,0,0,0,0
1685					
1686			:*ITEM 104		
1687	002256	066703	EM104		:REGISTER CONTENTS INCORRECT AFTER A
1688					:DIAGNOSTIC SEEK
1689	002260	070250	DH26		:PC
1690					:CONT OF RHCS1
1691					:CONT OF RHCS2
1692					:CONT OF RHDS1
1693					:CONT OF RHER2
1694					:CONT OF RHER3
1695	002262	071736	DT26		:\$ERRPC,CS1,CS2,DS1,ER2,ER3
1696	002264	072167	DF26		:0,0,0,0,0,0,0

CZRJID, RP04/5/6 FCTNL CTLR1 MACY11 30A(1052) 25-MAY-79 10:30 ^{1 3} PAGE 35
CZRJID.P11 28-MAR-79 09:03 ERROR POINTER TABLE

SEQ 0034

1697

1698
 1699
 1700
 1701
 1702
 1703
 1704
 1705
 1706
 1707
 1708
 1709
 1710
 1711
 1712
 1713
 1714
 1715
 1716
 1717
 1718
 1719
 1720
 1721
 1722
 1723
 1724
 1725
 1726
 1727
 1728
 1729
 1730
 1731
 1732
 1733
 1734
 1735
 1736
 1737
 1738
 1739
 1740

002266 000254

 ;*RH11 REGISTERS
 ;******

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	US1= 1	;UNIT SELECT (BIT #0)
000002	US2= 2	;UNIT SELECT (BIT #1)
000004	US4= 4	;UNIT SELECT (BIT #2)
000010	BAI= 10	;BUS ADDRESS INCREMENT INHIBIT (BIT #3)
000020	UNIB= 20	;UNIBUS B DC LO (BIT #4)
000040	CLR= 40	;CLEAR (BIT #5)
000100	IR= 100	;INPUT READY (BIT #6)
000200	OR= 200	;OUTPUT READY (BIT #7)
000400	MPE= 400	;MASS BUS PARITY ERROR (BIT #8)
001000	MXF= 1000	;MISSED TRANSFER ERROR (BIT #9)
002000	PGE= 2000	;PROGRAM ERROR (BIT #10)
004000	NEM= 4000	;NON EXISTANT MEMORY (BIT #11)
010000	NED= 10000	;NON EXISTANT DRIVE (BIT #12)
020000	UPE= 20000	;UNIBUS PARITY ERROR (BIT #13)
040000	WCE= 40000	;WRITE CHECK ERROR (BIT #14)
100000	DLT= 100000	;DATA LATE (BIT #15)

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

```
1741 ;*****
1742 ;*RP04 REGISTERS
1743 ;*****
1744
1745
1746
1747 ;*CONTROL AND STATUS 1 REGISTER. (#00)
1748
1749 000001 GO= 1 ;GO (BIT #0)
1750 000100 IE= 100 ;INTERRUPT ENABLE (BIT #6)
1751 000200 RDY= 200 ;READY (BIT #7)
1752 000400 A16= 400 ;HIGH ORDER UNIBUS BITS (BIT #8)
1753 001000 A17= 1000 ;HIGH ORDER UNIBUS BITS (BIT #9)
1754 002000 PSEL= 2000 ;PORT SELECT (BIT #10)
1755 004000 DVA= 4000 ;DEVICE AVAILABLE (BIT #11)
1756 020000 MLPE= 20000 ;MASSBUSS PARITY ERROR (BIT #13)
1757 040000 TRE= 40000 ;TRANSFER ERROR (BIT #14)
1758 100000 SC= 100000 ;SPECIAL CONDITION (BIT #15)
1759
1760 ;*STATUS REGISTER (RHDS1) (#01)
1761
1762 000001 DFF5= 1 ;DRIVE FORWARD 5''/SEC. (BIT #0)
1763 000002 DFF20= 2 ;DRIVE FORWARD 20''/SEC. (BIT #1)
1764 000004 DIGB= 4 ;DRIVE TO INNER GAVRD BAND (BIT #2)
1765 000010 GRV= 10 ;GO REVERSE (BIT #3)
1766 000020 DL64= 20 ;DIFFERENCE LESS THAN 64 (BIT #4)
1767 000040 DE1= 40 ;DIFFERENCE EQUALS 1 (BIT #5)
1768 000100 VV= 100 ;VOLUME VALID (BIT #6)
1769 000200 DRY= 200 ;DRIVE READY (BIT #7)
1770 000400 DPR= 400 ;DRIVE PRESENT (BIT #8)
1771 001000 PROG= 1000 ;PROGRAMABLE (BIT #9)
1772 002000 LBT= 2000 ;LAST SECTOR TRANSFERRED (BIT #10)
1773 004000 WRL= 4000 ;WRITE LOCK (BIT #11)
1774 010000 MOL= 10000 ;MEDIUM ON-LINE (BIT #12)
1775 020000 PIP= 20000 ;POSITIONING OPERATION IN PROGRESS (BIT #13)
1776 040000 ERR= 40000 ;COMPOSIT ERROR. (BIT #14)
1777 100000 ATA= 100000 ;ATTENTION ACTIVE (BIT #15)
1778
1779 ;*ERROR REGISTER #01 (RHER1) (#02)
1780 000001 ILF= 1 ;ILLEGAL FUNCTION (BIT #0)
1781 000002 ILR= 2 ;ILLEGAL REGISTER (BIT #1)
1782 000004 RMR= 4 ;REGISTER MODIFICATION REFUSED (BIT #2)
1783 000010 PAR= 10 ;PARITY ERROR (BIT #3)
1784 000020 FER= 2J ;FORMAT ERROR (BIT #4)
1785 000040 WCF= 40 ;WRITE CLOCK FAIL (BIT #5)
1786 000100 ECH= 100 ;ECC HARD ERROR (BIT #6)
1787 000200 HCE= 200 ;HEADER COMPARE ERROR (BIT #7)
1788 000400 HCRC= 400 ;HEADER CRC ERROR (BIT #8)
1789 001000 AOE= 1000 ;ADDRESS OVERFLOW ERROR (BIT #9)
1790 002000 IAE= 2000 ;INVALID ADDRESS ERROR (BIT #10)
1791 004000 WLE= 4000 ;WRITE LOCK ERROR (BIT #11)
1792 010000 DTE= 10000 ;DRIVE TIMING ERROR (BIT #12)
1793 020000 OPI= 20000 ;OPERATION INCOMPLETE (BIT #13)
1794 040000 UNS= 40000 ;DRIVE UNSAFE (BIT #14)
1795 100000 DCK= 100000 ;DATA CHECK ERROR (BIT 15)
1796
```

```
1797 ;*MAINTAINABILITY REGISTER (RHMR)(#03)
1798
1799 000001 DMD= 1 ;DIAGINOSTIC MODE (BIT #0)
1800 000002 MCLK= 2 ;MAINTAINABILITY CLOCK (BIT #1)
1801 000004 MINX= 4 ;MAINTAINABILITY INDEX (BIT #2)
1802 000010 MSTCK= 10 ;MAINTAINABILITY SECTOR CLOCK (BIT #3)
1803 000020 MRD= 20 ;MAINTAINABILITY READ (BIT #4)
1804 000040 MWR= 40 ;MAINTAINABILITY WRITE (BIT #5)
1805 001000 DTSY= 1000 ;MAINTAINABILITY SYNC DETECTED (BIT #9)
1806
1807 ;*ATTENTION SUMMARY PSEUDO-REGISTER (RHAS) (#04)
1808
1809 000001 AT0= 1 ;DEVICE 0 (BIT #0)
1810 000002 AT1= 2 ;DEVICE 1 (BIT #1)
1811 000004 AT2= 4 ;DEVICE 2 (BIT #2)
1812 000010 AT3= 10 ;DEVICE 3 (BIT #3)
1813 000020 AT4= 20 ;DEVICE 4 (BIT #4)
1814 000040 AT5= 40 ;DEVICE 5 (BIT #5)
1815 000100 AT6= 100 ;DEVICE 6 (BIT #6)
1816 000200 AT7= 200 ;DEVICE 7 (BIT #7)
1817
1818 ;*DESIRED SECTOR/TRACK ADDRESS REGISTER (RHDST) (#1)
1819 ;*EACH BIT IS CALIED BY BIT NUMBER
1820
1821
1822
1823
1824
1825 ;*DRIVE TYPE REGISTER (RHDT) (#06)
1826 ;*EACH BIT IS CALLED BY BIT NUMBER
1827
1828
1829
1830
1831
1832 ;*LOOK-AHEAD REGISTER (RHLA) (#07)
1833
1834 000001 EXT1= 1 ;EXTENSION 1 (BIT #0)
1835 000002 EXT2= 2 ;EXTENSION 2 (BIT #1)
1836 000004 EXT4= 4 ;EXTENSION 3 (BIT #2)
1837 000010 EXT10= 10 ;EXTENSION 4 (BIT #3)
1838 000020 EXT20= 20 ;EXTENSION 5 (BIT #4)
1839 000040 EXT40= 40 ;EXTENSION 6 (BIT #5)
1840 000100 SC1= 100 ;SECTOR COUNT FIELD 0 (BIT #6)
1841 000200 SC2= 200 ;SECTOR COUNT FIELD 1 (BIT #7)
1842 000400 SC4= 400 ;SECTOR COUNT FIELD 2 (BIT #8)
1843 001000 SC10= 1000 ;SECTOR COUNT FIELD 3 (BIT #9)
1844 002000 SC20= 2000 ;SECTOR COUNT FIELD 4 (BIT #10)
1845 004000 TRK1= 4000 ;TRACK FIELD 1 (BIT #11)
1846 010000 TRK2= 10000 ;TRACK FIELD 2 (BIT #12)
1847 020000 TRK4= 20000 ;TRACK FIELD 3 (BIT #13)
1848 040000 TRK10= 40000 ;TRACK FIELD 4 (BIT #14)
1849 100000 TRK20= 100000 ;TRACK FIELD 5 (BIT #15)
1850
1851 ;*ERROR REGISTER #2 (RHER2) (#10)
1852
```

1853	000001	WCU= 1	;WRITE CURRENT UNSAFE (BIT #0)
1854	000002	CSF= 2	;CURRENT SINK FAILURE (BIT #1)
1855	000004	WSU= 4	;WRITE SELECT UNSAFE (BIT #2)
1856	000010	CSU= 10	;CURRENT SWITCH UNSAFE (BIT #3)
1857	000020	MSE= 20	;MOTOR SEQUENCE ERROR (BIT #4)
1858	000040	TDF= 40	;TRANSITIONS DETECTOR FAILURE (BIT #5)
1859	000100	TUF= 100	;TRANSITIONS UNSAFE (BIT #6)
1860	000200	FEN= 200	;FAILSAFE ENABLED (BIT #7)
1861	000400	WRU= 400	;WRITE READY UNSAFE (BIT #8)
1862	001000	MHS= 1000	;MULTIPLE HEAD SELECT (BIT #9)
1863	002000	NHS= 2000	;NO HEAD SELECTION (BIT #10)
1864	004000	IXE= 4000	;INDEX ERROR (BIT #11)
1865	010000	VU30= 10000	;30VOLT UNSAFE (BIT #12)
1866	020000	PLU= 20000	;PLO UNSAFE (BIT #13)
1867	100000	ACU= 100000	;ACUNSAFE (BIT #15)
1868			
1869			;*OFFSET REGISTER (RHOF) (#11)
1870			
1871	000001	OF25= 1	;OFFSET 25 MICRO INCHES (BIT #0)
1872	000002	OF50= 2	;OFFSET 50 MICRO INCHES (BIT #1)
1873	000004	OF100= 4	;OFFSET 100 MICRO INCHES (BIT #2)
1874	000010	OF200= 10	;OFFSET 200 MICRO INCHES (BIT #3)
1875	000020	OF400= 20	;OFFSET 400 MICRO INCHES (BIT #4)
1876	000040	OF800= 40	;OFFSET 800 MICRO INCHES (BIT #5)
1877			
1878	000200	OFREV= 200	;OFFSET NEGATIVE (REVERSE) (BIT #5)
1879	002000	MCI= 2000	;HEADER COMPARE INHIBIT (BIT #10)
1880	004000	ECI= 4000	;ERROR CORRECTION CODE INHIBIT (BIT #11)
1881	010000	FMT22= 10000	;FORMAT BIT (BIT #12)
1882			
1883			;*DESIRED CYLINDER ADDRESS (RHCA) (#12)
1884			;*EACH BIT IS CALLED BY BIT NUMBER.
1885			
1886			
1887			
1888			
1889			;*CURRENT CYLINDER ADDRESS (RHCC) (#13)
1890			;*EACH BIT IS CALLED BY BIT NUMBER
1891			
1892			
1893			
1894			
1895			;*SERIAL NUMBER REGISTER (RHSN) (#14)
1896			;*EACH IS CALLED BY BIT NUMBER
1897			
1898			
1899			
1900			
1901			;*ERROR REGISTER #03 (RHER3) (#15)
1902			
1903	000001	PSU= 1	;PACK SPEED UNSAFE (BIT #0)
1904	000002	VUF= 2	;VELOCITY UNSAFE (BIT #1)
1905	000010	UWR= 10	;ANY UNSAFE EXCEPT READ/WRITE (BIT #3)
1906	000020	PRE= 20	;DISK PACK ROTATION ERROR (BIT #4)
1907	000040	ACL= 40	;AC LOW (BIT #5)
1908	000100	DCL= 100	;DC LOW (BIT #6)

1909	020000	ACE= 20000	; ADDRESS CHANGE ERROR (BIT #13)
1910	040000	SKI= 40000	; SEEK INCOMPLETE (BIT #14)
1911	100000	OCYL= 100000	; OFF CYLINDER (BIT #15)
1912			
1913			
1914			
1915		; *ECC POSITION REGISTER (RHEC1) (#16)	
1916		; *EACH BIT IS CALLED BY BIT NUMBER	
1917			
1918			
1919			
1920			
1921		; *ECC PATTERN REGISTER (RHEC2) (#17)	
1922		; *EACH BIT IS CALLED BY BIT NUMBER	
1923			
1924			
1925			
1926			
1927			
1928			
1929			
1930			
1931			

1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974

.SBTTL REGISTER ADDRESSES

;*RP04/5/6 DISK I/O REGISTER LOCATED IN THE RH11 CONTROLLER

RHDB: 176722 ;DATA BUFFER
RHWC: 176702 ;WORD COUNT
RHBA: 176704 ;BUS ADDRESS
RHCS2: 176710 ;CONTROL AND STATUS 2

;*RP04/5/6 DISK I/O REGISTERS LOCATED IN THE DEVICE CONTROL LOGIC (DCL)

RHCS1: 176700 ;CONTROL AND STATUS 1
RHER1: 176714 ;ERROR #1
RHDST: 176706 ;DESIRED SECTOR/TRACK ADDRESS
RHER2: 176740 ;ERROR #2
RHOF: 176732 ;OFFSET
RHCA: 176734 ;DESIRED CYLINDER ADDRESS
RHER3: 176742 ;ERROR #3
RHAS: 176716 ;ATTENTION SUMMARY
RHMR: 176724 ;MAINTAINABILITY
RHDS1: 176712 ;DRIVE STATUS
RHDT: 176726 ;DRIVE TYPE
RHSN: 176730 ;SERIAL NUMBER
RHEC1: 176744 ;ECC POSITION
RHEC2: 176746 ;ECC PATTERN
RHCC: 176736 ;CURRENT CYLINDER ADDRESS
RHLA: 176720 ;LOOK-AHEAD

;*ADDITIONAL I/O REGISTERS LOCATED IN THE RH70 CONTROLLER LOGIC

RHBAE: 176750 ;BUS ADDRESS EXTENSION REGISTER
RHCS3: 176752 ;CONTROL AND STATUS REGISTER #3

;*P-CLOCK (KW11-P) I/O REGISTERS

PCLCSR: 172540 ;CONTROL AND STATUS REGISTER
PCLBUF: 172542 ;COUNT SET BUFFER
PCLCTR: 172544 ;COUNTER

1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014

;*THE FOLLOWING LOCATIONS ARE RESERVED FOR REGISTER SNAPSHOTS
;*ANY TIME THERE IS AN ERROR ALL THESE WILL BE FILLED

;*ONLY SOME MAY BE PRINTED BUT ALL WILL BE FILLED TRUE
;*FOR THE TIME JUST AFTER THE "ERROR" ERROR COMMAND

;*THIS ASSUMES THAT A REGISTER SNAPSHOT HAS BEEN TAKEN WHICH IS NOT
;*ALWAYS THE CASE - IF QUESTIONABLE CONTENTS APPEAR IN THE REGISTER
;*PRINTOUTS, CHECK THE INLINE TEST CODE TO SEE IF THE REGISTER SNAPSHOT
;*REFLECTS THE CURRENT STATE OF THE MACHINE

DB: 0 ;DATA BUFFER
WC: 0 ;WORD COUNT
BA: 0 ;BUS ADDRESS
CS2: 0 ;CONTROL AND STATUS 2

CS1: 0 ;CONTROL AND STATUS 1
ER1: 0 ;ERROR #1
DST: 0 ;DESIRED SECTOR/TRACK ADDRESS
ER2: 0 ;ERROR #2
OF: 0 ;OFFSET
CA: 0 ;DESIRED CYLINDER ADDRESS
ER3: 0 ;ERROR #3
AS: 0 ;ATTENTION SUMMARY
MR: 0 ;MAINTAINABILITY
DS1: 0 ;DRIVE STATUS
DT: 0 ;DRIVE TYPE
SN: 0 ;SERIAL NUMBER
EC1: 0 ;ECC POSITION
EC2: 0 ;ECC PATTERN
CC: 0 ;CURRENT CYLINDER ADDRESS
LA: 0 ;LOOK-AHEAD

002352 000000
002354 000000
002356 000000
002360 000000

002362 000000
002364 000000
002366 000000
002370 000000
002372 000000
002374 000000
002376 000000
002400 000000
002402 000000
002404 000000
002406 000000
002410 000000
002412 000000
002414 000000
002416 000000
002420 000000

2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047

;*FUNCTION EQUATES

;*TABLE OF FUNCTIONS FOR RHCS1 THEN "GO" BIT HAS TO BE SET

002422	FUTABL:		
002422	NOPERA:	0	;NO OPERATION
002424	UNLOAD:	2	;UNLOAD (STAND BY)
002426	RECALI:	6	;RECALIBRATE
002430	DCLEAR:	10	;DRIVE CLEAR
002432	RELEAS:	12	;RELEASE (DUAL-PORT OPERATION)
002434	SERCH:	30	;SEARCH COMMAND
002436	WRCHK:	50	;WRITE CHECK DATA
002440	WRCHDT:	52	;WRITE CHECK HEADER AND DATA
002442	WRIDAT:	60	;WRITE DATA
002444	WRIFOR:	62	;WRITE HEADER AND DATA (FORMAT)
002446	READAT:	70	;READ DATA
002450	REFOR:	72	;READ HEADER AND DATA
002452	SEECOM:	4	;SEEK COMMAND
002454	OFSETC:	14	;OFFSET COMMAND
002456	RETCL:	16	;RETURN TO CENTERLINE
002460	PKACK:	22	;PACK ACKNOWLEDGE
002462	READIN:	20	;READ IN
002464	ILLEGL:	.WORD 0	;COMPUTED ILLEGAL FUNCTION

;*DATA BUFFERS FOR READ/WRITE

002470	WRFROM:	.BLKW	274.	;WRITE FROM THIS BUFFER
003534	REINTO:	.BLKW	274.	;READ INTO THIS BUFFER

```
2048
2049
2050 ;*RESERVED CORE LOCATIONS
2051
2052 004600 000000 REGADR: 0 ;SAVE REGISTER ADDRESS HERE
2053 004602 000000 ERWORD: 0 ;SAVE ERROR WORD NUMBER HERE
2054 004604 000000 TSTNM: 0 ;TEST NUMBER
2055 004606 000000 RP4VEC: 0 ;CONTAINS ADDRESS OF LOCATION
2056 ;WHERE AN RP04 INTERRUPT IS TO VECTOR TO
2057 ;THIS MUST BE MOVED INTO 'RPVEC' TO BE
2058 ;EFFECTIVE.
2059
2060 004610 000000 OFSTVL: 0 ;OFFSET VALUE USED IN OFFSET TEST
2061
2062
2063 004612 000024 SAVERE: .BLKW 20. ;BLOCK TO SAVE REGISTERS FOR PRETEST
2064 ;HARDWARE REGISTER SNAPSHOTS - THESE
2065 ;ARE USUALLY THEN CHANGED TO REFLECT
2066 ;EXPECTED CONDITIONS AFTER THE TEST
2067 004662 000000 FINALA: 0 ;SAVE LOOK AHEAD REGISTER AT END OF OPERATION
2068 004664 000000 FINACC: 0 ;SAVE CURRENT CYLINDER REGISTER AT END OF OPERATION
2069
2070
2071 ;*TABLE FOR ATTENTION BITS
2072 ;*ATTENTION TABLE
2073
2074 004666 001 002 004 ATABLE: .BYTE 1,2,4,10,20,40,100,200
2075 004671 010 020 040
2076 004674 100 200
2077
```

```
2078
2079
2080      ;*FLAGS AND INTERNAL PROGRAM CONTROL WORDS
2081
2082 004676 000010 UNITS: .BLKW 8.      ;THIS IS FILLED WITH -1
2083 004716 000000 UNIT: .WORD 0      ;UNIT UNDER TEST
2084 004720 000000 NOUNIT: .WORD 0    ;NUMBER OF UNITS PRESENT
2085      ;USED TO KEEP TRACK OF UNIT UNDER TEST
2086 004722 000000 NUNIT: .WORD 0    ;USED TO DETERMIN IF THERE ARE MORE
2087      ;THAN ONE UNIT
2088 004724 000000 NOPUSH: 0          ;ALL ONES INDICATE NONE OF THE OPERATOR
2089      ;INTERVENTION TESTS WILL BE PERFORMED
2090 004726 000000 SELECT: .WORD 0      ;ALL ONES INDICATE UNIT TO BE SELECTED
2091 004730 000000 UNITSL: .WORD 0    ;UNIT NO. SELECTED
2092 004732 000000 UBUSB: 0          ;IF ZERO UNIBUS PRESENT
2093      ;IF ONES NO UNIBUS B
2094 004734 000000 ERFLGS: 0          ;ERROR FLAG
2095 004736 000000 FIRST: 0          ;IF ZERO WILL TYPE HEADER
2096      ;IF ONES WILL NOT TYPE HEADER
2097
2098 004740 000000 ATTENT: 0          ;ATTENTION BIT FOR PRESENT UNIT
2099 004742 000000 TOTALAT:0        ;TOTAL ATTENTION BITS
2100
2101 004744 000000 RP06: 0          ;RP06 DEVICE TYPE FLAG LOCATION
2102 004746 000000 RP05: 0          ;MEMOREX RP04 DEVICE TYPE FLAG
2103 004750 000000 RH70: 0          ;IF = 1, PROGRAM IS RUNNING ON RWPO4 SYSTEM
2104      ;IF = 0, PROGRAM IS RUNNING ON RJPO4
2105 004752 000000 INUNIT: 0        ;INITIAL UNIT NO. - USED DURING
2106      ;CHECKING ALL ADDRESS PLUG ADDRESSES
2107
2108
2109
2110
2111
2112 004754 000000 TMP0: .WORD 0      ;TEMP STORAGE
2113 004756 000000 TMP1: .WORD 0
2114 004760 000000 TMP4: .WORD 0      ;TEMP STORAGE
```

```
2115 .SBTTL
2116 .SBTTL **DIAGNOSTIC CODE**
2117 .SBTTL
2118
2119
2120 .SBTTL SETUP TESTS
2121
2122
2123 004762 012737 177777 004724 BEGIN1: MOV # -1, @#NOPUSH ; JUMP OVER OPERATOR REQUIRED TESTS
2124 004770 005037 004726 CLR @#SELECT ; DO NOT SELECT UNIT
2125 004774 000412 BR START
2126 004776 012737 177777 004726 BEGIN2: MOV # -1, @#SELECT ; SELECT UNIT
2127 005004 005037 004724 CLR @#NOPUSH ; DO NOT JUMP OVER ANY TEST
2128 005010 000404 BR START
2129 005012 005037 004726 BEGIN: CLR @#SELECT ; DO NOT SELECT UNIT
2130 005016 005037 004724 CLR @#NOPUSH ; DO NOT JUMP OVER ANY OPERATOR
2131 ; INTERVENTION TESTS - NORMAL RUN
2132
2133 005022 START:
2134 005022 000005 RESET
2135 .SBTTL INITIALIZE THE COMMON TAGS
2136 ;; CLEAR THE COMMON TAGS ($CMTAG) AREA
2137 005024 012706 001100 MOV # $CMTAG, R6 ;; FIRST LOCATION TO BE CLEARED
2138 005030 005026 CLR (R6)+ ;; CLEAR MEMORY LOCATION
2139 005032 022706 001140 CMP # SWR, R6 ;; DONE?
2140 005036 001374 BNE -6 ;; LOOP BACK IF NO
2141 005040 012706 001000 MOV # STACK, SP ;; SETUP THE STACK POINTER
2142 ;; INITIALIZE A FEW VECTORS
2143 005044 012737 044776 000020 MOV # $SCOPE, @#IOTVEC ;; IOT VECTOR FOR SCOPE ROUTINE
2144 005052 012737 000340 000022 MOV # 340, @#IOTVEC+2 ;; LEVEL 7
2145 005060 012737 047174 000030 MOV # $ERROR, @#EMTVEC ;; EMT VECTOR FOR ERROR ROUTINE
2146 005066 012737 000340 000032 MOV # 340, @#EMTVEC+2 ;; LEVEL 7
2147 005074 012737 050720 000034 MOV # $TRAP, @#TRAPVEC ;; TRAP VECTOR FOR TRAP CALLS
2148 005102 012737 000340 000036 MOV # 340, @#TRAPVEC+2; LEVEL 7
2149 005110 012737 051004 000024 MOV # $PWRDN, @#PWRVEC ;; POWER FAILURE VECTOR
2150 005116 012737 000340 000026 MOV # 340, @#PWRVEC+2 ;; LEVEL 7
2151 005124 005037 001212 CLR $TIMES ;; INITIALIZE NUMBER OF ITERATIONS
2152 005130 005037 001214 CLR $ESCAPE ;; CLEAR THE ESCAPE ON ERROR ADDRESS
2153 005134 112737 000001 001115 MOV#B # 1, $ERMAX ;; ALLOW ONE ERROR PER TEST
2154 005142 012737 005142 001106 MOV # ., $LPADR ;; INITIALIZE THE LOOP ADDRESS FOR SCOPE
2155 005150 012737 005150 001110 MOV # ., $LPERR ;; SETUP THE ERROR LOOP ADDRESS
2156 ;; SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
2157 ;; EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
2158 005156 013746 000004 MOV @#ERRVEC, -(SP) ;; SAVE ERROR VECTOR
2159 005162 012737 005216 000004 MOV # 64$, @#ERRVEC ;; SET UP ERROR VECTOR
2160 005170 012737 177570 001140 MOV # DSWR, SWR ;; SETUP FOR A HARDWARE SWICH REGISTER
2161 005176 012737 177570 001142 MOV # DISP, DISPLAY ;; AND A HARDWARE DISPLAY REGISTER
2162 005204 022777 177777 173726 CMP # -1, @SWR ;; TRY TO REFERENCE HARDWARE SWR
2163 005212 001012 BNE 66$ ;; BRANCH IF NO TIMEOUT TRAP OCCURRED
2164 ;; AND THE HARDWARE SWR IS NOT - 1
2165 005214 000403 BR 65$ ;; BRANCH IF NO TIMEOUT
2166 005216 012716 005224 64$: MOV # 65$, (SP) ;; SET UP FOR TRAP RETURN
2167 005222 000002 RTI
2168 005224 012737 000176 001140 65$: MOV # SWREG, SWR ;; POINT TO SOFTWARE SWR
2169 005232 012737 000174 001142 MOV # DISPREG, DISPLAY
2170 005240 012637 000004 66$: MOV (SP)+, @#ERRVEC ;; RESTORE ERROR VECTOR
```

```

2171
2172
2173
2174
2175 005244 012737 000000 177776      MOV    #0,PS          ;SET PROCESSOR STATUS TO 0
2176 005252 012737 000200 000036      MOV    #200,@#TRAPVEC+2;TRAP PRIORITY = 4
2177 005260 013700 002266      MOV    @#RPVEC,R0     ;GET RP VECTOR ADDRESS
2178 005264 012720 044734      MOV    #RPVECT,(R0)+ ;THIS IS FOR UNTIMELY INTERRUPTS
2179 005270 012710 000340      MOV    #340,(R0)     ;DRIVE INTERRUPT SERVICE ROUTINE
2180                                     ;PRIORITY = 7
2181
2182 005274 004737 045734      JSR    PC,@#STKINT   ;INITIALIZE THE TTY KEYBOARD
2183 005300 005737 004736      TST    @#FIRST       ;IS THIS FIRST TIME ROUND ?
2184 005304 001001                BNE    1$            ;DON'T GIVE HEADER IF NOT
2185 005306 000402                BR     2$            ;HEADER 1ST TIME THROUGH
2186 005310 000137 006112      1$:   JMP    @#SND1       ;NO HEADER
2187
2188                                     2$:
2189 005314 104401 005322      TYPE    ,68$         ;;TYPE ASCIZ STRING
2190 005320 000435                BR     67$         ;;GET OVER THE ASCIZ
2191                                     ;;68$:
2192 005414                .ASCIZ <15><12>?RP04/5/6 FUNCTIONAL CONTROLLER TEST - PART I - CZRJI-D?
2193                                     67$:
2194 005414 104401 005422      TYPE    ,70$         ;;TYPE ASCIZ STRING
2195 005420 000431                BR     69$         ;;GET OVER THE ASCIZ
2196                                     ;;70$:
2197 005504                .ASCIZ <15><12>/ALL DCL'S MUST BE LOCKED ON THE CORRECT PORT/<15><12>
2198                                     69$:
2199 005504 104401 005512      TYPE    ,72$         ;;TYPE ASCIZ STRING
2200 005510 000424                BR     71$         ;;GET OVER THE ASCIZ
2201                                     ;;72$:
2202 005562                .ASCIZ <15><12>/PROGRAMMABLE DRIVES WILL NOT BE USED/
2203                                     71$:
2204 005562 104401 005570      TYPE    ,74$         ;;TYPE ASCIZ STRING
2205 005566 000433                BR     73$         ;;GET OVER THE ASCIZ
2206                                     ;;74$:
2207 005656                .ASCIZ <15><12>/IF CHANGE IS REQUIRED ON PORT SWITCH, THEN A CYCLE/
2208                                     73$:
2209 005656 104401 005664      TYPE    ,76$         ;;TYPE ASCIZ STRING
2210 005662 000435                BR     75$         ;;GET OVER THE ASCIZ
2211                                     ;;76$:
2212 005756                .ASCIZ <15><12>/UP SEQUENCE IS REQ FOR STROBING THE PORT SELECT FLOP/<15><12>
2213                                     75$:
2214 005756 104401 005764      TYPE    ,78$         ;;TYPE ASCIZ STRING
2215 005762 000426                BR     77$         ;;GET OVER THE ASCIZ
2216                                     ;;78$:
2217 006040                .ASCIZ <15><12>/ALL DCL'S NOT UNDER TEST MUST BE SWITCHED/
2218                                     77$:
2219 006040 104401 006046      TYPE    ,80$         ;;TYPE ASCIZ STRING
2220 006044 000422                BR     79$         ;;GET OVER THE ASCIZ
2221                                     ;;80$:
2222 006112                .ASCIZ <15><12>/OFF, OR LOCKED ON THE OTHER PORT/
2223                                     79$:
2224 006112 012737 177777 004736  SND1:  MOV    #-1,@#FIRST   ;NEXT TIME DO NOT GIVE HEADER
2225                                     .SBTTL GET VALUE FOR SOFTWARE SWITCH REGISTER
2226 006120 005737 000042      TST    @#42         ;;ARE WE RUNNING UNDER XXDP/ACT?
2227 006124 001006                BNE    64$         ;;BRANCH IF YES
2228 006126 023727 001140 000176  CMP    SWR,#SWREG   ;;SOFTWARE SWITCH REG SELECTED?
2229 006134 001005                BNE    65$         ;;BRANCH IF NO
  
```



```

2227 006136 104406          GTSWR          ;;GET SOFT-SWR SETTINGS
2228 006140 000403          BR             65$
2229 006142 112737 000001 001134 64$: MOVB      #1,$AUTOB  ;;SET AUTO-MODE INDICATOR
2230 006150          65$:
2231
2232 006150 032777 010000 172762 RH70CK: BIT     #SW12,$SWR  ;LOOK TO SEE IF USING RH70
2233 006156 001403          BEQ           3$         ;IF SW12 = 0, SKIP NEXT
2234 006160 012737 000001 004750          MOV           #1,$RH70   ;IF SW12 = 1, CU IS AN RH70
2235 006166          3$:
2236          ;*****
2237          ;*IS THERE A P-CLOCK (KW11-P) ON THE SYSTEM
2238          ;*IF SO MAKE 'WAT' TRAPS GO TO 'WAIT.P'
2239          ;*IF SO MAKE RP04 INTERRUPTS GO TO 'TIME 1'
2240          ;*IF NOT MAKE 'WAT' TRAPS GO TO 'WAIT.T'
2241          ;*IF NOT MAKE RP04 INTERRUPTS GO TO 'TIME 2'
2242
2243          ;*THE NEXT LINE IS TO BE ADDED LATER
2244          ;*AND THE JUMP AND NOP REMOVED
2245          ;*FOR NOW NO CLOCK WILL BE USED
2246
2247          :      MOV      @#1$,$ERRVEC  ;SET TIME-OUT VECTOR
2248          :
2249          :      JMP      @#1$          ;DO NOT USE CLOCK
2250          :      NOP
2251          :      TST      @#PCLCSR     ;REFERENCE P-CLOCK STATUS REGISTER
2252          :          ;ADDRESS = 172540
2253          :      MOV      #WAIT.P,$STRPAD+20 ;THERE IS A P-CLOCK
2254          :      MOV      #TIME1,$RP4VEC ;THERE IS A P CLOCK SO
2255          :          ;VECTOR TO TIME1
2256          :      BR       2$
2257          :1$: MOV      #WAIT.T,$STRPAD+20 ;THERE IS NO P-CLOCK
2258          :*****
2259
2260
2261
2262 006166 012737 041736 004606          MOV      #TIME2,$RP4VEC ;RP04/5/6 INTERRUPTS GO TO 'TIME 2'
2263 006174 012737 177777 047342 2$: MOV      #-1,$PRITEM  ;CLEAR PREVIOUS ITEM NUMBER
2264
2265
2266
2267
2268 006202 005737 004726          TST      @#SELECT      ;WAS IT A 200 START
2269 006206 001442          BEQ      TST1          ;BRANCH IF STARTING FROM 200
2270 006210 104401 006216          TYPE     ,65$         ;;TYPE ASCIZ STRING
2271 006214 000424          BR       64$         ;;GET OVER THE ASCIZ
2272          ;;65$: .ASCIZ <15><12>/SELECT UNIT NUMBER TO BE TESTED ? /<15><12>
2273          64$:
2274 006266 104412          RDOCT
2275 006270 042716 177770          BIC      #177770,(SP)  ;ONLY KEEP LAST 3 BITS
2276 006274 011637 004716          MOV      (SP),@#UNIT  ;SAVE UNIT TO BE TESTED
2277 006300 012637 004730          MOV      (SP)+,@#UNITSL ;SAVE UNIT TO BE TESTED
2278
2279
2280
2281 006304 001403          BEQ      TST1          ;BRANCH IF STARTING FROM 200
2282 006306 013737 004730 004716          MOV      @#UNITSL,$UNIT ;SET UNIT NUMBER

```

CZRJDO, RP04/5/6 FCTNL CTLR1
CZRJID.P11 28-MAR-79 09:03

MACY11 30A(1052) 25-MAY-79 10:30 ^{J 4} PAGE 49
GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0048

2283

```

2284
2285
2286      ;*****
2287      ;*TEST 1      REFERENCE EACH REGISTER
2288
2289      ;*
2290      REFERENCE EACH REGISTER BY A MOVE INSTRUCTION
2291      ;*****
2291 006314 000004 TST1: SCOPE
2292 006316 012737 000001 001212 MOV #1,$TIMES ;DO 1 ITERATION
2293 006324 012737 000001 004604 MOV #2-1,@#TSTNM ;THIS SAVES TEST NUMBER
2294 006332 012706 001000 MOV #STACK, SP ;SET UP STACK POINTER
2295 006336 012737 047204 000030 MOV #REGSA1,@#EMTVEC;ERROR VECTOR SO THAT
2296 ;NO REGISTERS ARE SAVED
2297 006344 012737 006372 000004 MOV #2$,@#ERRVEC ;SET UP FOR BUS TIMEOUT
2298
2299 006352 012700 000024 MOV #24,R0 ;THERE ARE 24 REG TO TEST
2300 006356 012701 002270 MOV #RHDB,R1 ;R1 NOW HAS ADDR OF ADDR OF FIRST REG.
2301 006362 013102 1$: MOV @(R1)+,R2 ;READ HARDWARE REG.
2302 006364 005300 DEC R0 ;COUNT DOWN
2303 006366 001375 BNE 1$ ;BRANCH IF 24 NOT DONE
2304 006370 000454 BR 3$ ;BRANCH IF 24 DONE
2305 006372 012737 000006 000004 2$: MOV #ERRVEC+2,@#ERRVEC ;RESTORE TRAP CATCHER
2306 006400 022626 CMP (SP)+,(SP)+ ;CLEAN STACK
2307 006402 016137 177776 001200 MOV -2(R1),$TMP1 ;STORE FAILING REG ADDR
2308 006410 104007 ERROR 7 ;REGISTER NON EXISTANT
2309 006412 032777 020000 172520 BIT #SW:3,@SWR ;INHIBIT ERROR PRINTOUT ?
2310 006420 001036 BNE 4$ ;BRANCH IF YES
2311
2312 006422 104401 006430 TYPE ,65$ ;:TYPE ASCIZ STRING
2313 006426 000427 BR 64$ ;:GET OVER THE ASCIZ
2314 ;:65$: .ASCIZ <15><12>/TO CHANGE BASE ADDRESS RESTART AT ADDRESS /
2315 006506 64$:
2316
2317 006506 012746 000204 MOV #ADDMOD,-(SP) ;GET READY TO TYPE STARTING ADDRESS
2318 ;OF 'CHANGE OF BASE ADDRESS' ROUTINE
2319 006512 104402 TYPDC
2320 006514 000000 HALT ;STOP TO FORCE THE CORRECT RESTART
2321 006516 000137 041052 4$: JMP @#SEJP ;GO TO END OF PROGRAM ----->
2322
2323 006522 012737 006602 000004 3$: MOV #TRP,@#4 ;INITIALIZE VECTOR
2324 006530 005037 004750 CLR RH70 ;INIT RH INDICATOR ++ C.W
2325 006534 005777 173600 TST @RHBAE ;ADDRESS RPB4E(RH11/RH70?)
2326 006540 005237 004750 INC RH70 ;FOUND AN RH70-SET MASK
2327 006544 104401 006552 TYPE ,67$ ;:TYPE ASCIZ STRING
2328 006550 000413 BR 66$ ;:GET OVER THE ASCIZ
2329 ;:67$: .ASCIZ <15><12>/RH70 CONTROLLER /
2330 006600 66$:
2331 006600 000420 BR RTN ;AND GET OUT
2332 006602 005726 TRP: TST (SP)+
2333 006604 005726 TST (SP)+ ;RESTORE THE STACK
2334 006606 104401 006614 TYPE ,65$ ;:TYPE ASCIZ STRING
2335 006612 000413 BR 64$ ;:GET OVER THE ASCIZ
2336 ;:65$: .ASCIZ <15><12>/RH11 CONTROLLER /
2337 006642 64$:
2338 006642 012737 047174 000030 RTN: MOV #ERROR,@#EMTVEC;RESTORE ERROR VECTOR
2339 ;SO THAT REGISTERS ARE SAVED
  
```

CZRJID, RPO4/5/6 FCTNL CTLR1 MACY11 30A(1052) 25-MAY-79 10:30 ^{L 4} PAGE 51
CZRJID.P11 28-MAR-79 09:03 T1 REFERENCE EACH REGISTER

SEQ 0050

2340 006650 012737 000006 000004 MOV #ERRVEC+2,@#ERRVEC ;RESTORE TRAP CATCHER
2341

```

2342
2343
2344      ;:*****
2345      ;*TEST 2      PARTIAL TEST OF RHAS FOR UNIT NUMBERS PRESENT
2346
2347      ;*      CHECK THAT RHAS CAN BE CLEARED BY MOVING ALL ONES INTO IT
2348
2349      ;:*****
2350      006656 000004      TST2:  SCOPE
2351      006660 012737 000001 001212      MOV      #1,$TIMES      ;;DO 1 ITERATION
2352      006666 012706 001000      MOV      #STACK,SP      ;SET STACK POINTER
2353      006672 012737 000002 004604      MOV      #3-1,@#TSTNM      ;THIS SAVES TEST NUMBER
2354
2355      006700 013701 002316      MOV      @#RHAS,R1      ;R1 HAS ADDRESS OF RHAS
2356      006704 012711 177777      MOV      #-1,@R1      ;THIS WRITES ALL ONES INTO RHAS
2357      006710 105711      TSTB    @R1      ;TEST RHAS FOR ALL 0'S
2358      006712 001407      BEQ     TST3      ; BRANCH IF GOOD
2359      006714 011137 001126      MOV      @R1,@#$BDDAT      ;BAD DATA
2360      006720 005037 001124      CLR     @#$GDDAT      ;GOOD DATA
2361      006724 010137 004600      MOV      R1,@#REGADR      ;FAILING REG. (RHAS)
2362
2363      006730 104005      ERROR  5      ;RHAS DOES NOT CLEAR BY WRITING
2364                          ;ALL ONES INTO IT
2365
  
```

```
2366  
2367  
2368  
2369  
2370  
2371  
2372  
2373  
2374  
2375  
2376  
2377  
2378  
2379  
2380 006732 000004  
2381 006734 012737 000001 001212  
2382 006742 012737 000003 004604  
2383 006750 000005  
2384 006752 004737 045734  
2385  
2386 006756 032777 020000 172154  
2387 006764 001026  
2388 006766 104401 006774  
2389 006772 000423  
2390  
2391 007042  
2392 007042 013701 002316  
2393 007046 013702 002276  
2394 007052 005012  
2395 007054 012700 000010  
2396 007060 013704 002302  
2397 007064 012714 177777  
2398 007070 005212  
2399 007072 005300  
2400 007074 001373  
2401  
2402 007076 111137 004742  
2403  
2404 007102 105037 004743  
2405 007106 105711  
2406 007110 001402  
2407 007112 000137 007464  
2408  
2409 007116 032777 020000 172014 2S:  
2410 007124 001402  
2411 007126 000137 010066  
2412  
2413  
2414 007132 3S:  
2415 007132 104401 007140  
2416 007136 000420  
2417  
2418 007200  
2419 007200 104401 007206  
2420 007204 000430  
2421
```

```
*****  
: *TEST 3 TEST FOR DRIVES PRESENT USING RHAS AND RHCS2  
: * THE NUMBER OF DRIVES PRESENT IS FOUND  
: * BY MOVING ALL ONES INTO RHER1 WITH UNIT NUMBER  
: * IN RHCS2 INCREMENTED FROM ZERO TO SEVEN  
: * THE BITS SET IN RHAS SHOULD GIVE DRIVES PRESENT  
: * THE DRIVE TYPE IS CHECKED TO BE AN RP04/RP06 AND THEN  
: * UNITS PRESENT ARE STORED IN A TABLE CALLED 'UNITS'  
*****  
TST3: SCOPE  
MOV #1,$TIMES ;;DO 1 ITERATION  
MOV #4-1,@#TSTNM ;;THIS SAVES TEST NUMBER  
RESET ;;START WITH AN INIT  
JSR PC,@#STKINT ;;INITIALIZE TTY KEYBOARD  
BIT #SW13,@SWR ;;INHIBIT ERROR TYPEOUT?  
BNE 4S ;;BRANCH IF YES  
TYPE ,65$ ;;TYPE ASCIZ STRING  
BR 64$ ;;GET OVER THE ASCIZ  
;;65$: .ASCIZ <15><12><15><12>/LOOKING AT RHAS - DRIVES PRESENT /  
64$:  
4S: MOV @#RHAS,R1 ;R1 HAS ADDR. OF RHAS  
MOV @#RHCS2,R2 ;R2 HAS ADDR. OF RHCS2  
CLR @R2 ;CLEAR RHCS2  
MOV #8,,R0 ;COUNT  
MOV @#RHER1,R4 ;R4 HAS ADDR. OF RHER1  
1S: MOV #-1,@R4 ;MOVE ERRORS INTO RHER1  
INC @R2 ;INCREMENT UNIT NO.  
DEC R0 ;COUNT DOWN DRIVE COUNTER  
BNE 1S ;DO NEXT UNIT IF 8 NOT DONE  
MOVB @R1,@#TOTALAT ;SAVE TOTAL ATTENTION  
;USED IN DRIVE CLEAR TEST  
CLRB @#TOTALAT+1 ;CLEAR UPPER BYTE  
TSTB @R1 ;TEST 'RHAS' FOR ANY DRIVES PRESENT  
BEQ 2S ;NONE RESPONDING - TYPE THE MESSAGE  
JMP XE2 ;SOME THERE - GO FILL 'UNITS' TABLE  
2S: BIT #SW13,@SWR ;INHIBIT ERROR TYPE OUT?  
BEQ 3S ;'NO DRIVES' MESSAGE IF NOT  
JMP SELTST ;CHECK FOR SELECTED UNIT START AND LOAD  
;'UNITS' TABLE WITH SELECTED ONE IF SO  
3S:  
TYPE ,67$ ;;TYPE ASCIZ STRING  
BR 66$ ;;GET OVER THE ASCIZ  
;;67$: .ASCIZ <15><12>/NO DRIVES PRESENT - RHAS = 0/  
66$:  
TYPE ,69$ ;;TYPE ASCIZ STRING  
BR 68$ ;;GET OVER THE ASCIZ  
;;69$: .ASCIZ <15><12>/WRITING ONES INTO RHER1 FOR ALL UNIT NUMBERS/
```

```

CZRJ10, RP04/5/6 FCTNL CTLR1 MACY11 30A(1052) 25-MAY-79 10:30 PAGE 54
CZRJID.P11 28-MAR-79 09:03 T3 TEST FOR DRIVES PRESENT USING RHAS AND RHCS2 SEQ 0053

```

2422	007266			68\$:			
2423	007266	104401	007274		TYPE	,71\$::TYPE ASCIZ STRING
2424	007272	000430			BR	70\$::GET OVER THE ASCIZ
2425				::71\$:	.ASCIZ	<15><12>/DOES NOT SET ANY BIT IN RHAS SO ABORT PROGRAM/	
2426	007354			70\$:			
2427	007354	104401	007362		TYPE	,73\$::TYPE ASCIZ STRING
2428	007360	000437			BR	72\$::GET OVER THE ASCIZ
2429				::73\$:	.ASCIZ	<15><12>/TO LOOP ON THIS TEST WO PRINTOUT SET SWITCHES 13, 8, 1, & 0/	
2430	007460			72\$:			
2431							
2432	007460	000137	041052		JMP	@#SEOP	:GO OUT----->
2433							
2434							
2435							
2436							
2437							
2438	007464						
2439	007464	012700	000010	XE2:	MOV	#8.,R0	:COUNTER
2440	007470	012703	004676	2\$:	MOV	#UNITS,R3	:POINTER
2441	007474	012723	177777	3\$:	MOV	#-1,(R3)+	:PRESET BLOCK TO ALL ONES
2442	007500	005300			DEC	R0	:COUNT
2443	007502	001374			BNE	3\$:BRANCH IF 8 NOT DONE
2444	007504	012703	004676		MOV	#UNITS,R3	:POINTER
2445	007510	005005			CLR	R5	:INITIALIZE UNIT NO. TO 0
2446	007512	005037	004720		CLR	@#NOUNIT	:NO. OF UNITS PRESENT
2447	007516	012700	000010		MOV	#8.,R0	:COUNTER
2448	007522	011137	001176		MOV	@R1,@#STMPO	:TEMPORARY STORAGE
2449	007526	006037	001176	4\$:	ROR	@#STMPO	:SET CARRY IF ONE IN 0 BIT
2450	007532	103135			BCC	5\$:CHECK NEXT UNIT IF ONE NOT IN BIT 0
2451							
2452	007534	010577	172536		MOV	R5,@RHCS2	:INSERT UNIT NUMBER INTO RHCS2 UA BITS
2453	007540	022777	024020	172556	CMP	#24020,@RHDT	:IS THIS A DUAL PORT RP04 ?
2454	007546	001425			BEQ	7\$:TYPE THE UNIT NO. IF YES
2455	007550	022777	020020	172546	CMP	#20020,@RHDT	:IS THIS A SINGLE PORT RP04 ?
2456	007556	001421			BEQ	7\$:TYPE UNIT NO. IF YES
2457							
2458							
2459	007560	022777	024021	172536	::*****		
2460	007566	001415			CMP	#24021,@RHDT	:DUAL PORT RP05 ?
2461	007570	022777	020021	172526	BEQ	7\$:TYPE UNIT NO. IF SO
2462	007576	001411			CMP	#20021,@RHDT	:SINGLE PORT RP05 ?
2463					BEQ	7\$:TYPE UNIT NO. IF SO
2464	007600	022777	024022	172516	CMP	#24022,@RHDT	:IS THIS A DUAL PORT RP06 ?
2465	007606	001405			BEQ	7\$:TYPE THE UNIT NO. IF SO
2466	007610	022777	020022	172506	CMP	#20022,@RHDT	:IS THIS A SINGLE PORT RP06 ?
2467	007616	001401			BEQ	7\$:TYPE UNIT NO. IF SO
2468	007620	000414			BR	9\$	
2469	007622	032777	001000	172472	7\$:	BIT	#BIT09,@RHDS1
2470	007630	001001			BNE	8\$:IS THE DRIVE PROGRAMMABLE?
2471	007632	000466			BR	6\$:BRANCH IF YES
2472	007634	104401	001223		8\$:	TYPE	,%CRLF
2473	007640	010546			MOV	R5, -(SP)	
2474	007642	104405			TYPDS		:TYPE THE DRIVE NUMBER
2475	007644	104401	066766		TYPE	,NOUSE	:REPORT THIS DRIVE WILL NOT BE USED(DRIVE MUST BE LOCKED
2476	007650	000466			BR	5\$	
2477	007652				9\$:		

```
2478  
2479  
2480  
2481  
2482  
2483  
2484 007652 104401 007660  
2485 007656 000410  
2486  
2487 007700  
2488 007700 010546  
2489 007702 104405  
2490 007704 104401 007712  
2491 007710 000406  
2492  
2493 007726  
2494 007726 017746 172372  
2495 007732 104402  
2496 007734 104401 007742  
2497 007740 000422  
2498  
2499 010006  
2500 010006 000407  
2501  
2502 010010 010523  
2503 010012 104401 001223  
2504 010016 010546  
2505 010020 104405  
2506 010022 005237 004720  
2507  
2508 010026 005205  
2509 010030 005300  
2510 010032 001235  
2511  
2512 010034 005737 004720  
2513 010040 001002  
2514 010042 000137 041052  
2515 010046  
2516  
2517 010046 013737 004676 004716  
2518 010054 013737 004720 004722  
2519 010062 005337 004722  
2520  
2521  
2522 010066 005737 004726  
2523 010072 001403  
2524 010074 013737 004730 004716
```

;*NO...IT'S NOT AN RP04/RP05/RP06 DEVICE SO TYPE
;*OUT THE DEVICE TYPE
TYPE ,65\$;:TYPE ASCIZ STRING
BR 64\$;:GET OVER THE ASCIZ
65\$: .ASCIZ <15><12>/UNIT NUMBER /
64\$:
MOV R5,-(SP) ;GET READY TO TYPE UNIT NUMBER
TYPDS
TYPE ,67\$;:TYPE ASCIZ STRING
BR 66\$;:GET OVER THE ASCIZ
67\$: .ASCIZ /, RMDT = /
66\$:
MOV @RMDT,-(SP) ;GET READY TO TYPE RMDT
TYPDC
TYPE ,69\$;:TYPE ASCIZ STRING
BR 68\$;:GET OVER THE ASCIZ
69\$: .ASCIZ ? - NOT AN RP04/RP05/RP06 DEVICE !!?
68\$:
BR 5\$;NO RP04/RP05/RP06 FOUND SO TEST NEXT UNIT
6\$: MOV R5,(R3)+ ;LOAD TABLE POSITION AND INCR IT
TYPE ,5CRLF
MOV R5,-(SP) ;PUT DRIVE NO. ON STACK
TYPDS ;TYPE DRIVE NO.
INC @#NOUNIT ;INCR THE TOTAL NO. OF UNITS
5\$: INC R5 ;'RHCS2' UNIT ADDRESS
DEC R0 ;DRIVE COUNTER DOWN ONE
BNE 4\$;TEST AND DO NEXT UNIT IF 8 NOT DONE
TST @#NUNIT ;IF THERE ARE ANY UNITS...
BNE 10\$;CONTINUE
JMP @#SEOP ;ELSE GO TO END OF PASS
10\$:
MOV @#UNITS,@#UNIT ;SET UNIT NO. TO FIRST ONE FOUND OR 0
MOV @#NOUNIT,@#NUNIT ;SAVE NO. OF UNITS
DEC @#NUNIT ;IF NUNIT = 0 THEN ONLY ONE UNIT
;IF NUNIT > 0 THEN MORE THAN ONE UNIT
SELTST: TST @#SELECT ;STARTING ADDRESS 200
BEQ TST4 ;SKIP NEXT IF STARTING FROM 200
MOV @#UNITSL,@#UNIT ;CHANGE UNIT NUMBER TO SELECTED ONE

2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580

```
*****  
;*TEST 4 TYPE SERIAL NUMBER AND DRIVE TYPE  
*****  
;* SET APPROPRIATE ATTENTION BIT OF UNIT UNDER TEST IN 'ATTENT'  
;* TYPE UNIT UNDER TEST  
*****  
;* READ SERIAL NUMBER AND DRIVE TYPE REGISTERS  
;* TYPE THEM OUT AND PROCEED  
*****  
;* TO LOOP HERE SET SWITCH 8, AND THIS TEST NUMBER ON  
;* SWITCHES 0 THRU 7 AND RESTART  
*****  
TST4: SCOPE  
MOV #1,STIMES ;:DO 1 ITERATION  
MOV #1$,SLPADR ;:SET SCOPE LOOP ADDRESS  
MOV #STACK,SP ;:RESET STACK  
MOV #4,@#TSTNM ;:SAVE TEST NUMBER  
JSR PC,@#CLDISK ;:SET R1-RHCS1, R2-RHCS2  
;:R3-RHDS1, R4-RHER1  
;:GIVE RH-11 INITIALIZE  
;:SETUP UNIT NUMBER  
;:CLEAR  
CLR @#ATTENT  
TST @#UNIT ;:IS "UNIT 0" NEXT IN THE UNITS TABLE ?  
BNE 9$ ;:IF NOT TEST THIS UNIT  
MOV #41,R0 ;:IF SO, CHECK THE LOAD MEDIA LOCATION  
CMPB #11,(R0) ;:WAS IT AN RP04/5/6 ?  
BNE 9$ ;:NO - GO AHEAD WITH TESTING UNIT #0  
TST @#SELECT ;:WAS UNIT #0 SELECTED ?  
;:(IE. WAS IT A 210 START ?)  
BNE 12$ ;:IF SO, CHANGE PACKS  
;:*INCREMENT THE UNITS TABLE TO NEXT DRIVE (IF ANY)  
;:*DECREMENT THE "NOUNITS" PRESENT (TO BE TESTED)  
MOV #UNITS,R0 ;:LOAD UNITS TABLE POINTER  
TST (R0)+ ;:SELECT THE NEXT UNIT IN THE TABLE  
;:(DOUBLE INCREMENT THE POINTER, R0)  
CMP #-1,(R0) ;:IS THERE ANOTHER TABLE ENTRY PRESENT ?  
BNE 11$ ;:IF SO, USE THE NEXT DRIVE & DEC "NOUNITS"  
;:IF NOT, MUST USE DRIVE #0 & CHANGE PACK  
12$:  
TYPE ,65$ ;:TYPE ASCIZ STRING  
BR 64$ ;:GET OVER THE ASCIZ  
;:65$: .ASCIZ <15><12><15><12>/DISMOUNT PACK FROM UNIT #0 AND MOUNT A SCRATCH PACK/  
64$:  
TYPE ,67$ ;:TYPE ASCIZ STRING  
BR 66$ ;:GET OVER THE ASCIZ  
;:67$: .ASCIZ <15><12>/PRESS CONTINUE WHEN FINISHED/<15><12>  
66$:
```

```

2581 010352 000000          HALT
2582 010354 000404          BR          9$          ;CONTINUE, USING SCRATCH PACK ON UNIT #0
2583
2584 010356 011037 004716  11$:  MOV      (R0),@#UNIT    ;SET UP TO BE THE UNIT UNDER TEST
2585 010362 005337 004720          DEC      @#NOUNITS      ;DECREMENT BECAUSE UNIT #0 WON'T BE TESTED
2586
2587 010366 013700 004716  9$:   MOV      @#UNIT,R0    ;R0 CONTAINS UNIT UNDER TEST
2588
2589
2590
2591      ;:*****
2592 010372 005037 004744          CLR      @#RP06          ;CLEAR RP06 DEVICE TYPE FLAG
2593 010376 010077 171674          MOV      R0,@#RHCS2     ;SET UP UNIT ADDRESSING
2594 010402 022777 024022 171714  CMP      #24022,@#RHDT   ;IS IT A DUAL PORT RP06 ?
2595 010410 001405          BEQ      2$             ;YES...SET THE FLAG
2596 010412 022777 020022 171704  CMP      #20022,@#RHDT   ;IS IT A SINGLE PORT RP06 ?
2597 010420 001401          BEQ      2$             ;YES...SET FLAG
2598 010422 000404          BR       3$             ;DON'T SET FLAG - CHECK FOR RP04
2599 010424 012737 177777 004744  2$:   MOV      #-1,@#RP06     ;SET THE FLAG
2600 010432 000416          BR       8$             ;DON'T CHECK FOR RP04, IT WAS RP06
2601
2602 010434 005037 004746  3$:   CLR      @#RP05          ;CLEAR MEMOREX RP04 DEVICE FLAG
2603 010440 022777 024021 171656  CMP      #24021,@#RHDT   ;IS IT A DUAL PORT MEMOREX RP04 ?
2604 010446 001405          BEQ      7$             ;YES..SET THE FLAG FOR ADDR PLUG TESTS
2605 010450 022777 020021 171646  CMP      #20021,@#RHDT   ;IS IT A SINGLE PORT MEMOREX RP04 ?
2606 010456 001401          BEQ      7$             ;YES..SET THE FLAG FOR ADDR PLUG TESTS
2607 010460 000403          BR       8$             ;DON'T SET FLAG - NOT MEMOREX DRIVE
2608 010462 012737 177777 004746  7$:   MOV      #-1,@#RP05     ;SET THE FLAG
2609 010470          8$:   ;ASSUME THE NEXT UNIT IS AN RP04
2610      ;:*****
2611
2612
2613
2614 010470 116037 004666 004740  MOVB     ATABLE(R0),@#ATTENT ;SET APPROPRIATE ATTENTION BIT
2615 010476 104401 010504          TYPE     ,69$           ;;TYPE ASCIZ STRING
2616 010502 000414          BR       68$           ;;GET OVER THE ASCIZ
2617      ;;69$: .ASCIZ <15><12>/TESTING DRIVE NUMBER/
2618          68$:
2619 010534          MOV      @#UNIT,-(SP)    ;UNIT NO. TO STACK
2620 010540 104405          TYPDS    ;TYPE DRIVE NO.
2621 010542 104401 010550          TYPE     ,71$           ;;TYPE ASCIZ STRING
2622 010546 000410          BR       70$           ;;GET OVER THE ASCIZ
2623      ;;71$: .ASCIZ <15><12>/SERIAL NO. = /
2624          70$:
2625 010570          MOV      @#RHSN,-(SP)   ;;SAVE @#RHSN FOR TYPEOUT
2626 010574 104402          TYPOC    ;GO TYPE--OCTAL ASCII(ALL DIGITS)
2627 010576 104401 010604          TYPE     ,73$           ;;TYPE ASCIZ STRING
2628 010602 000410          BR       72$           ;;GET OVER THE ASCIZ
2629      ;;73$: .ASCIZ <15><12>/DRIVE TYPE = /
2630          72$:
2631 010624          MOV      @#RHDT,-(SP)   ;;SAVE @#RHDT FOR TYPEOUT
2632 010630 104402          TYPOC    ;GO TYPE--OCTAL ASCII(ALL DIGITS)
2633

```

```

2634 ;TYPE OUT THE DRIVE TYPE IN ASCII
2635
2636
2637 ;*****
2638 010632 022777 024020 171464      CMP      #24020,@RHDT      ;DUAL PORT RP04 ?
2639 010640 001425                    BEQ      4$              ;TYPE ASCII MSG OUT
2640 010642 022777 020020 171454      CMP      #20020,@RHDT      ;SINGLE PORT RP04 ?
2641 010650 001421                    BEQ      4$              ;TYPE THE MESSAGE
2642
2643 010652 022777 024021 171444      CMP      #24021,@RHDT      ;DUAL PORT RP05 ?
2644 010660 001453                    BEQ      6$              ;TYPE THE MESSAGE
2645 010662 022777 020021 171434      CMP      #20021,@RHDT      ;SINGLE PORT RP05 ?
2646 010670 001447                    BEQ      6$              ;TYPE THE MESSAGE
2647
2648 010672 022777 024022 171424      CMP      #24022,@RHDT      ;DUAL PORT RP06 ?
2649 010700 001424                    BEQ      5$              ;TYPE THE MESSAGE
2650 010702 022777 020022 171414      CMP      #20022,@RHDT      ;SINGLE PORT RP06 ?
2651 010710 001420                    BEQ      5$              ;TYPE THE MESSAGE
2652 010712 000454                    BR       1$              ;DRIVE IS NOT AN RP04/RP05/RP06
2653 ;DON'T TYPE ASCII MESSAGE OUT
2654
2655 ;-SHOULD NEVER HAPPEN AT THIS POINT
2656 ;UNLESS DRIVE GOT SICK WHILE TESTING
2657 ;WAS IN PROGRESS
2658
2659 010714 4$:
2660 010714 104401 010722      TYPE      ,75$          ;;TYPE ASCII STRING
2661 010720 000413      BR       74$          ;;GET OVER THE ASCII
2662 ;:75$: .ASCIIZ <15><12>/DRIVE IS AN RP04/<15><12>
2663 010750 74$:
2664 010750 000435      BR       1$          ;SKIP NEXT MESSAGE
2665 010752 5$:
2666 010752 104401 010760      TYPE      ,77$          ;;TYPE ASCII STRING
2667 010756 000413      BR       76$          ;;GET OVER THE ASCII
2668 ;:77$: .ASCIIZ <15><12>/DRIVE IS AN RP06/<15><12>
2669 011006 76$:
2670 011006 000416      BR       1$          ;SKIP NEXT MESSAGE
2671 011010 6$:
2672 011010 104401 011016      TYPE      ,79$          ;;TYPE ASCII STRING
2673 011014 000413      BR       78$          ;;GET OVER THE ASCII
2674 ;:79$: .ASCIIZ <15><12>/DRIVE IS AN RP05/<15><12>
2675 011044 78$:
2676 ;*****
2677
2678
2679
2680
2681 011044 005777 171256      1$:  TST      @RHSN      ;READ SERIAL NO. AND DRIVE TYPE
2682 011050 005777 171250      TST      @RHDT      ;THESE TWO ARE TO HELP SCOPE LOOPS
2683 011054 017737 171246 002410      MOV      @RHSN,@#SN    ;SAVE TO CHECK IF DRIVE CLEAR CLEARS ANY BITS
2684 011062 017737 171236 002406      MOV      @RHDT,@#DT    ;SAVE TO CHECK IF DRIVE CLEAR CLEARS ANY BITS
  
```

2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719

011070 000004
011072 012737 000005 004604
011100 004737 041524
011104 032713 010000
011110 001151
011112 104401 011120
011116 000420

011160
011160 104401 011166
011164 000424

011236
011236 104401 011244
011242 000431

011326
011326 032713 010000
011332 001775
011334 104401 011342
011340 000435

011434

```
*****  
: *TEST 5 CHECK MOL TO BE HIGH  
  
: * MAKE SURE THAT DRIVE IS ON LINE BEFORE STARTING PROGRAM  
: * IF DRIVE IS OFFLINE, THEN AFTER TYPE OUT THE PROGRAM WILL  
: * HANG FOR EVER WAITING FOR DRIVE TO GO ON LINE.  
  
*****  
TST5: SCOPE  
MOV #6-1,@#TSTNM ;THIS SAVES TEST NUMBER  
JSR PC,@#CLDISK ;GIVE INITIALIZE  
BIT #MOL,@R3 ;CHECK MOL IN RHDS1  
BNE TST6 ; BRANCH IF MOL HIGH  
TYPE ,65$ ;;TYPE ASCIZ STRING  
BR 64$ ;;GET OVER THE ASCIZ  
65$: .ASCIZ <15><12>/DRIVE IS OFFLINE - MOL IS LOW/  
64$:  
TYPE ,67$ ;;TYPE ASCIZ STRING  
BR 66$ ;;GET OVER THE ASCIZ  
66$: .ASCIZ <15><12>/HIT START ON DRIVE TO GET IT ON LINE/  
67$:  
TYPE ,69$ ;;TYPE ASCIZ STRING  
BR 68$ ;;GET OVER THE ASCIZ  
68$: .ASCIZ <15><12>/PROGRAM WILL HANG TESTING MOL TILL MOL IS HIGH/  
1$: BIT #MOL,@R3 ;CHECK MOL IN RHDS1  
BEQ 1$ ;BRANCH IF MOL IS HIGH  
TYPE ,71$ ;;TYPE ASCIZ STRING  
BR 70$ ;;GET OVER THE ASCIZ  
70$: .ASCIZ <15><12>/GOOD - MOL IS NOW HIGH. PROGRAM WILL NOW BE EXECUTED/<15><12>  
71$:  
70$:
```

```
2720
2721
2722          :*****
2723          :*TEST 6          PROGRAM INTERRUPT
2724          :
2725          :*          PROGRAM INTERRUPT IS TESTED BY SETTING RDY AND IE
2726          :*          IN RHCS1 AT THE SAME TIME
2727          :*          THIS SHOULD INTERRUPT THROUGH LOCATION 254
2728          :*          THE PROCESSOR PRIORITY IS SET TO 4
2729          :*****
2729 011434 000004          TST6: SCOPE
2730
2731 011436 012737 000006 004604          MOV      #7-1,@#TSTNM          ;THIS SAVES TEST NUMBER
2732 011444 012706 001000          MOV      #STACK,SP          ;RESET STACK
2733
2734 011450 004737 041524          JSR      PC,@#CLDISK          ;SET R1-RHCS1, R2-RHCS2
2735          ;R3-RHDS1, R4-RHER1
2736          ;GIVE RH-11 INITIALIZE
2737          ;SETUP UNIT NUMBER
2738
2739 011454 013700 002266          MOV      @#RPVEC,R0          ;GET RP VECTOR ADDRESS
2740 011460 012720 011526          MOV      #RPTRP1,(R0)+      ;THIS IS FOR TIMELY INTERRUPTS
2741 011464 012710 000340          MOV      #340,(R0)          ;RPO4 INTERRUPT SERVICE ROUTINE
2742          ;PRIORITY = 7
2743 011470 012737 000200 177776          MOV      #200,PS          ;SET PROCESSOR PRIORITY @ 4 (DISK @ 5)
2744 011476 012711 000300          MOV      #RDY!IE,@R1          ;RDY, IE IN RHCS1 SHOULD CAUSE INTERRUPT
2745
2746 011502 013737 042222 001200          MOV      @#TIMCNT,@#STMP1 ;COUNTER
2747 011510 005337 001200          1$: DEC      @#STMP1          ;WAIT FOR INTERRUPT
2748 011514 001375          BNE      1$                  ;BRANCH IF NOT ZERO
2749          ;BEFORE THIS IS ZERO INTERRUPT SHOULD OCCUR
2750 011516 104065          ERROR   65                  ;INTERRUPT DID NOT OCCUR
2751 011520 012712 000040          MOV      #40,@R2          ;CLEAR CONTROLLER VIA CS2 CLR BIT
2752 011524 000407          BR       TST7 ;              ;BRANCH TO NEXT TEST -----)
2753
2754 011526 022626          RPTRP1: CMP      (SP)+,(SP)+      ;RESTORE STACK
2755 011530 022711 004200          CMP      #DVA!RDY,@R1      ;IE SHOULD BE LOW AS RUPT OCCURRED
2756 011534 001403          BEQ     TST7 ;              ;CONTINUE IF GOOD -----)
2757 011536 104065          ERROR   65                  ;INTERRUPT OCCURED BUT
2758          ;IE FAILED TO RESET
2759 011540 012712 000040          MOV      #40,@R2          ;CLEAR CONTROLLER VIA CS2 CLR BIT
```

```

2760
2761
2762          ;*****
2763          ;*TEST 7          INTERRUPT AT PROCESSOR AND DISK PRIORITY SAME
2764
2765          ;*          PROCESSOR PRIORITY IS SET AT 5 (SAME AS THE DISK)
2766          ;*          IE AND RDY IS SET. THIS SHOULD NOT ALLOW INTERRUPT
2767          ;*****
2768 011544 000004          TST7:  SCOPE
2769
2770 011546 012737 000007 004604          MOV    #10-1,@#TSTNM ;THIS SAVES TEST NUMBER
2771 011554 012706 001000          MOV    #STACK,SP    ;RESET STACK
2772
2773 011560 004737 041524          JSR    PC,@#CLDISK  ;SET R1-RHCS1, R2-RHCS2
2774                                     ;R3-RHDS1, R4-RHER1
2775                                     ;GIVE RH-11 INITIALIZE
2776                                     ;SETUP UNIT NUMBER
2777
2778 011564 013700 002266          MOV    @#RPVEC,R0   ;GET RP VECTOR ADDRESS
2779 011570 012720 011634          MOV    #RPTRP2,(R0)+ ;THIS IS FOR UNTIMELY INTERRUPTS
2780 011574 012710 000340          MOV    #340,(R0)   ;RP04 INTERRUPT SERVICE ROUTINE
2781                                     ;PRIORITY = 7
2782 011600 012737 000240 177776          MOV    #240,PS     ;SET PROCESSOR PRIORITY @ 5
2783 011606 012711 0003C0          MOV    #RDY!IE,@R1 ;RDY, IE IN RHSC1 WHOULD CAUSE INTERRUPT
2784
2785 011612 013737 042222 001200          MOV    @#TIMCNT,@#STMP1 ;COUNTER
2786 011620 005337 001200          1$:  DEC    @#STMP1   ;WAIT FOR INTERRUPT
2787 011624 001375          BNE    1$         ;BRANCH IF NOT ZERO
2788                                     ;BEFORE THIS IS ZERO INTERRUPT WHOULD
2789                                     ;OCCUR
2790 011626 012712 000040          MOV    #40,@R2    ;CLEAR THE CONTROLLER VIA CS2 CLR BIT
2791 011632 000404          BR     TST10     ;NO INTERRUPT SO CONTINUE -----)
2792
2793 011634 022626          RPTRP2: CMP    (SP)+,(SP)+ ;RESTORE STACK
2794 011636 104065          ERROR 65        ;INTERRUPT OCCURRED WITH PROCESSOR
2795                                     ;PROCESSOR STATUS SAME AS DISK
2796 011640 012712 000040          MOV    #40,@R2    ;CLEAR THE CONTROLLER VIA CS2 CLR BIT
2797
2798
2799
2800
2801
  
```

```
2802 ;*****  
2803 ;*TEST 10 PACK ACKNOWLEDGE  
2804  
2805 ;* IF STARTING ADDRESS 220 IS USED THIS TEST WILL NOT BE PERFORMED  
2806 ;*  
2807 ;* IF THE PROGRAM WORKS UNDER ACT-11 MONITOR  
2808 ;* THEN THIS TEST IS NOT PERFORMED  
2809 ;*  
2810 ;* IF NO ACT-11 MONITOR IS PRESENT  
2811 ;* THEN THIS TEST IS PERFORMED ONLY ON THE FIRST PASS  
2812 ;* ON SUBSEQUENT PASSES THIS TEST IS NOT DONE  
2813 ;*  
2814 ;* THIS TESTS THE ACKNOWLEDGE COMMAND=44  
2815 ;* VV BIT - RHDS1 BIT #6  
2816 ;* MOL BIT - RHDS1 BIT #12  
2817 ;* DVA BIT - RHCS1 BIT #11  
2818 ;* THE DRIVE IS STOPPED MOL IS CHECKED TO BE 0  
2819 ;* AND DVA SHOULD BE 0  
2820 ;* THE DRIVE IS POWERED UP  
2821 ;* VV SHOULD BE 0, MOL SHOULD BE 1, DVA SHOULD BE 1  
2822 ;* GO SHOULD BE 0  
2823 ;* ALL REGISTERS EXCEPT RHDB, RHLA AND RHCC ARE STORED  
2824 ;* PACK ACKNOWLEDGE IS ISSUED  
2825 ;* ALL STORED REGISTERS SHOULD BE UNCHANGED  
2826 ;* EXCEPT VV
```

```
2827 ;*****  
2828 ;*  
2829 011644 000004 TST10: SCOPE  
2830 011646 012706 001000 MOV #STACK,SP ;RESET STACK  
2831 011652 012737 000010 004604 MOV #10,@#TSTNM ;SAVE TEST NUMBER  
2832  
2833 011660 004737 041524 JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2  
2834 ;R3-RHDS1, R4-RHER1  
2835 ;GIVE RH-11 INITIALIZE  
2836 ;SETUP UNIT NUMBER  
2837 011664 012737 000000 177776 MOV #0,PS ;SET PROCESSER STATUS TO 0  
2838  
2839 ;*THIS CODE CHECKS TO SEE IF MANUAL INTERVENTION TESTS ARE OK  
2840  
2841  
2842 011672 005737 004724 TST @#NOPUSH ;IS THIS A 220 START ?  
2843 011676 001007 BNE 1$ ;SKIP THIS TEST IF SO  
2844 011700 005737 000042 TST @#42 ;MONITOR (ACT 1i) RETURN ADDRESS ?  
2845 011704 001004 BNE 1$ ;SKIP THIS TEST  
2846 011706 005737 001100 TST @#SPASS ;FIRST PASS ?  
2847 011712 001001 BNE 1$ ;SKIP THIS TEST IF NOT  
2848 011714 000402 BR 2$ ;CONTINUE WITH THIS TEST  
2849  
2850 011716 1$:  
2851 011716 000137 012424 JMP TST11 ; JUMP TO NEXT TEST -----)  
2852 011722 2$:  
2853 011722 104401 011730 TYPE ,65$ ;:TYPE ASCIZ STRING  
2854 011726 000407 BR 64$ ;:GET OVER THE ASCIZ  
2855 ;:65$: .ASCIZ <15><12>/STOP DRIVE /  
2856 011746 64$:  
2857 011746 013746 004716 MOV @#UNIT,-(SP) ;GET UNIT UNDER TEST
```

```

2858 011752 104405          TYPDS
2859 011754 104401 011762  TYPE      ,67$          ;;TYPE ASCIZ STRING
2860 011760 000413          BR        66$          ;;GET OVER THE ASCIZ
2861                ;;67$: .ASCIZ / THEN HIT CONTINUE/<15><12>
2862 012010                66$:
2863 012010 000000          HALT
2864 012012 032713 010000  BIT      #MOL,@R3      ;MOL IN RHDS1 SHOULD BE = 0
2865 012016 001403          BEQ      3$           ;BRANCH IF MOL=0
2866 012020 010337 001122  MOV      R3,@#SBDADR   ;FAILING REGISTER ADDRESS-RHDS1
2867 012024 104010          ERROR     10          ;ON SPINPLE POWERED DOWN
2868                ;MOL SHOULD BE 0
2869 012026 013746 004716  3$:  MOV      @#UNIT,-(SP) ;UNIT NUMBER
2870 012032 052716 000100  BIS      #IR,(SP)     ;INCLUDE IR
2871 012036 022612          CMP      (SP)+,@R2    ;ONLY UNIT NO. AND IR SHOULD BE
2872                ;HIGH IN RHCS2
2873 012040 001403          BEQ      4$           ;BRANCH IF RHCS2 GOOD
2874 012042 010237 001122  MOV      R2,@#SBDADR   ;FAILING REGISTER ADDRESS-RHCS2
2875 012046 104011          ERROR     11          ;WITH SPINDLE POWERED DOWN
2876                ;ONLY UNIT NO. AND IR SHOULD BE
2877                ;HIGH
2878
2879 012050                4$:
2880
2881 012050 004737 041524  JSR      PC,@#CLDISK   ;SET R1-RHCS1, R2-RHCS2
2882                ;R3-RHDS1, R4-RHER1
2883                ;GIVE RH-11 INITIALIZE
2884                ;SETUP UNIT NUMBER
2885 012054 104401 012062  TYPE      ,69$          ;;TYPE ASCIZ STRING
2886 012060 000407          BR        68$          ;;GET OVER THE ASCIZ
2887                ;;69$: .ASCIZ <15><12>/START DRIVE/
2888                68$:
2889 012100                MOV      @#UNIT,-(SP) ;GET UNIT UNDER TEST
2890 012104 104405          TYPDS
2891 012106 104401 012114  TYPE      ,71$          ;;TYPE ASCIZ STRING
2892 012112 000420          BR        70$          ;;GET OVER THE ASCIZ
2893                ;;71$: .ASCIZ / AFTER HEAD LOAD HIT CONTINUE/<15><12>
2894                70$:
2895 012154 000000          HALT
2896 012156 032713 010000  BIT      #MOL,@R3      ;MOL IN RHDS1 SHOULD BE = 1
2897 012162 001411          BEQ      5$           ;BRANCH IF MOL = 0
2898 012164 032713 000400  BIT      #DPR,@R3     ;DPR IN RHDS1 SHOULD BE - 1
2899 012170 001406          BEQ      5$           ;BRANCH IF DPR = 0
2900 012172 032713 000200  BIT      #DRY,@R3     ;DRY IN RHDS1 SHOULD BE = 1
2901 012176 001403          BEQ      5$           ;BRANCH IF DRY = 0
2902 012200 032713 000100  BIT      #VV,@R3      ;VV IN RHDS1 SHOULD BE = 0
2903 012204 001403          BEQ      6$           ;BRANCH IF VV = 0 (GOOD)
2904 012206 010337 001122  5$:  MOV      R3,@#SBDADR   ;FAILING REGISTER ADDRESS - RHDS1
2905 012212 104012          ERROR     12          ;WITH SPINDLE POWERED UP
2906                ;RHDS1 SHOULD HAVE VV = 0, MOL = 1
2907
2908 012214 011100                6$:  MOV      @R1,R0          ;GET RHCS1 CONTENTS
2909 012216 042700 160076  BIC      #SC!TRE!MCPE.76,R0 ;CLEAR SC,TRE,MCPE AND
2910                ;ALL FUNCTION BITS
2911 012222 022700 004200  CMP      #DVA'RDY,R0   ;RHCS1 SHOULD HAVE
2912                ;GO = 0, DVA = 1, RDY = 1
2913 012226 001403          BEQ      7$           ;BRANCH IF RHCS1 IS GOOD
  
```



```

2914 012230 010137 001122      MOV    R1,@#SBDADR      ; FAILING REGISTER RHCS1
2915 012234 104013              ERROR  13              ; AFTER A POWER UP WITHOUT ANY
2916                                ; INIT RHCS1 SHOULD HAVE
2917                                ; GO = 0, DVA = 1, RDY = 1, IE = 0
2918                                ; DISREGARD ALL OTHER BITS
2919
2920 012236 005777 170072      7$:   TST    @RHCC        ; TEST RHCC, IT SHOULD = 0
2921 012242 001411              BEQ    10$              ; BRANCH IF RHCC = 0
2922 012244 013737 002334 004600  MOV    @#RHCC,@#REGADR ; FAILING REGISTER RHCC
2923 012252 005037 001124      CLR    @#SGDDAT        ; RHCC SHOULD BE = 0
2924 012256 017737 170052 001126  MOV    @RHCC,@#SBDAT   ; BAD DATA
2925 012264 104014              ERROR  14              ; AFTER POWER UP RHCC
2926                                ; SHOULD BE 0
2927
2928 012266                        10$:
2929
2930 012266 004737 041524      JSR    PC,@#CLDISK    ; SET R1-RHCS1, R2-RHCS2
2931                                ; R3-RHDS1, R4-RHER1
2932                                ; GIVE RH-11 INITIALIZE
2933                                ; SETUP UNIT NUMBER
2934 012272 004737 041562      JSR    PC,@#CHECK     ; CHECK THAT DVA,RDY,MOL,DPR,DRY - 1
2935                                ; AND THAT NO STATUS BITS ARE STUCK = 1
2936 012276 104401 067033      TYPE   ,CPHALT        ; CANNOT CONTINUE TESTS IF THEY AREN'T
2937 012302 000000              HALT                    ; STOP
2938 012304 013777 002460 167766  MOV    @#PKACK,@RHCS1 ; GET READY FOR PKACK
2939                                ; PACK ACKNOWLEDGE WITH 22 IN RHCS1
2940
2941
2942                                ; *NOW SAVE REGISTERS FOR COMPARISON AFTER PACK ACKNOWLEDGE
2943
2944 012312 004037 041672      JSR    R0,@#SAVER     ; SAVE REGISTERS
2945 012316 002272              RHCW                    ; RHCW IS THE FIRST REGISTER SAVED
2946 012320 004612              SAVERE                  ; STARTING ADDRESS OF WHERE
2947                                ; THE REGISTERS ARE SAVED
2948 012322 000022              18.                    ; NUMBER OF REGISTERS
2949                                ; SAVED = 18.
2950
2951 012324 013777 004606 167734  MOV    @#RP4VEC,@RPVEC ; SET RP04 VECTOR ADDRESS
2952                                ; TO 'TIME1' IF P-CLOCK IS PRESENT
2953                                ; OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
2954                                ; 'TIME' WILL ONLY SAVE
2955                                ; CURRENT CYLINDER ADDRESS
2956                                ; AND LOOK AHEAD REGISTERS
2957
2958
2959 012332 013746 002460      MOV    @#PKACK,-(SP)   ; GET READY TO MOVE COMMAND
2960 012336 052716 000001      BIS    #GO,(SP)       ; GET READY TO SET GO
2961                                ; WITHOUT INTERRUPT ENABLE
2962 012342 012677 167732      MOV    (SP)+,@RHCS1   ; GO WITH
2963                                ; 22 IN RHCS1 FOR PACK ACKNOWLEDGE
2964                                ; WITH INTERRUPT DISABLED
2965
2966
2967
2968 012346 104413              WAT                    ; WAIT FOR VV BIT TO SET
2969 012350 002322              RHDS1                  ; WAIT FOR RHDS1 REGISTER
  
```

```
2970 012352 000100          VV          ;WAIT FOR VV BIT IN RHDS1 REGISTER
2971 012354 000001          1.          ;ALLOW 10 MICRO SECONDS
2972 012356 000001          1.          ;VV MUST SET BETWEEN
2973                                     ;00 AND 20 MICRO SECONDS
2974
2975
2976 012360 004037 042404    JSR      RO,@#CHREG    ;CHANGE BITS IN SAVED REGISTER
2977 012364 002322          RHDS1        ;CHANGE RHDS1 REGISTER
2978
2979 012366 000001          1           ;1 BIT/BITS TO BE CHANGED
2980 012370 000001          1           ;NEW VALUE OF VV IS 1
2981 012372 000100          VV          ;CHANGE VV BIT
2982
2983                                     ;*NOW COMPARE REGISTERS SO THAT NO REGISTERS
2984                                     ;*CHANGED EXCEPT VV IN RHDS1 AND IE IN RHCS1
2985
2986
2987 012374 004037 042512    JSR      RO,@#COMREG    ;COMPARE SAVED REGISTERS WITH
2988                                     ;PRESENT VALUE
2989 012400 004612          SAVERE        ;GOOD DATA SAVED IN 'SAVERE'
2990 012402 002354          WC           ;TEST DATA STARTING FROM 'RHWC'
2991 012404 000022          18.          ;18. REGISTERS TO BE COMPARED
2992 012406 012412          11$          ;RETURN TO 11$ ON ERROR
2993 012410 012416          12$          ;RETURN TO 12$ ON NO ERROR
2994
2995 012412 104015          11$:      ERROR 15    ;GIVING A PACK ACKNOWLEDGE
2996 012414 000207          RTS      PC      ;CAUSED AN ERROR
2997                                     ;PACK ACKNOWLEDGE SHOULD
2998                                     ;SET VV IN RHDS1
2999                                     ;INTERRUPT SHOULD MAKE
3000                                     ;IE = 0
3001                                     ;NO OTHER REGISTERS SHOULD
3002                                     ;CHANGE
3003                                     ;GOOD DATA GIVES
3004                                     ;CONTENTS OF REGISTER BEFORE
3005                                     ;PACK ACKNOWLEDGE
3006                                     ;RECEIVED DATA GIVES
3007                                     ;CONTENTS OF REGISTER
3008                                     ;AFTER PACK ACKNOWLEDGE
3009
3010 012416 012737 177777 047342 12$:  MOV      #-1,@#PRITEM ;CLEAR PREVIOUS ITEM NUMBER
3011
```

```
3012
3013
3014
3015
3016 *****
3017 *TEST 11      SET VV BIT #6 IN RHDS1
3018 *           THIS TEST SETS VV IN RHDS1 INCASE
3019 *           ACT-11 MONITOR IS PRESENT AND THE PREVIOUS TEST
3020 *           IS NOT PERFORMED
3021 *           THERE IS A RESET AT THE BEGINING OF THIS TEST
3022 *           FOR ERROR RECOVERY ONLY.
3023 *****
3024 012424 000004      TST11: SCOPE
3025
3026 ;*IN CASE THERE IS ANY DRIVE ERRORS DURING POWER UP
3027 ;*OR POWER DOWN OR ANY PARITY ERRORS A RESET IS GIVEN
3028 012426 000005      RESET
3029 012430 004737 045734 JSR    PC,@#STKINT    ;INITILIZE TK
3030
3031
3032 012434 012706 001000 MOV    #STACK,SP      ;RESET STACK
3033 012440 012737 000011 004604 MOV    #11,@#TSTNM   ;SAVE TEST NUMBER
3034
3035 012446 004737 041524 JSR    PC,@#CLDISK   ;SET R1-RHCS1, R2-RHCS2
3036 ;R3-RHDS1, R4-RHER1
3037 ;GIVE RH-11 INITIALIZE
3038 ;SETUP UNIT NUMBER
3039 012452 004737 041562 JSR    PC,@#CHECK    ;CHECK THAT DVA,RDY,MOL,DPR,DRY = 1
3040 ;AND THAT NO STATUS BITS ARE STUCK = 1
3041 012456 104401 067033 TYPE   ,CPHALT       ;CANNOT CONTINUE TESTS IF THEY AREN'T
3042 012462 000000      HALT
3043 012464 013777 002460 167606 MOV    @#PKACK,@RHCS1 ;GET READY FOR PKACK
3044 ;PACK ACKNOWLEDGE WITH 22 IN RHCS1
3045
3046
3047 ;*NOW SAVE REGISTERS FOR COMPARISON AFTER PACK ACKNOWLEDGE
3048
3049 012472 004037 041672 JSR    RO,@#SAVER    ;SAVE REGISTERS
3050 012476 002272      RHCW    ;RHCW IS THE FIRST REGISTER SAVED
3051 012500 004612      SAVERE  ;STARTING ADDRESS OF WHERE
3052 ;THE REGISTERS ARE SAVED
3053 012502 000022      18.    ;NUMBER OF REGISTERS
3054 ;SAVED = 18.
3055
3056 012504 013777 004606 167554 MOV    @#RP4VEC,@RPVEC ;SET RPO4 VECTOR ADDRESS
3057 ;TO 'TIME1' IF P-CLOCK IS PRESENT
3058 ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
3059 ;'TIME' WILL ONLY SAVE
3060 ;CURRENT CYLINDER ADDRESS
3061 ;AND LOOK AHEAD REGISTERS
3062
3063
3064 012512 013746 002460 MOV    @#PKACK,-(SP) ;GET READY TO MOVE COMMAND
3065 012516 052716 000001 BIS    #GO,(SP)      ;GET READY TO SET GO
3066 ;WITHOUT INTERRUPT ENABLE
3067 012522 012677 167552 MOV    (SP)+,@RHCS1 ;GO WITH
```

```
3068 ;22 IN RHCS1 FOR PACK ACKNOWLEDGE
3069 ;WITH INTERRUPT DISABLED
3070
3071 012526 011100 MOV @R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
3072 012530 011305 MOV @R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
3073
3074
3075 012532 104413 WAT ;WAIT FOR VV BIT TO SET
3076 012534 002322 RHDS1 ;WAIT FOR RHDS1 REGISTER
3077 012536 000100 VV ;WAIT FOR VV BIT IN RHDS1 REGISTER
3078 012540 000001 1. ;ALLOW 10 MICRO SECONDS
3079 012542 000001 1. ;VV MUST SET BETWEEN
3080 ;00 AND 20 MICRO SECONDS
3081
3082 ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
3083 ;*RO AND R5 IMMEDIATELY AFTER GO
3084
3085 012544 013746 002460 MOV @#PKACK,-(SP) ;SAVE COMMAND
3086 012550 052716 004200 BIS #DVA!RDY,(SP) ;INCLUDE DVA!RDY
3087 012554 011637 001124 MOV (SP),@#SGDDAT ;SAVE FOR PRINTOUT
3088 012560 022600 CMP (SP)+,R0 ;DURING ABOVE OPERATION ONLY DVA!RDY
3089 ;AND COMMAND SHOULD BE SET
3090 012562 001405 BEQ 64$ ;BRANCH IF GOOD
3091 012564 010037 001126 MOV R0,@#SBDDAT ;BAD DATA
3092 012570 010137 004600 MOV R1,@#REGADR ;FAILING REGISTER RHCS1
3093 012574 104021 ERROR 21 ;DURING ABOVE OPERATION ONLY
3094 ;COMMAND AND DVA!RDY SHOULD BE SET
3095 012576 012746 010700 64$: MOV #MOL!DPR!DRY!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
3096 012602 011637 001124 MOV (SP),@#SGDDAT ;SAVE FOR PRINTOUT
3097 012606 022605 CMP (SP)+,R5 ;DURING ABOVE OPERATION ONLY MOL.DPR!DRY VV
3098 ;SHOULD BE SET
3099 012610 001405 BEQ 66$ ;BRANCH IF GOOD
3100 012612 010537 001126 MOV R5,@#SBDDAT ;BAD DATA
3101 012616 010337 004600 MOV R3,@#REGADR ;FAILING REGISTER RHDS1
3102 012622 104063 ERROR 63 ;DURING ABOVE OPERATION ONLY
3103 ;MOL!DPR!DRY!VV SHOULD BE SET
3104 012624 66$:
3105
3106 012624 004037 042404 JSR R0,@#CHREG ;CHANGE BITS IN SAVED REGISTER
3107 012630 002322 RHDS1 ;CHANGE RHDS1 REGISTER
3108
3109 012632 000001 1 ;1 BIT/BITS TO BE CHANGED
3110 012634 000001 1 ;NEW VALUE OF VV IS 1
3111 012636 000100 VV ;CHANGE VV BIT
3112
3113 ;*NOW COMPARE REGISTERS BEFORE PACK ACKNOWLEDGE
3114 ;*WITH AFTER PACK ACKNOWLEDGE
3115
3116
3117
3118 012640 004037 042512 JSR R0,@#COMREG ;COMPARE SAVED REGISTERS WITH
3119 ;PRESENT VALUE
3120 012644 004612 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
3121 012646 002354 WC ;TEST DATA STARTING FROM 'RHWC'
3122 012650 000022 18. ;18. REGISTERS TO BE COMPARED
3123 012652 012656 1$ ;RETURN TO 1$ ON ERROR
```

3124	012654	012662		2\$;RETURN TO 2\$ ON NO ERROR
3125						
3126	012656	104015	1\$:	ERROR	15	;GIVING A PACK ACKNOWLEDGE
3127	012660	000207		RTS	PC	;CAUSED AN ERROR
3128						;PACK ACKNOWLEDGE SHOULD
3129						;SET VV IN RHDS1
3130						;INTERRUPT SHOULD MAKE
3131						;IE = 0
3132						;NO OTHER REGISTERS SHOULD
3133						;CHANGE
3134						;GOOD DATA GIVES CONTENTS
3135						;OF REGISTER BEFORE COMMAND
3136						;RECEIVED DATA GIVES CONTENTS
3137						;OF REGISTER AFTER COMMAND
3138	012662		2\$:			

```

3139
3140 ;:*****
3141 012662 005737 004744      TST    @#RP06      ;TEST FOR RPO6 DRIVE
3142 012666 001005          BNE    3$          ;IF = 1 DO 'MAKECYL' 777
3143 012670 005737 004746      TST    @#RPO5      ;TEST FOR RPO5 DRIVE
3144 012674 001004          BNE    4$          ;IF = 1 DO 'MAKECYL' 377
3145 012676 000137 015022      JMP    TST16      ; IF BOTH = 0, DON'T DO EITHER 'MAKECYL' -----)
3146                                     ;OR THE ADDRESS PLUG TESTS
3147 012702          3$:
3148 012702 000137 012746      JMP    TST13      ; DO 'MAKECYL' 777 -----)
3149 012706          4$:
3150 ;:*****
3151
3152
3153
3154 ;:*****
3155 ;*TEST 12      MAKE CURRENT CYLINDER = 377
3156 ;:*****
3157 012706 000004      TST12: SCOPE
3158 012710 012706 001000      MOV    #STACK,SP  ;RESET STACK
3159 012714 012737 000012 004604      MOV    #13-1,@#TSTNM ;THIS SAVES TEST NUMBER
3160
3161 012722 004737 041524          JSR    PC,@#CLDISK ;INIT DRIVE
3162 012726 012777 000001 167364      MOV    #DMD,@RHMR  ;SET DIAGNOSTIC MODE
3163 012734 004037 041174          JSR    RO,@#MAKECYL ;SUBROUTINE TO GIVE A SEEK
3164                                     ;COMMAND FOLLOWED BY A INIT
3165                                     ;THIS SHOULD CHANGE RHCC TO 377
3166 012740 000377          377
3167
3168
3169 012742 000137 013002      JMP    TST14      ; SKIP NEXT 'MAKECYL' -----)
3170
3171
3172
3173 ;:*****
3174 ;*TEST 13      MAKE CURRENT CYLINDER - 777
3175 ;:*****
3176 012746 000004      TST13: SCOPE
3177 012750 012706 001000      MOV    #STACK,SP  ;RESET STACK
3178 012754 012737 000013 004604      MOV    #14-1,@#TSTNM ;THIS SAVES TEST NUMBER
3179
3180 012762 004737 041524          JSR    PC,@#CLDISK ;INIT DRIVE
3181 012766 012777 000001 167324      MOV    #DMD,@RHMR  ;SET DIAGNOSTIC MODE
3182 012774 004037 041174          JSR    RO,@#MAKECYL ;SUBROUTINE TO GIVE A SEEK
3183                                     ;COMMAND FOLLOWED BY A INIT
3184                                     ;THIS SHOULD CHANGE RHCC TO 777
3185 013000 000777          777
3186
3187

```

```
3188
3189
3190 ;*****
3191 ;*TEST 14 ADDRESS PLUG CHANGE ERROR
3192
3193 ;* CHECK PROPER ADDRESS PLUG FUNCTIONALITY
3194 ;* BY PULLING THE ADDRESS PLUG DURING A COMMAND ISSUED
3195 ;* IN DIAGNOSTIC MODE (TO GUARANTEE COMMAND IS STILL ACTIVE WHEN
3196 ;* PLUG IS PULLED) AND VERIFYING THAT THE DRIVE GOES OFF LINE
3197
3198 ;* THE ADDRESS PLUG IS THEN REPLACED AND RETURN OF THE PROPER
3199 ;* RESPONDING DRIVE IS VERIFIED, AS WELL AS THE FACT THAT
3200 ;* ATTENTION BITS COME UP IN THE PROPER BIT LOCATION(?????), AND THE
3201 ;* ADDRESS CHANGE ERROR (ACE) - RHER3 BIT #13 IS SET, AS WELL AS
3202 ;* SC,TRE,ERR,AND MCPE; VV IS RESET AND RHCC = 000 (DRIVE RECALIBRATED)
3203
3204 ;* IN ADDITION VERIFICATION IS ALSO MADE THAT NO OTHER DRIVE
3205 ;* NUMBERS APPEAR ON THE BUSS AFTER THE PLUG IS REPLACED. (??????)
3206
3207 ;*****
3208 013002 000004 TST14: SCOPE
3209 013004 012737 000001 001212 MOV #1,$TIMES ;DO 1 ITERATION
3210
3211 ;*****
3212 013012 005737 004744 TST @#RP06 ;TEST FOR RP06 DRIVE
3213 013016 001005 BNE 4$ ;IF = 1, DO THIS TEST
3214 013020 005737 004746 TST @#RP05 ;TEST FOR MEMOREX RP04
3215 013024 001002 BNE 4$ ;IF = 1, DO THIS TEST
3216 ;IF NEITHER FLAG IS UP, ASSUME THE
3217 ;DRIVE IS AN ISS RP04 AND SKIP TEST
3218 013026 000137 014014 JMP TST15 ; JUMP TO NEXT TEST -----)
3219 013032 4$:
3220 ;*****
3221
3222 013032 012706 001000 MOV #STACK,SP ;RESET STACK
3223 013036 012737 000014 004604 MOV #14,@#TSTNM ;SAVE TEST NUMBER
3224
3225 013044 004737 041524 JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
3226 ;R3-RHDS1, R4-RHER1
3227 ;GIVE RH-11 INITIALIZE
3228 ;SETUP UNIT NUMBER
3229 013050 004737 041604 JSR PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
3230 ;AND THAT NO STATUS BITS ARE STUCK = 1
3231 013054 104401 067033 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF ANY OF
3232 ;THE FIRST SET OF BITS DON'T - 1
3233 013060 000000 HALT ;STOP
3234
3235 ;*THIS CODE CHECKS TO SEE IF MANUAL INTERVENTION TESTS ARE OK
3236
3237 013062 005737 004724 TST @#NOPUSH ;IS THIS A 220 START ?
3238 013066 001007 BNE 1$ ;SKIP THIS TEST IF SO
3239 013070 005737 000042 TST @#42 ;MONITOR (ACT 11) RETURN ADDRESS ?
3240 013074 001004 BNE 1$ ;SKIP THIS TEST
3241 013076 005737 001100 TST @#$PASS ;FIRST PASS ?
3242 013102 001001 BNE 1$ ;SKIP THIS TEST IF NOT
3243 013104 000402 BR 2$ ;CONTINUE WITH THIS TEST
```

```

3244
3245 013106 1$:
3246 013106 000137 014014 2$: JMP TST15 ; JUMP TO NEXT TEST -----)
3247 013112
3248
3249 ;*SET DIAGNOSTIC MODE TO ENABLE A COMMAND ACTIVE WHILE
3250 ;*THE PLUG IS PULLED
3251
3252 013112 052777 000001 167200 BIS #DMD,@RHMR ;SET UP DIAGNOSTIC MODE
3253
3254 ;*TAKE AN INITIAL REGISTER SNAPSHOT
3255
3256 013120 004037 041672 JSR RO,@#SAVER ;SAVE REGISTERS
3257 013124 002272 RHWC ;RHWC IS THE FIRST REGISTER SAVED
3258 013126 004612 SAVERE ;STARTING ADDRESS OF WHERE
3259 ;THE REGISTERS ARE SAVED
3260 013130 000022 18. ;NUMBER OF REGISTERS
3261 ;SAVED = 18.
3262
3263
3264 ;*ISSUE A COMMAND AND THE 'GO' BIT (NOT POSITIONING COMMAND)
3265 ;*TO VERIFY COMMAND ABORT IF PLUG IS PULLED
3266
3267
3268
3269 ;*ISSUE SOME CLOCKS TO GET THE COMMAND STARTED
3270 ;*(USE "SEARCH" WITH "DTETST" FLAG UP TO STOP CLOCKING ?)
3271
3272
3273
3274 013132 104401 013140 TYPE ,65$ ;;TYPE ASCIZ STRING
3275 013136 000420 BR 64$ ;;GET OVER THE ASCIZ
3276 ;;65$: .ASCIZ <15><12>/REMOVE ADDRESS PLUG ON DRIVE/
3277 013200 64$:
3278 013200 013746 004716 MOV @#UNIT,-(SP) ;GET THE UNIT NO. UNDER TEST
3279 013204 104405 TYPDS ;TYPE IT OUT
3280 013206 104401 013214 TYPE ,67$ ;;TYPE ASCIZ STRING
3281 013212 C00415 BR 66$ ;;GET OVER THE ASCIZ
3282 ;;67$: .ASCIZ / THEN PRESS CONTINUE/
3283 66$:
3284 013246 000000 HALT ;WAIT FOR OPERATOR PLUG CHANGE
3285
3286 ;*CHECK THAT THE UNIT NO. UNDER TEST HAS GONE OFFLINE
3287
3288 013250 017700 167030 MOV @RHDST,RO ;ATTEMPT TO ADDRESS THE DRIVE
3289 013254 004737 043436 JSR PC,@#PUTREG ;TAKE REGISTER SNAPSHOTS
3290 013260 032737 010000 002360 BIT #NED,@#CS2 ;TEST FOR NON EXISTENT DRIVE
3291 013266 001001 BNE 7$ ;CONTINUE IF 'NED' BIT SET (UNIT
3292 ;IS OFFLINE AS IT SHOULD BE)
3293 013270 104101 ERROR 101 ;UNIT DID NOT GO OFFLINE WHEN ADDRESS
3294 ;PLUG WAS REMOVED
3295 013272 7$:

```



```
3296
3297                                     ;*NOW REPLACE THE ADDRESS PLUG
3298
3299
3300 013272 104401 013300             TYPE      ,69$                ;;TYPE ASCIZ STRING
3301 013276 000420                   BR        68$                ;;GET OVER THE ASCIZ
3302                                     ;;69$: .ASCIZ <15><12>/REPLACE ADDRESS PLUG ON DRIVE/
3303 013340                               68$:
3304 013340 013746 004716             MOV      @#UNIT,-(SP)      ;GET THE UNIT UNDER TEST
3305 013344 104405                   TYPDS                    ;TYPE IT OUT
3306 013346 104401 013354             TYPE      ,71$                ;;TYPE ASCIZ STRING
3307 013352 000416                   BR        70$                ;;GET OVER THE ASCIZ
3308                                     ;;71$: .ASCIZ / THEN PRESS CONTINUE/<15><12>
3309 013410                               70$:
3310 013410 000000                   HALT                      ;WAIT FOR OPERATOR PLUG REPLACEMENT
3311
3312                                     ;*CHECK THAT THE ORIGINAL UNIT HAS COME BACK ON LINE
3313
3314 013412 004737 043436             JSR      PC,@#PUTREG      ;TAKE NEW REGISTER SNAPSHOTS
3315 013416 032737 000400 002404     BIT      #DPR,@#DS1      ;TEST THAT 'DPR' = 1
3316 013424 001411                   BEQ     9$                ;ERROR - DRIVE SHOULD BE PRESENT
3317 013426 032737 004000 002362     BIT      #DVA,@#CS1      ;TEST THAT DEVICE IS NOW AVAILABLE
3318 013434 001405                   BEQ     9$                ;ERROR - 'DVA' SHOULD = 1
3319 013436 032737 000200 002404     BIT      #DRY,@#DS1      ;TEST THAT DRIVE READY IS - 1
3320 013444 001401                   BEQ     9$                ;ERROR - 'DRY' SHOULD - 1
3321 013446 000403                   BR      8$                ;A-OK: 'DPR' = 1, 'DVA' - 1, & 'DRY' 1
3322 013450 104102                   9$: ERROR 102            ;UNIT NOT AVAILABLE AFTER
3323                                     ;ADDRESS PLUG WAS REPLACED
3324 013452 000137 014014             JMP      TST15           ;JUMP TO NEXT TEST -----)
3325
3326 013456                               8$: ;CHANGE THE INITIAL REGISTER SNAPSHOT TO EXPECTED VALUES
3327                                     ;*AFTER THE PLUG CHANGE
3328
3329
3330 013456 004037 042404             JSR      R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
3331 013462 002300                   RHCS1                    ;CHANGE RHCS1 REGISTER
3332
3333 013464 000004                   4                        ;4 BIT/BITS TO BE CHANGED
3334 013466 000001                   1                        ;NEW VALUE OF RDY IS 1
3335 013470 000200                   RDY                      ;CHANGE RDY BIT
3336 013472 000001                   1                        ;NEW VALUE OF SC IS 1
3337 013474 100000                   SC                        ;CHANGE SC BIT
3338 013476 000001                   1                        ;NEW VALUE OF TRE IS 1
3339 013500 040000                   TRE                      ;CHANGE TRE BIT
3340 013502 000001                   1                        ;NEW VALUE OF MCPE IS 1
3341 013504 020000                   MCPE                     ;CHANGE MCPE BIT
3342
3343 013506 004037 042404             JSR      R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
3344 013512 002322                   RHDS1                    ;CHANGE RHDS1 REGISTER
3345
3346 013514 000003                   3                        ;3 BIT/BITS TO BE CHANGED
3347 013516 000001                   1                        ;NEW VALUE OF ATA IS 1
3348 013520 100000                   ATA                      ;CHANGE ATA BIT
3349 013522 000001                   1                        ;NEW VALUE OF ERR IS 1
3350 013524 040000                   ERR                      ;CHANGE ERR BIT
3351 013526 000000                   0                        ;NEW VALUE OF VV IS 0
```



```
3387
3388
3389 ;*NOW CLEAR OUT THE CONTROLLER AND DRIVE
3390
3391 013616 68:
3392
3393 013616 004737 041524 JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
3394 ;R3-RHDS1, R4-RHER1
3395 ;GIVE RH-11 INITIALIZE
3396 ;SETUP UNIT NUMBER
3397
3398 ;*CHANGE THE REGISTER SNAPSHOT TO EXPECTED VALUES AFTER THE CLEAR
3399
3400
3401 013622 004037 042404 JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
3402 013626 002300 RHCS1 ;CHANGE RHCS1 REGISTER
3403
3404 013630 000003 3 ;3 BIT/BITS TO BE CHANGED
3405 013632 000000 0 ;NEW VALUE OF SC IS 0
3406 013634 100000 SC ;CHANGE SC BIT
3407 013636 000000 0 ;NEW VALUE OF TRE IS 0
3408 013640 040000 TRE ;CHANGE TRE BIT
3409 013642 000000 0 ;NEW VALUE OF MCPE IS 0
3410 013644 020000 MCPE ;CHANGE MCPE BIT
3411
3412 013646 004037 042404 JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
3413 013652 002322 RHDS1 ;CHANGE RHDS1 REGISTER
3414
3415 013654 000002 2 ;2 BIT/BITS TO BE CHANGED
3416 013656 000000 0 ;NEW VALUE OF ATA IS 0
3417 013660 100000 ATA ;CHANGE ATA BIT
3418 013662 000000 0 ;NEW VALUE OF ERR IS 0
3419 013664 040000 ERR ;CHANGE ERR BIT
3420
3421 013666 004037 042404 JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
3422 013672 002276 RHCS2 ;CHANGE RHCS2 REGISTER
3423
3424 013674 000001 1 ;1 BIT/BITS TO BE CHANGED
3425 013676 000000 0 ;NEW VALUE OF NED IS 0
3426 013700 010000 NED ;CHANGE NED BIT
3427
3428 013702 004037 042404 JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
3429 013706 002314 RHER3 ;CHANGE RHER3 REGISTER
3430
3431 013710 000001 1 ;1 BIT/BITS TO BE CHANGED
3432 013712 000000 0 ;NEW VALUE OF ACE IS 0
3433 013714 020000 ACE ;CHANGE ACE BIT
3434
3435 013716 004037 042404 JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
3436 013722 002320 RHMR ;CHANGE RHMR REGISTER
3437
3438 013724 000001 1 ;1 BIT/BITS TO BE CHANGED
3439 013726 000000 0 ;NEW VALUE OF DMD IS 0
3440 013730 000001 DMD ;CHANGE DMD BIT
3441
3442 013732 043737 004740 004636 BIC @#ATTENT,@#SAVERE+24 ;UNIT UNDER TEST ATTENTION BIT
```

```
3443
3444
3445 ;*TAKE ANOTHER REGISTER SNAPSHOT AND COMPARE RESULTS
3446 ;*WITH THE EXPECTED VALUES
3447
3448
3449 013740 004037 042512 JSR RO,@#COMREG ;COMPARE SAVED REGISTERS WITH
3450 ;PRESENT VALUE
3451 013744 004612 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
3452 013746 002354 WC ;TEST DATA STARTING FROM 'RHWC'
3453 013750 000022 18. ;18. REGISTERS TO BE COMPARED
3454 013752 013756 10$ ;RETURN TO 10$ ON ERROR
3455 013754 013762 11$ ;RETURN TO 11$ ON NO ERROR
3456
3457 013756 104100 10$: ERROR 100 ;ADDRESS PLUG CHANGE CAUSED SOME
3458 013760 000207 RTS PC ;INCORRECT REGISTER RESULT
3459
3460 013762 11$:
3461
3462 ;*(USE NED METHOD TO VERIFY
3463 ;*THAT ATTENTION BIT COMES UP IN THE PROPER LOCATION ??)
3464
3465
3466
3467
3468 ;*SET 'VV' IN RHDS1 AFTER RESET FROM THE RECALIBRATE
3469 ;*CAUSED BY PULLING THE PLUG
3470
3471
3472 013762 013746 002460 MOV @#PKACK,-(SP) ;GET READY TO MOVE COMMAND
3473 013766 052716 000001 BIS #GO,(SP) ;GET READY TO SET GO
3474 ;WITHOUT INTERRUPT ENABLE
3475 013772 012677 166302 MOV (SP)+,@RHCS1 ;GO WITH
3476 ;22 IN RHCS1 FOR PACK ACKNOWLEDGE
3477 ;WITH INTERRUPT DISABLED
3478
3479 013776 011100 MOV @R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
3480 014000 011305 MOV @R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
3481
3482 014002 104413 WAT ;WAIT FOR VV BIT TO SET
3483 014004 002322 RHDS1 ;WAIT FOR RHDS1 REGISTER
3484 014006 000100 VV ;WAIT FOR VV BIT IN RHDS1 REGISTER
3485 014010 000001 1. ;ALLOW 10 MICRO SECONDS
3486 014012 000001 1. ;VV MUST SET BETWEEN
3487 ;00 AND 20 MICRO SECONDS
3488
```

```
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3500
3501 014014 000004
3502 014016 012737 000001 001212
3503
3504
3505 014024 005737 004744
3506 014030 001005
3507 014032 005737 004746
3508 014036 001002
3509
3510
3511 014040 000137 015022
3512 014044
3513
3514
3515
3516
3517 014044 032777 000040 165066
3518 014052 001002
3519 014054 000137 015022
3520 014060
3521
3522
3523
3524
3525 014060 005737 004724
3526 014064 001007
3527 014066 005737 000042
3528 014072 001004
3529 014074 005737 001100
3530 014100 001001
3531 014102 000402
3532
3533 014104
3534 014104 000137 015022
3535 014110
3536 014110 012706 001000
3537 014114 012737 000015 004604
3538
3539 014122 004737 041524
3540
3541
3542
3543 014126 004737 041604
3544
```

```
*****
;*TEST 15 CHECK ALL ADDRESS PLUG ADDRESSES
*****
;* CHECK THAT ALL ADDRESS PLUG NUMBERS FUNCTION PROPERLY
*****
;* THIS TEST IS DONE FOR MEMOREX RP04'S AND RP06'S ONLY
;* THIS TEST IS SELECTED BY SW#5 AND THE DEFAULT IS NOT TO DO IT
*****
TST15: SCOPE
MOV #1,$TIMES ;;DO 1 ITERATION
*****
TST @#RP06 ;TEST FOR RP06 DRIVE
BNE 4$ ;IF = 1, OK TO DO THIS TEST
TST @#RP05 ;TEST FOR MEMOREX RP04
BNE 4$ ;IF = 1, OK TO DO THIS TEST
;NOT AN RP06 OR MEMOREX RP04 SO
;ASSUME AN ISS RP04 AND SKIP TEST
JMP TST16 ; JUMP TO NEXT TEST -----)
4$:
*****
;*CHECK TO SEE IF THIS TEST HAS BEEN SELECTED WITH SW5
BIT #SW5,@SWR ;TEST THE SWITCH
BNE 5$ ;IF 0, TEST HAS NOT BEEN SELECTED
JMP TST16 ; JUMP TO NEXT TEST -----)
5$:
;TEST SELECTED, CONTINUE IT
*****
;*THIS CODE CHECKS TO SEE IF MANUAL INTERVENTION TESTS ARE OK
TST @#NOPUSH ;IS THIS A 220 START ?
BNE 1$ ;SKIP THIS TEST IF SO
TST @#42 ;MONITOR (ACT 11) RETURN ADDRESS ?
BNE 1$ ;SKIP THIS TEST
TST @#$PASS ;FIRST PASS ?
BNE 1$ ;SKIP THIS TEST IF NOT
BR 2$ ;CONTINUE WITH THIS TEST
1$:
JMP TST16 ; JUMP TO NEXT TEST -----)
2$:
MOV #STACK,SP ;RESET STACK
MOV #15,@#TSTNM ;SAVE TEST NUMBER
JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
;R3-RHDS1, R4-RHER1
;GIVE RH-11 INITIALIZE
;SETUP UNIT NUMBER
JSR PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
;AND THAT NO STATUS BITS ARE STUCK = 1
```

```

3545 014132 104401 067033      TYPE      ,CPHALT      ;CANNOT CONTINUE TESTING IF ANY OF
3546                                ;THE FIRST SET OF BITS DON'T = 1
3547 014136 000000      HALT                                ;STOP
3548
3549
3550 014140 104401 014146      TYPE      ,65$          ;;TYPE ASCIZ STRING
3551 014144 000426      BR        64$          ;;GET OVER THE ASCIZ
3552                                ;;65$: .ASCIZ <15><12>/ALL DISK DRIVES EXCEPTING THE UNIT UNDER/
3553 014222                                64$:
3554 014222 104401 014230      TYPE      ,67$          ;;TYPE ASCIZ STRING
3555 014226 000417      BR        66$          ;;GET OVER THE ASCIZ
3556                                ;;67$: .ASCIZ <15><12>/TEST MUST BE POWERED DOWN/<15><12>
3557 014266                                66$:
3558
3559
3560
3561 014266 013737 004716 004752  MOV      @#UNIT,@#INUNIT ;MAKE THE INITIAL UNIT NO. = "UNIT"
3562
  
```

3563
3564
3565
3566
3567
3568
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3610
3611
3612
3613
3614
3615

014274
014274 104401 014302
014300 000420
014342
014342 013746 004716
014346 104405
014350 104401 014356
014354 000421
014420
014420 104401 014426
014424 000417
014464
014464 000000
014466 005037 047342
014472 005237 004716
014476 042737 177770 004716
014504 004737 041524
014510 017700 165570
014514 004737 043436
014520 032737 010000 002360
014526 001423
014530 104102
014532 104401 014540
014536 000407
014556
014556 013746 004716
014562 104405
014564 104401 014572
014570 000402
014576

```
;;*CHANGE ADDRESS PLUG ON THE UNIT UNDER TEST  
6$:  
TYPE ,69$ ;;TYPE ASCIZ STRING  
BR 68$ ;;GET OVER THE ASCIZ  
;;69$: .ASCIZ <15><12>/REMOVE ADDRESS PLUG ON DRIVE/  
68$:  
MOV @#UNIT,-(SP) ;GET THE UNIT UNDER TEST  
TYPDS ;TYPE IT OUT  
TYPE ,71$ ;;TYPE ASCIZ STRING  
BR 70$ ;;GET OVER THE ASCIZ  
;;71$: .ASCIZ / REPLACE WITH NEXT HIGHER ADDRESS/  
70$:  
TYPE ,73$ ;;TYPE ASCIZ STRING  
BR 72$ ;;GET OVER THE ASCIZ  
;;73$: .ASCIZ <15><12>/PLUG - THEN PRESS CONTINUE/  
72$:  
HALT  
;;*HOUSEKEEPING  
CLR @#PRITEM ;CLEAR THE PREVIOUS ERROR NUMBER  
INC @#UNIT ;ADD ONE TO THE UNIT NO.  
BIC #^C7,@#UNIT ;TRUNCATE TO LOW ORDER 3 BITS  
JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2  
;R3-RHDS1, R4-RHER1  
;GIVE RH-11 INITIALIZE  
;SETUP UNIT NUMBER  
;;*ATTEMPT TO ADDRESS THE NEW UNIT NUMBER  
MOV @#RHDST,R0 ;ATTEMPT TO ADDRESS THE NEW DRIVE NO.  
JSR PC,@#PUTREG ;TAKE REG. SNAPSHOT IN CASE OF ERROR  
BIT #NED,@#CS2 ;TEST FOR NON EXISTENT DRIVE  
BEQ 7$ ;CONTINUE IF 'NED' IS NOT SET - DRIVE  
;SHOULD BE EXISTENT ON THE BUSS  
ERROR 102 ;UNIT NOT AVAILABLE AFTER ADDRESS  
;PLUG REPLACED  
TYPE ,75$ ;;TYPE ASCIZ STRING  
BR 74$ ;;GET OVER THE ASCIZ  
;;75$: .ASCIZ <15><12>/UNIT NUMBER/  
74$:  
MOV @#UNIT,-(SP) ;GET THE BAD UNIT NUMBER  
TYPDS ;TYPE IT OUT  
TYPE ,77$ ;;TYPE ASCIZ STRING  
BR 76$ ;;GET OVER THE ASCIZ  
;;77$: .ASCIZ <15><12>/ /  
76$:
```

```

3616
3617
3618 ;*CHECK IF ALL UNIT NUMBERS HAVE BEEN TRIED
3619 014576 023737 004716 004752 7$: CMP @#UNIT,@#INUNIT ;HAVE WE INCREMENTED BACK TO THE
3620 ;ORIGINAL UNIT NO. YET ?
3621 014604 001233 BNE 6$ ;NO..DO NEXT ADDRESS PLUG
3622 ;YES..CONTINUE WITH TESTS
3623
3624
3625
3626 ;*SET 'VV' IN RHDS1 AFTER RESET FROM THE RECALIBRATE
3627 ;*CAUSED BY PULLING THE ADDRESS PLUGS OUT
3628
3629
3630 014606 013746 002460 MOV @#PKACK,-(SP) ;GET READY TO MOVE COMMAND
3631 014612 052716 000001 BIS #GO,(SP) ;GET READY TO SET GO
3632 ;WITHOUT INTERRUPT ENABLE
3633 014616 012677 165456 MOV (SP)+,@RHCS1 ;GO WITH
3634 ;22 IN RHCS1 FOR PACK ACKNOWLEDGE
3635 ;WITH INTERRUPT DISABLED
3636
3637 014622 011100 MOV @R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
3638 014624 011305 MOV @R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
3639
3640 014626 104413 WAT ;WAIT FOR VV BIT TO SET
3641 014630 002322 RHDS1 ;WAIT FOR RHDS1 REGISTER
3642 014632 000100 VV ;WAIT FOR VV BIT IN RHDS1 REGISTER
3643 014634 000001 1. ;ALLOW 10 MICRO SECONDS
3644 014636 000001 1. ;VV MUST SET BETWEEN
3645 ;00 AND 20 MICRO SECONDS
3646
3647 014640 104401 014646 TYPE ,82$ ;;TYPE ASCIZ STRING
3648 014644 000436 BR 81$ ;;GET OVER THE ASCIZ
3649 ;;82$: .ASCIZ <15><12><15><12>/ALL DISK DRIVES WHICH WERE POWERED UP WHEN THE PROGRAM/
3650 81$:
3651 014742 104401 014750 TYPE ,84$ ;;TYPE ASCIZ STRING
3652 014746 000425 BR 83$ ;;GET OVER THE ASCIZ
3653 ;;84$: .ASCIZ <15><12>/WAS STARTED MUST BE POWERED UP AGAIN/<15><12>
3654 83$:
015022

```



```

3655
3656
3657
3658          .SBTTL
3659          .SBTTL **DRIVE COMMAND TESTS**
3660          .SBTTL
3661
3662
3663
3664
3665
3666          ;*****
3667          ;*TEST 16      NO OPERATION FUNCTION TEST
3668          ;*          ALL POSSIBLE REGISTERS ARE CLEARED THEN A 'NOP'=0
3669          ;*          IS GIVEN NO CHANGE SHOULD HAPPEN
3670          ;*          ALL POSSIBLE REGISTERS ARE FILLED WITH ONES THEN A 'NOP'
3671          ;*          IS GIVEN NO CHANGE SHOULD HAPPEN
3672
3673          ;*****
3674          TST16:  SCOPE
3675          015022 000004          MOV      #STACK,SP          ;RESET STACK
3676          015024 012706 001000  MOV      #16,@#TSTNM        ;SAVE TEST NUMBER
3677          015030 012737 000016 004604
3678          015036 004737 041524          JSR      PC,@#CLDISK        ;SET R1-RHCS1, R2-RHCS2
3679          ;R3-RHDS1, R4-RHER1
3680          ;GIVE RH-11 INITIALIZE
3681          ;SETUP UNIT NUMBER
3682          015042 004737 041604          JSR      PC,@#CHECKT        ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
3683          ;AND THAT NO STATUS BITS ARE STUCK = 1
3684          015046 104401 067033          TYPE     ,CPHALT           ;CANNOT CONTINUE TESTING IF ANY OF
3685          ;THE FIRST SET OF BITS DON'T = 1
3686          015052 000000          HALT
3687          ;STOP
3688          015054 013777 002422 165216  MOV      @#NOPERA,@RHCS1    ;GET READY FOR NOPERA
3689          ;NO OPERATION WITH 0 IN RHCS1
3690
3691
3692          ;*NOW SAVE REGISTERS FOR COMPARISON AFTER NO OPERATION
3693
3694          015062 004037 041672          JSR      RO,@#SAVER         ;SAVE REGISTERS
3695          015066 002272          RHWC
3696          015070 004612          SAVERE
3697          ;RHWC IS THE FIRST REGISTER SAVED
3698          ;STARTING ADDRESS OF WHERE
3699          ;THE REGISTERS ARE SAVED
3700          ;NUMBER OF REGISTERS
3701          ;SAVED = 18.
3702          015074 013777 004606 165164  MOV      @#RP4VEC,@RPVEC    ;SET RPO4 VECTOR ADDRESS
3703          ;TO 'TIME1' IF P-CLOCK IS PRESENT
3704          ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
3705          ;'TIME' WILL ONLY SAVE
3706          ;CURRENT CYLINDER ADDRESS
3707          ;AND LOOK AHEAD REGISTERS
3708
3709          015102 013746 002422          MOV      @#NOPERA,-(SP)     ;GET READY TO MOVE COMMAND
3710          015106 052716 000001          BIS      #GO,(SP)          ;GET READY TO SET GO
  
```



```
3744
3745 ;*NOW REPEAT TEST BY MOVING IN ALL POSSIBLE ONES
3746
3747 015152 2$:
3748
3749 015152 004737 041524 JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
3750 ;R3-RHDS1, R4-RHER1
3751 ;GIVE RH-11 INITIALIZE
3752 ;SETUP UNIT NUMBER
3753 015156 012700 002272 MOV #RHWC,R0 ;ADDR. OF ADDR OF RHWC IN R0
3754 015162 012730 177777 MOV #177777,@(R0)+ ;LOAD 177777 INTO RHWC
3755
3756 015166 012730 177777 MOV #177777,@(R0)+ ;LOAD 177777 INTO RHBA
3757
3758 015172 052730 043010 BIS #43010,@(R0)+ ;LOAD 43010 INTO RHCS2
3759 015176 012730 001400 MOV #1400,@(R0)+ ;LOAD 1400 INTO RHCS1
3760
3761 015202 012730 000000 MOV #0,@(R0)+ ;LOAD 0 INTO RHER1
3762
3763 015206 012730 177777 MOV #177777,@(R0)+ ;LOAD 177777 INTO RHDST
3764
3765 015212 012730 000000 MOV #0,@(R0)+ ;LOAD 0 INTO RHER2
3766
3767 015216 012730 177777 MOV #177777,@(R0)+ ;LOAD 177777 INTO RHOF
3768
3769 015222 012730 177777 MOV #177777,@(R0)+ ;LOAD 177777 INTO RHCA
3770
3771 015226 012730 000000 MOV #0,@(R0)+ ;LOAD 0 INTO RHER3
3772
3773 ;*NOW SET BITS IN RHAS FOR ALL DRIVES PRESENT
3774
3775 015232 010046 MOV R0,-(SP) ;;PUSH R0 ON STACK
3776 015234 011446 MOV @R4,-(SP) ;SAVE RHER1 TO REINSTATE LATER
3777 015236 011246 MOV @R2,-(SP) ;SAVE RHCS2 TO BE REINSTATED
3778 ;AFTER ALL ATA BITS HAVE BEEN SET
3779 015240 013700 004742 MOV @#TOTALAT,R0 ;GET DRIVES PRESENT
3780 015244 005012 CLR @R2 ;CLEAR RHCS2 AND CARRY
3781 015246 012705 000010 MOV #8.,R5 ;COUNTER
3782 015252 006000 87$: ROR R0 ;GET BIT INTO CARRY
3783 015254 103002 BCC 88$ ;BRANCH IF NO UNIT ON THIS BIT
3784 015256 012714 177777 MOV #-1,@R4 ;MOVE INTO ERROR REGISTER
3785 ;TO SET ATA BIT
3786 015262 005212 88$: INC @R2 ;INCREMENT RHCS2 TO NEXT UNIT
3787 015264 005305 DEC R5 ;COUNT
3788 015266 001371 BNE 87$ ;BRANCH IF 8 NOT DONE
3789 015270 012612 MOV (SP)+,@R2 ;REINSTATE RHCS2
3790 015272 012614 MOV (SP)+,@R4 ;REINSTATE RHER1
3791 015274 012600 MOV (SP)+,R0 ;;POP STACK INTO R0
3792 015276 005720 TST (R0)+ ;GET OVER PHAS IN R0
3793
3794
3795 015300 012730 177776 MOV #177776,@(R0)+ ;LOAD 177776 INTO RHMR
3796
3797
3798 015304 004737 041604 JSR PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
3799 ;AND THAT NO STATUS BITS ARE STUCK = 1
```

```

3800 015310 104401 067033 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF ANY OF
3801 ;THE FIRST SET OF BITS DON'T = 1
3802 015314 000000 HALT ;STOP
3803 015316 013777 002422 164754 MOV @#NOPERA,@RHCS1 ;GET READY FOR NOPERA
3804 ;NO OPERATION WITH 0 IN RHCS1
3805
3806
3807 ;*NOW SAVE REGISTERS FOR COMPARISON AFTER A NO-OP
3808
3809 015324 004037 041672 JSR RO,@#SAVER ;SAVE REGISTERS
3810 015330 002272 RHWC ;RHWC IS THE FIRST REGISTER SAVED
3811 015332 004612 SAVERE ;STARTING ADDRESS OF WHERE
3812 ;THE REGISTERS ARE SAVED
3813 015334 000022 18. ;NUMBER OF REGISTERS
3814 ;SAVED = 18.
3815
3816 015336 013777 004606 164722 MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
3817 ;TO 'TIME1' IF P-CLOCK IS PRESENT
3818 ;OR TO 'TIME?' IF P-CLOCK IS NOT PRESENT
3819 ;'TIME' WILL ONLY SAVE
3820 ;CURRENT CYLINDER ADDRESS
3821 ;AND LOOK AHEAD REGISTERS
3822
3823
3824 015344 013746 002422 MOV @#NOPERA,-(SP) ;GET READY TO MOVE COMMAND
3825 015350 052716 000001 BIS #GO,(SP) ;GET READY TO SET GO
3826 ;WITHOUT INTERRUPT ENABLE
3827 015354 012677 164720 MOV (SP)+,@RHCS1 ;GO WITH
3828 ;0 IN RHCS1 FOR NO-OPERATION
3829 ;WITH INTERRUPT DISABLED
3830
3831
3832
3833 015360 104413 WAT ;WAIT FOR RDY BIT TO SET
3834 015362 002300 RHCS1 ;WAIT FOR RHCS1 REGISTER
3835 015364 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
3836 015366 000001 1. ;ALLOW 10 MICRO SECONDS
3837 015370 000001 1. ;RDY MUST SET BETWEEN
3838 ;00 AND 20 MICRO SECONDS
3839
3840 ;*CHANGE REGISTERS TO EXPECTED VALUES
3841
3842 015372 005737 004750 TST @#RH70 ;RUNNING ON AN RH70 ?
3843 015376 001406 BEQ 5$ ;IF NOT, SKIP NEXT
3844 015400 005737 004722 TST @#NUNIT ;TESTING MORE THAN ONE DRIVE ?
3845 015404 001003 BNE 5$ ;SKIP NEXT IF SO
3846 015406 042737 100000 004620 BIC #SC,@#SAVERE+6 ;CLEAR 'SC' IN RHCS1
3847
3848 015414 5$:
3849
3850 015414 043737 004740 004636 BIC @#ATTENT,@#SAVERE+24 ;CLEAR APPROPRIATE ATA BITS
3851 ;FOR WORKING DRIVE IN SAVED RHAS
3852
3853 015422 004037 042404 JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
3854 015426 002322 RHDS1 ;CHANGE RHDS1 REGISTER
3855

```

```
3856 015430 000001      1      ;1 BIT/BITS TO BE CHANGED
3857 015432 000000      0      ;NEW VALUE OF ATA IS 0
3858 015434 100000      ATA     ;CHANGE ATA BIT
3859
3860
3861                      ;*NOW COMPARE REGISTERS BEFORE NO-OP WITH
3862                      ;*AFTER NO-OP COMMAND
3863
3864
3865 015436 004037 042512 JSR     RO,@#COMREG   ;COMPARE SAVED REGISTERS WITH
3866                                     ;PRESENT VALUE
3867 015442 004612      SAVERE   ;GOOD DATA SAVED IN 'SAVERE'
3868 015444 002354      WC        ;TEST DATA STARTING FROM 'RHWC'
3869 015446 000022      18.      ;18. REGISTERS TO BE COMPARED
3870 015450 015454      3$       ;RETURN TO 3$ ON ERROR
3871 015452 015460      4$       ;RETURN TO 4$ ON NO ERROR
3872
3873 015454 104016      3$:     ERROR 16   ;GIVING A NO-OP COMMAND
3874 015456 000207      RTS     PC    ;CAUSED AN ERROR
3875                                     ;NO REGISTERS SHOULD CHANGE
3876                                     ;GOOD DATA GIVES REGISTER
3877                                     ;CONTENTS BEFORE COMMAND
3878                                     ;RECEIVED DATA GIVES REGISTER
3879                                     ;CONTENTS AFTER COMMAND
3880 015460      4$:
3881
3882
```

```
3883 .....  
3884 *TEST 17 DRIVE CLEAR  
3885 * ALL WRITE BITS OF ALL REGISTERS ARE FILLED  
3886 * WITH ONES EXCEPT GO,CLR,IE,PAT,MCPE,UPE  
3887 * THEN A DRIVE CLEAR IS GIVEN  
3888 * THEN ALL REGISTERS ARE CHECKED TO HAVE APPROPRIATE VALUE  
3889 .....  
3890 015460 000004 TST17: SCOPE  
3891 015462 012706 001000 MOV #STACK,SP ;RESET STACK  
3892 015466 012737 000017 004604 MOV #17,@#TSTNM ;SAVE TEST NUMBER  
3893  
3894 015474 004737 041524 JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2  
3895 ;R3-RHDS1, R4-RHER1  
3896 ;GIVE RH-11 INITIALIZE  
3897 ;SETUP UNIT NUMBER  
3898 015500 004737 041604 JSR PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1  
3899 ;AND THAT NO STATUS BITS ARE STUCK = 1  
3900 015504 104401 067033 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF ANY OF  
3901 ;THE FIRST SET OF BITS DON'T = 1  
3902 015510 000000 HALT ;STOP  
3903  
3904 ;*WRITE ALL WRITABLE REGISTER BITS  
3905  
3906 015512 012700 002272 MOV #RHWC,R0 ;ADDR. OF ADDR. OF RHWC IN R0  
3907 015516 012730 177777 MOV #177777,@(R0)+ ;LOAD 177777 INTO RHWC  
3908  
3909 015522 012730 177776 MOV #177776,@(R0)+ ;LOAD 177776 INTO RHBA  
3910  
3911 015526 052730 043010 BIS #43010,@(R0)+ ;LOAD 43010 INTO RHCS2  
3912 015532 012730 001400 MOV #1400,@(R0)+ ;LOAD 1400 INTO RHCS1  
3913  
3914 015536 012730 000000 MOV #0,@(R0)+ ;LOAD 0 INTO RHER1  
3915  
3916 015542 012730 177777 MOV #177777,@(R0)+ ;LOAD 177777 INTO RHDS1  
3917  
3918 015546 012730 000000 MOV #0,@(R0)+ ;LOAD 0 INTO RHER2  
3919  
3920 015552 012730 177777 MOV #177777,@(R0)+ ;LOAD 177777 INTO RHOF  
3921  
3922 015556 012730 177777 MOV #177777,@(R0)+ ;LOAD 177777 INTO RHCA  
3923  
3924 015562 012730 000000 MOV #0,@(R0)+ ;LOAD 0 INTO RHER3  
3925  
3926 ;*NOW SET BITS IN RHAS FOR ALL DRIVES PRESENT  
3927  
3928 015566 010046 MOV R0,-(SP) ;:PUSH R0 ON STACK  
3929 015570 011446 MOV @R4,-(SP) ;:SAVE RHER1 TO REINSTATE LATER  
3930 015572 011246 MOV @R2,-(SP) ;:SAVE RHCS2 TO BE REINSTATED  
3931 ;:AFTER ALL ATA BITS HAVE BEEN SET  
3932 015574 013700 004742 MOV @#TOTALAT,R0 ;:GET DRIVES PRESENT  
3933 015600 005012 CLR @R2 ;:CLEAR RHCS2 AND CARRY  
3934 015602 012705 000010 MOV #8.,R5 ;:COUNTER  
3935 015606 006000 84$: ROR R0 ;:GET BIT INTO CARRY  
3936 015610 103002 BCC 85$ ;:BRANCH IF NO UNIT ON THIS BIT  
3937 015612 012714 177777 MOV #-1,@R4 ;:MOVE INTO ERROR REGISTER  
3938 ;:TO SET ATA BIT
```

```

3939 015616 005212      85$: INC @R2 ; INCREMENT RHCS2 TO NEXT UNIT
3940 015620 005305      DEC R5 ; COUNT
3941 015622 001371      BNE 84$ ; BRANCH IF 8 NOT DONE
3942 015624 012612      MOV (SP)+,@R2 ; REINSTATE RHCS2
3943 015626 012614      MOV (SP)+,@R4 ; REINSTATE RHER1
3944 015630 012600      MOV (SP)+,R0 ; POP STACK INTO R0
3945 015632 005720      TST (R0)+ ; GET OVER PHAS IN R0
3946
3947
3948 015634 012730 177776 MOV #177776,@(R0)+ ; LOAD 177776 INTO RHMR
3949
3950
3951 015640 017737 164470 004654 MOV @RHCC,@#SAVERE+42 ; SAVE RHCC IN SAVERE TABLE
3952 015646 013777 004606 164412 MOV @#RP4VEC,@RPVEC ; SET RP04 VECTOR ADDRESS
3953 ; TO 'TIME1' IF P-CLOCK IS PRESENT
3954 ; OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
3955 ; 'TIME' WILL ONLY SAVE
3956 ; CURRENT CYLINDER ADDRESS
3957 ; AND LOOK AHEAD REGISTERS
3958
3959
3960 015654 013746 002430 MOV @#DCLEAR,-(SP) ; GET READY TO MOVE COMMAND
3961 01566C 052716 000001 BIS #GO,(SP) ; GET READY TO SET GO
3962 ; WITHOUT INTERRUPT ENABLE
3963 015664 012677 164410 MOV (SP)+,@RHCS1 ; GO WITH
3964 ; 10 IN RHCS1 FOR DRIVE CLEAR
3965 ; WITH INTERRUPT DISABLED
3966
3967
3968
3969 015670 104413 WAT ; WAIT FOR RDY BIT TO SET
3970 015672 002300 RHCS1 ; WAIT FOR RHCS1 REGISTER
3971 015674 000200 RDY ; WAIT FOR RDY BIT IN RHCS1 REGISTER
3972 015676 000001 1. ; ALLOW 10 MICRO SECONDS
3973 015700 000001 1. ; RDY MUST SET BETWEEN
3974 ; 00 AND 20 MICRO SECONDS
3975
3976 ; *NOW LOAD 'SAVERE' REGISTER SNAPSHOT WITH EXPECTED VALUES
3977
3978 015702 004037 041426 JSR R0,@#FILLRE ; MOV 177777 INTO SAVED RHWC
3979 015706 002272 RHWC ; SAVED REGISTER TO CHANGE
3980 015710 177777 177777 ; DATA
3981 015712 004037 041426 JSR R0,@#FILLRE ; MOV 177776 INTO SAVED RHBA
3982 015716 002274 RHBA ; SAVED REGISTER TO CHANGE
3983 015720 177776 177776 ; DATA
3984 015722 005037 004616 CLR @#SAVERE+4 ; CLEAR LOCATION FOR RHCS2
3985 015726 053737 004716 004616 BIS @#UNIT,@#SAVERE+4 ; PUT UNIT # BACK IN THE SAVED RHCS2
3986
3987 015734 005737 004750 TST @#RH70 ; RUNNING ON AN RH70 CONTROLLER ?
3988 015740 001021 BNE 8$ ; IF SO SKIP NEXT RH11 CODE
3989
3990
3991 015742 004037 042404 JSR R0,@#CHREG ; CHANGE BITS IN SAVED REGISTER
3992 015746 002276 RHCS2 ; CHANGE RHCS2 REGISTER
3993
3994 015750 000003 3 ; 3 BIT/BITS TO BE CHANGED
  
```

```

3995 015752 000001          1          ;NEW VALUE OF IR IS 1
3996 015754 000100          IR          ;CHANGE IR BIT
3997 015756 000001          1          ;NEW VALUE OF BAI IS 1
3998 015760 000010          BAI         ;CHANGE BAI BIT
3999 015762 000001          1          ;NEW VALUE OF MXF IS 1
4000 015764 001000          MXF         ;CHANGE MXF BIT
4001
4002 015766 004037 042404    JSR         RO,@#CHREG    ;CHANGE BITS IN SAVED REGISTER
4003 015772 002300          RHCS1       ;CHANGE RHCS1 REGISTER
4004
4005 015774 000001          1          ;1 BIT/BITS TO BE CHANGED
4006 015776 000001          1          ;NEW VALUE OF SC IS '
4007 016000 100000          SC          ;CHANGE SC BIT
4008 016002 000416          BR         9$          ;SKIP NEXT RH70 CODE
4009
4010 016004          8$:
4011
4012 016004 004037 042404    JSR         RO,@#CHREG    ;CHANGE BITS IN SAVED REGISTER
4013 016010 002276          RHCS2       ;CHANGE RHCS2 REGISTER
4014
4015 016012 000002          2          ;2 BIT/BITS TO BE CHANGED
4016 016014 000001          1          ;NEW VALUE OF IR IS 1
4017 016016 000100          IR          ;CHANGE IR BIT
4018 016020 000001          1          ;NEW VALUE OF BAI IS 1
4019 016022 000010          BAI         ;CHANGE BAI BIT
4020 016024 005737 004722    TST         @#NUNIT       ;TESTING MORE THAN ONE DRIVE ?
4021 016030 001003          BNE         9$          ;SKIP NEXT IF SO
4022 016032 042737 100000 004620  BIC         #SC,@#SAVERE+6 ;CLEAR 'SC' IF NOT
4023 016040          9$:
4024 016040 004037 041426    JSR         RO,@#FILLRE   ;MOV 0 INTO SAVED RHER1
4025 016044 002302          RHER1       ;SAVED REGISTER TO CHANGE
4026 016046 000000          0          ;DATA
4027 016050 004037 041426    JSR         RO,@#FILLRE   ;MOV 17437 INTO SAVED RHDST
4028 016054 002304          RHDST       ;SAVED REGISTER TO CHANGE
4029 016056 017437          17437       ;DATA
4030 016060 004037 041426    JSR         RO,@#FILLRE   ;MOV 0 INTO SAVED RHER2
4031 016064 002306          RHER2       ;SAVED REGISTER TO CHANGE
4032 016066 000000          0          ;DATA
4033 016070 004037 041426    JSR         RO,@#FILLRE   ;MOV 116000 INTO SAVED RHOF
4034 016074 002310          RHOF        ;SAVED REGISTER TO CHANGE
4035 016076 116000          116000      ;DATA
4036 016100 004037 041426    JSR         RO,@#FILLRE   ;MOV 1777 INTO SAVED RHCA
4037 016104 002312          RHCA        ;SAVED REGISTER TO CHANGE
4038 016106 001777          1777        ;DATA
4039 016110 004037 041426    JSR         RO,@#FILLRE   ;MOV 0 INTO SAVED RHER3
4040 016114 002314          RHER3       ;SAVED REGISTER TO CHANGE
4041 016116 000000          0          ;DATA
4042 016120 013746 004742    MOV         @#TOTALAT,-(SP) ;GET ALL BITS OF DRIVE & PRESENT
4043          ;IN RHAS
4044 016124 043716 004740    BIC         @#ATTENT,(SP) ;CLEAR WORKING DRIVE BIT
4045 016130 012637 004636    MOV         (SP)+,@#SAVERE+24 ;MOVE THIS INTO RHAS POSITION
4046 016134 004037 041426    JSR         RO,@#FILLRE   ;MOV 400 INTO SAVED RHMR
4047 016140 002320          RHMR        ;SAVED REGISTER TO CHANGE
4048 016142 000400          400         ;DATA
4049
4050 016144          3$:

```



```

4051 016144 004037 041426      JSR    R0,@#FILLRE      ;MOV 10700 INTO SAVED RHDS1
4052 016150 002322              RHDS1                    ;SAVED REGISTER TO CHANGE
4053 016152 010700              10700                   ;DATA
4054
4055 016154 013737 002406 004644 4$:  MOV    @#DT,@#SAVERE+32 ;MOVE DRIVE TYPE INTO RMDT
4056                                ;POSITION OF SAVRE TABLE
4057 016162 013737 002410 004646      MOV    @#SN,@#SAVERE+34 ;MOVE SERIAL NUMBER INTO RHSN
4058                                ;POSITION OF SAVERE TABLE
4059
4060 016170 004037 041426      JSR    R0,@#FILLRE      ;MOV 0 INTO SAVED RHEC1
4061 016174 002330              RHEC1                   ;SAVED REGISTER TO CHANGE
4062 016176 000000              0                        ;DATA
4063 016200 004037 041426      JSR    R0,@#FILLRE      ;MOV 0 INTO SAVED RHEC2
4064 016204 002332              RHEC2                   ;SAVED REGISTER TO CHANGE
4065 016206 000000              0                        ;DATA
4066
4067 016210 004037 042404      JSR    R0,@#CHREG       ;CHANGE BITS IN SAVED REGISTER
4068 016214 002300              RHCS1                   ;CHANGE RHCS1 REGISTER
4069
4070 016216 000001              1                        ;1 BIT/BITS TO BE CHANGED
4071 016220 000001              1                        ;NEW VALUE OF PAR IS 1
4072 016222 000010              PAR                     ;CHANGE PAR BIT
4073
4074                                ;*NOW THAT SAVERE TABLE HAS BEEN LOADED WITH
4075                                ;*EXPECTED VALUES, THE REGISTERS WILL BE COMPARED
4076                                ;*WITH SAVERE TABLE
4077
4078
4079 016224 004037 042512      JSR    R0,@#COMREG      ;COMPARE SAVED REGISTERS WITH
4080                                ;PRESENT VALUE
4081 016230 004612              SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
4082 016232 002354              WC                      ;TEST DATA STARTING FROM 'RHWC'
4083 016234 000022              18.                    ;18. REGISTERS TO BE COMPARED
4084 016236 016242              5$                      ;RETURN TO 5$ ON ERROR
4085 016240 016246              6$                      ;RETURN TO 6$ ON NO ERROR
4086
4087 016242 104017              5$:  ERROR 17            ;DRIVE CLEAR COMMAND
4088 016244 000207              RTS    PC               ;GAVE AN ERROR
4089                                ;GOOD DATA HAS WHAT SHOULD
4090                                ;BE IN REGISTER AFTER A
4091                                ;DRIVE CLEAR
4092                                ;RECEIVED DATA HAS WHAT
4093                                ;THE REGISTER ACTUALLY
4094                                ;CONTAINED
4095 016246              6$:
4096

```

4097
4098
4099
4100
4101
4102
4103
4104
4105
4106
4107
4108
4109
4110
4111
4112
4113
4114
4115
4116
4117
4118
4119
4120
4121
4122
4123
4124
4125
4126
4127
4128
4129
4130
4131
4132
4133
4134
4135
4136
4137
4138
4139
4140
4141
4142
4143
4144
4145
4146
4147
4148
4149
4150
4151
4152

```
*****  
*TEST 20 READ-IN-PRESET  
* IF STARTING ADDRESS 220 IS USED THIS TEST WILL NOT BE PERFORMED  
*  
* IF THE PROGRAM WORKS UNDER ACT-11 MONITOR  
* THEN THIS TEST IS NOT PERFORMED  
*  
* IF NO ACT-11 MONITOR IS PRESENT THEN  
* THIS TEST IS PERFORMED ONLY ON THE FIRST PASS  
* ON SUBSEQUENT PASSES THIS TEST IS NOT DONE  
*  
* THIS TESTS THE READ-IN-PRESET COMMAND  
* FIRST THE DRIVE IS POWERED DOWN AND UP IN ORDER TO  
* RESET VV-BIT #6 IN RHDS1  
* THEN ALL WRITE BITS OF ALL REGISTERS ARE FILLED  
* WITH ONES EXCEPT GO,CLR,IE,PAT,MCPE,UPE  
* ATA FOR DRIVE UNDER TEST IS MADE = 0  
*  
* THEN A READ-IN-PRESET COMMAND = 20 IS GIVEN  
* THEN ALL REGISTERS ARE TESTED  
* THE FOLLOWING SHOULD BE THE REGISTER CONTENTS  
* RHCA = 0, RHDST = 0, RHOF SHOULD HAVE  
* FMT22 = 0, ECI = 0, HCI = 0, RHDS1 SHOULD HAVE VV = 1  
* ALL OTHER REGISTERS SHOULD BE UNCHANGED  
*****
```

```
*****  
TST20: SCOPE  
MOV #STACK,SP ;RESET STACK  
MOV #20,@TSTNM ;SAVE TEST NUMBER  
  
JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2  
;R3-RHDS1, R4-RHER1  
;GIVE RH-11 INITIALIZE  
;SETUP UNIT NUMBER  
  
;*THIS CODE CHECKS TO SEE IF MANUAL INTERVENTION TESTS ARE OK  
  
TST @#NOPUSH ;IS THIS A 220 START ?  
BNE 1$ ;SKIP THIS TEST IF SO  
TST @#42 ;MONITOR (ACT 11) RETURN ADDRESS ?  
BNE 1$ ;SKIP THIS TEST  
TST @#SPASS ;FIRST PASS ?  
BNE 1$ ;SKIP THIS TEST IF NOT  
BR 2$ ;CONTINUE WITH THIS TEST  
  
1$:  
JMP TST21 ; JUMP TO NEXT TEST -----)  
2$:  
TYPE ,65$ ;;TYPE ASCIZ STRING  
BR 64$ ;;GET OVER THE ASCIZ  
;;65$: .ASCIZ <15><12>/STOP DRIVE /  
64$:
```



```

4209 016524 013700 004742      MOV    @#TOTALAT,R0      ;GET DRIVES PRESENT
4210 016530 005012              CLR    @R2                ;CLEAR RHCS2 AND CARRY
4211 016532 012705 000010      MOV    #8.,R5            ;COUNTER
4212 016536 006000              88$:  ROR    R0            ;GET BIT INTO CARRY
4213 016540 103002              BCC   89$                ;BRANCH IF NO UNIT ON THIS BIT
4214 016542 012714 177777      MOV    #-1,@R4           ;MOVE INTO ERROR REGISTER
4215                                ;TO SET ATA BIT
4216 016546 005212              89$:  INC    @R2           ;INCREMENT RHCS2 TO NEXT UNIT
4217 016550 005305              DEC    R5                 ;COUNT
4218 016552 001371              BNE   88$                ;BRANCH IF 8 NOT DONE
4219 016554 012612              MOV    (SP)+,@R2         ;REINSTATE RHCS2
4220 016556 012614              MOV    (SP)+,@R4         ;REINSTATE RHER1
4221 016560 012600              MOV    (SP)+,R0          ;POP STACK INTO R0
4222 016562 005720              TST   (R0)+              ;GET OVER PHAS IN R0
4223
4224
4225 016564 012730 177776      MOV    #177776,@(R0)+   ;LOAD 177776 INTO RHMR
4226
4227
4228 016570 013777 004740 163520  MOV    @#ATTENT,@RHAS    ;CLEAR WORKING DRIVE 'ATA'
4229 016576 013777 002462 163474  MOV    @#READIN,@RHCS1  ;GET READY FOR READIN
4230                                ;READ IN WITH 20 IN RHCS1
4231
4232
4233                                ;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ-IN COMMAND
4234
4235 016604 004037 041672      JSR    R0,@#SAVER        ;SAVE REGISTERS
4236 016610 002272              RHWC                                ;RHWC IS THE FIRST REGISTER SAVED
4237 016612 004612              SAVERE                            ;STARTING ADDRESS OF WHERE
4238                                ;THE REGISTERS ARE SAVED
4239 016614 000022              18.                                ;NUMBER OF REGISTERS
4240                                ;SAVED = 18.
4241
4242 016616 013777 004606 163442  MOV    @#RP4VEC,@RPVEC   ;SET RP04 VECTOR ADDRESS
4243                                ;TO 'TIME1' IF P-CLOCK IS PRESENT
4244                                ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
4245                                ;'TIME' WILL ONLY SAVE
4246                                ;CURRENT CYLINDER ADDRESS
4247                                ;AND LOOK AHEAD REGISTERS
4248
4249 016624 005737 004750      TST   @#RH70             ;RUNNING ON AN RH70 CONTROLLER ?
4250 016630 001411              BEQ   7$                 ;SKIP NEXT FOR RH70 IF NOT
4251
4252 016632 013746 002462      MOV    @#READIN,-(SP)    ;GET READY TO MOVE COMMAND
4253 016636 052716 000001      BIS   #GO,(SP)           ;GET READY TO SET GO
4254                                ;WITHOUT INTERRUPT ENABLE
4255 016642 012677 163432      MOV    (SP)+,@RHCS1     ;GO WITH
4256                                ;20 IN RHCS1 FOR READ IN
4257                                ;WITH INTERRUPT DISABLED
4258
4259 016646 011100              MOV    @R1,R0            ;SAVE RHCS1 DURING ABOVE OPERATION
4260 016650 011305              MOV    @R3,R5            ;SAVE RHDS1 DURING ABOVE OPERATION
4261 016652 000406              BR    8$                 ;SKIP NEXT FOR RH11
4262
4263 016654              7$:
4264

```


4337
4338
4339
4340
4341
4342
4343
4344
4345
4346
4347
4348
4349
4350
4351
4352
4353
4354
4355
4356
4357
4358
4359
4360
4361
4362
4363
4364
4365
4366
4367
4368
4369
4370
4371
4372
4373
4374
4375
4376
4377
4378
4379
4380
4381
4382
4383
4384
4385
4386
4387
4388
4389
4390
4391
4392

: *TEST 21 READ-IN-PRESET

: * THIS TEST IS THE SAME AS THE PREVIOUS TEST EXCEPT
: * THAT IT DOES NOT TEST THE SETTING OF VV
: *
: * THIS TEST IS HERE BECAUSE IF ACT-11 MONITOR IS PRESENT
: * THEN THE PREVIOUS TEST WILL NOT BE PERFORMED
: * THIS TESTS THE READ-IN-PRESET COMMAND
: * ALL WRITE BITS OF ALL REGISTERS ARE FILLED
: * WITH ONES EXCEPT GO,CLR,IE,PAT,MCPE,UPE
: * ATA FOR DRIVE UNDER TEST IS MADE = 0
: *
: * THEN A READ-IN-PRESET COMMAND = 20 IS GIVEN
: * THEN ALL REGISTERS ARE TESTED
: * THE FOLLOWING SHOULD BE THE REGISTER CONTENTS
: * RHCA = 0, RHDST = 0, RHOF SHOULD HAVE
: * FMT22 = 0, ECI = 0, HCI = 0, RHDS1 SHOULD HAVE VV = 1
: * ALL OTHER REGISTERS SHOULD BE UNCHANGED

TST21: SCOPE

MOV #STACK,SP ;RESET STACK
MOV #21,@#TSTNM ;SAVE TEST NUMBER
JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
;R3-RHDS1, R4-RHER1
;GIVE RH-11 INITIALIZE
;SETUP UNIT NUMBER
JSR PC,@#CHECK ;CHECK THAT DVA,RDY,MOL,DPR,DRY - 1
;AND THAT NO STATUS BITS ARE STUCK 1
TYPE ,CPHALT ;CANNOT CONTINUE TESTS IF THEY AREN'T
HALT ;STOP
MOV #RHWC,RO ;ADDR. OF ADDR. OF RHWC IN RO

: *INITIALIZE ALL THE REGISTERS

MOV #177777,@(RO)+ ;LOAD 177777 INTO RHWC
MOV #177777,@(RO)+ ;LOAD 177777 INTO RHBA
BIS #43010,@(RO)+ ;LOAD 43010 INTO RHCS2
MOV #1400,@(RO)+ ;LOAD 1400 INTO RHCS1
MOV #0,@(RO)+ ;LOAD 0 INTO RHER1
MOV #177777,@(RO)+ ;LOAD 177777 INTO RHDST
MOV #0,@(RO)+ ;LOAD 0 INTO RHER2
MOV #177777,@(RO)+ ;LOAD 177777 INTO RHOF
MOV #177777,@(RO)+ ;LOAD 177777 INTO RHCA

017012 000004
017014 012706 001000
017020 012737 000021 004604
017026 004737 041524
017032 004737 041562
017036 104401 067033
017042 000000
017044 012700 002272

017050 012730 177777
017054 012730 177777
017060 052730 043010
017064 012730 001400
017070 012730 000000
017074 012730 177777
017100 012730 000000
017104 012730 177777
017110 012730 177777

```

4393
4394 017114 012730 000000      MOV      #0,@(RO)+      ;LOAD 0 INTO RHER3
4395
4396 017120 012730 177777      MOV      #-1,@(RO)+    ;CLEAR ALL BITS OF RHAS
4397 017124 012730 177776      MOV      #177776,@(RO)+;LOAD 177776 INTO RHMR
4398
4399
4400 017130 013777 004740 163160  MOV      @#ATTENT,@RHAS ;CLEAR WORKING DRIVE 'ATA'
4401 017136 013777 002462 163134  MOV      @#READIN,@RHCS1;GET READY FOR READIN
4402                                     ;READ IN WITH 20 IN RHCS1
4403
4404
4405                                     ;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ-IN COMMAND
4406
4407 017144 004037 041672      JSR      RO,@#SAVER    ;SAVE REGISTERS
4408 017150 002272              RHWC              ;RHWC IS THE FIRST REGISTER SAVED
4409 017152 004612              SAVERE           ;STARTING ADDRESS OF WHERE
4410                                     ;THE REGISTERS ARE SAVED
4411 017154 000022              18.              ;NUMBER OF REGISTERS
4412                                     ;SAVED = 18.
4413
4414 017156 013777 004606 163102  MOV      @#RP4VEC,@RPVEC;SET RP04 VECTOR ADDRESS
4415                                     ;TO 'TIME1' IF P-CLOCK IS PRESENT
4416                                     ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
4417                                     ;'TIME' WILL ONLY SAVE
4418                                     ;CURRENT CYLINDER ADDRESS
4419                                     ;AND LOOK AHEAD REGISTERS
4420
4421 017164 005737 004750      TST      @#RH70        ;RUNNING ON AN RH70 CONTROLLER ?
4422 017170 001411              BEQ      9%           ;SKIP NEXT IF NOT
4423
4424 017172 013746 002462      MOV      @#READIN,-(SP);GET READY TO MOVE COMMAND
4425 017176 052716 000001      BIS      #GO,(SP)     ;GET READY TO SET GO
4426                                     ;WITHOUT INTERRUPT ENABLE
4427 017202 012677 163072      MOV      (SP)+,@RHCS1 ;GO WITH
4428                                     ;20 IN RHCS1 FOR READ IN
4429                                     ;WITH INTERRUPT DISABLED
4430
4431 017206 011100              MOV      @R1,R0        ;SAVE RHCS1 DURING ABOVE OPERATION
4432 017210 011305              MCV      @R3,R5        ;SAVE RHDS1 DURING ABOVE OPERATION
4433 017212 000406              BR       10%          ;SKIP NEXT RH11 CODE
4434
4435 017214                      9%:
4436
4437 017214 013746 002462      MOV      @#READIN,-(SP);GET READY TO MOVE COMMAND
4438 017220 052716 000101      BIS      #GO!IE,(SP)  ;GET READY TO SET 'GO' AND
4439                                     ;ENABLE INTERRUPT
4440 017224 012677 163050      MOV      (SP)+,@RHCS1 ;GO WITH
4441                                     ;20 IN RHCS1 FOR READ IN
4442                                     ;WITH INTERRUPT ENABLED
4443
4444 017230                      10%:
4445
4446 017230 104413              WAT              ;WAIT FOR RDY BIT TO SET
4447 017232 002300              RHCS1           ;WAIT FOR RHCS1 REGISTER
4448 017234 000200              RDY             ;WAIT FOR RDY BIT IN RHCS1 REGISTER
  
```


4505
4506
4507
4508 017344

68:

;RHDS1 SHOULD HAVE VV = 1
;ALL OTHER BITS SHOULD
;BE UNCHANGED

```

4509 ;:*****
4510 017344 005737 004744          TST   @#RP06      ;TEST FOR RP06 DRIVE
4511 017350 001401                   BEQ   7$          ;IF = 0, TREAT DRIVE AS AN RP04
4512 017352 000402                   BR    8$          ;TREAT AS RP06 - DO NEXT 'MAKECL'
4513 017354 000137 017420       7$:   JMP   @#DOG      ;DO SECOND FOLLOWING 'MAKECL'
4514 017360       8$:
4515 ;:*****
4516
4517
4518
4519
4520 ;:*****
4521 ;*TEST 22      MAKE CURRENT CYLINDER = 777
4522 ;:*****
4523 017360 000004          TST22: SCOPE
4524 017362 012706 001000          MOV   #STACK,SP  ;RESET STACK
4525 017366 012737 000022 004604          MOV   #23-1,@#TSTNM ;THIS SAVES TEST NUMBER
4526
4527 017374 004737 041524          JSR   PC,@#CLDISK ;INIT DRIVE
4528 017400 012777 000001 162712          MOV   #DMD,@RHMR  ;SET DIAGNOSTIC MODE
4529 017406 004037 041174          JSR   RO,@#MAKECYL ;SUBROUTINE TO GIVE A SEEK
4530                                     ;COMMAND FOLLOWED BY A INIT
4531                                     ;THIS SHOULD CHANGE RHCC TO 777
4532 017412 000777          777
4533
4534 017414 000137 017454          JMP   @#FISH      ;DON'T DO NEXT 'MAKECL'
4535
4536
4537 017420          DOG:
4538
4539 ;:*****
4540 ;*TEST 23      MAKE CURRENT CYLINDER = 377
4541 ;:*****
4542 017420 000004          TST23: SCOPE
4543 017422 012706 001000          MOV   #STACK,SP  ;RESET STACK
4544 017426 012737 000023 004604          MOV   #24-1,@#TSTNM ;THIS SAVES TEST NUMBER
4545
4546 017434 004737 041524          JSR   PC,@#CLDISK ;INIT DRIVE
4547 017440 012777 000001 162652          MOV   #DMD,@RHMR  ;SET DIAGNOSTIC MODE
4548 017446 004037 041174          JSR   RO,@#MAKECYL ;SUBROUTINE TO GIVE A SEEK
4549                                     ;COMMAND FOLLOWED BY A INIT
4550                                     ;THIS SHOULD CHANGE RHCC TO 377
4551 017452 000377          377
4552
4553
  
```

4554
4555
4556
4557 017454
4558
4559
4560
4561
4562
4563
4564
4565
4566
4567 017454 000004
4568 017456 012706 001000
4569 017462 012737 000024 004604
4570
4571 017470 004737 041524
4572
4573
4574
4575 017474 004737 041604
4576
4577 017500 04401 067033
4578
4579 017504 000000
4580 017506 012700 002272
4581 017512 012730 177777
4582
4583 017516 012730 177776
4584
4585 017522 052730 000010
4586 017526 012730 001400
4587
4588 017532 012730 000000
4589
4590 017536 012730 177777
4591
4592 017542 012730 000000
4593
4594 017546 012730 177777
4595
4596 017552 012730 177777
4597
4598 017556 012730 000000
4599
4600
4601
4602 017562 010046
4603 017564 011446
4604 017566 011246
4605
4606 017570 013700 004742
4607 017574 005012
4608 017576 012705 000010
4609 017602 006000

FISH:

: *TEST 24 RECALIBRATE COMMAND

: * ALL POSSIBLE REGISTERS ARE FILLED WITH ONES
: * THEN A RECALIBRATE = 6 COMMAND IS GIVEN

: * NO REGISTERS SHOULD CHANGE EXCEPT RHCC = 0

TST24: SCOPE
MOV #STACK,SP ;RESET STACK
MOV #24,@#TSTNM ;SAVE TEST NUMBER
JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
;R3-RHDS1, R4-RHER1
;GIVE RH-11 INITIALIZE
;SETUP UNIT NUMBER
JSR PC,@#CHECKT ;CHECK DVA, RDY, MOL, DPR, DRY, VV = 1
;AND THAT NO STATUS BITS ARE STUCK = 1
TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF ANY OF
;THE FIRST SET OF BITS DON'T = 1
HALT ;STOP
MOV #RHWC,RO ;ADDR. OF ADDR. OF RHWC IN RO
MOV #177777,@(RO)+ ;LOAD 177777 INTO RHWC
MOV #177776,@(RO)+ ;LOAD 177776 INTO RHBA
BIS #010,@(RO)+ ;LOAD 010 INTO RHCS2
MOV #1400,@(RO)+ ;LOAD 1400 INTO RHCS1
MOV #0,@(RO)+ ;LOAD 0 INTO RHER1
MOV #177777,@(RO)+ ;LOAD 177777 INTO RHDST
MOV #0,@(RO)+ ;LOAD 0 INTO RHER2
MOV #177777,@(RO)+ ;LOAD 177777 INTO RHOF
MOV #177777,@(RO)+ ;LOAD 177777 INTO RHCA
MOV #0,@(RO)+ ;LOAD 0 INTO RHER3
: *NOW SET BITS IN RHAS FOR ALL DRIVES PRESENT
MOV RO,-(SP) ;;PUSH RO ON STACK
MOV @R4,-(SP) ;SAVE RHER1 TO REINSTATE LATER
MOV @R2,-(SP) ;SAVE RHCS2 TO BE REINSTATED
;AFTER ALL ATA BITS HAVE BEEN SET
MOV @#TOTALAT,RO ;GET DRIVES PRESENT
CLR @R2 ;CLEAR RHCS2 AND CARRY
MOV #8.,R5 ;COUNTER
ROR RO ;GET BIT INTO CARRY

848:

```

4610 017604 103002          BCC      85$      ;BRANCH IF NO UNIT ON THIS BIT
4611 017606 012714 177777  MOV      #-1,@R4  ;MOVE INTO ERROR REGISTER
4612                                ;TO SET ATA BIT
4613 017612 005212          85$: INC      @R2    ;INCREMENT RHCS2 TO NEXT UNIT
4614 017614 005305          DEC      R5      ;COUNT
4615 017616 001371          BNE     84$      ;BRANCH IF 8 NOT DONE
4616 017620 012612          MOV     (SP)+,@R2 ;REINSTATE RHCS2
4617 017622 012614          MOV     (SP)+,@R4 ;REINSTATE RHER1
4618 017624 012600          MOV     (SP)+,R0  ;POP STACK INTO R0
4619 017626 005720          TST     (R0)+    ;GET OVER PHAS IN R0
4620
4621
4622 017630 012730 177776  MOV     #177776,@(R0)+ ;LOAD 177776 INTO RHMR
4623
4624
4625 017634 013777 002426 162436 MOV     @#RECALI,@RHCS1 ;GET READY FOR RECALI
4626                                ;RECALIBRATE WITH 6 IN RHCS1
4627
4628
4629                                ;*NOW SAVE REGISTERS FOR COMPARISON AFTER RECALIBRATE
4630
4631 017642 004037 041672  JSR     R0,@#SAVER ;SAVE REGISTERS
4632 017646 002272          RHWC     ;RHWC IS THE FIRST REGISTER SAVED
4633 017650 004612          SAVERE  ;STARTING ADDRESS OF WHERE
4634                                ;THE REGISTERS ARE SAVED
4635 017652 000022          18.    ;NUMBER OF REGISTERS
4636                                ;SAVED = 18.
4637 017654 013777 004606 162404 MOV     @#RP4VEC,@RPVEC ;SET RPO4 VECTOR ADDRESS
4638                                ;TO 'TIME1' IF P-CLOCK IS PRESENT
4639                                ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
4640                                ;'TIME' WILL ONLY SAVE
4641                                ;CURRENT CYLINDER ADDRESS
4642                                ;AND LOOK AHEAD REGISTERS
4643
4644 017662 013746 002426  MOV     @#RECALI,-(SP) ;GET READY TO MOVE COMMAND
4645 017666 052716 000101  BIS     #GO!IE,(SP)  ;GET READY TO SET 'GO' AND
4646                                ;ENABLE INTERRUPT
4647 017672 012677 162402  MOV     (SP)+,@RHCS1 ;GO WITH
4648                                ;6 IN RHCS1 FOR RECALIBRATE
4649                                ;WITH INTERRUPT ENABLED
4650 017676 011100          MOV     @R1,R0    ;SAVE RHCS1 DURING ABOVE OPERATION
4651 017700 011305          MOV     @R3,R5    ;SAVE RHDS1 DURING ABOVE OPERATION
4652
4653 017702 104413          WAT     ;WAIT FOR DRY BIT TO SET
4654 017704 002322          RHDS1  ;WAIT FOR RHDS1 REGISTER
4655 017706 000200          DRY    ;WAIT FOR DRY BIT IN RHDS1 REGISTER
4656 017710 076377          31999. ;ALLOW 319990 MICRO SECONDS
4657 017712 056701          24001. ;DRY MUST SET BETWEEN
4658                                ;79980 AND 560000 MICRO SECONDS
4659
4660                                ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
4661                                ;*R0 AND R5 IMMEDIATELY AFTER GO
4662
4663 017714 013746 002426  MOV     @#RECALI,-(SP) ;SAVE COMMAND
4664 017720 052716 004301  BIS     #DVA!GO!IE!RDY,(SP) ;INCLUDE DVA!GO!IE!RDY
4665 017724 005737 004722  TST     @#NUNIT    ;ARE THERE MORE THAN ONE UNIT

```

```

4666 017730 001413          BEQ      89$          ;BRANCH IF ONLY ONE UNIT
4667 017732 010037 004760    MOV      R0,@#TMP4    ;GET RHCS1
4668 017736 042737 177677    BIC      #^CIE,@#TMP4 ;KEEP IE BIT
4669 017744 042716 000100    BIC      #IE,(SP)     ;CLEAR IE IN GOOD DATA
4670 017750 053716 004760    BIS      @#TMP4,(SP)  ;GET IE AS IS
4671 017754 052716 100000    BIS      #SC,(SP)    ;SET SC IN RHCS1
4672 017760          89$:
4673 017760 011637 001124    MOV      (SP),@#SGDDAT ;SAVE FOR PRINTOUT
4674 017764 022600          CMP      (SP)+,R0    ;DURING ABOVE OPERATION ONLY DVA.GO!IE!RDY
4675          ;AND COMMAND SHOULD BE SET
4676 017766 001405          BEQ      88$          ;BRANCH IF GOOD
4677 017770 010037 001126    MOV      R0,@#SBDDAT  ;BAD DATA
4678 017774 010137 004600    MOV      R1,@#REGADR  ;FAILING REGISTER RHCS1
4679 020000 104021          ERROR   21          ;DURING ABOVE OPERATION ONLY
4680          ;COMMAND AND DVA!GO!IE!RDY SHOULD BE SET
4681 020002 012746 030500    MCV      #MOL!DPR!VV!PIP,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
4682 020006 011637 001124    MOV      (SP),@#SGDDAT ;SAVE FOR PRINTOUT
4683 020012 022605          CMP      (SP)+,R5    ;DURING ABOVE OPERATION ONLY MOL!DPR!VV!PIP
4684          ;SHOULD BE SET
4685 020014 001405          BEQ      90$          ;BRANCH IF GOOD
4686 020016 010537 001126    MOV      R5,@#SBDDAT  ;BAD DATA
4687 020022 010337 004600    MOV      R3,@#REGADR  ;FAILING REGISTER RHDS1
4688 020026 104063          ERROR   63          ;DURING ABOVE OPERATION ONLY
4689          ;MOL!DPR!VV!PIP SHOULD BE SET
4690 020030          90$:
4691
4692          ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
4693
4694 020030 004037 041426    JSR      R0,@#FILLRE  ;MOV 0 INTO SAVED RHCC
4695 020034 002334          RHCC              ;SAVED REGISTER TO CHANGE
4696 020036 000000          0                ;DATA
4697 020040 004037 041426    JSR      R0,@#FILLRE  ;MOV 116377 INTO SAVED RHOF
4698 020044 002310          RHOF              ;SAVED REGISTER TO CHANGE
4699 020046 116377          116377           ;DATA
4700
4701 020050 053737 004740 004636  BIS      @#ATTENT,@#SAVERE+24 ;SET APPROPRIATE 'ATA' BITS
4702          ;FOR WORKING DRIVE IN
4703          ;SAVED RHAS LOACTION
4704 020056 004037 041426    JSR      R0,@#FILLRE  ;MOV 104206 INTO SAVED RHCS1
4705 020062 002300          RHCS1            ;SAVED REGISTER TO CHANGE
4706 020064 104206          104206           ;DATA
4707 020066 004037 041426    JSR      R0,@#FILLRE  ;MOV 110700 INTO SAVED RHDS1
4708 020072 002322          RHDS1            ;SAVED REGISTER TO CHANGE
4709 020074 110700          110700           ;DATA
4710
4711
4712          ;*NOW COMPARE REGISTERS AFTER A RECALIBRATE COMMAND
4713
4714
4715 020076 004037 042512    JSR      R0,@#COMREG  ;COMPARE SAVED REGISTERS WITH
4716          ;PRESENT VALUE
4717 020102 004012          SAVERE            ;GOOD DATA SAVED IN 'SAVERE'
4718 020104 002354          WC              ;TEST DATA STARTING FROM 'RHWC'
4719 020106 000022          18.             ;18. REGISTERS TO BE COMPARED
4720 020110 020114          1$              ;RETURN TO 1$ ON ERROR
4721 020112 020120          2$              ;RETURN TO 2$ ON NO ERROR

```

4722						
4723	020114	104064	18:	ERROR	64	;RECALIBRATE COMMAND CAUSED
4724	020116	000207		RTS	PC	;AN ERROR
4725						;GOOD DATA GIVES WHAT SHOULD
4726						;BE THERE
4727						;RECEIVED DATA GIVES WHAT WAS
4728						;THERE AFTER COMMAND
4729						
4730	020120		28:			

```

4731
4732
4733 020120 005737 004744
4734 020124 001401
4735 020126 000402
4736 020150 000137 020174
4737 020134
4738
4739
4740
4741
4742
4743
4744
4745 020134 000004
4746 020136 012706 001000
4747 020142 012737 000025 004604
4748
4749 020150 004737 041524
4750 020154 012777 000001 162136
4751 020162 004037 041174
4752
4753
4754 020166 000777
4755
4756 020170 000137 020230
4757
4758
4759 020174
4760
4761
4762
4763
4764 020174 000004
4765 020176 012706 001000
4766 020202 012737 000026 004604
4767
4768 020210 004737 041524
4769 020214 012777 000001 162076
4770 020222 004037 041174
4771
4772
4773 020226 000377
4774

```

```

*****
;
;*****
TST @#RP06 ;TEST FOR RP06 DRIVE
BEQ 3$ ;IF = 0, TREAT DRIVE AS AN RP04
BR 4$ ;TREAT AS RP06 - DO NEXT 'MAKECL'
3$: JMP @#CAT ;DO SECOND FOLLOWING 'MAKECL'
4$:
*****
;
;*****
;*TEST 25 MAKE CURRENT CYLINDER = 777
;*****
TST25: SCOPE
MOV #STACK,SP ;RESET STACK
MOV #26-1,@#TSTNM ;THIS SAVES TEST NUMBER
JSR PC,@#CLDISK ;INIT DRIVE
MOV #DMD,@#RHMR ;SET DIAGNOSTIC MODE
JSR RO,@#MAKECYL ;SUBROUTINE TO GIVE A SEEK
;COMMAND FOLLOWED BY A INIT
;THIS SHOULD CHANGE RHCC TO 777
777
JMP @#BIKD ;DON'T DO NEXT 'MAKECL'
CAT:
*****
;*TEST 26 MAKE CURRENT CYLINDER = 377
;*****
TST26: SCOPE
MOV #STACK,SP ;RESET STACK
MOV #27-1,@#TSTNM ;THIS SAVES TEST NUMBER
JSR PC,@#CLDISK ;INIT DRIVE
MOV #DMD,@#RHMR ;SET DIAGNOSTIC MODE
JSR RO,@#MAKECYL ;SUBROUTINE TO GIVE A SEEK
;COMMAND FOLLOWED BY A INIT
;THIS SHOULD CHANGE RHCC TO 377
377

```


4775
4776
4777
4778 020230
4779
4780
4781
4782
4783
4784
4785
4786
4787
4788 020230 000004
4789 020232 012706 001000
4790 020236 012737 000027 004604
4791
4792 020244 004737 041524
4793
4794
4795
4796 020250 004737 041604
4797
4798 020254 104401 067033
4799
4800 020260 000000
4801 020262 012700 002272
4802
4803 020266 012730 000000
4804
4805 020272 012730 000000
4806
4807 020276 052730 000000
4808 020302 012730 000000
4809
4810 020306 012730 000000
4811
4812 020312 012730 000000
4813
4814 020316 012730 000000
4815
4816 020322 012730 000000
4817
4818 020326 012730 000000
4819
4820 020332 012730 000000
4821
4822 020336 012730 177777
4823 020342 012730 000000
4824
4825 020346 013777 002426 161724
4826
4827
4828
4829
4830

BIRD:

; *TEST 27 RECALIBRATE COMMAND

; * ALL POSSIBLE REGISTERS ARE FILLED WITH 0
; * THEN A RECALIBRATE =6 COMMAND IS GIVEN

; * NO REGISTERS SHOULD CHANGE EXCEPT RHCC=0

TST27: SCOPE

MOV #STACK,SP ;RESET STACK
MOV #27,@TSTNM ;SAVE TEST NUMBER
JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
;R3-RHDS1, R4-RHER1
;GIVE RH-11 INITIALIZE
;SETUP UNIT NUMBER
JSR PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
;AND THAT NO STATUS BITS ARE STUCK = 1
TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF ANY OF
;THE FIRST SET OF BITS DON'T = 1
HALT ;STOP
MOV #RHWC,R0 ;ADDR. OF ADDR OF RHWC IN R0
MOV #0,@(R0)+ ;LOAD 0 INTO RHWC
MOV #0,@(R0)+ ;LOAD 0 INTO RHBA
BIS #0,@(R0)+ ;LOAD 0 INTO RHCS2
MOV #0,@(R0)+ ;LOAD 0 INTO RHCS1
MOV #0,@(R0)+ ;LOAD 0 INTO RHER1
MOV #0,@(R0)+ ;LOAD 0 INTO RHD5T
MOV #0,@(R0)+ ;LOAD 0 INTO RHER2
MOV #0,@(R0)+ ;LOAD 0 INTO RHOF
MOV #0,@(R0)+ ;LOAD 0 INTO RHCA
MOV #0,@(R0)+ ;LOAD 0 INTO RHER3
MOV #-1,@(R0)+ ;CLEAR ALL BITS OF RHAS
MOV #0,@(R0)+ ;LOAD 0 INTO RHMR
MOV @#RECALI,@RHCS1 ;GET READY FOR RECALI
;RECALIBRATE WITH 6 IN RHCS1

; *NOW SAVE REGISTERS FOR COMPARISON AFTER RECALIBRATE

```

4831 020354 004037 041672 JSR RO,@#SAVER ;SAVE REGISTERS
4832 020360 002272 RHC ;RHC IS THE FIRST REGISTER SAVED
4833 020362 004612 SAVERE ;STARTING ADDRESS OF WHERE
4834 ;THE REGISTERS ARE SAVED
4835 020364 000022 18. ;NUMBER OF REGISTERS
4836 ;SAVED = 18.
4837 020366 013777 004606 161672 MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
4838 ;TO 'TIME1' IF P-CLOCK IS PRESENT
4839 ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
4840 ;'TIME' WILL ONLY SAVE
4841 ;CURRENT CYLINDER ADDRESS
4842 ;AND LOOK AHEAD REGISTERS
4843
4844 020374 013746 002426 MOV @#RECALI,-(SP) ;GET READY TO MOVE COMMAND
4845 020400 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET 'GO' AND
4846 ;ENABLE INTERRUPT
4847 020404 012677 161670 MOV (SP)+,@RHCS1 ;GO WITH
4848 ;6 IN RHCS1 FOR RECALIBRATE
4849 ;WITH INTERRUPT ENABLED
4850 020410 011100 MOV @R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
4851 020412 011305 MOV @R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
4852
4853 020414 104413 WAT ;WAIT FOR DRY BIT TO SET
4854 020416 002322 RHDS1 ;WAIT FOR RHDS1 REGISTER
4855 020420 000200 DRY ;WAIT FOR DRY BIT IN RHDS1 REGISTER
4856 020422 076377 31999. ;ALLOW 319990 MICRO SECONDS
4857 020424 056701 24001. ;DRY MUST SET BETWEEN
4858 ;79980 AND 560000 MICRO SECONDS
4859
4860 ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
4861 ;*R0 AND R5 IMMEDIATELY AFTER GO
4862
4863 020426 013746 002426 MOV @#RECALI,-(SP) ;SAVE COMMAND
4864 020432 052716 004301 BIS #DVA!GO!IE!RDY,(SP) ;INCLUDE DVA!GO!IE!RDY
4865 020436 011637 001124 MOV (SP),@#SGDDAT ;SAVE FOR PRINTOUT
4866 020442 022600 CMP (SP)+,R0 ;DURING ABOVE OPERATION ONLY DVA!GO!IE!RDY
4867 ;AND COMMAND SHOULD BE SET
4868 020444 001405 BEQ 88$ ;BRANCH IF GOOD
4869 020446 010037 001126 MOV R0,@#SBDDAT ;BAD DATA
4870 020452 010137 004600 MOV R1,@#REGADR ;FAILING REGISTER RHCS1
4871 020456 104021 ERROR 21 ;DURING ABOVE OPERATION ONLY
4872 ;COMMAND AND DVA!GO!IE!RDY SHOULD BE SET
4873 020460 012746 030500 88$: MOV #MOL!DPR!VV!PIP,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
4874 020464 011637 001124 MOV (SP),@#SGDDAT ;SAVE FOR PRINTOUT
4875 020470 022605 CMP (SP)+,R5 ;DURING ABOVE OPERATION ONLY MOL!DPR!VV!PIP
4876 ;SHOULD BE SET
4877 020472 001405 BEQ 90$ ;BRANCH IF GOOD
4878 020474 010537 001126 MOV R5,@#SBDDAT ;BAD DATA
4879 020500 010337 004600 MOV R3,@#REGADR ;FAILING REGISTER RHDS1
4880 020504 104063 ERROR 63 ;DURING ABOVE OPERATION ONLY
4881 ;MOL!DPR!VV!PIP SHOULD BE SET
4882 020506 90$:
4883 ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
4884
4885 020506 004037 041426 JSR R0,@#FILLRE ;MOV 0 INTO SAVED RHCC
4886 020512 002334 RHC ;SAVED REGISTER TO CHANGE
  
```

```

4887 020514 000000 0 ;DATA
4888
4889 020516 053737 004740 004636 BIS @#ATTENT,@#SAVERE+24 ;SET APPROPRIATE 'ATA' BITS
4890 ;FOR WORKING DRIVE IN
4891 ;SAVED RNAS LOACTION
4892
4893 020524 004037 042404 JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
4894 020530 002322 RHDS1 ;CHANGE RHDS1 REGISTER
4895
4896 020532 000001 1 ;1 BIT/BITS TO BE CHANGED
4897 020534 000001 1 ;NEW VALUE OF ATA IS 1
4898 020536 100000 ATA ;CHANGE ATA BIT
4899
4900 020540 004037 042404 JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
4901 020544 002300 RHCS1 ;CHANGE RHCS1 REGISTER
4902
4903 020546 000001 1 ;1 BIT/BITS TO BE CHANGED
4904 020550 000001 1 ;NEW VALUE OF SC IS 1
4905 020552 100000 SC ;CHANGE SC BIT
4906
4907 ;*NOW COMPARE REGISTERS AFTER A RECALIBRATE COMMAND
4908
4909
4910 020554 004037 042512 JSR RO,@#COMREG ;COMPARE SAVED REGISTERS WITH
4911 ;PRESENT VALUE
4912 020560 004612 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
4913 020562 002354 WC ;TEST DATA STARTING FROM 'RHWC'
4914 020564 000022 18. ;18. REGISTERS TO BE COMPARED
4915 020566 020572 1$ ;RETURN TO 1$ ON ERROR
4916 020570 020576 2$ ;RETURN TO 2$ ON NO ERROR
4917
4918 020572 104064 1$: ERROR 64 ;RECALIBRATE COMMAND CAUSED
4919 020574 000207 RTS PC ;AN ERROR
4920 ;GOOD DATA GIVES WHAT SHOULD BE
4921 ;THERE
4922 ;RECEIVED DATA GIVES WHAT WAS
4923 020576 2$: ;THERE AFTER A RECALIBRATE
4924

```

```
4925 .....  
4926 :*TEST 30 UNLOAD COMMAND  
4927  
4928 :* IF STARTING ADDRESS 220 IS USED THIS TEST WILL NOT BE PERFORMED  
4929 :*  
4930 :* IF THE PROGRAM WORKS UNDER ACT-11 MONITOR  
4931 :* THEN THIS TEST IS NOT PERFORMED  
4932 :*  
4933 :* IF NO ACT-11 MONITOR IS PRESENT  
4934 :* THEN THIS TEST IS PERFORMED ONLY ON THE FIRST PASS  
4935 :* ON SUBSEQUENT PASSES THIS TEST IS NOT DONE  
4936 :*  
4937 :*  
4938 :* ALL POSSIBLE REGISTERS ARE FILLED WITH ONES  
4939 :* THEN AN UNLOAD COMMAND =2 IS GIVEN  
4940 :* NO REGISTERS SHOULD CHANGE EXCEPT MOL SHOULD=0  
4941 :* THEN THE DRIVE IS POWERED UP BY OPERATOR  
4942 :* AND A PACK ACKNOWLEDGE COMMAND (ALREADY TESTED)  
4943 :* SETS VV-IN RHDS1  
4944 :*  
4945 .....  
4946 020576 000004 TST30: SCOPE  
4947  
4948  
4949 :*THIS CODE CHECKS TO SEE IF MANUAL INTERVENTION TESTS ARE OK  
4950  
4951 020600 005737 004724 TST @#NOPUSH ;IS THIS A 220 START ?  
4952 020604 001007 BNE 1$ ;SKIP THIS TEST IF SO  
4953 020606 005737 000042 TST @#42 ;MONITOR (ACT 11) RETURN ADDRESS ?  
4954 020612 001004 BNE 1$ ;SKIP THIS TEST  
4955 020614 005737 001100 TST @#$PASS ;FIRST PASS ?  
4956 020620 001001 BNE 1$ ;SKIP THIS TEST IF NOT  
4957 020622 000402 BR 2$ ;CONTINUE WITH THIS TEST  
4958  
4959 020624 1$:  
4960 020624 000137 022112 JMP TST31 ; JUMP TO NEXT TEST -----)  
4961 020630 2$:  
4962 020630 012706 001000 MOV #STACK,SP ;RESET STACK  
4963 020634 012737 000030 004604 MOV #30,@#TSTNM ;SAVE TEST NUMBER  
4964  
4965 020642 004737 041524 JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2  
4966 ;R3-RHDS1, R4-RHER1  
4967 ;GIVE RH-11 INITIALIZE  
4968 ;SETUP UNIT NUMBER  
4969 020646 004737 041604 JSR PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1  
4970 ;AND THAT NO STATUS BITS ARE STUCK - 1  
4971 020652 104401 067033 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF ANY OF  
4972 ;THE FIRST SET OF BITS DON'T = 1  
4973 020656 000000 HALT ;STOP  
4974  
4975 :*THIS SETTING OF VV IS FOR LOOP ON ERROR ONLY  
4976 :*WHERE UNLOAD TAKES EFFECT AND CYCLE UP BRINGS VV DOWN  
4977  
4978 020660 017746 161436 MOV @RHDS1,-(SP) ;PUSH RHDS1 ONTO STACK  
4979 020664 042716 167677 BIC #167677,(SP) ;CLEAR EVERYTHING EXCEPT VV AND MOL  
4980 020670 022726 010100 CMP #VV!MOL,(SP)+ ;ARE VV AND MOL SET ?
```

```

4981 020674 001504 BEQ 68 ;CONTINUE IF YES
4982 020676 104401 020704 TYPE ,658 ;:TYPE ASCIZ STRING
4983 020702 000427 BR 648 ;:GET OVER THE ASCIZ
4984 ;:658: .ASCIZ <15><12>/GET DRIVE HEADS LOADED THEN HIT "CONTINUE"/
4985 648:
4986 020762 104401 020770 TYPE ,678 ;:TYPE ASCIZ STRING
4987 020766 000424 BR 668 ;:GET OVER THE ASCIZ
4988 ;:678: .ASCIZ <15><12>/IF ALREADY LOADED THEN HIT "CONTINUE"/
4989 668:
4990 021040 000000 HALT ;WAIT FOR CONTINUE
4991
4992
4993 021042 004737 041524 JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
4994 ;R3-RHDS1, R4-RHER1
4995 ;GIVE RH-11 INITIALIZE
4996 ;SETUP UNIT NUMBER
4997 021046 004737 041562 JSR PC,@#CHECK ;CHECK THAT DVA,RDY,MOL,DPR,DRY = 1
4998 ;AND THAT NO STATUS BITS ARE STUCK = 1
4999 021052 104401 067033 TYPE ,CPHALT ;CANNOT CONTINUE TESTS IF THEY AREN'T
5000 021056 000000 HALT ;STOP
5001
5002 ;*SET VV IN RHDS1 WITH PACK ACKNOWLEDGE
5003
5004
5005 021060 013746 002460 MOV @#PKACK,-(SP) ;GET READY TO MOVE COMMAND
5006 021064 052716 000001 BIS #GO,(SP) ;GET READY TO SET GO
5007 ;WITHOUT INTERRUPT ENABLE
5008 021070 012677 161204 MOV (SP)+,@RHCS1 ;GO WITH
5009 ;22 IN RHCS1 FOR PACK ACKNOWLEDGE
5010 ;WITH INTERRUPT DISABLED
5011
5012
5013
5014 021074 104413 WAT ;WAIT FOR VV BIT TO SET
5015 021076 002322 RHDS1 ;WAIT FOR RHDS1 REGISTER
5016 021100 000100 VV ;WAIT FOR VV BIT IN RHDS1 REGISTER
5017 021102 000001 1. ;ALLOW 10 MICRO SECONDS
5018 021104 000001 1. ;VV MUST SET BETWEEN
5019 ;00 AND 20 MICRO SECONDS
5020
5021 021106 68:
5022 021106 004737 041604 JSR PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR & VV = 1
5023 021112 000240 NOP ;CHECK THAT ALL OTHER BITS = 0
5024 021114 000240 NOP ;UNLIKE THE OTHER STATUS BIT TESTS,
5025 021116 000240 NOP ;THERE IS NO HALT IF IT FAILS - IT IS
5026 ;USED IN THE MIDDLE OF A SINGLE TEST
5027
5028 021120 012700 002272 MOV #RHWC,R0 ;ADDR. OF ADDR OF RHWC IN R0
5029
5030 ;*LOAD ALL POSSIBLE REGISTERS WITH ONES
5031
5032
5033 021124 012730 177777 MOV #177777,@(R0)+ ;LOAD 177777 INTO RHWC
5034
5035
5036

```

```

5037 021130 012730 177777      MOV      #177777,@(R0)+ ;LOAD 177777 INTO RHBA
5038
5039
5040
5041 021134 052730 000010      BIS      #10,@(R0)+ ;LOAD 10 INTO RHCS2
5042
5043
5044 021140 012730 001400      MOV      #1400,@(R0)+ ;LOAD 1400 INTO RHCS1
5045
5046
5047
5048 021144 012730 000000      MOV      #0,@(R0)+ ;LOAD 0 INTO RHER1
5049
5050
5051
5052 021150 012730 177777      MOV      #177777,@(R0)+ ;LOAD 177777 INTO RHDST
5053
5054
5055
5056 021154 012730 000000      MOV      #0,@(R0)+ ;LOAD 0 INTO RHER2
5057
5058
5059
5060 021160 012730 177777      MOV      #177777,@(R0)+ ;LOAD 177777 INTO RHOF
5061
5062
5063
5064 021164 012730 177777      MOV      #177777,@(R0)+ ;LOAD 177777 INTO RHCA
5065
5066
5067
5068 02117C 012730 000000      MOV      #0,@(R0)+ ;LOAD 0 INTO RHER3
5069
5070
5071
5072
5073      ;*NOW SET BITS IN RHAS FOR ALL DRIVES PRESENT
5074 021174 010046      MOV      R0,-(SP) ;:PUSH R0 ON STACK
5075 021176 011446      MOV      @R4,-(SP) ;:SAVE RHER1 TO REINSTATE LATER
5076 021200 011246      MOV      @R2,-(SP) ;:SAVE RHCS2 TO BE REINSTATED
5077      ;:AFTER ALL ATA BITS HAVE BEEN SET
5078 021202 013700 004742      MOV      @#TOTALAT,R0 ;:GET DRIVES PRESENT
5079 021206 005012      CLR      @R2 ;:CLEAR RHCS2 AND CARRY
5080 021210 012705 000010      MOV      #8.,R5 ;:COUNTER
5081 021214 006000      91$: ROR      R0 ;:GET BIT INTO CARRY
5082 021216 103002      BCC      92$ ;:BRANCH IF NO UNIT ON THIS BIT
5083 021220 012714 177777      MOV      #-1,@R4 ;:MOVE INTO ERROR REGISTER
5084      ;:TO SET ATA BIT
5085 021224 005212      92$: INC      @R2 ;:INCREMENT RHCS2 TO NEXT UNIT
5086 021226 005305      DEC      R5 ;:COUNT
5087 021230 001371      BNE      91$ ;:BRANCH IF 8 NOT DONE
5088 021232 012612      MOV      (SP)+,@R2 ;:REINSTATE RHCS2
5089 021234 012614      MOV      (SP)+,@R4 ;:REINSTATE RHER1
5090 021236 012600      MOV      (SP)+,R0 ;:POP STACK INTO R0
5091 021240 005720      TST      (R0)+ ;:GET OVER PHAS IN R0
5092

```



```

5136
5137
5138
5139
5140 021326 013746 002424      MOV    @#UNLOAD,-(SP)  ;PUSH COMMAND ON STACK
5141 021332 052716 004201      BIS    #DVA!GO!RDY,(SP) ;INCLUDE THESE BITS SET
5142 021336 005737 004722      TST    @#NUNIT        ;IS THERE MORE THAN ONE UNIT ?
5143 021342 001413              BEQ    9$              ;SKIP NEXT IF ONLY ONE UNIT
5144 021344 010037 004760      MOV    R0,@#TMP4      ;PUT SAVED RHCSI INTO TMP4
5145 021350 042737 177677 004760 BIC    #^CIE,@#TMP4    ;MASK ALL BUT THE 'IE' BIT IN RHCSI
5146 021356 042716 000100      BIC    #IE,(SP)        ;CLEAR 'IE' IN EXPECTED DATA
5147 021362 053716 004760      BIS    @#TMP4,(SP)     ;SET 'IE' STATE FROM ACTUAL RHCSI DATA
5148 021366 052716 100000      BIS    #SC,(SP)        ;SET 'SC' IN RHCSI SAVED DATA
5149
5150 021372 011637 001124      9$:   MOV    (SP),@#SGDDAT  ;SAVE EXPECTED DATA FOR PRINTOUT
5151 021376 022600              CMP    (SP)+,R0        ;COMPARE EXPECTED DATA WITH SAVED
5152                                ;RHCSI DATA AND RESET THE STACK
5153 021400 001405              BEQ    10$             ;CHECK NEXT BITS IF THESE OK
5154 021402 010037 001126      MOV    R0,@#SBDDAT    ;RHCSI IS BAD - PRINT IT OUT
5155 021406 010137 004600      MOV    R1,@#REGADR    ;REGISTER ADDRESS
5156 021412 104021              ERROR  21             ;DURING ABOVE OPERATION ONLY THE
5157                                ;'DVA', 'GO', 'RDY' AND COMMAND BITS
5158                                ;SHOULD BE SET
5159
5160 021414 012746 020400      10$:  MOV    #PIP!DPR,-(SP)  ;PUT SOME EXPECTED RHDS1 BITS ON STACK
5161 021420 010537 004760      MOV    R5,@#TMP4      ;PUT SAVED RHDS1 INTO TMP4
5162 021424 042737 167677 004760 BIC    #^C<MOL!VV>,@#TMP4 ;MASK ALL BUT 'MOL' & 'VV' IN RHDS1
5163 021432 042716 010100      BIC    #MOL!VV,(SP)   ;CLEAR 'MOL' & 'VV' IN EXPECTED RHDS1
5164 021436 053716 004760      BIS    @#TMP4,(SP)    ;SET EXPECTED 'MOL' & 'VV' BIT STATES
5165                                ;FROM THE ACTUAL DATA (DON'T CARE)
5166 021442 011637 001124      MOV    (SP),@#SGDDAT  ;SAVE EXPECTED DATA FOR PRINTOUT
5167 021446 022605              CMP    (SP)+,R5        ;COMPARE EXPECTED DATA WITH SAVED
5168                                ;RHDS1 DATA AND RESET THE STACK
5169 021450 001405              BEQ    11$             ;CONTINUE IF EXPECTED=SAVED
5170 021452 010537 001126      MOV    R5,@#SBDDAT    ;RHDS1 IS BAD - PRINT IT OUT
5171 021456 010337 004600      MOV    R3,@#REGADR    ;REGISTER ADDRESS
5172 021462 104063              ERROR  63             ;DURING THE ABOVE OPERATION, ONLY 'PIP
5173                                ;AND 'DPR' SHOULD BE SET
5174                                ;'MOL' & 'VV' ARE DON'T CARES
5175 021464              11$:
5176
5177 021464 104401 021472      TYPE    ,96$          ;;TYPE ASCIZ STRING
5178 021470 000425              BR      95$           ;;GET OVER THE ASCIZ
5179                                ;;96$: .ASCIZ <15><12>/IF STANDBY NOT LIT - ERROR AFTER UNLOAD/
5180 021544      95$:
5181                                ;THIS PROVIDES A 1 SECOND "STALL"
5182

```



```

5183
5184
5185 ;*NOW CHANGE REGISTERS SAVED BEFORE UNLOAD COMMAND
5186 ;*TO EXPECTED VALUES AFTER UNLOAD COMMAND
5187
5188 ;* - AGAIN 'MOL' & 'VV' ARE DON'T CARES
5189
5190
5191 021544 012746 020400 MOV #PIP!DPR,-(SP) ;SET EXPECTED FINAL RHDS1 BITS
5192 021550 017737 160546 004760 MOV @RHDS1,@#TMP4 ;GET PRESENT ACTUAL RHDS1 CONTENTS
5193 021556 042737 167677 004760 BIC #^C<MOL!VV>,@#TMP4 ;MASK OUT ALL BUT 'MOL' & 'VV'
5194 021564 042716 010100 BIC #MOL!VV,(SP) ;CLEAR 'MOL' & 'VV' IN EXPECTED RHDS1
5195 021570 053716 004760 BIS @#TMP4,(SP) ;SET EXPECTED 'MOL' & 'VV' STATES
5196 ;FROM THE ACTUAL (DON'T CARE COND.)
5197 021574 042716 100200 BIC #ATA!DRY,(SP) ;CLEAR THESE ADDITIONAL RHDS1 BITS
5198 021600 012637 004642 MOV (SP)+,@#SAVERE+30 ;CHANGE THE SAVED RHDS1 REGISTER
5199 ;AND ADJUST THE STACK
5200
5201
5202 021604 004037 042404 JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
5203 021610 002300 RHCS1 ;CHANGE RHCS1 REGISTER
5204
5205 021612 000001 1 ;1 BIT/BITS TO BE CHANGED
5206 021614 000001 1 ;NEW VALUE OF GO IS 1
5207 021616 000001 GO ;CHANGE GO BIT
5208
5209 021620 005737 004722 TST @#NUNIT ;IS THERE MORE THAN ONE UNIT ?
5210 021624 001006 BNE 7$ ;SKIP NEXT IF MORE THAN ONE UNIT
5211
5212 021626 004037 042404 JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
5213 021632 002300 RHCS1 ;CHANGE RHCS1 REGISTER
5214
5215 021634 000001 1 ;1 BIT/BITS TO BE CHANGED
5216 021636 000000 0 ;NEW VALUE OF SC IS 0
5217 021640 100000 SC ;CHANGE SC BIT
5218 021642 7$:
5219
5220 021642 043737 004740 004636 BIC @#ATTENT,@#SAVERE+24 ;CLEAR APPROPRIATE ATA BITS
5221 ;FOR WORKING DRIVE IN SAVED RHAS
5222
5223 ;*NOW COMPARE REGISTERS AFTER THE UNLOAD COMMAND
5224 ;*WITH EXPECTED VALUES
5225
5226
5227 021650 004037 042512 JSR RO,@#COMREG ;COMPARE SAVED REGISTERS WITH
5228 ;PRESENT VALUE
5229 021654 004612 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
5230 021656 002354 WC ;TEST DATA STARTING FROM 'RHWC'
5231 021660 000021 17. ;17. REGISTERS TO BE COMPARED
5232 021662 021666 3$ ;RETURN TO 3$ ON ERROR
5233 021664 021672 4$ ;RETURN TO 4$ ON NO ERROR
5234
5235 021666 104023 3$: ERROR 23 ;UNLOAD COMMAND GAVE
5236 021670 000207 RTS PC ;AN ERROR
5237 ;GOOD DATA GIVES WHAT SHOULD
5238 ;BE THERE

```


5283 ;*****
5284 ;*TEST 31 OFFSET AND RETURN TO CENTER LINE COMMAND
5285
5286 ;* THIS TESTS TWO COMMANDS: (1)OFFSET, (2)RETURN-TO-CENTER-LINE
5287
5288 ;* ALL POSSIBLE REGISTERS ARE FILLED WITH ONES (EXCEPT RHOF)
5289 ;* AND AN OFFSET IS GIVEN
5290 ;* ALL REGISTERS ARE COMPARED, ONLY ATA SHOULD SET
5291 ;* ALL OTHER REGISTERS SHOULD REMAIN UNCHANGED
5292
5293 ;* THEN A RETURN-TO-CENTER-LINE IS GIVEN
5294 ;* ALL REGISTERS ARE COMPARED ONLY ATA SHOULD SET
5295 ;* AND RHOF SHOULD CLEAR (EXCEPT HCI,ECI,FMT22)
5296 ;* ALL OTHER REGISTERS SHOULD REMAIN UNCHANGED
5297
5298 ;* THE ABOVE PROCESS IS REPEATED FOR OFFSET REGISTER
5299 ;* VALUES OF 1 TO 377 IE. 377 TIMES

5300
5301 ;*****
5302 022112 000004 TST31: SCOPE
5303 022114 012737 022166 001106 MOV #1\$, \$LPADR ;;SET SCOPE LOOP ADDRESS
5304 022122 012706 001000 MOV #STACK, SP ;RESET STACK
5305 022126 012737 000031 004604 MOV #31, @#TSTNM ;SAVE TEST NUMBER
5306
5307 022134 004737 041524 JSR PC, @#CLDISK ;SET R1-RHCS1, R2-RHCS2
5308 ;R3-RHDS1, R4-RHER1
5309 ;GIVE RH-11 INITIALIZE
5310 ;SETUP UNIT NUMBER
5311 022140 004737 041562 JSR PC, @#CHECK ;CHECK THAT DVA, RDY, MOL, DPR, DRY = 1
5312 ;AND THAT NO STATUS BITS ARE STUCK 1
5313 022144 104401 067033 TYPE ,CPHALT ;CANNOT CONTINUE TESTS IF THEY AREN'T
5314 022150 000000 HALT ;STOP
5315 022152 112737 000001 004610 MOVB #1, @#OFSTVL ;SET OFFSET VALUE TO 1
5316 022160 112737 000034 004611 MOVB #34, @#OFSTVL+1 ;SET HCI, ECI, FMT22
5317
5318
5319 022166 1\$:
5320
5321 022166 004737 041524 JSR PC, @#CLDISK ;SET R1-RHCS1, R2-RHCS2
5322 ;R3-RHDS1, R4-RHER1
5323 ;GIVE RH-11 INITIALIZE
5324 ;SETUP UNIT NUMBER
5325 022172 004737 041604 JSR PC, @#CHECKT ;CHECK DVA, RDY, MOL, DPR & VV - 1
5326 022176 000240 NOP ;CHECK THAT ALL OTHER BITS = 0
5327 022200 000240 NOP ;UNLIKE THE OTHER STATUS BIT TESTS,
5328 022202 000240 NOP ;THERE IS NO HALT IF IT FAILS - IT IS
5329 ;USED IN THE MIDDLE OF A SINGLE TEST
5330 022204 012700 002272 MOV #RHWC, R0 ;ADDR. OF ADDR OF RHWC IN R0
5331
5332 022210 012730 177777 MOV #177777, @ (R0)+ ;LOAD 177777 INTO RHWC
5333
5334
5335 022214 012730 177777 MOV #177777, @ (R0)+ ;LOAD 177777 INTO RHBA
5336
5337
5338 022220 052730 000010 BIS #10, @ (R0)+ ;LOAD 10 INTO RHCS2

```

5339
5340 022224 012730 001400      MOV      #1400,@(RO)+      ;LOAD 1400 INTO RHCS1
5341
5342
5343 022230 012730 000000      MOV      #0,@(RO)+        ;LOAD 0 INTO RHER1
5344
5345
5346 022234 012730 177777      MOV      #177777,@(RO)+   ;LOAD 177777 INTO RHDST
5347
5348
5349 022240 012730 000000      MOV      #0,@(RO)+        ;LOAD 0 INTO RHER2
5350
5351      ;*THE OFFSET REGISTER WILL BE INCREMENTED FROM 0 TO 377
5352
5353 022244 013730 004610      MOV      @#OFSTVL,@(RO)+  ;SET OFFSET REGISTER
5354
5355 022250 012730 177777      MOV      #177777,@(RO)+   ;LOAD 177777 INTO RHCA
5356
5357 022254 012730 000000      MOV      #0,@(RO)+        ;LOAD 0 INTO RHER3
5358
5359      ;*NOW SET BITS IN RHAS FOR ALL DRIVES PRESENT
5360
5361 022260 010046      MOV      RO,-(SP)          ;;PUSH RO ON STACK
5362 022262 011446      MOV      @R4,-(SP)         ;SAVE RHER1 TO REINSTATE LATER
5363 022264 011246      MOV      @R2,-(SP)         ;SAVE RHCS2 TO BE REINSTATED
5364      ;AFTER ALL ATA BITS HAVE BEEN SET
5365 022266 013700 004742      MOV      @#TOTALAT,R0     ;GET DRIVES PRESENT
5366 022272 005012      CLR      @R2              ;CLEAR RHCS2 AND CARRY
5367 022274 012705 00C010      MOV      #8.,R5           ;COUNTER
5368 022300 006000      82$:  ROR      R0           ;GET BIT INTO CARRY
5369 022302 103002      BCC      83$              ;BRANCH IF NO UNIT ON THIS BIT
5370 022304 012714 177777      MOV      #-1,@R4          ;MOVE INTO ERROR REGISTER
5371      ;TO SET ATA BIT
5372 022310 005212      83$:  INC      @R2          ;INCREMENT RHCS2 TO NEXT UNIT
5373 022312 005305      DEC      R5               ;COUNT
5374 022314 001371      BNE      82$              ;BRANCH IF 8 NOT DONE
5375 022316 012612      MOV      (SP)+,@R2        ;REINSTATE RHCS2
5376 022320 012614      MOV      (SP)+,@R4        ;REINSTATE RHER1
5377 022322 012600      MOV      (SP)+,R0         ;;POP STACK INTO RO
5378 022324 005720      TST      (RO)+           ;GET OVER PHAS IN RO
5379
5380
5381 022326 012730 177776      MOV      #177776,@(RO)+   ;LOAD 177776 INTO RHMR
5382
5383 022332 013777 002454 157740  MOV      @#OFSETC,@RHCS1  ;GET READY FOR OFSETC
5384
5385
5386      ;*NOW SAVE REGISTERS FOR COMPARISON AFTER OFFSET
5387
5388 022340 004037 041672      JSR      RO,@#SAVER        ;SAVE REGISTERS
5389 022344 002272      RHC      ;RHC IS THE FIRST REGISTER SAVED
5390 022346 004612      SAVERE   ;STARTING ADDRESS OF WHERE
5391      ;THE REGISTERS ARE SAVED
5392 022350 000022      18.      ;NUMBER OF REGISTERS
5393      ;SAVED = 18.
5394

```

```

5395 022352 013777 004606 157706      MOV      @#RP4VEC,@RPVEC      ;SET RP04 VECTOR ADDRESS
5396                                     ;TO 'TIME1' IF P-CLOCK IS PRESENT
5397                                     ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
5398                                     ;'TIME' WILL ONLY SAVE
5399                                     ;CURRENT CYLINDER ADDRESS
5400                                     ;AND LOOK AHEAD REGISTERS
5401
5402
5403 022360 013746 002454      MOV      @#OFSETC,-(SP)      ;GET READY TO MOVE COMMAND
5404 022364 052716 000101      BIS      #GO!IE,(SP)        ;GET READY TO SET 'GO' AND
5405                                     ;ENABLE INTERRUPT
5406 022370 012677 157704      MOV      (SP)+,@RHCS1       ;GO WITH
5407                                     ;14 IN RHCS1 FOR OFFSET
5408                                     ;WITH INTERRUPT ENABLED
5409 022374 011100      MOV      @R1,R0              ;SAVE RHCS1 DURING ABOVE OPERATION
5410 022376 011305      MOV      @R3,R5              ;SAVE RHDS1 DURING ABOVE OPERATION
5411
5412
5413 022400 104413      WAT                                     ;WAIT FOR DRY BIT TO SET
5414 022402 002322      RHDS1                             ;WAIT FOR RHDS1 REGISTER
5415 022404 000200      DRY                                 ;WAIT FOR DRY BIT IN RHDS1 REGISTER
5416 022406 001750      1000.                              ;ALLOW 10000 MICRO SECONDS
5417 022410 000454      300.                                ;DRY MUST SET BETWEEN
5418                                     ;7000 AND 13000 MICRO SECONDS
5419
5420                                     ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
5421                                     ;*R0 AND R5 IMMEDIATELY AFTER GO
5422
5423 022412 013746 002454      MOV      @#OFSETC,-(SP)      ;SAVE COMMAND
5424 022416 052716 004301      BIS      #DVA!GO!IE!RDY,(SP) ;INCLUDE DVA!GO!IE!RDY
5425 022422 005737 004722      TST      @#NUNIT             ;ARE THERE MORE THAN ONE UNIT
5426 022426 001413      BEQ      87$                 ;BRANCH IF ONLY ONE UNIT
5427 022430 010037 004760      MOV      R0,@#TMP4           ;GET RHCS1
5428 022434 042737 177677 004760      BIC      #^CIE,@#TMP4        ;KEEP IE BIT
5429 022442 042716 000100      BIC      #IE,(SP)            ;CLEAR IE IN GOOD DATA
5430 022446 053716 004760      BIS      @#TMP4,(SP)         ;GET IE AS IS
5431 022452 052716 100000      BIS      #SC,(SP)            ;SET SC IN RHCS1
5432 022456                                     87$:
5433 022456 011637 001124      MOV      (SP),@#SGDDAT       ;SAVE FOR PRINTOUT
5434 022462 022600      CMP      (SP)+,R0            ;DURING ABOVE OPERATION ONLY DVA!GO!IE!RDY
5435                                     ;AND COMMAND SHOULD BE SET
5436 022464 001405      BEQ      86$                 ;BRANCH IF GOOD
5437 022466 010037 001126      MOV      R0,@#BDDAT          ;BAD DATA
5438 022472 010137 004600      MOV      R1,@#REGADR         ;FAILING REGISTER RHCS1
5439 022476 104021      ERROR   21                   ;DURING ABOVE OPERATION ONLY
5440                                     ;COMMAND AND DVA!GO!IE!RDY SHOULD BE SET
5441 022500 012746 030500      86$: MOV      #PIP!MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
5442 022504 011637 001124      MOV      (SP),@#SGDDAT       ;SAVE FOR PRINTOUT
5443 022510 022605      CMP      (SP)+,R5            ;DURING ABOVE OPERATION ONLY PIP.MOL!DPR.VV
5444                                     ;SHOULD BE SET
5445 022512 001405      BEQ      88$                 ;BRANCH IF GOOD
5446 022514 010537 001126      MOV      R5,@#BDDAT          ;BAD DATA
5447 022520 010337 004600      MOV      R3,@#REGADR         ;FAILING REGISTER RHDS1
5448 022524 104063      ERROR   63                   ;DURING ABOVE OPERATION ONLY
5449                                     ;PIP!MOL.DPR!VV SHOULD BE SET
5450 022526                                     88$:

```

```

5451
5452                ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
5453
5454
5455 022526 004037 042404 JSR    RO,@#CHREG    ;CHANGE BITS IN SAVED REGISTER
5456 022532 002300 RHCS1  ;CHANGE RHCS1 REGISTER
5457
5458 022534 000001      1      ;1 BIT/BITS TO BE CHANGED
5459 022536 000001      1      ;NEW VALUE OF SC IS 1
5460 022540 100000 SC      ;CHANGE SC BIT
5461
5462 022542 004037 042404 JSR    RO,@#CHREG    ;CHANGE BITS IN SAVED REGISTER
5463 022546 002322 RHDS1  ;CHANGE RHDS1 REGISTER
5464
5465 022550 000001      1      ;1 BIT/BITS TO BE CHANGED
5466 022552 000001      1      ;NEW VALUE OF ATA IS 1
5467 022554 100000 ATA     ;CHANGE ATA BIT
5468
5469 022556 053737 004740 004636 BIS    @#ATTENT,@#SAVERE+24 ;SET APPROPRIATE 'ATA' BITS
5470                                     ;FOR WORKING DRIVE IN
5471                                     ;SAVED RHAS LOACTION
5472
5473                ;*NOW COMPARE REGISTERS AFTER AN OFFSET COMMAND
5474
5475
5476 022564 004037 042512 JSR    RO,@#COMREG    ;COMPARE SAVED REGISTERS WITH
5477                                     ;PRESENT VALUE
5478 022570 004612 SAVERE  ;GOOD DATA SAVED IN 'SAVERE'
5479 022572 002354 WC      ;TEST DATA STARTING FROM 'RHWC'
5480 022574 000022 18.    ;18. REGISTERS TO BE COMPARED
5481 022576 022602 2$     ;RETURN TO 2$ ON ERROR
5482 022600 022606 3$     ;RETURN TO 3$ ON NO ERROR
5483
5484 022602 104024 2$:    ERROR 24 ;OFFSET COMMAND CAUSED AN ERROR
5485 022604 000207 RTS    PC  ;GOOD DATA IS WHAT SHOULD BE THERE
5486                                     ;RECEIVED DATA GIVES WHAT WAS THERE
5487                                     ;AFTER AN OFFSET COMMAND
5488
5489 022606 013777 004740 157502 3$: MOV    @#ATTENT,@RHAS ;CLEAR WORKING DRIVE ATTENTION
5490
5491
5492
5493
5494                ;*NOW A RETURN TO CENTER LINE COMMAND WILL BE GIVEN
5495                ;:*****
5496
5497 022614 004737 041604 JSR    PC,@#CHECKT   ;CHECK DVA,RDY,MOL,DPR & VV = 1
5498 022620 000240 NOP     ;CHECK THAT ALL OTHER BITS = 0
5499 022622 000240 NOP     ;UNLIKE THE OTHER STATUS BIT TESTS,
5500 022624 000240 NOP     ;THERE IS NO HALT IF IT FAILS - IT IS
5501                                     ;USED IN THE MIDDLE OF A SINGLE TEST
5502 022626 013777 002456 157444 MOV    @#RETCL,@RHCS1 ;GET READY FOR RETCL
5503                                     ;RETURN TO CENTER LINE WITH 16 IN RHCS1
5504
5505
5506                ;*NOW REGISTERS ARE SAVED FOR COMPARISON AFTER COMMAND

```

```

5507
5508 022634 004037 041672      JSR      RO,@#SAVER      ;SAVE REGISTERS
5509 022640 002272              RHWC      ;RHWC IS THE FIRST REGISTER SAVED
5510 022642 004612              SAVERE    ;STARTING ADDRESS OF WHERE
5511                          ;THE REGISTERS ARE SAVED
5512 022644 000022              18.      ;NUMBER OF REGISTERS
5513                          ;SAVED = 18.
5514
5515 022646 013777 004606 157412  MOV      @#RP4VEC,@RPVEC ;SET RPO4 VECTOR ADDRESS
5516                          ;TO 'TIME1' IF P-CLOCK IS PRESENT
5517                          ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
5518                          ;'TIME' WILL ONLY SAVE
5519                          ;CURRENT CYLINDER ADDRESS
5520                          ;AND LOOK AHEAD REGISTERS
5521
5522
5523 022654 013746 002456      MOV      @#RETCL,-(SP)   ;GET READY TO MOVE COMMAND
5524 022660 052716 000101      BIS      #GO!IE,(SP)    ;GET READY TO SET 'GO' AND
5525                          ;ENABLE INTERRUPT
5526 022664 012677 157410      MOV      (SP)+,@RHCS1   ;GO WITH
5527                          ;16 IN RHCS1 FOR RETURN TO CENTER LINE
5528                          ;WITH INTERRUPT ENABLED
5529 022670 011100              MOV      @R1,R0         ;SAVE RHCS1 DURING ABOVE OPERATION
5530 022672 011305              MOV      @R3,R5         ;SAVE RHDS1 DURING ABOVE OPERATION
5531
5532
5533 022674 104413              WAT                      ;WAIT FOR DRY BIT TO SET
5534 022676 002322              RHDS1                   ;WAIT FOR RHDS1 REGISTER
5535 022700 000200              DRY                      ;WAIT FOR DRY BIT IN RHDS1 REGISTER
5536 022702 001750              1000.                   ;ALLOW 1000 MICRO SECONDS
5537 022704 001750              1000                    ;DRY MUST SET BETWEEN
5538                          ;00 AND 20000 MICRO SECONDS
5539
5540                          ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
5541                          ;*R0 AND R5 IMMEDIATELY AFTER GO
5542
5543 022706 013746 002456      MOV      @#RETCL,-(SP)   ;SAVE COMMAND
5544 022712 052716 004301      BIS      #DVA!GO!IE!RDY,(SP) ;INCLUDE DVA!GO!IE!RDY
5545 022716 005737 004722      TST      @#NUNIT        ;ARE THERE MORE THAN ONE UNIT
5546 022722 001413              BEQ      90$            ;BRANCH IF ONLY ONE UNIT
5547 022724 010037 004760      MOV      RO,@#TMP4      ;GET RHCS1
5548 022730 042737 177677 004760  BIC      #^CIE,@#TMP4   ;KEEP IE BIT
5549 022736 042716 000100      BIC      #IE,(SP)       ;CLEAR IE IN GOOD DATA
5550 022742 053716 004760      BIS      @#TMP4,(SP)    ;GET IE AS IS
5551 022746 052716 100000      BIS      #SC,(SP)      ;SET SC IN RHCS1
5552 022752                      90$:
5553 022752 011637 001124      MOV      (SP),@#SGDDAT   ;SAVE FOR PRINTOUT
5554 022756 022600              CMP      (SP)+,R0       ;DURING ABOVE OPERATION ONLY DVA!GO!IE!RDY
5555                          ;AND COMMAND SHOULD BE SET
5556 022760 001405              BEQ      89$            ;BRANCH IF GOOD
5557 022762 010037 001126      MOV      RO,@#SBDDAT    ;BAD DATA
5558 022766 010137 004600      MOV      R1,@#REGADR    ;FAILING REGISTER RHCS1
5559 022772 104021              ERROR   21             ;DURING ABOVE OPERATION ONLY
5560                          ;COMMAND AND DVA!GO!IE!RDY SHOULD BE SET
5561 022774 012746 030500      89$: MOV      #PIP!MOL'DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
5562 023000 011637 001124      MOV      (SP),@#SGDDAT ;SAVE FOR PRINTOUT

```

B 10

CZRJ10, RP04/5/6 FCTNL CTLR1 MACY11 30A(1052) 25-MAY-79 10:30 PAGE 119
 CZRJ10.P11 28-MAR-79 09:03 T31 OFFSET AND RETURN TO CENTER LINE COMMAND SEQ 0118

```

5563 023004 022605      CMP      (SP)+,R5      ;DURING ABOVE OPERATION ONLY PIP!MOL!DPR!VV
5564                                     ;SHOULD BE SET
5565 023006 001405      BEQ      91$          ;BRANCH IF GOOD
5566 023010 010537 001126  MOV      R5,@#$BDDAT ;BAD DATA
5567 023014 010337 004600  MOV      R3,@#REGADR  ;FAILING REGISTER RHDS1
5568 023020 104063      ERROR    63          ;DURING ABOVE OPERATION ONLY
5569                                     ;PIP!MOL!DPR!VV SHOULD BE SET
5570 023022                91$:
5571
5572                                     ;*NOW CHANGE SAVED REGISTER TO EXPECTED VALUE
5573
5574
5575 023022 004037 042404  JSR      RO,@#CHREG   ;CHANGE BITS IN SAVED REGISTER
5576 023026 002300      RHCS1                ;CHANGE RHCS1 REGISTER
5577
5578 023030 000001      1                  ;1 BIT/BITS TO BE CHANGED
5579 023032 000001      1                  ;NEW VALUE OF SC IS 1
5580 023034 100000      SC                  ;CHANGE SC BIT
5581
5582 023036 004037 042404  JSR      RO,@#CHREG   ;CHANGE BITS IN SAVED REGISTER
5583 023042 002322      RHDS1                ;CHANGE RHDS1 REGISTER
5584
5585 023044 000001      1                  ;1 BIT/BITS TO BE CHANGED
5586 023046 000001      1                  ;NEW VALUE OF ATA IS 1
5587 023050 100000      ATA                 ;CHANGE ATA BIT
5588
5589 023052 053737 004740 004636  BIS      @#ATTENT,@#SAVERE+24 ;SET APPROPRIATE 'ATA' BITS
5590                                     ;FOR WORKING DRIVE IN
5591                                     ;SAVED RHAS LOACTION
5592 023060 004037 041426  JSR      RO,@#FILLRE  ;MOV BIT15!HCI!ECI!FMT22 INTO SAVED RHOF
5593 023064 002310      RHOF                 ;SAVED REGISTER TO CHANGE
5594 023066 116000      BIT15!HCI!ECI.FMT22 ;DATA
5595
5596                                     ;*NOW COMPARE REGISTERS AFTER RETURN-TO-CENTER-LINE
5597
5598 023070 004037 042512  JSR      RO,@#COMREG  ;COMPARE SAVED REGISTERS WITH
5599                                     ;PRESENT VALUE
5600 023074 004612      SAVERE                ;GOOD DATA SAVED IN 'SAVERE'
5601 023076 002354      WC                  ;TEST DATA STARTING FROM 'RHWC'
5602 023100 000022      18.                 ;18. REGISTERS TO BE COMPARED
5603 023102 023106      4$                   ;RETURN TO 4$ ON ERROR
5604 023104 023112      5$                   ;RETURN TO 5$ ON NO ERROR
5605 023106 104025      4$: ERROR 25        ;RETURN TO CENTER-LINE
5606 023110 000207      RTS PC              ;COMMAND CAUSED AN ERROR
5607                                     ;GOOD DATA HAS WHAT SHOULD
5608                                     ;BE THERE
5609                                     ;RECEIVED DATA HAS WHAT WAS
5610                                     ;THERE AFTER COMMAND
5611
5612 023112                5$:
5613
5614 023112 004737 041524  JSR      PC,@#CLDISK  ;SET R1-RHCS1, R2-RHCS2
5615                                     ;R3-RHDS1, R4-RHER1
5616                                     ;GIVE RH-11 INITIALIZE
5617                                     ;SETUP UNIT NUMBER
5618 023116 105237 004610  INCB     @#OFSTVL     ;GET NEXT OFFSET VALUE

```


CZRJ1D0, RP04/5/6 FCTNL CTLR1
CZRJ1D.P11 28-MAR-79 09:03

MACY11 30A(1052) 25-MAY-79 10:30 PAGE 120
T31 OFFSET AND RETURN TO CENTER LINE COMMAND

SEQ 0119

5619	023122	132737	000100	004610		BITB	#100,@#OFSTVL	:SEE IF UNUSED BIT 6 IS ON
5620	023130	001403				BEQ	7\$:NO SO DO SOME MORE
5621	023132	062737	000100	004610		ADD	#100,@#OFSTVL	:YES SO BY-PASS IT
5622	023140	105737	004610		7\$:	TSTB	@#OFSTVL	:IF ZERO ALL COMBINATIONS ARE
5623								:COMPLETE
5624	023144	001001				BNE	6\$:BRANCH IF 377 NOT DONE
5625	023146	000402				BR	TST32 ;	BRANCH TO NEXT TEST
5626	023150	000137	022166		6\$:	JMP	@#1\$:JUMP BECAUSE 377 NOT DONE
5627								

```
5628 ;*****
5629 ;*TEST 32      OFFSET COMMAND
5630
5631 ;*      THIS TEST WILL ONLY GIVE REPEATED OFFSETS
5632
5633 ;*****
5634 023154 000004 TST32: SCOPE
5635 023156 012737 000764 004756 MOV # 500.,@#TMP1 ;COUNTER
5636 023164 1$:
5637 023164 012706 001000 MOV #STACK,SP ;RESET STACK
5638 023170 012737 000032 004604 MOV #32,@#TSTNM ;SAVE TEST NUMBER
5639
5640 023176 004737 041524 JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
5641 ;R3-RHDS1, R4-RHER1
5642 ;GIVE RH-11 INITIALIZE
5643 ;SETUP UNIT NUMBER
5644 023202 004737 041604 JSR PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
5645 ;AND THAT NO STATUS BITS ARE STUCK = 1
5646 023206 104401 067033 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF ANY OF
5647 ;THE FIRST SET OF BITS DON'T = 1
5648 023212 000000 HALT ;STOP
5649 023214 004037 041510 JSR RO,@#OFFSET ;OFFSET
5650 023220 000260 260 ;260 IN OFFSET REGISTER
5651 ;OFFSET -1200 MICRO INCHES
5652
5653
5654 023222 013746 002454 MOV @#OFFSETC,-(SP) ;GET READY TO MOVE COMMAND
5655 023226 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET 'GO' AND
5656 ;ENABLE INTERRUPT
5657 023232 012677 157042 MOV (SP)+,@#RHCS1 ;GO WITH
5658 ;14 IN RHCS1 FOR OFFSET
5659 ;WITH INTERRUPT ENABLED
5660
5661 023236 104413 WAT ;WAIT FOR DRY BIT TO SET
5662 023240 002322 RHDS1 ;WAIT FOR RHDS1 REGISTER
5663 023242 000200 DRY ;WAIT FOR DRY BIT IN RHDS1 REGISTER
5664 023244 000536 350. ;ALLOW 3500 MICRO SECONDS
5665 023246 000536 350. ;DRY MUST SET BETWEEN
5666 ;00 AND 7000 MICRO SECONDS
5667 023250 032777 040000 157044 BIT #ERR,@#RHDS1 ;IS ERR SET?
5668 023256 001417 BEQ 2$ ;NO
5669 023260 104026 ERROR 26 ;REPEATED OFFSETS CAUSED AN ERROR
5670 023262 000004 SCOPE
5671
5672 023264 004737 041524 JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
5673 ;R3-RHDS1, R4-RHER1
5674 ;GIVE RH-11 INITIALIZE
5675 ;SETUP UNIT NUMBER
5676
5677 023270 013746 002426 MOV @#RECALI,-(SP) ;GET READY TO MOVE COMMAND
5678 023274 052716 000101 BIS #GO.IE,(SP) ;GET READY TO SET 'GO' AND
5679 ;ENABLE INTERRUPT
5680 023300 012677 156774 MOV (SP)+,@#RHCS1 ;GO WITH
5681 ;6 IN RHCS1 FOR RECALIBRATE
5682 ;WITH INTERRUPT ENABLED
5683
```



```
5694 .SBTTL READ/WRITE TESTS USING MEDIA
5695 :*****
5696 :*TEST 33 WRITE/READ HEADER AND DATA (0'S)
5697
5698 :* WRITE HEADER AND DATA CYLINDER 0, FORMAT 16 BITS PER WORD
5699 :* TRACK 0, SECTOR 0, KEYS=0, NUMBER OF WORDS 256 WORDS
5700 :* OF 0
5701 :* THEN READ HEADER AND DATA FOR ABOVE.
5702 :* WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH
5703 :* 10000,0,0,0, AND 256 OF 0
5704 :* THE WRITE COMMAND IS THEN LOADED INTO THE REGISTERS EXCEPT
5705 :* THE GO BIT, AND ALL THE REGISTERS ARE SAVED
5706 :* THEN GO IS GIVEN FOR WRITE HEADER AND DATA
5707
5708 :* THEN ALL REGISTERS ARE COMPARED TO CHECK FOR IMPROPER CHANGED
5709 :* THEN WRITE FROM BUFFER IS CHECKED TO SEE THAT NOTHING CHANGED
5710
5711 :*
5712 :* NOW FOR THE READ COMMAND READ INTO BUFFER IS FILLED
5713 :* WITH ALL ONES, COMMAND IS LOADED INTO REGISTERS EXCEPT
5714 :* GO BIT AND ALL REGISTERS ARE SAVED
5715 :* GO IS GIVEN FOR THE READ COMMAND
5716
5717 :* ALL REGISTERS ARE CHECKED FOR IMPROPER CHANGE
5718 :* THEN THE READ DATA IS COMPARED.
5719
5720 :*****
5721 023326 000004 TST33: SCOPE
5722 023330 012706 001000 MOV #STACK,SP ;RESET STACK
5723 023334 012737 000033 004604 MOV #33,@#TSTNM ;SAVE TEST NUMBER
5724
5725 023342 004737 041524 JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
5726 ;R3-RHDS1, R4-RHER1
5727 ;GIVE RH-11 INITIALIZE
5728 ;SETUP UNIT NUMBER
5729
5730 ;*FILL WRITE FROM BUFFER WITH HEADER
5731
5732
5733 023346 004037 041350 JSR R0,@#FLHEAD ;SAVE HEADER DATA IN WRFROM
5734 023352 002470 WRFROM ;LOCATION WHERE SAVED
5735 023354 000004 4 ;NUMBER OF WORDS SAVED
5736 023356 010000 10000 ;FIRST DATA WORD
5737 023360 000000 0 ;SECOND DATA WORD
5738 023362 000000 0 ;THIRD DATA WORD
5739 023364 000000 0 ;FOURTH DATA WORD
5740
5741 ;*FILL WRITE FROM BUFFER WITH DATA
5742
5743 023366 004037 041374 JSR R0,@#CLAREA ;CLEAR 256. WORDS, FROM WRFROM+10
5744 023372 002500 WRFROM+10 ;STARTING FROM WRFROM+10
5745 023374 000400 256. ;256. WORDS
5746 023376 000000 0 ;FILL WITH 0
5747
5748 ;*NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA
5749
```

```
5750 ;*AS WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
5751 ;*CAN BE MADE TO MAKE SURE THAT WRITE DID NOT
5752 ;*CHANGE WRITE FROM BUFFER
5753
5754
5755 023400 004037 041350 JSR RO,@#FLHEAD ;SAVE HEADER DATA IN REINTO
5756 023404 003534 REINTO ;LOCATION WHERE SAVED
5757 023406 000004 4 ;NUMBER OF WORDS SAVED
5758 023410 010000 10000 ;FIRST DATA WORD
5759 023412 000000 0 ;SECOND DATA WORD
5760 023414 000000 0 ;THIRD DATA WORD
5761 023416 000000 0 ;FOURTH DATA WORD
5762 023420 004037 041374 JSR RO,@#CLAREA ;CLEAR 256. WORDS, FROM REINTO+10
5763 023424 003544 REINTO+10 ;STARTING FROM REINTO+10
5764 023426 000400 256. ;256. WORDS
5765 023430 000000 0 ;FILL WITH 0
5766
5767
5768 ;*NOW THE WRITE HEADER AND DATA COMMAND WILL BE FILLED
5769
5770
5771 023432 004037 043476 JSR RO,@#RUN ;SETUP TO RUN FOR DATA COMMAND
5772 023436 000000 0 ;CYLINDER 0
5773 023440 000 .BYTE 0 ;SECTOR 0
5774 023441 000 .BYTE 0 ;TRACK 0
5775 023442 177374 -256.-4 ;WORD COUNT (DATA) = 256. +
5776 ;4 HEADER WORDS
5777 023444 002470 WRFROM ;BUS ADDRESS
5778 ;STARTING ADDRESS OF DATA
5779 ;BUFFER = WRFROM
5780 023446 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCHEMENT
5781 023450 010000 FMT22 ;16 BITS PER WORD FORMAT
5782 ;DO NOT INHIBIT ECC CORRECTION
5783 ;DO NOT INHIBIT HEADER COMPARE
5784 023452 002444 WRIFOR ;GET READY TO DO A WRIFOR
5785 ;WRITE HEADER AND DATA WITH 62 IN RHCS1
5786
5787
5788 ;*NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE HEADER AND DATA
5789
5790 023454 004037 041672 JSR RO,@#SAVER ;SAVE REGISTERS
5791 023460 002272 RHWC ;RHWC IS THE FIRST REGISTER SAVED
5792 023462 004612 SAVERE ;STARTING ADDRESS OF WHERE
5793 ;THE REGISTERS ARE SAVED
5794 023464 000021 17. ;NUMBER OF REGISTERS
5795 ;SAVED = 17.
5796
5797 023466 004737 041604 JSR PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
5798 ;AND THAT NO STATUS BITS ARE STUCK = 1
5799 023472 104401 067033 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF ANY OF
5800 ;THE FIRST SET OF BITS DON'T 1
5801 023476 000000 HALT ;STOP
5802
5803 023500 013777 004606 156560 MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
5804 ;TO 'TIME1' IF P-CLOCK IS PRESENT
5805 ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
```

```
5806                                     ;'TIME' WILL ONLY SAVE
5807                                     ;CURRENT CYLINDER ADDRESS
5808                                     ;AND LOOK AHEAD REGISTERS
5809
5810
5811 023506 013746 002444      MOV    @#WRIFOR,-(SP)    ;GET READY TO MOVE COMMAND
5812 023512 052716 000101      BIS    #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
5813                                     ;ENABLE INTERRUPT
5814 023516 012677 156556      MOV    (SP)+,@RHCS1    ;GO WITH
5815                                     ;62 IN RHCS1 FOR WRITE HEADER AND DATA
5816                                     ;WITH INTERRUPT ENABLED
5817 023522 011100      MOV    @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
5818 023524 011305      MOV    @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
5819
5820                                     ;*ONE REVOLUTION=16670 MICRO SEC, ONE SECTOR = 760 MICRO SEC
5821
5822
5823 023526 104413      WAT                                     ;WAIT FOR RDY BIT TO SET
5824 023530 002300      RHCS1                                ;WAIT FOR RHCS1 REGISTER
5825 023532 000200      RDY                                    ;WAIT FOR RDY BIT IN RHCS1 REGISTER
5826 023534 001614      908.                                  ;ALLOW 9080 MICRO SECONDS
5827 023536 001507      839.                                  ;RDY MUST SET BETWEEN
5828                                     ;690 AND 17470 MICRO SECONDS
5829
5830                                     ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
5831                                     ;*R0 AND R5 IMMEDIATELY AFTER GO
5832
5833 023540 013746 002444      MOV    @#WRIFOR,-(SP)    ;SAVE COMMAND
5834 023544 052716 004101      BIS    #IE!GO!DVA,(SP)  ;INCLUDE IE!GO!DVA
5835 023550 011637 001124      MOV    (SP),@#$GDDAT    ;SAVE FOR PRINTOUT
5836 023554 022600      CMP    (SP)+,R0         ;DURING ABOVE OPERATION ONLY IE.GO.DVA
5837                                     ;AND COMMAND SHOULD BE SET
5838 023556 001405      BEQ    64$              ;BRANCH IF GOOD
5839 023560 010037 001126      MOV    R0,@#$BDDAT      ;BAD DATA
5840 023564 010137 004600      MOV    R1,@#REGADR      ;FAILING REGISTER RHCS1
5841 023570 104021      ERROR  21              ;DURING ABOVE OPERATION ONLY
5842                                     ;COMMAND AND IE!GO!DVA SHOULD BE SET
5843 023572 012746 010500      64$: MOV    #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
5844 023576 011637 001124      MOV    (SP),@#$GDDAT    ;SAVE FOR PRINTOUT
5845 023602 022605      CMP    (SP)+,R5         ;DURING ABOVE OPERATION ONLY MOL.DPR!VV
5846                                     ;SHOULD BE SET
5847 023604 001405      BEQ    66$              ;BRANCH IF GOOD
5848 023606 010537 001126      MOV    R5,@#$BDDAT      ;BAD DATA
5849 023612 010337 004600      MOV    R3,@#REGADR      ;FAILING REGISTER RHDS1
5850 023616 104063      ERROR  63              ;DURING ABOVE OPERATION ONLY
5851                                     ;MOL!DPR!VV SHOULD BE SET
5852 023620      66$:
5853
5854                                     ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
5855
5856 023620 004037 041426      JSR    R0,@#FILLRE      ;MOV 0 INTO SAVED RHWC
5857 023624 002272      RHWC                                ;SAVED REGISTER TO CHANGE
5858 023626 000000      0                                    ;DATA
5859 023630 004037 041426      JSR    R0,@#FILLRE      ;MOV WRFROM+<260.*2> INTO SAVED RHBA
5860 023634 002274      RHBA                                ;SAVED REGISTER TO CHANGE
5861 023636 003500      WRFROM+<260.*2>        ;DATA
```

```

5862 023640 004037 041426 JSR RO,@#FILLRE ;MOV 1 INTO SAVED RHDST
5863 023644 002304 RHDST ;SAVED REGISTER TO CHANGE
5864 023646 000001 1 ;DATA
5865
5866 ;*NOW COMPARE REGISTERS BEFORE WRITE HEADER AND DATA
5867 ;*WITH REGISTERS AFTER COMMAND
5868
5869
5870 023650 004037 042512 JSR RO,@#COMREG ;COMPARE SAVED REGISTERS WITH
5871 ;PRESENT VALUE
5872 023654 004612 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
5873 023656 002354 WC ;TEST DATA STARTING FROM 'RHWC'
5874 023660 000021 17. ;17. REGISTERS TO BE COMPARED
5875 023662 023666 1$ ;RETURN TO 1$ ON ERROR
5876 023664 023672 2$ ;RETURN TO 2$ ON NO ERROR
5877
5878 023666 104027 1$: ERROR 27 ;WRITE HEADER AND DATA
5879 023670 000207 RTS PC ;CAUSED IMPROPER REGISTER
5880 ;CHANGE
5881 ;GOOD DATA GIVES WHAT SHOULD
5882 ;BE THERE
5883 ;RECEIVED DATA GIVES WHAT
5884 ;WAS THERE AFTER COMMAND
5885
5886 ;*NOW WRITE FROM BUFFER WILL BE CHECKED TO SEE THAT
5887 ;*NOTHING GOT CHANGED
5888
5889 023672 2$:
5890
5891 023672 004037 043542 JSR RO,@#COMPAR ;COMPARE TWO BLOCKS OF MEMORY
5892 023676 003534 REINTO ;GOOD DATA STARTS FROM REINTO
5893 023700 002470 WRFROM ;TEST DATA STARTS FROM WRFROM
5894 023702 000404 260. ;260. WORDS TO BE COMPARED
5895 023704 023710 3$ ;RETURN TO 3$ ON ERROR
5896 023706 023714 4$ ;RETURN TO 4$ ON NO ERROR
5897
5898
5899 023710 104030 3$: ERROR 30 ;WRITE HEADER AND DATA
5900 023712 000207 RTS PC ;CHANGED WRITE FROM BUFFER
5901
5902 ;*NOW A READ HEADER AND DATA COMMAND WILL BE GIVEN
5903 ;*READ INTO BUFFER IS FILLED WITH ONES
5904
5905 023714 4$:
5906
5907 023714 004737 041524 JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
5908 ;R3-RHDS1, R4-RHER1
5909 ;GIVE RH-11 INITIALIZE
5910 ;SETUP UNIT NUMBER
5911 023720 004037 041374 JSR RO,@#CLAREA ;CLEAR 260. WORDS, FROM REINTO
5912 023724 003534 REINTO ;STARTING FROM REINTO
5913 023726 000404 260. ;260. WORDS
5914 023730 177777 -1 ;FILL WITH -1
5915
5916
5917 ;*NOW FILL COMMAND
  
```

```

5918
5919
5920 023732 004037 043476      JSR      RO,@#RUN      ;SETUP TO RUN FOR DATA COMMAND
5921 023736 000000              0              ;CYLINDER 0
5922 023740      000          .BYTE 0              ;SECTOR 0
5923 023741      000          .BYTE 0              ;TRACK 0
5924 023742 177374          -256.-4         ;WORD COUNT (DATA) = 256. +
5925                                ;4 HEADER WORDS
5926 023744 003534      REINTO          ;BUS ADDRESS
5927                                ;STARTING ADDRESS OF DATA
5928                                ;BUFFER = REINTO
5929 023746 000000              0              ;DO NOT INHIBIT BUS ADDRESS INCREMENT
5930 023750 014000      ECI!FMT22          ;16 BITS PER WORD FORMAT
5931                                ;INHIBIT ECC CORRECTION
5932                                ;DO NOT INHIBIT HEADER COMPARE
5933 023752 002450      REFOR            ;GET READY TO DO A REFOR
5934                                ;READ HEADER AND DATA WITH 72 IN RHCS1
5935
5936
5937                                ;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ HEADER AND DATA
5938
5939 023754 004037 041672      JSR      RO,@#SAVER    ;SAVE REGISTERS
5940 023760 002272      RHWC            ;RHWC IS THE FIRST REGISTER SAVED
5941 023762 004612      SAVERE          ;STARTING ADDRESS OF WHERE
5942                                ;THE REGISTERS ARE SAVED
5943 023764 000022          18.            ;NUMBER OF REGISTERS
5944                                ;SAVED = 18.
5945
5946 023766 004737 041604      JSR      PC,@#CHECKT   ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
5947                                ;AND THAT NO STATUS BITS ARE STUCK 1
5948 023772 104401 067033      TYPE      ,CPHALT     ;CANNOT CONTINUE TESTING IF ANY OF
5949                                ;THE FIRST SET OF BITS DON'T = 1
5950 023776 000000      HALT            ;STOP
5951
5952 024000 013777 004606 156260  MOV      @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
5953                                ;TO 'TIME1' IF P-CLOCK IS PRESENT
5954                                ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
5955                                ;'TIME' WILL ONLY SAVE
5956                                ;CURRENT CYLINDER ADDRESS
5957                                ;AND LOOK AHEAD REGISTERS
5958
5959
5960 024006 013746 002450      MOV      @#REFOR,-(SP) ;GET READY TO MOVE COMMAND
5961 024012 052716 000101      BIS      #GO!IE,(SP)  ;GET READY TO SET 'GO' AND
5962                                ;ENABLE INTERRUPT
5963 024016 012677 156256      MOV      (SP)+,@RHCS1 ;GO WITH
5964                                ;72 IN RHCS1 FOR READ DATA
5965                                ;WITH INTERRUPT ENABLED
5966 024022 011100      MOV      @R1,R0       ;SAVE RHCS1 DURING ABOVE OPERATION
5967 024024 011305      MOV      @R3,R5       ;SAVE RHDS1 DURING ABOVE OPERATION
5968
5969
5970 024026 104413      WAT            ;WAIT FOR RDY BIT TO SET
5971 024030 002300      RHCS1         ;WAIT FOR RHCS1 REGISTER
5972 024032 000200      RDY           ;WAIT FOR RDY BIT IN RHCS1 REGISTER
5973 024034 001614          908.          ;ALLOW 9080 MICRO SECONDS

```



```
6030 ;THERE AFTER COMMAND
6031
6032 ;*NOW READ INTO BUFFER WILL BE CHECKED TO SEE
6033 ;*THE READ WAS GOOD
6034
6035 024172 6$:
6036
6037 024172 004037 043542 JSR RO,@#COMPAR ;COMPARE TWO BLOCKS OF MEMORY
6038 024176 002470 WRFROM ;GOOD DATA STARTS FROM WRFROM
6039 024200 003534 REINTO ;TEST DATA STARTS FROM REINTO
6040 024202 000404 260. ;260. WORDS TO BE COMPARED
6041 024204 024210 7$ ;RETURN TO 7$ ON ERROR
6042 024206 024214 10$ ;RETURN TO '0$ ON NO ERROR
6043
6044
6045 024210 104032 7$: ERROR 32 ;WRITE HEADER AND DATA
6046 024212 000207 RTS PC ;FOLLOWED BY A READ HEADER
6047 ;AND DATA GAVE A READ ERROR
6048 ;ERROR MAY BE IN READ OR WRITE
6049 024214 10$:
6050
```

```
6051
6052
6053
6054
6055
6056
6057
6058
6059
6060
6061
6062
6063 024214 000004
6064 024216 012706 001000
6065 024222 012737 000034 004604
6066
6067 024230 004737 041524
6068
6069
6070
6071
6072
6073
6074 024234 004037 041374
6075 024240 002470
6076 024242 000400
6077 024244 000000
6078
6079
6080
6081
6082
6083 024246 004037 043476
6084 024252 000000
6085 024254 000
6086 024255 000
6087 024256 177400
6088 024260 003534
6089
6090
6091 024262 000000
6092 024264 014000
6093
6094
6095 024266 002446
6096
6097
6098
6099
6100
6101 024270 004037 041672
6102 024274 002272
6103 024276 004612
6104
6105 024300 000022
6106
```

```
*****
;*TEST 34 READ DATA (O'S)
*****
;* THIS TEST READS DATA WRITTEN BY THE PREVIOUS TEST
;* THE WRITE FROM BUFFER IS FILLED WITH ALL OS
;* THE COMMAND IS FILLED, THEN ALL REGISTERS SAVED
;* THEN READ DATA COMMAND IS GIVEN
;* ALL REGISTERS ARE COMPARED FOR PROPER VALUES
;* READ DATA INTO 'READ INTO' BUFFER IS CHECKED
*****
TST34: SCOPE
MOV #STACK,SP ;RESET STACK
MOV #34,@#TSTNM ;SAVE TEST NUMBER
JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
;R3-RHDS1, R4-RHER1
;GIVE RH-11 INITIALIZE
;SETUP UNIT NUMBER
;*FILL WRITE FROM BUFFER WITH EXPECTED DATA
JSR RO,@#CLAREA ;CLEAR 256. WORDS, FROM WRFROM
WRFROM ;STARTING FROM WRFROM
256. ;256. WORDS
0 ;FILL WITH 0
;*NOW THE READ DATA COMMAND WILL BE FILLED
JSR RO,@#RUN ;SETUP TO RUN FOR DATA COMMAND
0 ;CYLINDER 0
.BYTE 0 ;SECTOR 0
.BYTE 0 ;TRACK 0
-256. ;WORD COUNT = 256.
REINTO ;BUS ADDRESS
;STARTING ADDRESS OF DATA
;BUFFER = REINTO
0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
ECI!FMT2 ;16 BITS PER WORD FORMAT
;INHIBIT ECC CORRECTION
;DO NOT INHIBIT HEADER COMPARE
READAT ;GET READY TO DO A READAT
;READ DATA WITH 70 IN RHCS1
;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ DATA COMMAND
JSR RO,@#SAVER ;SAVE REGISTERS
RHWC ;RHWC IS THE FIRST REGISTER SAVED
SAVERE ;STARTING ADDRESS OF WHERE
;THE REGISTERS ARE SAVED
18. ;NUMBER OF REGISTERS
;SAVED = 18.
```

```

6107
6108 024302 004737 041604 JSR PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
6109 ;AND THAT NO STATUS BITS ARE STUCK = 1
6110 024306 104401 067033 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF ANY OF
6111 ;THE FIRST SET OF BITS DON'T = 1
6112 024312 000000 HALT ;STOP
6113
6114 024314 013777 004606 155744 MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
6115 ;TO 'TIME1' IF P-CLOCK IS PRESENT
6116 ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
6117 ;'TIME' WILL ONLY SAVE
6118 ;CURRENT CYLINDER ADDRESS
6119 ;AND LOOK AHEAD REGISTERS
6120
6121
6122 024322 013746 002446 MOV @#READAT,-(SP) ;GET READY TO MOVE COMMAND
6123 024326 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET 'GO' AND
6124 ;ENABLE INTERRUPT
6125 024332 012677 155742 MOV (SP)+,@RHCS1 ;GO WITH
6126 ;70 IN RHCS1 FOR READ DATA
6127 ;WITH INTERRUPT ENABLED
6128 024336 011100 MOV @R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
6129 024340 011305 MOV @R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
6130
6131
6132 024342 104413 WAT ;WAIT FOR RDY BIT TO SET
6133 024344 002300 RHCS1 ;WAIT FOR RHCS1 REGISTER
6134 024346 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
6135 024350 001614 908. ;ALLOW 9080 MICRO SECONDS
6136 024352 001507 839. ;RDY MUST SET BETWEEN
6137 ;690 AND 17470 MICRO SECONDS
6138
6139 ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
6140 ;*R0 AND R5 IMMEDIATELY AFTER GO
6141
6142 024354 013746 002446 MOV @#READAT,-(SP) ;SAVE COMMAND
6143 024360 052716 004101 BIS #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
6144 024364 011637 001124 MOV (SP),@#SGDDAT ;SAVE FOR PRINTOUT
6145 024370 022600 CMP (SP)+,R0 ;DURING ABOVE OPERATION ONLY IE!GO.DVA
6146 ;AND COMMAND SHOULD BE SET
6147 024372 001405 BEQ 64$ ;BRANCH IF GOOD
6148 024374 010037 001126 MOV R0,@#SBDDAT ;BAD DATA
6149 024400 010137 004600 MOV R1,@#REGADR ;FAILING REGISTER RHCS1
6150 024404 104021 ERROR 21 ;DURING ABOVE OPERATION ONLY
6151 ;COMMAND AND IE!GO!DVA SHOULD BE SET
6152 024406 012746 010500 64$: MOV #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
6153 024412 011637 001124 MOV (SP),@#SGDDAT ;SAVE FOR PRINTOUT
6154 024416 022605 CMP (SP)+,R5 ;DURING ABOVE OPERATION ONLY MOL!DPR.VV
6155 ;SHOULD BE SET
6156 024420 001405 BEQ 66$ ;BRANCH IF GOOD
6157 024422 010537 001126 MOV R5,@#SBDDAT ;BAD DATA
6158 024426 010337 004600 MOV R3,@#REGADR ;FAILING REGISTER RHDS1
6159 024432 104063 ERROR 63 ;DURING ABOVE OPERATION ONLY
6160 ;MOL!DPR.VV SHOULD BE SET
6161 024434 66$:
6162

```

```
6163 ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
6164
6165 024434 004037 041426 JSR RO,@#FILLRE ;MOV 0 INTO SAVED RHWC
6166 024440 002272 RHWC ;SAVED REGISTER TO CHANGE
6167 024442 000000 0 ;DATA
6168 024444 004037 041426 JSR RO,@#FILLRE ;MOV REINTO+<256.*2> INTO SAVED RHBA
6169 024450 002274 RHBA ;SAVED REGISTER TO CHANGE
6170 024452 004534 REINTO+<256.*2> ;DATA
6171 024454 004037 041426 JSR RO,@#FILLRE ;MOV 1 INTO SAVED RHDST
6172 024460 002304 RHDST ;SAVED REGISTER TO CHANGE
6173 024462 000001 1 ;DATA
6174
6175 ;*NOW COMPARE REGISTERS BEFORE READ DATA WITH
6176 ;*AFTER COMMAND
6177
6178
6179 024464 004037 042512 JSR RO,@#COMREG ;COMPARE SAVED REGISTERS WITH
6180 ;PRESENT VALUE
6181 024470 004612 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
6182 024472 002354 WC ;TEST DATA STARTING FROM 'RHWC'
6183 024474 000022 18. ;18. REGISTERS TO BE COMPARED
6184 024476 024502 1$ ;RETURN TO 1$ ON ERROR
6185 024500 024506 2$ ;RETURN TO 2$ ON NO ERROR
6186
6187 024502 104033 1$: ERROR 33 ;READ DATA CAUSED IMPROPER REGISTER
6188 024504 000207 RTS PC ;CHANGE
6189 ;GOOD DATA GIVES WHAT SHOULD BE THERE
6190 ;RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND
6191 ;*NOW READ INTO BUFFER WILL BE CHECKED TO SEE THAT READ
6192 ;*WAS GOOD
6193
6194 024506 2$:
6195
6196 024506 004037 043542 JSR RO,@#COMPAR ;COMPARE TWO BLOCKS OF MEMORY
6197 024512 002470 WRFROM ;GOOD DATA STARTS FROM WRFROM
6198 024514 003534 REINTO ;TEST DATA STARTS FROM REINTO
6199 024516 000400 256. ;256. WORDS TO BE COMPARED
6200 024520 024524 3$ ;RETURN TO 3$ ON ERROR
6201 024522 024530 4$ ;RETURN TO 4$ ON NO ERROR
6202
6203
6204 024524 104034 3$: ERROR 34 ;READ DATA COMMAND
6205 024526 000207 RTS PC ;READ INCORRECTLY
6206
6207 024530 4$:
```

6208
6209
6210
6211
6212
6213
6214
6215
6216
6217
6218
6219
6220
6221
6222
6223
6224
6225
6226
6227
6228
6229
6230
6231
6232
6233
6234
6235
6236
6237
6238
6239
6240
6241
6242
6243
6244
6245
6246
6247
6248
6249
6250
6251
6252
6253
6254
6255
6256
6257
6258
6259
6260
6261
6262
6263

: *TEST 35 WRITE/READ DATA (1'S & 125252)

: * THIS TEST GIVES A WRITE DATA COMMAND FRO CYLINDER 0
: * TRACK 0, SECTOR 0, KEYS 0, 200 WORDS OF ALL ONES
: * THIS SECTOR IS FORMATED BY PREVIOUS TEST
: * THEN READ DATA COMMAND IS GIVEN FOR 256 WORDS IN
: * SAME CYLINDER, TRACK, SECTOR, KEYS
: * WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH
: * 200 ONES AND 56 125252
: * THE WRITE DATA COMMAND IS LOADED EXCEPT GO AND IE
: * ALL REGISTERS ARE SAVED AND THEN GO IS GIVEN TO WRITE DATA

: * THEN ALL REGISTERS ARE COMPARED TO CHECK FOR IMPROPER
: * CHANGE, THEN WRITE FROM BUFFER IS CHECKED FOR NO CHANGE

: * THEN READ INTO BUFFER IS FILLED WITH 200 OF ZEROS
: * AND 56 OF 377, WRITE FROM BUFFER IS FILLED WITH 200 ONES
: * AND 56 OF 377.
: * THE COMMAND EXCEPT GO IS LOADED, ALL REGISTERS ARE SAVED
: * GO IS GIVEN

: * ALL REGISTERS ARE CHECKED
: * READ DATA IS CHECKED

TST35: SCOPE
MOV #STACK,SP ;RESET STACK
MOV #35,@#TSTNM ;SAVE TEST NUMBER

JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
;R3-RHDS1, R4-RHER1
;GIVE RH-11 INITIALIZE
;SETUP UNIT NUMBER

: *NOW FILL WRITE FROM BUFFER -200 OF 1'S AND 56 OF 125252

JSR RO,@#CLAREA ;CLEAR 200. WORDS, FROM WRFROM
WRFROM ;STARTING FROM WRFROM
200. ;200. WORDS
-1 ;FILL WITH -1

JSR RO,@#CLAREA ;CLEAR 56. WORDS, FROM WRFROM+<200.*2>
WRFROM+<200.*2> ;STARTING FROM WRFROM+<200.*2>
56. ;56. WORDS
125252 ;FILL WITH 125252

: *NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA AS
: *WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
: *CAN BE MADE TO DETERMINE THAT WRITE DID NOT CHANGE BUFFER

JSR RO,@#CLAREA ;CLEAR 200. WORDS, FROM REINTO

024530 000004
024532 012706 001000
024536 012737 000035 004604

024544 004737 041524

024550 004037 041374
024554 002470
024556 000310
024560 177777

024562 004037 041374
024566 003310
024570 000070
024572 125252

024574 004037 041374

```
6264 024600 003534 REINTO ;STARTING FROM REINTO
6265 024602 000310 200. ;200. WORDS
6266 024604 177777 -1 ;FILL WITH -1
6267
6268 024606 004037 041374 JSR RO,@#CLAREA ;CLEAR 56. WORDS, FROM REINTO+<200.*2>
6269 024612 004354 REINTO+<200.*2> ;STARTING FROM REINTO+<200.*2>
6270 024614 000070 56. ;56. WORDS
6271 024616 125252 125252 ;FILL WITH 125252
6272
6273
6274 ;*NOW WRITE DATA COMMAND WILL BE LOADED
6275
6276
6277 024620 004037 043476 JSR RO,@#RUN ;SETUP TO RUN FOR DATA COMMAND
6278 024624 000000 0 ;CYLINDER 0
6279 024626 000 .BYTE 0 ;SECTOR 0
6280 024627 000 .BYTE 0 ;TRACK 0
6281 024630 177470 -200. ;WORD COUNT = 200.
6282 024632 002470 WRFROM ;BUS ADDRESS
6283 ;STARTING ADDRESS OF DATA
6284 ;BUFFER = WRFROM
6285 024634 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
6286 024636 010000 FMT22 ;16 BITS PER WORD FORMAT
6287 ;DO NOT INHIBIT ECC CORRECTION
6288 ;DO NOT INHIBIT HEADER COMPARE
6289 024640 002442 WRIDAT ;GET READY TO DO A WRIDAT
6290 ;WRITE DATA WITH 60 IN RHCS1
6291
6292
6293 ;*NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE DATA
6294
6295 024642 004037 041672 JSR RO,@#SAVER ;SAVE REGISTERS
6296 024646 002272 RHWC ;RHWC IS THE FIRST REGISTER SAVED
6297 024650 004612 SAVERE ;STARTING ADDRESS OF WHERE
6298 6298 ;THE REGISTERS ARE SAVED
6299 024652 000022 18. ;NUMBER OF REGISTERS
6300 ;SAVED = 18.
6301
6302 024654 004737 041604 JSR PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
6303 ;AND THAT NO STATUS BITS ARE STUCK = 1
6304 024660 104401 067033 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF ANY OF
6305 ;THE FIRST SET OF BITS DON'T = 1
6306 024664 000000 HALT ;STOP
6307
6308 024666 013777 004606 155372 MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
6309 ;TO 'TIME1' IF P-CLOCK IS PRESENT
6310 ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
6311 ;'TIME' WILL ONLY SAVE
6312 ;CURRENT CYLINDER ADDRESS
6313 ;AND LOOK AHEAD REGISTERS
6314
6315
6316 024674 013746 002442 MOV @#WRIDAT,-(SP) ;GET READY TO MOVE COMMAND
6317 024700 052716 000101 BIS #GO.IE,(SP) ;GET READY TO SET 'GO' AND
6318 ;ENABLE INTERRUPT
6319 024704 012677 155370 MOV (SP)+,@RHCS1 ;GO WITH
```

```

6320                                     ;60 IN RHCS1 FOR WRITE DATA
6321                                     ;WITH INTERRUPT ENABLED
6322 024710 011100                       MOV @R1,R0                       ;SAVE RHCS1 DURING ABOVE OPERATION
6323 024712 011305                       MOV @R3,R5                       ;SAVE RHDS1 DURING ABOVE OPERATION
6324
6325                                     ;*ONE REVOLUTION = 16670 MICRO SEC, ONE SECTOR=760 MICRO SEC
6326
6327
6328 024714 104413                       WAT                               ;WAIT FOR RDY BIT TO SET
6329 024716 002300                       RHCS1                            ;WAIT FOR RHCS1 REGISTER
6330 024720 000200                       RDY                               ;WAIT FOR RDY BIT IN RHCS1 REGISTER
6331 024722 001614                       908.                            ;ALLOW 9080 MICRO SECONDS
6332 024724 001507                       839.                            ;RDY MUST SET BETWEEN
6333                                     ;690 AND 17470 MICRO SECONDS
6334
6335                                     ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
6336                                     ;*R0 AND R5 IMMEDIATELY AFTER GO
6337
6338 024726 013746 002442                 MOV @#WRIDAT,-(SP)              ;SAVE COMMAND
6339 024732 052716 004101                 BIS #IE!GO!DVA,(SP)           ;INCLUDE IE!GO!DVA
6340 024736 011637 001124                 MOV (SP),@#SGDDAT             ;SAVE FOR PRINTOUT
6341 024742 022600                       CMP (SP)+,R0                   ;DURING ABOVE OPERATION ONLY IE.GO!DVA
6342                                     ;AND COMMAND SHOULD BE SET
6343 024744 001405                       BEQ 64$                        ;BRANCH IF GOOD
6344 024746 010037 001126                 MOV R0,@#SBDDAT              ;BAD DATA
6345 024752 010137 004600                 MOV R1,@#REGADR              ;FAILING REGISTER RHCS1
6346 024756 104021                       ERROR 21                      ;DURING ABOVE OPERATION ONLY
6347                                     ;COMMAND AND IE!GO!DVA SHOULD BE SET
6348 024760 012746 010500                 64$: MOV #MOL!DPR!VV,-(SP)      ;SAVE BITS SET DURING OPERATION IN RHDS1
6349 024764 011637 001124                 MOV (SP),@#SGDDAT           ;SAVE FOR PRINTOUT
6350 024770 022605                       CMP (SP)+,R5                  ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
6351                                     ;SHOULD BE SET
6352 024772 001405                       BEQ 66$                        ;BRANCH IF GOOD
6353 024774 010537 001126                 MOV R5,@#SBDDAT              ;BAD DATA
6354 025000 010337 004600                 MOV R3,@#REGADR              ;FAILING REGISTER RHDS1
6355 025004 104063                       ERROR 63                      ;DURING ABOVE OPERATION ONLY
6356                                     ;MOL!DPR!VV SHOULD BE SET
6357 025006                               66$:
6358
6359                                     ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
6360
6361 025006 004037 041426                 JSR R0,@#FILLRE              ;MOV 0 INTO SAVED RHWC
6362 025012 002272                       RHWC                          ;SAVED REGISTER TO CHANGE
6363 025014 000000                       0                             ;DATA
6364 025016 004037 041426                 JSR R0,@#FILLRE              ;MOV WRFROM+<200.*2> INTO SAVED RHBA
6365 025022 002274                       RHBA                          ;SAVED REGISTER TO CHANGE
6366 025024 003310                       WRFROM+<200.*2>              ;DATA
6367 025026 004037 041426                 JSR R0,@#FILLRE              ;MOV 1 INTO SAVED RHDST
6368 025032 002304                       RHDST                         ;SAVED REGISTER TO CHANGE
6369 025034 000001                       1                             ;DATA
6370
6371                                     ;*NOW COMPARE REGISTERS BEFORE WRITE DATA WITH REGISTERS
6372                                     ;*AFTER COMMAND
6373
6374
6375 025036 004037 042512                 JSR R0,@#COMREG              ;COMPARE SAVED REGISTERS WITH

```



```

6376                                     ;PRESENT VALUE
6377 025042 004612                       ;GOOD DATA SAVED IN 'SAVERE'
6378 025044 002354                       ;TEST DATA STARTING FROM 'RHWC'
6379 025046 000022                       ;18. REGISTERS TO BE COMPARED
6380 025050 025054                       ;1$
6381 025052 025060                       ;2$
6382                                     ;
6383 025054 104035 1$: ERROR 35          ;WRITE DATA COMMAND CAUSED
6384 025056 000207                       ;IMPROPER REGISTER CHANGE
6385                                     ;GOOD DATA GIVES WHAT SHOULD
6386                                     ;BE
6387                                     ;RECEIVED DATA GIVES WHAT WAS
6388                                     ;THERE AFTER COMMAND
6389                                     ;
6390                                     ;*NOW WRITE FROM BUFFER WILL BE CHECKED FOR NO CHANGE
6391                                     ;
6392 025060 2$:
6393                                     ;
6394 025060 004037 043542                JSR    RO,@#COMPAR      ;COMPARE TWO BLOCKS OF MEMORY
6395 025064 003534                       REINTO   ;GOOD DATA STARTS FROM REINTO
6396 025066 002470                       WRFROM   ;TEST DATA STARTS FROM WRFROM
6397 025070 000400                       256.    ;256. WORDS TO BE COMPARED
6398 025072 025076                       3$      ;RETURN TO 3$ ON ERROR
6399 025074 025102                       4$      ;RETURN TO 4$ ON NO ERROR
6400                                     ;
6401                                     ;
6402 025076 104036 3$: ERROR 36          ;WRITE DATA COMMAND CHANGED
6403 025100 000207                       RTS     PC  ;WRITE FROM BUFFER
6404                                     ;
6405                                     ;*NOW A READ DATA COMMAND WILL BE GIVEN
6406                                     ;
6407                                     ;*FILL READ INTO BUFFER WITH 200 ZEROS AND 56 OF 377
6408                                     ;
6409 025102 4$:
6410                                     ;
6411 025102 004737 041524                JSR    PC,@#CLDISK    ;SET R1-RHCS1, R2-RHCS2
6412                                     ;R3-RHDS1, R4-RHER1
6413                                     ;GIVE RH-11 INITIALIZE
6414                                     ;SETUP UNIT NUMBER
6415 025106 004037 041374                JSR    RO,@#CLAREA    ;CLEAR 200. WORDS, FROM REINTO
6416 025112 003534                       REINTO   ;STARTING FROM REINTO
6417 025114 000310                       200.    ;200. WORDS
6418 025116 000000                       0       ;FILL WITH 0
6419                                     ;
6420 025120 004037 041374                JSR    RO,@#CLAREA    ;CLEAR 56. WORDS, FROM REINTO+<200.*2>
6421 025124 004354                       REINTO+<200.*2>      ;STARTING FROM REINTO+<200.*2>
6422 025126 000070                       56.     ;56. WORDS
6423 025130 000377                       377     ;FILL WITH 377
6424                                     ;
6425                                     ;
6426                                     ;*FILL WRITE FROM BUFFER WITH 200 ONES AND 56 OF 377
6427                                     ;
6428 025132 004037 041374                JSR    RO,@#CLAREA    ;CLEAR 200. WORDS, FROM WRFROM
6429 025136 002470                       WRFROM   ;STARTING FROM WRFROM
6430 025140 000310                       200.    ;200. WORDS
6431 025142 177777                       -1      ;FILL WITH -1

```

6432							
6433	025144	004037	041374	JSR	RO,@#CLAREA	:	CLEAR 56. WORDS, FROM WRFROM+<200.*2>
6434	025150	003310			WRFROM+<200.*2>	:	STARTING FROM WRFROM+<200.*2>
6435	025152	000070			56.	:	56. WORDS
6436	025154	000377			377	:	FILL WITH 377
6437							
6438							
6439							
6440							
6441							
6442	025156	004037	043476	JSR	RO,@#RUN	:	SETUP TO RUN FOR DATA COMMAND
6443	025162	000000			0	:	CYLINDER 0
6444	025164	000		.BYTE	0	:	SECTOR 0
6445	025165	000		.BYTE	0	:	TRACK 0
6446	025166	177470			-200.	:	WORD COUNT = 200.
6447	025170	003534			REINTO	:	BUS ADDRESS
6448						:	STARTING ADDRESS OF DATA
6449						:	BUFFER = REINTO
6450	025172	000000			0	:	DO NOT INHIBIT BUS ADDRESS INCREMENT
6451	025174	014000			ECI:FMT22	:	16 BITS PER WORD FORMAT
6452						:	INHIBIT ECC CORRECTION
6453						:	DO NOT INHIBIT HEADER COMPARE
6454	025176	002446			READAT	:	GET READY TO DO A READAT
6455						:	READ DATA WITH 70 IN RHCS1
6456							
6457							
6458							
6459							
6460	025200	004037	041672	JSR	RO,@#SAVER	:	SAVE REGISTERS
6461	025204	002272			RHWC	:	RHWC IS THE FIRST REGISTER SAVED
6462	025206	004612			SAVERE	:	STARTING ADDRESS OF WHERE
6463						:	THE REGISTERS ARE SAVED
6464	025210	000022			18.	:	NUMBER OF REGISTERS
6465						:	SAVED = 18.
6466							
6467	025212	004737	041604	JSR	PC,@#CHFCKT	:	CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
6468						:	AND THAT NO STATUS BITS ARE STUCK - 1
6469	025216	104401	067033	TYPE	,CPHALT	:	CANNOT CONTINUE TESTING IF ANY OF
6470						:	THE FIRST SET OF BITS DON'T = 1
6471	025222	000000			HALT	:	STOP
6472							
6473	025224	013777	004606 155034	MOV	@#RP4VEC,@RPVEC	:	SET RP04 VECTOR ADDRESS
6474						:	TO 'TIME1' IF P-CLOCK IS PRESENT
6475						:	OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
6476						:	'TIME' WILL ONLY SAVE
6477						:	CURRENT CYLINDER ADDRESS
6478						:	AND LOOK AHEAD REGISTERS
6479							
6480							
6481	025232	013746	002446	MOV	@#READAT,-(SP)	:	GET READY TO MOVE COMMAND
6482	025236	052716	000101	BIS	#GO!IE,(SP)	:	GET READY TO SET 'GO' AND
6483						:	ENABLE INTERRUPT
6484	025242	012677	155032	MOV	(SP)+,@RHCS1	:	GO WITH
6485						:	70 IN RHCS1 FOR READ DATA
6486						:	WITH INTERRUPT ENABLED
6487	025246	011100		MOV	@R1,RO	:	SAVE RHCS1 DURING ABOVE OPERATION

```
6488 025250 011305      MOV      @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
6489
6490
6491 025252 104413      WAT                      ;WAIT FOR RDY BIT TO SET
6492 025254 002300      RHCS1                   ;WAIT FOR RHCS1 REGISTER
6493 025256 000200      RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
6494 025260 001614      908.                   ;ALLOW 9080 MICRO SECONDS
6495 025262 001507      839.                   ;RDY MUST SET BETWEEN
6496                                     ;690 AND 17470 MICRO SECONDS
6497
6498                                     ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
6499                                     ;*R0 AND R5 IMMEDIATELY AFTER GO
6500
6501 025264 013746 002446  MOV      @#READAT,-(SP)  ;SAVE COMMAND
6502 025270 052716 004101  BIS      #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
6503 025274 011637 001124  MOV      (SP),@#SGDDAT  ;SAVE FOR PRINTOUT
6504 025300 022600      CMP      (SP)+,R0      ;DURING ABOVE OPERATION ONLY IE.GO.DVA
6505                                     ;AND COMMAND SHOULD BE SET
6506 025302 001405      BEQ      67$           ;BRANCH IF GOOD
6507 025304 010037 001126  MOV      R0,@#SBDDAT    ;BAD DATA
6508 025310 010137 004600  MOV      R1,@#REGADR    ;FAILING REGISTER RHCS1
6509 025314 104021      ERROR    21           ;DURING ABOVE OPERATION ONLY
6510                                     ;COMMAND AND IE!GO!DVA SHOULD BE SET
6511 025316 012746 010500  67$:  MOV      #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
6512 025322 011637 001124  MOV      (SP),@#SGDDAT  ;SAVE FOR PRINTOUT
6513 025326 022605      CMP      (SP)+,R5      ;DURING ABOVE OPERATION ONLY MOL!DPR.VV
6514                                     ;SHOULD BE SET
6515 025330 001405      BEQ      69$           ;BRANCH IF GOOD
6516 025332 010537 001126  MOV      R5,@#SBDDAT    ;BAD DATA
6517 025336 010337 004600  MOV      R3,@#REGADR    ;FAILING REGISTER RHDS1
6518 025342 104063      ERROR    63           ;DURING ABOVE OPERATION ONLY
6519                                     ;MOL!DPR!VV SHOULD BE SET
6520 025344      69$:
6521
6522                                     ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
6523
6524 025344 004037 041426  JSR      R0,@#FILLRE    ;MOV 0 INTO SAVED RHWC
6525 025350 002272      RHWC                   ;SAVED REGISTER TO CHANGE
6526 025352 000000      0                       ;DATA
6527 025354 004037 041426  JSR      R0,@#FILLRE    ;MOV REINTO+<200.*2> INTO SAVED RHBA
6528 025360 002274      RHBA                   ;SAVED REGISTER TO CHANGE
6529 025362 004354      REINTO+<200.*2>        ;DATA
6530 025364 004037 041426  JSR      R0,@#FILLRE    ;MOV 1 INTO SAVED RHDST
6531 025370 002304      RHDST                   ;SAVED REGISTER TO CHANGE
6532 025372 000001      1                       ;DATA
6533
6534                                     ;*COMPARE REGISTERS BEFORE READ DATA COMMAND
6535                                     ;*WITH REGISTERS AFTER COMMAND
6536
6537
6538 025374 004037 042512  JSR      R0,@#COMREG    ;COMPARE SAVED REGISTERS WITH
6539                                     ;PRESENT VALUE
6540 025400 004612      SAVERE                   ;GOOD DATA SAVED IN 'SAVERE'
6541 025402 002354      WC                       ;TEST DATA STARTING FROM 'RHWC'
6542 025404 000022      18.                     ;18. REGISTERS TO BE COMPARED
6543 025406 025412      5$                       ;RETURN TO 5$ ON ERROR
```

CZRJID0, RP04/5/6 FCTNL CTLR1
CZRJID.P11 28-MAR-79 09:03

MACY11 30A(1052) 25-MAY-79 10:30 PAGE 139
T35 WRITE/READ DATA (1'S & 125252)

SEQ 0138

```

6544 025410 025416          6$          ;RETURN TO 6$ ON NO ERROR
6545
6546 025412 104033        5$:  ERROR 33          ;READ DATA CAUSED IMPROPER
6547 025414 000207        RTS    PC          ;REGISTER CHANGE
6548                                ;GOOD DATA GIVES WHAT SHOULD BE THERE
6549                                ;RECEIVED DATA GIVES WHAT WAS THERE
6550                                ;AFTER COMMAND
6551
6552                                ;*NOW READ INTO BUFFER IS CHECKED FOR GOOD READ
6553
6554 025416          6$:
6555
6556 025416 004037 043542  JSR    RO,@#COMPAR  ;COMPARE TWO BLOCKS OF MEMORY
6557 025422 002470        WRFROM          ;GOOD DATA STARTS FROM WRFROM
6558 025424 003534        REINTO          ;TEST DATA STARTS FROM REINTO
6559 025426 000400        256.          ;256. WORDS TO BE COMPARED
6560 025430 025434        7$          ;RETURN TO 7$ ON ERROR
6561 025432 025440        10$         ;RETURN TO 10$ ON NO ERROR
6562
6563
6564 025434 104034        7$:  ERROR 34          ;INCORRECT DATA AFTER
6565 025436 000207        RTS    PC          ;WRITE DATA FOLLOWED BY A
6566                                ;READ DATA
6567 025440          10$:
6568
6569
6570

```

6571
6572
6573
6574
6575
6576
6577
6578
6579
6580
6581
6582
6583
6584
6585
6586
6587
6588
6589
6590
6591
6592
6593
6594
6595
6596
6597
6598
6599
6600
6601
6602
6603
6604
6605
6606
6607
6608
6609
6610
6611
6612
6613
6614
6615
6616
6617
6618
6619
6620
6621
6622
6623
6624
6625
6626

*TEST 36 WRITE/READ DATA (125252)

* THIS TEST GIVES A WRITE DATA COMMAND FOR CYLINDER 0
* TRACK 0, SECTOR 0, KEYS 0, 256 WORDS OF 125252
* THIS SECTOR IS FORMATED BY PREVIOUS TEST
* THEN READ DATA COMMAND IS GIVEN FOR 256 WORDS IN
* SAME CYLINDER, TRACK, SECTOR, KEYS
* WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH
* 256 OF 125252
* THE WRITE DATA COMMAND IS LOADED EXCEPT GO AND IE
* ALL REGISTERS ARE SAVED AND THEN GO IS GIVEN TO WRITE DATA

* THEN ALL REGISTERS ARE COMPARED TO CHECK FOR IMPROPER
* CHANGE, THEN WRITE FROM BUFFER IS CHECKED FOR NO CHANGE

* THEN READ INTO BUFFER IS FILLED WITH 256 OF ZEROS
* WRITE FROM BUFFER IS FILLED WITH
* 256 OF 125252
* THE COMMAND EXCEPT GO IS LOADED, ALL REGISTERS ARE SAVED
* GO IS GIVEN

* ALL REGISTERS ARE CHECKED
* READ DATA IS CHECKED

TST36: SCOPE
MOV #STACK,SP ;RESET STACK
MOV #36,@#TSTNM ;SAVE TEST NUMBER
JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
;R3-RHDS1, R4-RHER1
;GIVE RH-11 INITIALIZE
;SETUP UNIT NUMBER

*NOW FILL WRITE FROM BUFFER - 256 OF 125252
JSR RO,@#CLAREA ;CLEAR 256. WORDS, FROM WRFROM
WRFROM ;STARTING FROM WRFROM
256. ;256. WORDS
125252 ;FILL WITH 125252

*NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA AS
*WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
*CAN BE MADE TO DETERMINE THAT WRITE DID NOT CHANGE BUFFER

JSR RO,@#CLAREA ;CLEAR 256. WORDS, FROM REINTO
REINTO ;STARTING FROM REINTO
256. ;256. WORDS
125252 ;FILL WITH 125252

*NOW WRITE DATA COMMAND WILL BE LOADED

025440 000004
025442 012706 001000
025446 012737 000036 004604
025454 004737 041524
025460 004037 041374
025464 002470
025466 000400
025470 125252
025472 004037 041374
025476 003534
025500 000400
025502 125252

```

6627
6628
6629 025504 004037 043476 JSR RO,@#RUN ;SETUP TO RUN FOR DATA COMMAND
6630 025510 000000 0 ;CYLINDER 0
6631 025512 000 ;SECTOR 0
6632 025513 000 ;TRACK 0
6633 025514 177400 ;WORD COUNT = 256.
6634 025516 002470 WRFROM ;BUS ADDRESS
6635 ;STARTING ADDRESS OF DATA
6636 ;BUFFER = WRFROM
6637 025520 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
6638 025522 010000 FMT22 ;16 BITS PER WORD FORMAT
6639 ;DO NOT INHIBIT ECC CORRECTION
6640 ;DO NOT INHIBIT HEADER COMPARE
6641 025524 002442 WRIDAT ;GET READY TO DO A WRIDAT
6642 ;WRITE DATA WITH 60 IN RHCS1
6643
6644
6645 ;*NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE DATA
6646
6647 025526 004037 041672 JSR RO,@#SAVER ;SAVE REGISTERS
6648 025532 002272 RHWC ;RHWC IS THE FIRST REGISTER SAVED
6649 025534 004612 SAVERE ;STARTING ADDRESS OF WHERE
6650 ;THE REGISTERS ARE SAVED
6651 025536 000022 18. ;NUMBER OF REGISTERS
6652 ;SAVED = 18.
6653
6654 025540 004737 041604 JSR PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
6655 ;AND THAT NO STATUS BITS ARE STUCK = 1
6656 025544 104401 067033 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF ANY OF
6657 ;THE FIRST SET OF BITS DON'T - 1
6658 025550 000000 HALT ;STOP
6659
6660 025552 013777 004606 154506 MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
6661 ;TO 'TIME1' IF P-CLOCK IS PRESENT
6662 ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
6663 ;'TIME' WILL ONLY SAVE
6664 ;CURRENT CYLINDER ADDRESS
6665 ;AND LOOK AHEAD REGISTERS
6666
6667
6668 025560 013746 002442 MOV @#WRIDAT,-(SP) ;GET READY TO MOVE COMMAND
6669 025564 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET 'GO' AND
6670 ;ENABLE INTERRUPT
6671 025570 012677 154504 MOV (SP)+,@RHCS1 ;GO WITH
6672 ;60 IN RHCS1 FOR WRITE DATA
6673 ;WITH INTERRUPT ENABLED
6674 025574 011100 MOV @R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
6675 025576 011305 MOV @R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
6676
6677 ;*ONE REVOLUTION=16670 MICROSEC, ONE SECTOR=760 MICROSEC
6678
6679
6680 025600 104413 WAT ;WAIT FOR RDY BIT TO SET
6681 025602 002300 RHCS1 ;WAIT FOR RHCS1 REGISTER
6682 025604 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
    
```

```
6683 025606 001614          908.          ;ALLOW 9080 MICRO SECONDS
6684 025610 001507          839.          ;RDY MUST SET BETWEEN
6685                                     ;690 AND 17470 MICRO SECONDS
6686
6687                                     ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
6688                                     ;*RO AND R5 IMMEDIATELY AFTER GO
6689
6690 025612 013746 002442     MOV    @#WRIDAT,-(SP)    ;SAVE COMMAND
6691 025616 052716 004101     BIS    #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
6692 025622 011637 001124     MOV    (SP),@#SGDDAT   ;SAVE FOR PRINTOUT
6693 025626 022600          CMP    (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA
6694                                     ;AND COMMAND SHOULD BE SET
6695 025630 001405          BEQ    64$             ;BRANCH IF GOOD
6696 025632 010037 001126     MOV    R0,@#SBDDAT     ;BAD DATA
6697 025636 010137 004600     MOV    R1,@#REGADR     ;FAILING REGISTER RHCS1
6698 025642 104021          ERROR  21             ;DURING ABOVE OPERATION ONLY
6699                                     ;COMMAND AND IE!GO!DVA SHOULD BE SET
6700 025644 012746 010500     64$: MOV    #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
6701 025650 011637 001124     MOV    (SP),@#SGDDAT   ;SAVE FOR PRINTOUT
6702 025654 022605          CMP    (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
6703                                     ;SHOULD BE SET
6704 025656 001405          BEQ    66$             ;BRANCH IF GOOD
6705 025660 010537 001126     MOV    R5,@#SBDDAT     ;BAD DATA
6706 025664 010337 004600     MOV    R3,@#REGADR     ;FAILING REGISTER RHDS1
6707 025670 104063          ERROR  63             ;DURING ABOVE OPERATION ONLY
6708                                     ;MOL!DPR!VV SHOULD BE SET
6709 025672          66$:
6710
6711                                     ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
6712
6713 025672 004037 041426     JSR    R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
6714 025676 002272          RHWC                  ;SAVED REGISTER TO CHANGE
6715 025700 000000          0                      ;DATA
6716 025702 004037 041426     JSR    R0,@#FILLRE     ;MOV WRFROM+<256.*2> INTO SAVED RHBA
6717 025706 002274          RHBA                  ;SAVED REGISTER TO CHANGE
6718 025710 003470          WRFROM+<256.*2>       ;DATA
6719 025712 004037 041426     JSR    R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
6720 025716 002304          RHDST                 ;SAVED REGISTER TO CHANGE
6721 025720 000001          1                      ;DATA
6722
6723                                     ;*NOW COMPARE REGISTERS BEFORE WRITE DATA WITH REGISTERS
6724                                     ;*AFTER COMMAND
6725
6726
6727 025722 004037 042512     JSR    R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
6728                                     ;PRESENT VALUE
6729 025726 004612          SAVERE                ;GOOD DATA SAVED IN 'SAVERE'
6730 025730 002354          WC                    ;TEST DATA STARTING FROM 'RHWC'
6731 025732 000022          18.                   ;18. REGISTERS TO BE COMPARED
6732 025734 025740          1$                    ;RETURN TO 1$ ON ERROR
6733 025736 025744          2$                    ;RETURN TO 2$ ON NO ERROR
6734
6735 025740 104035     1$: ERROR  35          ;WRITE DATA COMMAND CAUSED
6736 025742 000207          RTS    PC             ;IMPROPER REGISTER CHANGE
6737                                     ;GOOD DATA GIVES WHAT SHOULD
6738                                     ;BE
```



```

6795 026036 002446 READAT ;GET READY TO DO A READAT
6796 ;READ DATA WITH 70 IN RHCS1
6797
6798
6799 ;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ DATA COMMAND
6800
6801 026040 004037 041672 JSR RO,@#SAVER ;SAVE REGISTERS
6802 026044 002272 RHWC ;RHWC IS THE FIRST REGISTER SAVED
6803 026046 004612 SAVERE ;STARTING ADDRESS OF WHERE
6804 ;THE REGISTERS ARE SAVED
6805 026050 000022 18. ;NUMBER OF REGISTERS
6806 ;SAVED = 18.
6807
6808 026052 004737 041604 JSR PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
6809 ;AND THAT NO STATUS BITS ARE STUCK = 1
6810 026056 104401 067033 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF ANY OF
6811 ;THE FIRST SET OF BITS DON'T = 1
6812 026062 000000 HALT ;STOP
6813
6814 026064 013777 004606 154174 MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
6815 ;TO 'TIME1' IF P-CLOCK IS PRESENT
6816 ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
6817 ;'TIME' WILL ONLY SAVE
6818 ;CURRENT CYLINDER ADDRESS
6819 ;AND LOOK AHEAD REGISTERS
6820
6821
6822 026072 013746 002446 MOV @#READAT,-(SP) ;GET READY TO MOVE COMMAND
6823 026076 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET 'GO' AND
6824 ;ENABLE INTERRUPT
6825 026102 012677 154172 MOV (SP)+,@RHCS1 ;GO WITH
6826 ;70 IN RHCS1 FOR READ DATA
6827 ;WITH INTERRUPT ENABLED
6828 026106 011100 MOV @R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
6829 026110 011305 MOV @R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
6830
6831
6832 026112 104413 WAT ;WAIT FOR RDY BIT TO SET
6833 026114 002300 RHCS1 ;WAIT FOR RHCS1 REGISTER
6834 026116 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
6835 026120 001614 908. ;ALLOW 9080 MICRO SECONDS
6836 026122 001507 839. ;RDY MUST SET BETWEEN
6837 ;690 AND 17470 MICRO SECONDS
6838
6839 ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
6840 ;*R0 AND R5 IMMEDIATELY AFTER GO
6841
6842 026124 013746 002446 MOV @#READAT,-(SP) ;SAVE COMMAND
6843 026130 052716 004101 BIS #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
6844 026134 011637 001124 MOV (SP),@#SGDDAT ;SAVE FOR PRINTOUT
6845 026140 022600 CMP (SP)+,R0 ;DURING ABOVE OPERATION ONLY ;E!GO.DVA
6846 ;AND COMMAND SHOULD BE SET
6847 026142 001405 BEQ 67$ ;BRANCH IF GOOD
6848 026144 010037 001126 MOV R0,@#SBDDAT ;BAD DATA
6849 026150 010137 004600 MOV R1,@#REGADR ;FAILING REGISTER RHCS1
6850 026154 104021 ERROR 21 ;DURING ABOVE OPERATION ONLY

```

```

6851
6852 026156 012746 010500      67$:  MOV    #MOL!DPR!VV, -(SP)      ;COMMAND AND IE!GO!DVA SHOULD BE SET
6853 026162 011637 001124      MOV    (SP), @#$GDDAT      ;SAVE BITS SET DURING OPERATION IN RHDS1
6854 026166 022605              CMP    (SP)+, R5          ;SAVE FOR PRINTOUT
6855                                ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
6856 026170 001405              BEQ    69$                ;SHOULD BE SET
6857 026172 010537 001126      MOV    R5, @#$BDDAT      ;BRANCH IF GOOD
6858 026176 010337 004600      MOV    R3, @#$REGADR     ;BAD DATA
6859 026202 104063              ERROR  63                ;FAILING REGISTER RHDS1
6860                                ;DURING ABOVE OPERATION ONLY
6861 026204      69$:              ;MOL!DPR!VV SHOULD BE SET
6862
6863                                ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
6864
6865 026204 004037 041426      JSR    RO, @#$FILLRE     ;MOV 0 INTO SAVED RHWC
6866 026210 002272              RHWC                      ;SAVED REGISTER TO CHANGE
6867 026212 000000              0                          ;DATA
6868 026214 004037 041426      JSR    RO, @#$FILLRE     ;MOV REINTO+<256.*2> INTO SAVED RHBA
6869 026220 002274              RHBA                      ;SAVED REGISTER TO CHANGE
6870 026222 004534              REINTO+<256.*2>          ;DATA
6871 026224 004037 041426      JSR    RO, @#$FILLRE     ;MOV 1 INTO SAVED RHDST
6872 026230 002304              RHDST                     ;SAVED REGISTER TO CHANGE
6873 026232 000001              1                          ;DATA
6874
6875                                ;*COMPARE REGISTERS BEFORE READ DATA COMMAND
6876                                ;*WITH REGISTERS AFTER COMMAND
6877
6878
6879 026234 004037 042512      JSR    RO, @#$COMREG     ;COMPARE SAVED REGISTERS WITH
6880                                ;PRESENT VALUE
6881 026240 004612              SAVERE                    ;GOOD DATA SAVED IN 'SAVERE'
6882 026242 002354              WC                        ;TEST DATA STARTING FROM 'RHWC'
6883 026244 000022              18.                       ;18. REGISTERS TO BE COMPARED
6884 026246 026252              5$                        ;RETURN TO 5$ ON ERROR
6885 026250 026256              6$                        ;RETURN TO 6$ ON NO ERROR
6886
6887 026252 104033      5$:  ERROR  33              ;READ DATA CAUSED IMPROPER
6888 026254 000207      RTS    PC                  ;REGISTER CHANGE
6889                                ;GOOD DATA GIVES WHAT SHOULD BE THE
6890                                ;RECEIVED DATA GIVES WHAT WAS THERE
6891                                ;AFTER COMMAND
6892
6893                                ;*NOW READ INTO BUFFER IS CHECKED FOR GOOD READ
6894
6895 026256      6$:
6896
6897 026256 004037 043542      JSR    RO, @#$COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
6898 026262 002470              WRFROM                    ;GOOD DATA STARTS FROM WRFROM
6899 026264 003534              REINTO                     ;TEST DATA STARTS FROM REINTO
6900 026266 000400              256.                      ;256. WORDS TO BE COMPARED
6901 026270 026274              7$                        ;RETURN TO 7$ ON ERROR
6902 026272 026300              10$                       ;RETURN TO 10$ ON NO ERROR
6903
6904
6905 026274 104034      7$:  ERROR  34              ;INCORRECT DATA AFTER
6906 026276 000207      RTS    PC                  ;WRITE DATA FOLLOWED BY A
    
```

CZRJID, RP04/5/6 FCTNL CTLR1
CZRJID.P11 28-MAR-79 09:03

MACY11 30A(1052) 25-MAY-79 10:30 PAGE 146
T36 WRITE/READ DATA (125252)

C 12

SEQ 0145

6907
6908 026300
6909

108:

;READ DATA

6910
6911
6912
6913
6914
6915
6916
6917
6918
6919
6920
6921
6922
6923
6924
6925
6926
6927
6928
6929
6930
6931
6932
6933
6934
6935
6936
6937
6938
6939
6940
6941
6942
6943
6944
6945
6946
6947
6948
6949
6950
6951
6952
6953
6954
6955
6956
6957
6958
6959
6960
6961
6962
6963
6964
6965

: *TEST 37 WRITE/READ DATA (052525)

: * THIS TEST GIVES A WRITE DATA COMMAND FOR CYLINDER 0
: * TRACK 0, SECTOR 0, KEYS 0, 200 WORDS OF 052525
: * THIS SECTOR IS FORMATED BY PREVIOUS TEST
: * THEN READ DATA COMMAND IS GIVEN FOR 256 WORDS IN
: * SAME CYLINDER, TRACK, SECTOR, KEYS
: * WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH
: * 200 OF 052525 AND 56 OF 377
: * THE WRITE DATA COMMAND IS LOADED EXCEPT GO AND IE
: * ALL REGISTERS ARE SAVED AND THEN GO IS TO WRITE DATA

: * THEN ALL REGISTERS ARE COMPARED TO CHECK FOR IMPROPER
: * CHANGE, THEN WRITE FROM BUFFER IS CHECKED FOR NO CHANGE

: * THEN READ INTO BUFFER IS FILLED WITH 200 IF ZEROS
: * AND 56 ALL ONES, WRITE FROM BUFFER IS FILLED WITH 200
: * WORDS OF 52525 AND 56 WORDS OF 0
: * THE COMMAND EXCEPT GO IS LOADED, ALL REGISTERS ARE SAVED
: * GO IS GIVEN

: * ALL REGISTER ARE CHECKED
: * READ DATA IS CHECKED

TST37: SCOPE

MOV #STACK,SP ;RESET STACK
MOV #37,@#TSTNM ;SAVE TEST NUMBER

JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
;R3-RHDS1, R4-RHER1
;GIVE RH-11 INITIALIZE
;SETUP UNIT NUMBER

: *NOW FILL WRITE FROM BUFFER-200 OF 52525 AND 56 OF 377

JSR RO,@#CLAREA ;CLEAR 200. WORDS, FROM WRFROM
WRFROM ;STARTING FROM WRFROM
200. ;200. WORDS
52525 ;FILL WITH 52525

JSR RO,@#CLAREA ;CLEAR 56. WORDS, FROM WRFROM+<200.*2>
WRFROM+<200.*2> ;STARTING FROM WRFROM+<200.*2>
56. ;56. WORDS
377 ;FILL WITH 377

: *NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA AS
: *WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
: *CAN BE MADE TO DETERMINE THAT WRITE DID NOT CHANGE BUFFER

JSR RO,@#CLAREA ;CLEAR 200. WORDS, FROM REINTO
REINTO ;STARTING FROM REINTO

026300 000004
026302 012706 001000
026306 012737 000037 004604
026314 004737 041524
026320 004037 041374
026324 002470
026326 000310
026330 052525
026332 004037 041374
026336 003310
026340 000070
026342 000377
026344 004037 041374
026350 003534

```

6966 026352 000310          200.          ;200. WORDS
6967 026354 052525          52525         ;FILL WITH 52525
6968
6969 026356 004037 041374  JSR      RO,@#CLAREA  ;CLEAR 56. WORDS, FROM REINTO+<200.*2>
6970 026362 004354          REINTO+<200.*2>    ;STARTING FROM REINTO+<200.*2>
6971 026364 000070          56.           ;56. WORDS
6972 026366 000377          377           ;FILL WITH 377
6973
6974
6975          ;*NOW WRITE DATA COMMAND WILL BE LOADED
6976
6977
6978 026370 004037 043476  JSR      RO,@#RUN     ;SETUP TO RUN FOR DATA COMMAND
6979 026374 000000          0             ;CYLINDER 0
6980 026376 000          .BYTE 0         ;SECTOR 0
6981 026377 000          .BYTE 0         ;TRACK 0
6982 026400 177470          -200.        ;WORD COUNT = 200.
6983 026402 002470          WRFROM       ;BUS ADDRESS
6984          ;STARTING ADDRESS OF DATA
6985          ;BUFFER = WRFROM
6986 026404 000000          0             ;DO NOT INHIBIT BUS ADDRESS INCREMENT
6987 026406 010000          FMT22        ;16 BITS PER WORD FORMAT
6988          ;DO NOT INHIBIT ECC CORRECTION
6989          ;DO NOT INHIBIT HEADER COMPARE
6990 026410 002442          WRIDAT       ;GET READY TO DO A WRIDAT
6991          ;WRITE DATA WITH 60 IN RHCS1
6992
6993
6994          ;*NOW SAVE REGISTER FOR COMPARISON AFTER WRITE DATA
6995
6996 026412 004037 041672  JSR      RO,@#SAVER   ;SAVE REGISTERS
6997 026416 002272          RHWC         ;RHWC IS THE FIRST REGISTER SAVED
6998 026420 004612          SAVERE       ;STARTING ADDRESS OF WHERE
6999          ;THE REGISTERS ARE SAVED
7000 026422 000022          18.         ;NUMBER OF REGISTERS
7001          ;SAVED = 18.
7002
7003 026424 004737 041604  JSR      PC,@#CHECKT  ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
7004          ;AND THAT NO STATUS BITS ARE STUCK = 1
7005 026430 104401 067033  TYPE     ,CPHALT     ;CANNOT CONTINUE TESTING IF ANY OF
7006          ;THE FIRST SET OF BITS DON'T = 1
7007 026434 000000          HALT        ;STOP
7008
7009 026436 013777 004606 153622 MOV      @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
7010          ;TO 'TIME1' IF P-CLOCK IS PRESENT
7011          ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
7012          ;'TIME' WILL ONLY SAVE
7013          ;CURRENT CYLINDER ADDRESS
7014          ;AND LOOK AHEAD REGISTERS
7015
7016
7017 026444 013746 002442  MOV      @#WRIDAT,-(SP) ;GET READY TO MOVE COMMAND
7018 026450 052716 000101  BIS      #GO.IE,(SP)  ;GET READY TO SET 'GO' AND
7019          ;ENABLE INTERRUPT
7020 026454 012677 153620  MOV      (SP)+,@RHCS1 ;GO WITH
7021          ;60 IN RHCS1 FOR WRITE DATA
  
```

```

7022                                     ;WITH INTERRUPT ENABLED
7023 026460 011100 MOV @R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
7024 026462 011305 MOV @R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
7025
7026 ;*ONE REVOLUTION=16670 MICROSEC, ONE SECTOR=760 MICROSEC
7027
7028
7029 026464 104413 WAT ;WAIT FOR RDY BIT TO SET
7030 026466 002300 RHCS1 ;WAIT FOR RHCS1 REGISTER
7031 026470 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
7032 026472 001614 908. ;ALLOW 9080 MICRO SECONDS
7033 026474 001507 839. ;RDY MUST SET BETWEEN
7034 ;690 AND 17470 MICRO SECONDS
7035
7036 ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
7037 ;*RO AND R5 IMMEDIATELY AFTER GO
7038
7039 026476 013746 002442 MOV @#WRIDAT,-(SP) ;SAVE COMMAND
7040 026502 052716 004101 BIS #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
7041 026506 011637 001124 MOV (SP),@#SGDDAT ;SAVE FOR PRINTOUT
7042 026512 022600 CMP (SP)+,R0 ;DURING ABOVE OPERATION ONLY IE.GO!DVA
7043 ;AND COMMAND SHOULD BE SET
7044 026514 001405 BEQ 64$ ;BRANCH IF GOOD
7045 026516 010037 001126 MOV R0,@#SBDDAT ;BAD DATA
7046 026522 010137 004600 MOV R1,@#REGADR ;FAILING REGISTER RHCS1
7047 026526 104021 ERROR 21 ;DURING ABOVE OPERATION ONLY
7048 ;COMMAND AND IE.GO!DVA SHOULD BE SET
7049 026530 012746 010500 64$: MOV #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
7050 026534 011637 001124 MOV (SP),@#SGDDAT ;SAVE FOR PRINTOUT
7051 026540 022605 CMP (SP)+,R5 ;DURING ABOVE OPERATION ONLY MOL.DPR.VV
7052 ;SHOULD BE SET
7053 026542 001405 BEQ 66$ ;BRANCH IF GOOD
7054 026544 010537 001126 MOV R5,@#SBDDAT ;BAD DATA
7055 026550 010337 004600 MOV R3,@#REGADR ;FAILING REGISTER RHDS1
7056 026554 104063 ERROR 63 ;DURING ABOVE OPERATION ONLY
7057 ;MOL!DPR!VV SHOULD BE SET
7058 026556 66$:
7059
7060 ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
7061
7062 026556 004037 041426 JSR R0,@#FILLRE ;MOV 0 INTO SAVED RHWC
7063 026562 002272 RHWC ;SAVED REGISTER TO CHANGE
7064 026564 000000 0 ;DATA
7065 026566 004037 041426 JSR R0,@#FILLRE ;MOV WRFROM+<200.*2> INTO SAVED R4BA
7066 026572 002274 RHBA ;SAVED REGISTER TO CHANGE
7067 026574 003310 WRFROM+<200.*2> ;DATA
7068 026576 004037 041426 JSR R0,@#FILLRE ;MOV 1 INTO SAVED RHDST
7069 026602 002304 RHDST ;SAVED REGISTER TO CHANGE
7070 026604 000001 1 ;DATA
7071
7072 ;*NOW COMPARE REGISTERS BEFORE WRITE DATA WITH REGISTERS
7073 ;*AFTER COMMAND
7074
7075
7076 026606 004037 042512 JSR R0,@#COMREG ;COMPARE SAVED REGISTERS WITH
7077 ;PRESENT VALUE

```

```

7078 026612 004612          SAVERE          ;GOOD DATA SAVED IN 'SAVERE'
7079 026614 002354          WC              ;TEST DATA STARTING FROM 'RHWC'
7080 026616 000022          18.            ;18. REGISTERS TO BE COMPARED
7081 026620 026624          1$             ;RETURN TO 1$ ON ERROR
7082 026622 026630          2$             ;RETURN TO 2$ ON NO ERROR
7083
7084 026624 104035          1$:           ERROR 35          ;WRITE DATA COMMAND CAUSED
7085 026626 000207          RTS      PC      ;IMPROPER REGISTER CHANGE
7086                                     ;GOOD DATA GIVES WHAT SHOULD
7087                                     ;BE
7088                                     ;RECEIVED DATA GIVES WHAT WAS
7089                                     ;THERE AFTER COMMAND
7090
7091                                     ;*NOW WRITE FROM BUFFER WILL BE CHECKED FOR NO CHANGE
7092
7093 026630          2$:
7094
7095 026630 004037 043542     JSR      RO,@#COMPAR ;COMPARE TWO BLOCKS OF MEMORY
7096 026634 003534          REINTO         ;GOOD DATA STARTS FROM REINTO
7097 026636 002470          WRFROM         ;TEST DATA STARTS FROM WRFROM
7098 026640 000400          256.          ;256. WORDS TO BE COMPARED
7099 026642 026646          3$             ;RETURN TO 3$ ON ERROR
7100 026644 026652          4$             ;RETURN TO 4$ ON NO ERROR
7101
7102
7103 026646 104036          3$:           ERROR 36          ;WRITE DATA COMMAND CHANGED
7104 026650 000207          RTS      PC      ;WRITE FROM BUFFER
7105
7106                                     ;*NOW A READ DATA COMMAND WILL BE GIVEN
7107
7108                                     ;*FILL READ INTO BUFFER WITH 200 ZEROS AND 56 OF ALL ONES
7109
7110 026652          4$:
7111
7112 026652 004737 041524     JSR      PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
7113                                     ;R3-RHDS1, R4-RHER1
7114                                     ;GIVE RH-11 INITIALIZE
7115                                     ;SETUP UNIT NUMBER
7116 026656 004037 041374     JSR      RO,@#CLAREA ;CLEAR 200. WORDS, FROM REINTO
7117 026662 003534          REINTO         ;STARTING FROM REINTO
7118 026664 000310          200.          ;200. WORDS
7119 026666 000000          0              ;FILL WITH 0
7120
7121 026670 004037 041374     JSR      RO,@#CLAREA ;CLEAR 56. WORDS, FROM REINTO+<200.*2>
7122 026674 004354          REINTO+<200.*2> ;STARTING FROM REINTO+<200.*2>
7123 026676 000070          56.            ;56. WORDS
7124 026700 000377          377           ;FILL WITH 377
7125
7126
7127                                     ;*FILL WRITE FROM BUFFER WITH 200 OF 52525 AND 56 OF 0
7128
7129 026702 004037 041374     JSR      RO,@#CLAREA ;CLEAR 200. WORDS, FROM WRFROM
7130 026706 002470          WRFROM         ;STARTING FROM WRFROM
7131 026710 000310          200.          ;200. WORDS
7132 026712 052525          52525         ;FILL WITH 52525
7133
  
```

7134	026714	004037	041374		JSR	RO,@#CLAREA	:CLEAR 56. WORDS, FROM WRFROM+<200.*2>
7135	026720	003310			WRFROM+<200.*2>		:STARTING FROM WRFROM+<200.*2>
7136	026722	000070			56.		:56. WORDS
7137	026724	000377			377		:FILL WITH 377
7138							
7139							
7140							:*NOW FILL COMMAND
7141							
7142							
7143	026726	004037	043476		JSR	RO,@#RUN	:SETUP TO RUN FOR DATA COMMAND
7144	026732	000000			0		:CYLINDER 0
7145	026734	000		.BYTE	0		:SECTOR 0
7146	026735	000		.BYTE	0		:TRACK 0
7147	026736	177470			-200.		:WORD COUNT = 200.
7148	026740	003534			REINTO		:BUS ADDRESS
7149							:STARTING ADDRESS OF DATA
7150							:BUFFER = REINTO
7151	026742	000000			0		:DO NOT INHIBIT BUS ADDRESS INCREMENT
7152	026744	014000			EC1:FMT22		:16 BITS PER WORD FORMAT
7153							:INHIBIT ECC CORRECTION
7154							:DO NOT INHIBIT HEADER COMPARE
7155	026746	002446			READAT		:GET READY TO DO A READAT
7156							:READ DATA WITH 70 IN RHCS1
7157							
7158							
7159							:*NOW SAVE REGISTERS FOR COMPARISON AFTER READ DATA COMMAND
7160							
7161	026750	004037	041672		JSR	RO,@#SAVER	:SAVE REGISTERS
7162	026754	002272			RHWC		:RHWC IS THE FIRST REGISTER SAVED
7163	026756	004612			SAVERE		:STARTING ADDRESS OF WHERE
7164							:THE REGISTERS ARE SAVED
7165	026760	000022			18.		:NUMBER OF REGISTERS
7166							:SAVED = 18.
7167							
7168	026762	004737	041604		JSR	PC,@#CHECKT	:CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
7169							:AND THAT NO STATUS BITS ARE STUCK 1
7170	026766	104401	067033		TYPE	,CPHALT	:CANNOT CONTINUE TESTING IF ANY OF
7171							:THE FIRST SET OF BITS DON'T = 1
7172	026772	000000			HALT		:STOP
7173							
7174	026774	013777	004606	153264	MOV	@#RP4VEC,@RPVEC	:SET RP04 VECTOR ADDRESS
7175							:TO 'TIME1' IF P-CLOCK IS PRESENT
7176							:OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
7177							: 'TIME' WILL ONLY SAVE
7178							:CURRENT CYLINDER ADDRESS
7179							:AND LOOK AHEAD REGISTERS
7180							
7181							
7182	027002	013746	002446		MOV	@#READAT,-(SP)	:GET READY TO MOVE COMMAND
7183	027006	052716	000101		BIS	#GO!IE,(SP)	:GET READY TO SET 'GO' AND
7184							:ENABLE INTERRUPT
7185	027012	012677	153262		MOV	(SP)+,@RHCS1	:GO WITH
7186							:70 IN RHCS1 FOR READ DATA
7187							:WITH INTERRUPT ENABLED
7188	027016	011100			MOV	@R1,R0	:SAVE RHCS1 DURING ABOVE OPERATION
7189	027020	011305			MOV	@R3,R5	:SAVE RHDS1 DURING ABOVE OPERATION


```

7190
7191
7192 027022 104413      WAT          ;WAIT FOR RDY BIT TO SET
7193 027024 002300      RHCS1        ;WAIT FOR RHCS1 REGISTER
7194 027026 000200      RDY          ;WAIT FOR RDY BIT IN RHCS1 REGISTER
7195 027030 001614      908.        ;ALLOW 9080 MICRO SECONDS
7196 027032 001507      839.        ;RDY MUST SET BETWEEN
7197                                     ;690 AND 17470 MICRO SECONDS
7198
7199                                     ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
7200                                     ;*RO AND R5 IMMEDIATELY AFTER GO
7201
7202 027034 013746 002446  MOV    @#READAT,-(SP) ;SAVE COMMAND
7203 027040 052716 004101  BIS    #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
7204 027044 011637 001124  MOV    (SP),@#SGDDAT ;SAVE FOR PRINTOUT
7205 027050 022600      CMP    (SP)+,RO      ;DURING ABOVE OPERATION ONLY IE!GO!DVA
7206                                     ;AND COMMAND SHOULD BE SET
7207 027052 001405      BEQ    67$          ;BRANCH IF GOOD
7208 027054 010037 001126  MOV    RO,@#SBDDAT   ;BAD DATA
7209 027060 010137 004600  MOV    R1,@#REGADR   ;FAILING REGISTER RHCS1
7210 027064 104021      ERROR  21          ;DURING ABOVE OPERATION ONLY
7211                                     ;COMMAND AND IE!GO!DVA SHOULD BE SET
7212 027066 012746 010500  67$:  MOV    #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
7213 027072 011637 001124  MOV    (SP),@#SGDDAT ;SAVE FOR PRINTOUT
7214 027076 022605      CMP    (SP)+,R5     ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
7215                                     ;SHOULD BE SET
7216 027100 001405      BEQ    69$          ;BRANCH IF GOOD
7217 027102 010537 001126  MOV    R5,@#SBDDAT   ;BAD DATA
7218 027106 010337 004600  MOV    R3,@#REGADR   ;FAILING REGISTER RHDS1
7219 027112 104063      ERROR  63          ;DURING ABOVE OPERATION ONLY
7220                                     ;MOL!DPR!VV SHOULD BE SET
7221 027114      69$:
7222
7223                                     ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
7224
7225 027114 004037 041426  JSR    RO,@#FILLRE   ;MOV 0 INTO SAVED RHWC
7226 027120 002272      RHWC          ;SAVED REGISTER TO CHANGE
7227 027122 000000      0            ;DATA
7228 027124 004037 041426  JSR    RO,@#FILLRE   ;MOV REINTO+<200.*2> INTO SAVED RHBA
7229 027130 002274      RHBA          ;SAVED REGISTER TO CHANGE
7230 027132 004354      REINTO+<200.*2> ;DATA
7231 027134 004037 041426  JSR    RO,@#FILLRE   ;MOV 1 INTO SAVED RHDST
7232 027140 002304      RHDST        ;SAVED REGISTER TO CHANGE
7233 027142 000001      1            ;DATA
7234
7235                                     ;*COMPARE REGISTERS BEFORE READ DATA COMMAND
7236                                     ;*WITH REGISTERS AFTER COMMAND
7237
7238
7239 027144 004037 042512  JSR    RO,@#COMREG   ;COMPARE SAVED REGISTERS WITH
7240                                     ;PRESENT VALUE
7241 027150 004612      SAVERE        ;GOOD DATA SAVED IN 'SAVERE'
7242 027152 002354      WC           ;TEST DATA STARTING FROM 'RHWC'
7243 027154 000022      18.          ;18. REGISTERS TO BE COMPARED
7244 027156 027162      5$           ;RETURN TO 5$ ON ERROR
7245 027160 027166      6$           ;RETURN TO 6$ ON NO ERROR

```

```
7246
7247 027162 104033      5$:  ERROR 33      ; READ DATA CAUSED IMPROPER
7248 027164 000207      RTS   PC      ; REGISTER CHANGE
7249                                     ; GOOD DATA GIVES WHAT SHOULD BE THE
7250                                     ; RECEIVED DATA GIVES WHAT WAS THERE
7251                                     ; AFTER COMMAND
7252
7253                                     ; *NOW READ INTO BUFFER IS CHECKED FOR GOOD READ
7254
7255 027166      6$:
7256
7257 027166 004037 043542 JSR   RO,@#COMPAR ; COMPARE TWO BLOCKS OF MEMORY
7258 027172 002470      WRFROM ; GOOD DATA STARTS FROM WRFROM
7259 027174 003534      REINTO ; TEST DATA STARTS FROM REINTO
7260 027176 000400      256. ; 256. WORDS TO BE COMPARED
7261 027200 027204      7$ ; RETURN TO 7$ ON ERROR
7262 027202 027210      10$ ; RETURN TO 10$ ON NO ERROR
7263
7264
7265 027204 104034      7$:  ERROR 34      ; INCORRECT DATA AFTER
7266 027206 000207      RTS   PC      ; WRITE DATA FOLLOWED BY A
7267                                     ; READ DATA
7268 027210      10$:
7269
7270
7271
7272
```

```
7273 .....  
7274 :*TEST 40 WRITE/READ DATA USING UNIBUS B  
7275  
7276 :* THIS TEST USES UNIBUS B IF CONNECTED TO THE RH  
7277 :* IF UNIBUS B IS NOT CONNECTED THEN THIS TEST IS NOT PERFORMED  
7278 :* THIS TEST GIVES A WRITE DATA COMMAND FOR CYLINDER 0  
7279 :* TRACK 0, SECTOR 0, KEYS 0, 200 WORDS OF 052525  
7280 :* THIS SECTOR IS FORMATED BY PREVIOUS TEST  
7281 :* THEN READ DATA COMMAND IS GIVEN FOR 256 WORDS IN  
7282 :* SAME CYLINDER, TRACK, SECTOR, KEYS  
7283 :* THESE COMMANDS USE UNIBUS B FOR DATA  
7284 :* WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH  
7285 :* 200 OF 052525 AND 56 OF 377  
7286 :* THE WRITE DATA COMMAND IS LOADED EXCEPT GO AND IE  
7287 :* ALL REGISTERS ARE SAVED AND THEN GO IS GIVEN TO WRITE DATA  
7288  
7289  
7290 :* THEN ALL REGISTERS ARE COMPARED TO CHECK FOR IMPROPER  
7291 :* CHANGE, THEN WRITE FROM BUFFER IS CHECKED FOR NO CHANGE  
7292  
7293 :* THEN READ INTO BUFFER IS FILLED WITH 200 IF ZEROS  
7294 :* AND 56 ALL ONES, WRITE FROM BUFFER IS FILLED WITH 200  
7295 :* WORDS OF 52525 AND 56 WORDS OF 0  
7296 :* THE COMMAND EXCEPT GO IS LOADED, ALL REGISTERS ARE SAVED  
7297 :* GO IS GIVEN  
7298  
7299  
7300 :* ALL REGISTER ARE CHECKED  
7301 :* READ DATA IS CHECKED
```

```
7302 .....  
7303 :*TEST40: SCOPE  
7304 027210 000004  
7305 027212 012737 000001 001212 MOV #1,$TIMES ;;DO 1 ITERATION  
7306 027220 012706 001000 MOV #STACK,SP ;RESET STACK  
7307 027224 012737 000040 004604 MOV #40,@#TSTNM ;SAVE TEST NUMBER  
7308  
7309 027232 004737 041524 JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2  
7310 ;R3-RHDS1, R4-RHER1  
7311 ;GIVE RH-11 INITIALIZE  
7312 ;SETUP UNIT NUMBER  
7313  
7314 :*CHECK TO SEE IF THE PROGRAM IS RUNNING WITH AN RH70  
7315  
7316 027236 005737 004750 TST @#RH70 ;TEST FOR RH70 CONTROLLER  
7317 027242 001402 BEQ 30$ ;IF FLAG = 1, THIS TEST IS SKIPPED  
7318 027244 000137 030360 JMP TST41 ; JUMP TO NEXT TEST -----)  
7319 027250 30$: ;IF FLAG = 1, DO THIS TEST  
7320  
7321 027250 005037 004732 CLR @#UBUSB ;CLEAR UNIBUS INDICATOR  
7322  
7323 :*NOW FILL WRITE FROM BUFFER-200 OF 52525 AND 56 OF 377  
7324  
7325 027254 004037 041374 JSR R0,@#CLAREA ;CLEAR 200. WORDS, FROM WRFROM  
7326 027260 002470 WRFROM ;STARTING FROM WRFROM  
7327 027262 000310 200. ;200. WORDS  
7328 027264 052525 52525 ;FILL WITH 52525
```

```

7329
7330 027266 004037 041374 JSR RO,@#CLAREA ;CLEAR 56. WORDS, FROM WRFROM+<200.*2>
7331 027272 003310 WRFROM+<200.*2> ;STARTING FROM WRFROM+<200.*2>
7332 027274 000070 56. ;56. WORDS
7333 027276 000377 377 ;FILL WITH 377
7334
7335
7336 ;*NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA AS
7337 ;*WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
7338 ;*CAN BE MADE TO DETERMINE THAT WRITE DID NOT CHANGE BUFFER
7339
7340 027300 004037 041374 JSR RO,@#CLAREA ;CLEAR 200. WORDS, FROM REINTO
7341 027304 003534 REINTO ;STARTING FROM REINTO
7342 027306 000310 200. ;200. WORDS
7343 027310 052525 52525 ;FILL WITH 52525
7344
7345 027312 004037 041374 JSR RO,@#CLAREA ;CLEAR 56. WORDS, FROM REINTO+<200.*2>
7346 027316 004354 REINTO+<200.*2> ;STARTING FROM REINTO+<200.*2>
7347 027320 000070 56. ;56. WORDS
7348 027322 000377 377 ;FILL WITH 377
7349
7350
7351 ;*NOW WRITE DATA COMMAND WILL BE LOADED
7352
7353
7354 027324 004037 043476 JSR RO,@#RUN ;SETUP TO RUN FOR DATA COMMAND
7355 027330 000000 0 ;CYLINDER 0
7356 027332 000 .BYTE 0 ;SECTOR 0
7357 027333 000 .BYTE 0 ;TRACK 0
7358 027334 177470 -200. ;WORD COUNT = 200.
7359 027336 002470 WRFROM ;BUS ADDRESS
7360 ;STARTING ADDRESS OF DATA
7361 ;BUFFER = WRFROM
7362 027340 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
7363 027342 010000 FMT22 ;16 BITS PER WORD FORMAT
7364 ;DO NOT INHIBIT ECC CORRECTION
7365 ;DO NOT INHIBIT HEADER COMPARE
7366 027344 002442 WRIDAT ;GET READY TO DO A WRIDAT
7367 ;WRITE DATA WITH 60 IN RHCS1
7368
7369 027346 052777 002000 152724 BIS #PSEL,@RHCS1 ;SET PORT B
7370 ;THAT IS UNIBUS B
7371
7372 ;*NOW SAVE REGISTER FOR COMPARISON AFTER WRITE DATA
7373
7374 027354 004037 041672 JSR RO,@#SAVER ;SAVE REGISTERS
7375 027360 002272 RHWC ;RHWC IS THE FIRST REGISTER SAVED
7376 027362 004612 SAVERE ;STARTING ADDRESS OF WHERE
7377 ;THE REGISTERS ARE SAVED
7378 027364 000022 18. ;NUMBER OF REGISTERS
7379 ;SAVED = 18.
7380
7381 027366 004737 041604 JSR PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
7382 ;AND THAT NO STATUS BITS ARE STUCK = 1
7383 027372 104401 067033 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF ANY OF
7384 ;THE FIRST SET OF BITS DON'T = 1

```

```

7385 027376 000000 HALT ;STOP
7386
7387 027400 013777 004606 152660 MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
7388 ;TO 'TIME1' IF P-CLOCK IS PRESENT
7389 ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
7390 ;'TIME' WILL ONLY SAVE
7391 ;CURRENT CYLINDER ADDRESS
7392 ;AND LOOK AHEAD REGISTERS
7393
7394 027406 013746 002442 MOV @#WRIDAT,-(SP) ;GET READY TO MOVE COMMAND
7395 027412 052716 002101 BIS #GO!IE!PSEL,(SP) ;GET READY TO SET 'GO' AND
7396 ;ENABLE INTERRUPT
7397 027416 012677 152656 MOV (SP)+,@RHCS1 ;GO WITH
7398 ;60 IN RHCS1 FOR WRITE DATA
7399 ;WITH INTERRUPT ENABLED
7400 027422 011100 MOV @R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
7401 027424 011305 MOV @R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
7402
7403 ;*ONE REVOLUTION=16670 MICROSEC, ONE SECTOR=760 MICROSEC
7404
7405
7406 027426 104413 WAT ;WAIT FOR RDY BIT TO SET
7407 027430 002300 RHCS1 ;WAIT FOR RHCS1 REGISTER
7408 027432 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
7409 027434 001614 908. ;ALLOW 9080 MICRO SECONDS
7410 027436 001507 839. ;RDY MUST SET BETWEEN
7411 ;690 AND 17470 MICRO SECONDS
7412
7413 ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
7414 ;*R0 AND R5 IMMEDIATELY AFTER GO
7415
7416 027440 013746 002442 MOV @#WRIDAT,-(SP) ;SAVE COMMAND
7417 027444 052716 006101 BIS #IE!GO!DVA!PSEL,(SP) ;INCLUDE IE!GO!DVA!PSEL
7418 027450 011637 001124 MOV (SP),@#SGDDAT ;SAVE FOR PRINTOUT
7419 027454 022600 CMP (SP)+,R0 ;DURING ABOVE OPERATION ONLY IE!GO!DVA!PSEL
7420 ;AND COMMAND SHOULD BE SET
7421 027456 001405 BEQ 64$ ;BRANCH IF GOOD
7422 027460 010037 001126 MOV R0,@#SBDDAT ;BAD DATA
7423 027464 010137 004600 MOV R1,@#REGADR ;FAILING REGISTER RHCS1
7424 027470 104021 ERROR 21 ;DURING ABOVE OPERATION ONLY
7425 ;COMMAND AND IE!GO!DVA!PSEL SHOULD BE SET
7426 027472 012746 010500 64$: MOV #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
7427 027476 011637 001124 MOV (SP),@#SGDDAT ;SAVE FOR PRINTOUT
7428 027502 022605 CMP (SP)+,R5 ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
7429 ;SHOULD BE SET
7430 027504 001405 BEQ 66$ ;BRANCH IF GOOD
7431 027506 010537 001126 MOV R5,@#SBDDAT ;BAD DATA
7432 027512 010337 004600 MOV R3,@#REGADR ;FAILING REGISTER RHDS1
7433 027516 104063 ERROR 63 ;DURING ABOVE OPERATION ONLY
7434 ;MOL!DPR!VV SHOULD BE SET
7435 027520 66$:
7436
7437
7438 ;*CHECK IF NEM NON EXISTANT MEMORY IS SET
7439 ;*IF SET IT MEANS UNIBUS B IS NOT CONNECTED
7440 ;*SO THIS TEST IS NOT PERFORMED

```

```

7441
7442 027520 032777 004000 152550 BIT #NEM,@RHCS2 ;TEST NEM
7443 027526 001441 BEQ 11$ ;BRANCH IF UNIBUS B THERE
7444 027530 012737 177777 004732 MOV #-1,@#UBUSB ;UNIBUS B NOT THERE
7445 027536 104401 027544 TYPE ,68$ ;:TYPE ASCIZ STRING
7446 027542 000425 BR 67$ ;:GET OVER THE ASCIZ
7447 ;:68$: .ASCIZ <15><12>//THE RH DOES NOT HAVE UNIBUS B CONNECTED/
7448 67$:
7449 027616 104401 001223 TYPE ,SCLF
7450 027622 104401 001223 TYPE ,SCLF
7451 027626 000137 030360 JMP @#10$ ;JUMP TO NEXT TEST - NO UNIBUS B
7452 027632 11$:
7453 027632 104401 027640 TYPE ,70$ ;:TYPE ASCIZ STRING
7454 027636 000424 BR 69$ ;:GET OVER THE ASCIZ
7455 ;:70$: .ASCIZ <15><12>//THE RH DOES HAVE UNIBUS B CONNECTED/
7456 69$:
7457 027710 104401 001223 TYPE ,SCLF
7458 027714 104401 001223 TYPE ,SCLF
7459
7460 ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
7461
7462 027720 004037 041426 JSR RO,@#FILLRE ;MOV 0 INTO SAVED RHWC
7463 027724 002272 RHWC ;SAVED REGISTER TO CHANGE
7464 027726 000000 0 ;DATA
7465 027730 004037 041426 JSR RO,@#FILLRE ;MOV WRFROM+<200.*2> INTO SAVED RHBA
7466 027734 002274 RHBA ;SAVED REGISTER TO CHANGE
7467 027736 003310 WRFROM+<200.*2> ;DATA
7468 027740 004037 041426 JSR RO,@#FILLRE ;MOV 1 INTO SAVED RHDST
7469 027744 002304 RHDST ;SAVED REGISTER TO CHANGE
7470 027746 000001 1 ;DATA
7471
7472 ;*NOW COMPARE REGISTERS BEFORE WRITE DATA WITH REGISTERS
7473 ;*AFTER COMMAND
7474
7475
7476 027750 004037 042512 JSR RO,@#COMREG ;COMPARE SAVED REGISTERS WITH
7477 ;PRESENT VALUE
7478 027754 004612 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
7479 027756 002354 WC ;TEST DATA STARTING FROM 'RHWC'
7480 027760 000022 18. ;18. REGISTERS TO BE COMPARED
7481 027762 027766 1$ ;RETURN TO 1$ ON ERROR
7482 027764 027772 2$ ;RETURN TO 2$ ON NO ERROR
7483
7484 027766 104074 1$: ERROR 74 ;WHILE USING UNIBUS B
7485 ;WRITE DATA COMMAND CAUSED
7486 027770 000207 RTS PC ;IMPROPER REGISTER CHANGE
7487 ;GOOD DATA GIVES WHAT SHOULD
7488 ;BE
7489 ;RECEIVED DATA GIVES WHAT WAS
7490 ;THERE AFTER COMMAND
7491
7492 ;*NOW WRITE FROM BUFFER WILL BE CHECKED FOR NO CHANGE
7493
7494 027772 2$:
7495
7496 027772 004037 043542 JSR RO,@#COMPAR ;COMPARE TWO BLOCKS OF MEMORY

```

```
7497 027776 003534 REINTO ;GOOD DATA STARTS FROM REINTO
7498 030000 00247J WRFROM ;TEST DATA STARTS FROM WRFROM
7499 030002 000400 256. ;256. WORDS TO BE COMPARED
7500 030004 030010 3$ ;RETURN TO 3$ ON ERROR
7501 030006 030014 4$ ;RETURN TO 4$ ON NO ERROR
7502
7503
7504 030010 104075 3$: ERROR 75 ;WHILE USING UNIBUS B
7505 ;WRITE DATA COMMAND CHANGED
7506 030012 000207 RTS PC ;WRITE FROM BUFFER
7507
7508 ;*NOW A READ DATA COMMAND WILL BE GIVEN
7509 ;*FILL READ INTO BUFFER WITH 200 ZEROS AND 56 OF ALL ONES
7510
7511 030014 4$:
7512
7513 030014 004737 041524 JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
7514 ;R3-RHDS1, R4-RHER1
7515 ;GIVE RH-11 INITIALIZE
7516 ;SETUP UNIT NUMBER
7517 030020 004037 041374 JSR RO,@#CLAREA ;CLEAR 200. WORDS, FROM REINTO
7518 030024 003534 REINTO ;STARTING FROM REINTO
7519 030026 000310 200. ;200. WORDS
7520 030030 000000 0 ;FILL WITH 0
7521
7522 030032 004037 041374 JSR RO,@#CLAREA ;CLEAR 56. WORDS, FROM REINTO+<200.*2>
7523 030036 004354 REINTO+<200.*2> ;STARTING FROM REINTO+<200.*2>
7524 030040 000070 56. ;56. WORDS
7525 030042 000377 377 ;FILL WITH 377
7526
7527
7528 ;*FILL WRITE FROM BUFFER WITH 200 OF 52525 AND 56 OF 0
7529
7530 030044 004037 041374 JSR RO,@#CLAREA ;CLEAR 200. WORDS, FROM WRFROM
7531 030050 002470 WRFROM ;STARTING FROM WRFROM
7532 030052 000310 200. ;200. WORDS
7533 030054 052525 52525 ;FILL WITH 52525
7534
7535 030056 004037 041374 JSR RO,@#CLAREA ;CLEAR 56. WORDS, FROM WRFROM+<200.*2>
7536 030062 003310 WRFROM+<200.*2> ;STARTING FROM WRFROM+<200.*2>
7537 030064 000070 56. ;56. WORDS
7538 030066 000377 377 ;FILL WITH 377
7539
7540
7541 ;*NOW FILL COMMAND
7542
7543
7544 030070 004037 043476 JSR RO,@#RUN ;SETUP TO RUN FOR DATA COMMAND
7545 030074 000000 0 ;CYLINDER 0
7546 030076 000 .BYTE 0 ;SECTOR 0
7547 030077 000 .BYTE 0 ;TRACK 0
7548 030100 177470 -200. ;WORD COUNT = 200.
7549 030102 003534 REINTO ;BUS ADDRESS
7550 ;STARTING ADDRESS OF DATA
7551 ;BUFFER = REINTO
7552 030104 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
```



```

7609 030222 001405          BEQ      71$          ;BRANCH IF GOOD
7610 030224 010037 001126    MOV      RO,@#$BDDAT  ;BAD DATA
7611 030230 010137 004600    MOV      R1,@#REGADR  ;FAILING REGISTER RHCS1
7612 030234 104021          ERROR    21          ;DURING ABOVE OPERATION ONLY
7613                          ;COMMAND AND IE!GO!DVA!PSEL SHOULD BE SET
7614 030236 012746 010500    71$:   MOV      #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
7615 030242 011637 001124    MOV      (SP),@#$GDDAT ;SAVE FOR PRINTOUT
7616 030246 022605          CMP      (SP)+,R5     ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
7617                          ;SHOULD BE SET
7618 030250 001405          BEQ      73$          ;BRANCH IF GOOD
7619 030252 010537 001126    MOV      R5,@#$BDDAT  ;BAD DATA
7620 030256 010337 004600    MOV      R3,@#REGADR  ;FAILING REGISTER RHDS1
7621 030262 104063          ERROR    63          ;DURING ABOVE OPERATION ONLY
7622                          ;MOL!DPR!VV SHOULD BE SET
7623 030264          73$:
7624
7625                          ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
7626
7627 030264 004037 041426    JSR      RO,@#FILLRE  ;MOV 0 INTO SAVED RHWC
7628 030270 002272          RHWC              ;SAVED REGISTER TO CHANGE
7629 030272 000000          0                ;DATA
7630 030274 004037 041426    JSR      RO,@#FILLRE  ;MOV REINTO+<200.*2> INTO SAVED RHBA
7631 030300 002274          RHBA              ;SAVED REGISTER TO CHANGE
7632 030302 004354          REINTO+<200.*2>  ;DATA
7633 030304 004037 041426    JSR      RO,@#FILLRE  ;MOV 1 INTO SAVED RHDST
7634 030310 002304          RHDST            ;SAVED REGISTER TO CHANGE
7635 030312 000001          1                ;DATA
7636
7637                          ;*COMPARE REGISTERS BEFORE READ DATA COMMAND
7638                          ;*WITH REGISTERS AFTER COMMAND
7639
7640
7641 030314 004037 042512    JSR      RO,@#COMREG  ;COMPARE SAVED REGISTERS WITH
7642                          ;PRESENT VALUE
7643 030320 004612          SAVERE            ;GOOD DATA SAVED IN 'SAVERE'
7644 030322 002354          WC              ;TEST DATA STARTING FROM 'RHWC'
7645 030324 000022          18.             ;18. REGISTERS TO BE COMPARED
7646 030326 030332          5$              ;RETURN TO 5$ ON ERROR
7647 030330 030336          6$              ;RETURN TO 6$ ON NO ERROR
7648
7649 030332 104072          5$:   ERROR    72          ;WHILE USING UNIBUS B
7650                          ;READ DATA CAUSED IMPROPER
7651 030334 000207          RTS      PC        ;REGISTER CHANGE
7652                          ;GOOD DATA GIVES WHAT SHOULD BE THE
7653                          ;RECEIVED DATA GIVES WHAT WAS THERE
7654                          ;AFTER COMMAND
7655
7656                          ;*NOW READ INTO BUFFER IS CHECKED FOR GOOD READ
7657
7658 030336          6$:
7659
7660 030336 004037 043542    JSR      RO,@#COMPAR  ;COMPARE TWO BLOCKS OF MEMORY
7661 030342 002470          WRFROM          ;GOOD DATA STARTS FROM WRFROM
7662 030344 003534          REINTO          ;TEST DATA STARTS FROM REINTO
7663 030346 000400          256.           ;256. WORDS TO BE COMPARED
7664 030350 030354          7$              ;RETURN TO 7$ ON ERROR
  
```


7679
7680
7681
7682
7683
7684
7685
7686
7687
7688
7689
7690
7691
7692
7693
7694
7695
7696
7697
7698
7699
7700
7701
7702
7703
7704
7705
7706
7707
7708
7709
7710
7711
7712
7713
7714
7715
7716
7717
7718
7719
7720
7721
7722
7723
7724
7725
7726
7727
7728
7729
7730
7731
7732
7733
7734

```
*****  
;*TEST 41 IMPLIED SEARCH  
*****  
;* ONLY NEW ADDITION !N THIS TEST IS AN IMPLIED SEARCH  
;* A WRITE HEADER AND DATA IS GIVEN FOR MORE THAN ONE SECTOR  
;* OF WORDS  
;* WRITE HEADER AND DATA CYLINDER 0, FORMAT 16 BITS PER WORD  
;* TRACK 0, SECTOR 0, KEYS=0, NUMBER OF WORDS  
;* 266 (4 HEADER 256 DATA=0 4 HEADER 2 DATA=1  
;* THEN READ HEADER AND DATA FOR ABOVE SECTOR 1 ONLY  
;* WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH  
;* 10000,0,0,0 AND 256 OF 0, 10000,1,0,0, AND 2 OF 1  
;* THE WRITE COMMAND IS THEN LOADED INTO THE REGISTERS EXCEPT  
;* THE GO BIT, AND ALL THE REGISTERS ARE SAVED  
;* THEN GO IS GIVEN FOR WRITE HEADER AND DATA  
*****  
;* THEN ALL REGISTERS ARE COMPARED TO CHECK FOR IMPROPER CHANGE  
;* THEN WRITE FROM BUFFER IS CHECKED TO SEE THAT NOTHING CHANGED  
*****  
;* NOW FOR THE READ COMMAND READ INTO BUFFER IS FILLED  
;* WITH 377, WRITE FROM BUFFER IS FILLED WITH 10000,1,0,0,1,1  
;* AND 254 OF ZEROS COMMAND IS LOADED INTO REGISTERS EXCEPT  
;* GO AND IE THEN ALL REGISTERS ARE SAVED  
;* GO IS GIVEN FOR THE READ COMMAND, 256 WORDS  
*****  
;* ALL REGISTERS ARE CHECKED FOR IMPROPER CHANGE  
;* THEN THE READ DATA IS COMPARED  
*****
```

```
*****  
TST41: SCOPE  
MOV #STACK,SP ;RESET STACK  
MOV #41,@#TSTNM ;SAVE TEST NUMBER  
*****  
JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2  
;R3-RHDS1, R4-RHER1  
;GIVE RH-11 INITIALIZE  
;SETUP UNIT NUMBER  
*****  
;*FILL WRITE FROM BUFFER WITH 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  
*****  
;*FILL WRITE FROM BUFFER WITH DATA  
*****  
JSR RO,@#CLAREA ;CLEAR 256. WORDS, FROM WRFROM+10  
WRFROM+10 ;STARTING FROM WRFROM+10  
256. ;256. WORDS  
0 ;FILL WITH 0  
*****
```

```

7735
7736 ;*FILL WRITE FROM BUFFER WITH NEXT SECTOR HEADER
7737
7738 030432 004037 041350 JSR RO,@#FLHEAD ;SAVE HEADER DATA IN WRFROM+<260.*2>
7739 030436 003500 WRFROM+<260.*2> ;LOCATION WHERE SAVED
7740 030440 000004 4 ;NUMBER OF WORDS SAVED
7741 030442 010000 10000 ;FIRST DATA WORD
7742 030444 000001 1 ;SECOND DATA WORD
7743 030446 000000 0 ;THIRD DATA WORD
7744 030450 000000 0 ;FOURTH DATA WORD
7745
7746 ;*FILL WRITE FROM BUFFER WITH NEXT SECTOR DATA
7747 030452 004037 041374 JSR RO,@#CLAREA ;CLEAR 2 WORDS, FROM WRFROM+<264.*2>
7748 030456 003510 WRFROM+<264.*2> ;STARTING FROM WRFROM+<264.*2>
7749 030460 000002 2 ;2 WORDS
7750 030462 000001 1 ;FILL WITH 1
7751
7752
7753 ;*NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA
7754 ;*AS WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
7755 ;*CAN BE MADE TO MAKE SURE THAT WRITE DID NOT
7756 ;*CHANGE WRITE FROM BUFFER.
7757
7758
7759 030464 004037 041350 JSR RO,@#FLHEAD ;SAVE HEADER DATA IN REINTO
7760 030470 003534 REINTO ;LOCATION WHERE SAVED
7761 030472 000004 4 ;NUMBER OF WORDS SAVED
7762 030474 010000 10000 ;FIRST DATA WORD
7763 030476 000000 0 ;SECOND DATA WORD
7764 030500 000000 0 ;THIRD DATA WORD
7765 030502 000000 0 ;FOURTH DATA WORD
7766 030504 004037 041374 JSR RO,@#CLAREA ;CLEAR 256. WORDS, FROM REINTO+10
7767 030510 003544 REINTO+10 ;STARTING FROM REINTO+10
7768 030512 000400 256. ;256. WORDS
7769 030514 000000 0 ;FILL WITH 0
7770
7771
7772 030516 004037 041350 JSR RO,@#FLHEAD ;SAVE HEADER DATA IN REINTO+<260.*2>
7773 030522 004544 REINTO+<260.*2> ;LOCATION WHERE SAVED
7774 030524 000004 4 ;NUMBER OF WORDS SAVED
7775 030526 010000 10000 ;FIRST DATA WORD
7776 030530 000001 1 ;SECOND DATA WORD
7777 030532 000000 0 ;THIRD DATA WORD
7778 030534 000000 0 ;FOURTH DATA WORD
7779 030536 004037 041374 JSR RO,@#CLAREA ;CLEAR 2 WORDS, FROM REINTO+<264.*2>
7780 030542 004554 REINTO+<264.*2> ;STARTING FROM REINTO+<264.*2>
7781 030544 000002 2 ;2 WORDS
7782 030546 000001 1 ;FILL WITH 1
7783
7784
7785 ;*NOW THE WRITE HEADER AND DATA COMMAND WILL BE FILLED
7786
7787
7788 030550 004037 043476 JSR RO,@#RUN ;SETUP TO RUN FOR DATA COMMAND
7789 030554 000000 0 ;CYLINDER 0
7790 030556 000 .BYTE 0 ;SECTOR 0
  
```

```

7791 030557 000 .BYTE 0 ;TRACK 0
7792 030560 177366 -262.-4 ;WORD COUNT (DATA) = 262. +
7793 ;4 HEADER WORDS
7794 030562 002470 WRFROM ;BUS ADDRESS
7795 ;STARTING ADDRESS OF DATA
7796 ;BUFFER = WRFROM
7797 030564 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
7798 030566 010000 FMT22 ;16 BITS PER WORD FORMAT
7799 ;DO NOT INHIBIT ECC CORRECTION
7800 ;DO NOT INHIBIT HEADER COMPARE
7801 030570 002444 WRIFOR ;GET READY TO DO A WRIFOR
7802 ;WRITE HEADER AND DATA WITH 62 IN RHCS1
7803
7804
7805 ;*NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE HEADER AND DAT1
7806 030572 004037 041672 JSR RO,@#SAVER ;SAVE REGISTERS
7807 030576 002272 RHWC ;RHWC IS THE FIRST REGISTER SAVED
7808 030600 004612 SAVERE ;STARTING ADDRESS OF WHERE
7809 ;THE REGISTERS ARE SAVED
7810 030602 000022 18. ;NUMBER OF REGISTERS
7811 ;SAVED = 18.
7812
7813 030604 004737 041604 JSR PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
7814 ;AND THAT NO STATUS BITS ARE STUCK = 1
7815 030610 104401 067033 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF ANY OF
7816 ;THE FIRST SET OF BITS DON'T = 1
7817 030614 000000 HALT ;STOP
7818
7819 030616 013777 004606 151442 MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
7820 ;TO 'TIME1' IF P-CLOCK IS PRESENT
7821 ;OR TO 'TIME2' IF P-CLOCK IS NO PRESENT
7822 ;'TIME' WILL ONLY SAVE
7823 ;CURRENT CYLINDER ADDRESS
7824 ;AND LOOK AHEAD REGISTERS
7825
7826
7827 030624 013746 002444 MOV @#WRIFOR,-(SP) ;GET READY TO MOVE COMMAND
7828 030630 052716 000101 BIS #GO.IE,(SP) ;GET READY TO SET 'GO' AND
7829 ;ENABLE INTERRUPT
7830 030634 012677 151440 MOV (SP)+,@RHCS1 ;GO WITH
7831 ;62 IN RHCS1 FOR WRITE HEADER AND DATA
7832 ;WITH INTERRUPT ENABLED
7833 030640 011100 MOV @R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
7834 030642 011305 MOV @R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
7835
7836 ;*ONE REVOLUTION = 16670 MICRO SEC, ONE SECTOR = 760 MICRO SEC
7837
7838
7839 030644 104413 WAT ;WAIT FOR RDY BIT TO SET
7840 030646 002300 RHCS1 ;WAIT FOR RHCS1 REGISTER
7841 030650 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
7842 030652 002001 1025. ;ALLOW 10250 MICRO SECONDS
7843 030654 001553 875. ;RDY MUST SET BETWEEN
7844 ;1500 AND 19000 MICRO SECONDS
7845
7846 ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
  
```

```

7847                                     ;*R0 AND R5 IMMEDIATELY AFTER GO
7848
7849 030656 013746 002444                MOV    @#WRIFOR,-(SP)  ;SAVE COMMAND
7850 030662 052716 004101                BIS    #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
7851 030666 011637 001124                MOV    (SP),@#SGDDAT  ;SAVE FOR PRINTOUT
7852 030672 022600                        CMP    (SP)+,R0       ;DURING ABOVE OPERATION ONLY IE.GO!DVA
7853                                     ;AND COMMAND SHOULD BE SET
7854 030674 001405                        BEQ    64$            ;BRANCH IF GOOD
7855 030676 010037 001126                MOV    R0,@#SBDDAT   ;BAD DATA
7856 030702 010137 004600                MOV    R1,@#REGADR   ;FAILING REGISTER RHCS1
7857 030706 104021                        ERROR  21            ;DURING ABOVE OPERATION ONLY
7858                                     ;COMMAND AND IE!GO!DVA SHOULD BE SET
7859 030710 012746 010500                64$: MOV    #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
7860 030714 011637 001124                MOV    (SP),@#SGDDAT ;SAVE FOR PRINTOUT
7861 030720 022605                        CMP    (SP)+,R5       ;DURING ABOVE OPERATION ONLY MOL.DPR!VV
7862                                     ;SHOULD BE SET
7863 030722 001405                        BEQ    66$            ;BRANCH IF GOOD
7864 030724 010537 001126                MOV    R5,@#SBDDAT   ;BAD DATA
7865 030730 010337 004600                MOV    R3,@#REGADR   ;FAILING REGISTER RHDS1
7866 030734 104063                        ERROR  63            ;DURING ABOVE OPERATION ONLY
7867                                     ;MOL!DPR!VV SHOULD BE SET
7868 030736                                66$:
7869
7870                                     ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
7871
7872 030736 004037 041426                JSR    R0,@#FILLRE   ;MOV 0 INTO SAVED RHWC
7873 030742 002272                        RHWC                                     ;SAVED REGISTER TO CHANGE
7874 030744 000000                        0                                       ;DATA
7875 030746 004037 041426                JSR    R0,@#FILLRE   ;MOV WRFROM+<266.*2> INTO SAVED RHBA
7876 030752 002274                        RHBA                                     ;SAVED REGISTER TO CHANGE
7877 030754 003514                        WRFROM+<266.*2> ;DATA
7878 030756 004037 041426                JSR    R0,@#FILLRE   ;MOV 2 INTO SAVED RHDST
7879 030762 002304                        RHDST                                     ;SAVED REGISTER TO CHANGE
7880 030764 000002                        2                                       ;DATA
7881
7882                                     ;*NOW COMPARE REGISTERS BEFORE WRITE HEADER AND DATA
7883                                     ;*WITH REGISTERS AFTER COMMAND
7884
7885
7886 030766 004037 042512                JSR    R0,@#COMREG   ;COMPARE SAVED REGISTERS WITH
7887                                     ;PRESENT VALUE
7888 030772 004612                        SAVERE                                     ;GOOD DATA SAVED IN 'SAVERE'
7889 030774 002354                        WC                                       ;TEST DATA STARTING FROM 'RHWC'
7890 030776 000022                        18.                                     ;18. REGISTERS TO BE COMPARED
7891 031000 031004                        1$                                       ;RETURN TO 1$ ON ERROR
7892 031002 031010                        2$                                       ;RETURN TO 2$ ON NO ERROR
7893
7894 031004 104027                        1$: ERROR  27        ;WRITE HEADER AND DATA
7895 031006 000207                        RTS    PC          ;CAUSED IMPROPER REGISTER
7896                                     ;CHANGE
7897                                     ;GOOD DATA GIVES WHAT SHOULD
7898                                     ;BE THERE
7899                                     ;RECEIVED DATA GIVES WHAT
7900                                     ;WAS THERE AFTER COMMANT
7901
7902                                     ;*NOW WRITE FROM BUFFER WILL BE CHECKED TO SEE THAT

```

```

7903                                     : *NOTHING GOT CHANGED
7904
7905 031010                               2$:
7906
7907 031010 004037 043542                 JSR    RO,@#COMPAR    ;COMPARE TWO BLOCKS OF MEMORY
7908 031014 003534                         REINTO                ;GOOD DATA STARTS FROM REINTO
7909 031016 002470                         WRFROM                ;TEST DATA STARTS FROM WRFROM
7910 031020 000412                         266.                  ;266. WORDS TO BE COMPARED
7911 031022 031026                         3$                    ;RETURN TO 3$ ON ERROR
7912 031024 031032                         4$                    ;RETURN TO 4$ ON NO ERROR
7913
7914
7915 031026 104030                           3$: ERROR 30          ;WRITE HEADER AND DATA
7916 031030 000207                         RTS    PC              ;CHANGED WRITE FROM BUFFER
7917
7918                                     : *NOW A READ HEADER AND DATA COMMAND WILL BE GIVEN
7919                                     : *FOR SECTOR 1, 256 WORDS
7920                                     : *READ INTO BUFFER IS FILLED WITH ONES
7921 031032                               4$:
7922
7923 031032 004737 041524                 JSR    PC,@#CLDISK    ;SET R1-RHCS1, R2-RHCS2
7924                                     ;R3-RHDS1, R4-RHER1
7925                                     ;GIVE RH-11 INITIALIZE
7926                                     ;SETUP UNIT NUMBER
7927 031036 004037 041374                 JSR    RO,@#CLAREA    ;CLEAR 260. WORDS, FROM REINTO
7928 031042 003534                         REINTO                ;STARTING FROM REINTO
7929 031044 000404                         260.                  ;260. WORDS
7930 031046 000377                         377                   ;FILL WITH 377
7931
7932
7933                                     : *WRITE FROM BUFFER IS FILLED WITH EXPECTED DATA
7934
7935
7936 031050 004037 041350                 JSR    RO,@#FLHEAD    ;SAVE HEADER DATA IN WRFROM
7937 031054 002470                         WRFROM                ;LOCATION WHERE SAVED
7938 031056 000006                         6                     ;NUMBER OF WORDS SAVED
7939 031060 010000                         10000                ;FIRST DATA WORD
7940 031062 000001                         1                     ;SECOND DATA WORD
7941 031064 000000                         0                     ;THIRD DATA WORD
7942 031066 000000                         0                     ;FOURTH DATA WORD
7943 031070 000001                         1                     ;FIFTH DATA WORD
7944 031072 000001                         1                     ;SIXTH DATA WORD
7945 031074 004037 041374                 JSR    RO,@#CLAREA    ;CLEAR 254 WORDS, FROM WRFROM+<6.*2>
7946 031100 002504                         WRFROM+<6.*2>         ;STARTING FROM WRFROM+<6.*2>
7947 031102 000254                         254                   ;254 WORDS
7948 031104 000000                         0                     ;FILL WITH 0
7949
7950
7951                                     : *NOW FILL COMMAND
7952
7953
7954 031106 004037 043476                 JSR    RO,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
7955 031112 000000                         0                     ;CYLINDER 0
7956 031114 001 .BYTE 1                     ;SECTOR 1
7957 031115 000 .BYTE 0                     ;TRACK 0
7958 031116 177374                         -256.-4              ;WORD COUNT (DATA) - 256. +

```



```

8015
8016 031214 013746 002450      MOV    @#REFOR,-(SP)    ;SAVE COMMAND
8017 031220 052716 004101      BIS    #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
8018 031224 011637 001124      MOV    (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
8019 031230 022600              CMP    (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO.DVA
8020                                ;AND COMMAND SHOULD BE SET
8021 031232 001405              BEQ    67$             ;BRANCH IF GOOD
8022 031234 010037 001126      MOV    R0,@#$BDDAT     ;BAD DATA
8023 031240 010137 004600      MOV    R1,@#REGADR     ;FAILING REGISTER RHCS1
8024 031244 104021              ERROR  21             ;DURING ABOVE OPERATION ONLY
8025                                ;COMMAND AND IE!GO!DVA SHOULD BE SET
8026 031246 012746 010500      67$: MOV    #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
8027 031252 011637 001124      MOV    (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
8028 031256 022605              CMP    (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
8029                                ;SHOULD BE SET
8030                                BEQ    69$             ;BRANCH IF GOOD
8031 031262 010537 001126      MOV    R5,@#$BDDAT     ;BAD DATA
8032 031266 010337 004600      MOV    R3,@#REGADR     ;FAILING REGISTER RHDS1
8033 031272 104063              ERROR  63             ;DURING ABOVE OPERATION ONLY
8034                                ;MOL!DPR!VV SHOULD BE SET
8035 031274              69$:
8036
8037                                ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
8038
8039 031274 004037 041426      JSR    R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
8040 031300 002272              RHWC                  ;SAVED REGISTER TO CHANGE
8041 031302 000000              0                     ;DATA
8042 031304 004037 041426      JSR    R0,@#FILLRE     ;MOV REINTO+<260.*2> INTO SAVED RHBA
8043 031310 002274              RHBA                  ;SAVED REGISTER TO CHANGE
8044 031312 004544              REINTO+<260.*2>      ;DATA
8045 031314 004037 041426      JSR    R0,@#FILLRE     ;MOV 2 INTO SAVED RHDST
8046 031320 002304              RHDST                 ;SAVED REGISTER TO CHANGE
8047 031322 000002              2                     ;DATA
8048
8049                                ;*COMPARE REGISTERS BEFORE READ HEADER AND DATA
8050                                ;*WITH REGISTERS AFTER COMMAND
8051
8052
8053 031324 004037 042512      JSR    R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
8054                                ;PRESENT VALUE
8055 031330 004612              SAVERE                ;GOOD DATA SAVED IN 'SAVERE'
8056 031332 002354              WC                     ;TEST DATA STARTING FROM 'RHWC'
8057 031334 000022              18.                   ;18. REGISTERS TO BE COMPARED
8058 031336 031342              5$                    ;RETURN TO 5$ ON ERROR
8059 031340 031346              6$                    ;RETURN TO 6$ ON NO ERROR
8060
8061
8062 031342 104031              5$: ERROR  31          ;READ HEADER AND DATA CAUSED
8063 031344 000207              RTS    PC              ;IMPROPER REGISTER CHANGE
8064                                ;GOOD DATA GIVES WHAT SHOULD
8065                                ;BE THERE
8066                                ;RECEIVED DATA GIVES WHAT WAS
8067                                ;THERE AFTER COMMAND
8068
8069                                ;*NOW READ INTO BUFFER WILL BE CHECKED TO SEE
8070                                ;*THAT READ WAS GOOD

```

```

8071
8072 031346          6$:
8073
8074 031346 004037 043542      JSR      RO,@#COMPAR      ;COMPARE TWO BLOCKS OF MEMORY
8075 031352 002470              WRFROM      ;GOOD DATA STARTS FROM WRFROM
8076 031354 003534              REINTO     ;TEST DATA STARTS FROM REINTO
8077 031356 000404              260.       ;260. WORDS TO BE COMPARED
8078 031360 031364              7$        ;RETURN TO 7$ ON ERROR
8079 031362 031370              10$       ;RETURN TO 10$ ON NO ERROR
8080
8081
8082
8083 031364 104032          7$:      ERROR  32      ;WRITE HEADER AND DATA
8084 031366 000207              RTS      PC      ;FOLLOWED BY A READ HEADER
8085                                     ;AND DATA GAVE A READ ERROR
8086                                     ;ERROR MAY BE IN READ OR WRITE
8087 031370          10$:

```

8088
8089
8090
8091
8092
8093
8094
8095
8096
8097
8098
8099
8100
8101
8102
8103
8104
8105
8106
8107
8108
8109
8110
8111
8112
8113
8114
8115
8116
8117
8118
8119
8120
8121
8122
8123
8124
8125
8126
8127
8128
8129
8130
8131
8132
8133
8134
8135
8136
8137
8138
8139
8140
8141
8142
8143

: *TEST 42 IMPLIED SEEK TO CYL 001

.SBTTL SEEK TESTS

: * ONLY NEW ADDITION IN THIS TEST IS AN IMPLIED SEEK FROM CYLINDER 0
: * TO CYLINDER 1, A WRITE HEADER AND DATA IS GIVEN FOR MORE THAN
: * ONE SECTOR OF WORDS

: * WRITE HEADER AND DATA CYLINDER 0, FORMAT 16 BITS PER WORD
: * TRACK 18, SECTOR 21, KEYS=0, NUMBER OF WORDS 266 WORDS
: * (4 HEADER, 256 DATA=1125, 4 HEADER 2 DATA = 2000
: * THEN READ HEADER AND DATA FOR ABOVE, TRACK 0, SECTOR 0, CYL=1
: * WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH
: * 10000,0,0,0 AND 256 OF 1125, 10001,0,0,0, AND 2 OF 2000
: * THE WRITE COMMAND IS THEN LOADED INTO THE REGISTERS EXCEPT
: * THE GO BIT, AND ALL THE REGISTERS ARE SAVED
: * THEN GO IS GIVEN FOR WRITE HEADER AND DATA

: * THEN ALL REGISTERS ARE COMPARED TO CHECK FOR IMPROPER CHANGE
: * THEN WRITE FROM BUFFER IS CHECKED TO SEE THAT NOTHING CHANGED

: * NOW FOR READ COMMAND READ INTO BUFFER IS FILLED
: * WITH ALL ONES, WRITE FROM BUFFER IS LOADED WITH
: * 10001,0,0,0 AND 2 OF 2000, 254 OF ZEROS COMMAND IS LOADED
: * INTO REGISTERS EXCEPT GO AND IE ALL REGISTERS ARE SAVED
: * GO IS GIVEN FOR THE READ COMMAND

: * ALL REGISTERS ARE CHECKED FOR IMPROPER CHANGE
: * THEN THE READ DATA IS COMPARED.

```
TST42: SCOPE
MOV #STACK,SP ;RESET STACK
MOV #42,@#TSTNM ;SAVE TEST NUMBER
JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
;R3-RHDS1, R4-RHER1
;GIVE RH-11 INITIALIZE
;SETUP UNIT NUMBER

: *FILL WRITE FROM BUFFER WITH HEADER
JSR R0,@#FLHEAD ;SAVE HEADER DATA IN WRFROM
WRFROM ;LOCATION WHERE SAVED
4 ;NUMBER OF WORDS SAVED
10000 ;FIRST DATA WORD
<18.*400>!<21.> ;SECOND DATA WORD
0 ;THIRD DATA WORD
0 ;FOURTH DATA WORD

: *FILL WRITE FROM BUFFER WITH DATA
JSR R0,@#CLAREA ;CLEAR 256. WORDS, FROM WRFROM+10
```

031370 000004
031372 012706 001000
031376 012737 000042 004604
031404 004737 041524
031410 004037 041350
031414 002470
031416 000004
031420 010000
031422 011025
031424 000000
031426 000000
031430 004037 041374

```
8144 031434 002500 WRFROM+10 ;STARTING FROM WRFROM+10
8145 031436 000400 256. ;256. WORDS
8146 031440 001125 <18.*40>!21. ;FILL WITH <18.*40>!21.
8147
8148
8149 ;*FILL WRITE FROM BUFFER WITH NEXT TRACK HEADER
8150
8151 031442 004037 041350 JSR RO,@#FLHEAD ;SAVE HEADER DATA IN WRFROM+<260.*2>
8152 031446 003500 WRFROM+<260.*2> ;LOCATION WHERE SAVED
8153 031450 000004 4 ;NUMBER OF WORDS SAVED
8154 031452 010001 10001 ;FIRST DATA WORD
8155 031454 000000 0 ;SECOND DATA WORD
8156 031456 000000 0 ;THIRD DATA WORD
8157 031460 000000 0 ;FOURTH DATA WORD
8158
8159 ;*FILL WRITE FROM BUFFER WITH NEXT TRACK DATA
8160 031462 004037 041374 JSR RO,@#CLAREA ;CLEAR 2 WORDS, FROM WRFROM+<264.*2>
8161 031466 003510 WRFROM+<264.*2> ;STARTING FROM WRFROM+<264.*2>
8162 031470 000002 2 ;2 WORDS
8163 031472 002000 2000 ;FILL WITH 2000
8164
8165
8166 ;*NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA
8167 ;*AS WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
8168 ;*CAN BE MADE TO MAKE SURE THAT WRITE DID NOT
8169 ;*CHANGE WRITE FROM BUFFER.
8170
8171
8172 031474 004037 041350 JSR RO,@#FLHEAD ;SAVE HEADER DATA IN REINTO
8173 031500 003534 REINTO ;LOCATION WHERE SAVED
8174 031502 000004 4 ;NUMBER OF WORDS SAVED
8175 031504 010000 10000 ;FIRST DATA WORD
8176 031506 011025 <18.*400>!<21.> ;SECOND DATA WORD
8177 031510 000000 0 ;THIRD DATA WORD
8178 031512 000000 0 ;FOURTH DATA WORD
8179 031514 004037 041374 JSR RO,@#CLAREA ;CLEAR 256. WORDS, FROM REINTO+10
8180 031520 003544 REINTO+10 ;STARTING FROM REINTO+10
8181 031522 000400 256. ;256. WORDS
8182 031524 001125 <18.*40>!21. ;FILL WITH <18.*40>!21.
8183
8184
8185 031526 004037 041350 JSR RO,@#FLHEAD ;SAVE HEADER DATA IN REINTO+<260.*2>
8186 031532 004544 REINTO+<260.*2> ;LOCATION WHERE SAVED
8187 031534 000004 4 ;NUMBER OF WORDS SAVED
8188 031536 010001 10001 ;FIRST DATA WORD
8189 031540 000000 0 ;SECOND DATA WORD
8190 031542 000000 0 ;THIRD DATA WORD
8191 031544 000000 0 ;FOURTH DATA WORD
8192 031546 004037 041374 JSR RO,@#CLAREA ;CLEAR 2 WORDS, FROM REINTO+<264.*2>
8193 031552 004554 REINTO+<264.*2> ;STARTING FROM REINTO+<264.*2>
8194 031554 000002 2 ;2 WORDS
8195 031556 002000 2000 ;FILL WITH 2000
8196
8197
8198 ;*NOW THE WRITE HEADER AND DATA COMMAND WILL BE FILLED
8199
```

```

8200
8201 031560 004037 043476      JSR      RO,@#RUN      ;SETUP TO RUN FOR DATA COMMAND
8202 031564 000000              0              ;CYLINDER 0
8203 031566      025      .BYTE 21.          ;SECTOR 21.
8204 031567      022      .BYTE 18.          ;TRACK 18.
8205 031570 177366              -262.-4       ;WORD COUNT (DATA) = 262. +
8206                          ;4 HEADER WORDS
8207 031572 002470      WRFROM      ;BUS ADDRESS
8208                          ;STARTING ADDRESS OF DATA
8209                          ;BUFFER = WRFROM
8210 031574 000000              0              ;DO NOT INHIBIT BUS ADDRESS INCREMENT
8211 031576 010000      FMT22      ;16 BITS PER WORD FORMAT
8212                          ;DO NOT INHIBIT ECC CORRECTION
8213                          ;DO NOT INHIBIT HEADER COMPARE
8214 031600 002444      WRIFOR      ;GET READY TO DO A WRIFOR
8215                          ;WRITE HEADER AND DATA WITH 62 IN RHCS1
8216
8217
8218                          ;*NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE HEADER AND DATA1
8219
8220 031602 004037 041672      JSR      RO,@#SAVER    ;SAVE REGISTERS
8221 031606 002272      RHWC      ;RHWC IS THE FIRST REGISTER SAVED
8222 031610 004612      SAVERE     ;STARTING ADDRESS OF WHERE
8223                          ;THE REGISTERS ARE SAVED
8224 031612 000022              18.          ;NUMBER OF REGISTERS
8225                          ;SAVED = 18.
8226
8227 031614 004737 041604      JSR      PC,@#CHECKT   ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
8228                          ;AND THAT NO STATUS BITS ARE STUCK - 1
8229 031620 104401 067033      TYPE      ,CPHALT    ;CANNOT CONTINUE TESTING IF ANY OF
8230                          ;THE FIRST SET OF BITS DON'T = 1
8231 031624 000000      HALT      ;STOP
8232
8233 031626 013777 004606 150432  MOV      @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
8234                          ;TO 'TIME1' IF P-CLOCK IS PRESENT
8235                          ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
8236                          ;'TIME' WILL ONLY SAVE
8237                          ;CURRENT CYLINDER ADDRESS
8238                          ;AND LOOK AHEAD REGISTERS
8239
8240
8241 031634 013746 002444      MOV      @#WRIFOR,-(SP) ;GET READY TO MOVE COMMAND
8242 031640 052716 000101      BIS      #GO!IE,(SP)  ;GET READY TO SET 'GO' AND
8243                          ;ENABLE INTERRUPT
8244 031644 012677 150430      MOV      (SP)+,@RHCS1 ;GO WITH
8245                          ;62 IN RHCS1 FOR WRITE HEADER AND DATA
8246                          ;WITH INTERRUPT ENABLED
8247 031650 011100      MOV      @R1,R0        ;SAVE RHCS1 DURING ABOVE OPERATION
8248 031652 011305      MOV      @R3,R5        ;SAVE RHDS1 DURING ABOVE OPERATION
8249
8250                          ;*ONE REVOLUTION = 16670 MICRO1 SEC, ONE SECTOR = 760 MICRO1 SEC
8251
8252
8253 031654 104413      WAT      ;WAIT FOR RDY BIT TO SET
8254 031656 002300      RHCS1   ;WAIT FOR RHCS1 REGISTER
8255 031660 000200      RDY     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
  
```

```

8256 031662 003237          1695.          ;ALLOW 16950 MICRO SECONDS
8257 031664 001515          845.          ;RDY MUST SET BETWEEN
8258                                     ;8500 AND 25400 MICRO SECONDS
8259
8260                                     ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
8261                                     ;*RO AND R5 IMMEDIATELY AFTER GO
8262
8263 031666 013746 002444    MOV    @#WRIFOR,-(SP) ;SAVE COMMAND
8264 031672 052716 004101    BIS    #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
8265 031676 011637 001124    MOV    (SP),@#SGDDAT ;SAVE FOR PRINTOUT
8266 031702 022600          CMP    (SP)+,R0 ;DURING ABOVE OPERATION ONLY IE!GO!DVA
8267                                     ;AND COMMAND SHOULD BE SET
8268 031704 001405          BEQ    64$ ;BRANCH IF GOOD
8269 031706 010037 001126    MOV    R0,@#SBDDAT ;BAD DATA
8270 031712 010137 004600    MOV    R1,@#REGADR ;FAILING REGISTER RHCS1
8271 031716 104021          ERROR  21 ;DURING ABOVE OPERATION ONLY
8272                                     ;COMMAND AND IE!GO!DVA SHOULD BE SET
8273 031720 012746 010500    64$: MOV    #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
8274 031724 011637 001124    MOV    (SP),@#SGDDAT ;SAVE FOR PRINTOUT
8275 031730 022605          CMP    (SP)+,R5 ;DURING ABOVE OPERATION ONLY MOL!DPR.VV
8276                                     ;SHOULD BE SET
8277 031732 001405          BEQ    66$ ;BRANCH IF GOOD
8278 031734 010537 001126    MOV    R5,@#SBDDAT ;BAD DATA
8279 031740 010337 004600    MOV    R3,@#REGADR ;FAILING REGISTER RHDS1
8280 031744 104063          ERROR  63 ;DURING ABOVE OPERATION ONLY
8281                                     ;MOL!DPR!VV SHOULD BE SET
8282 031746          66$:
8283
8284                                     ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
8285
8286 031746 004037 041426    JSR    R0,@#FILLRE ;MOV 0 INTO SAVED RHWC
8287 031752 002272          RHWC ;SAVED REGISTER TO CHANGE
8288 031754 000000          0 ;DATA
8289 031756 004037 041426    JSR    R0,@#FILLRE ;MOV WRFROM+<266.*2> INTO SAVED RHBA
8290 031762 002274          RHBA ;SAVED REGISTER TO CHANGE
8291 031764 003514          WRFROM+<266.*2> ;DATA
8292 031766 004037 041426    JSR    R0,@#FILLRE ;MOV 1 INTO SAVED RHCA
8293 031772 002312          RHCA ;SAVED REGISTER TO CHANGE
8294 031774 000001          1 ;DATA
8295 031776 004037 041426    JSR    R0,@#FILLRE ;MOV 1 INTO SAVED RHCC
8296 032002 002334          RHCC ;SAVED REGISTER TO CHANGE
8297 032004 000001          1 ;DATA
8298 032006 004037 041426    JSR    R0,@#FILLRE ;MOV 1 INTO SAVED RHDST
8299 032012 002304          RHDST ;SAVED REGISTER TO CHANGE
8300 032014 000001          1 ;DATA
8301
8302                                     ;*NOW COMARE REGISTERS BEFORE WRITE HEADER AND DATA
8303                                     ;*WITH REGISTERS AFTER COMMAND
8304
8305
8306 032016 004037 042512    JSR    R0,@#COMREG ;COMPARE SAVED REGISTERS WITH
8307                                     ;PRESENT VALUE
8308 032022 004612          SAVERE ;GOOD DATA SAVED IN 'SAVERE'
8309 032024 002354          WC ;TEST DATA STARTING FROM 'RHWC'
8310 032026 000022          18. ;18. REGISTERS TO BE COMPARED
8311 032030 032034          1$ ;RETURN TO 1$ ON ERROR

```

```
8312 032032 032040          2$:          ;RETURN TO 2$ ON NO ERROR
8313
8314 032034 104027          1$:      ERROR  27          ;WRITE HEADER AND DATA
8315 032036 000207          RTS      PC          ;CAUSED IMPROPER REGISTER
8316                          ;CHANGE
8317                          ;GOOD DATA GIVES WHAT SHOULD
8318                          ;BE THERE
8319                          ;RECEIVED DATA GIVES WHAT
8320                          ;WAS THERE AFTER COMMANT
8321
8322                          ;*NOW WRITE FROM BUFFER WILL BE CHECKED TO SEE THAT
8323                          ;*NOTHING GOT CHANGED
8324
8325 032040          2$:
8326
8327 032040 004037 043542      JSR      RO,@#COMPAR      ;COMPARE TWO BLOCKS OF MEMORY
8328 032044 003534          REINTO          ;GOOD DATA STARTS FROM REINTO
8329 032046 002470          WRFROM          ;TEST DATA STARTS FROM WRFROM
8330 032050 000412          266.          ;266. WORDS TO BE COMPARED
8331 032052 032056          3$          ;RETURN TO 3$ ON ERROR
8332 032054 032062          4$          ;RETURN TO 4$ ON NO ERROR
8333
8334
8335 032056 104030          3$:      ERROR  30          ;WRITE HEADER AND DATA
8336 032060 000207          RTS      PC          ;CHANGED WRITE FROM BUFFER
8337
8338                          ;*NOW A READ HEADER AND DATA COMMAND WILL BE GIVEN
8339                          ;*READ INTO BUFFER IS FILLED WITH ONES
8340
8341 032062          4$:
8342
8343 032062 004737 041524      JSR      PC,@#CLDISK      ;SET R1-RHCS1, R2-RHCS2
8344                          ;R3-RHDS1, R4-RHER1
8345                          ;GIVE RH-11 INITIALIZE
8346                          ;SETUP UNIT NUMBER
8347 032066 004037 041374      JSR      RO,@#CLAREA      ;CLEAR 260. WORDS, FROM REINTO
8348 032072 003534          REINTO          ;STARTING FROM REINTO
8349 032074 000404          260.          ;260. WORDS
8350 032076 177 77          -1          ;FILL WITH -1
8351
8352
8353                          ;*WRITE FROM BUFFER IS FILLED WITH 10001,0,0,0,2000,2000, AND 254 OF 0
8354
8355 032100 004037 041350      JSR      RO,@#FLHEAD      ;SAVE HEADER DATA IN WRFROM
8356 032104 002470          WRFROM          ;LOCATION WHERE SAVED
8357 032106 000006          6          ;NUMBER OF WORDS SAVED
8358 032110 010001          10001          ;FIRST DATA WORD
8359 032112 000000          0          ;SECOND DATA WORD
8360 032114 000000          0          ;THIRD DATA WORD
8361 032116 000000          0          ;FOURTH DATA WORD
8362 032120 002000          2000          ;FIFTH DATA WORD
8363 032122 002000          2000          ;SIXTH DATA WORD
8364 032124 004037 041374      JSR      RO,@#CLAREA      ;CLEAR 254. WORDS, FROM WRFROM+<6*2>
8365 032130 002504          WRFROM+<6*2>      ;STARTING FROM WRFROM+<6*2>
8366 032132 000376          254.          ;254. WORDS
8367 032134 000000          0          ;FILL WITH 0
```

```
8368
8369
8370 ;*NOW FILL COMMAND
8371
8372 032136 004037 043476 JSR RO,@#RUN ;SETUP TO RUN FOR DATA COMMAND
8373 032142 000001 1 ;CYLINDER 1
8374 032144 000 .BYTE 0 ;SECTOR 0
8375 032145 000 .BYTE 0 ;TRACK 0
8376 032146 177374 -256.-4 ;WORD COUNT (DATA) = 256. +
8377 ;4 HEADER WORDS
8378 032150 003534 REINTO ;BUS ADDRESS
8379 ;STARTING ADDRESS OF DATA
8380 ;BUFFER = REINTO
8381 032152 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
8382 032154 014000 ECI:FMT22 ;16 BITS PER WORD FORMAT
8383 ;INHIBIT ECC CORRECTION
8384 ;DO NOT INHIBIT HEADER COMPARE
8385 032156 002450 REFOR ;GET READY TO DO A REFOR
8386 ;READ HEADER AND DATA WITH 72 IN RHCS1
8387
8388
8389 ;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ HEADER AND DATA
8390
8391 032160 004037 041672 JSR RO,@#SAVER ;SAVE REGISTERS
8392 032164 002272 RHWC ;RHWC IS THE FIRST REGISTER SAVED
8393 032166 004612 SAVERE ;STARTING ADDRESS OF WHERE
8394 ;THE REGISTERS ARE SAVED
8395 032170 000022 18. ;NUMBER OF REGISTERS
8396 ;SAVED = 18.
8397
8398 032172 004737 041604 JSR PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
8399 ;AND THAT NO STATUS BITS ARE STUCK = 1
8400 032176 104401 067033 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF ANY OF
8401 ;THE FIRST SET OF BITS DON'T - 1
8402 032202 000000 HALT ;STOP
8403
8404 032204 013777 004606 150054 MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
8405 ;TO 'TIME1' IF P-CLOCK IS PRESENT
8406 ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
8407 ;'TIME' WILL ONLY SAVE
8408 ;CURRENT CYLINDER ADDRESS
8409 ;AND LOOK AHEAD REGISTERS
8410
8411
8412 032212 013746 002450 MOV @#REFOR,-(SP) ;GET READY TO MOVE COMMAND
8413 032216 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET 'GO' AND
8414 ;ENABLE INTERRUPT
8415 032222 012677 150052 MOV (SP)+,@RHCS1 ;GO WITH
8416 ;72 IN RHCS1 FOR READ DATA
8417 ;WITH INTERRUPT ENABLED
8418 032226 011100 MOV @R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
8419 032230 011305 MOV @R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
8420
8421
8422 032232 104413 WAT ;WAIT FOR RDY BIT TO SET
8423 032234 002300 RHCS1 ;WAIT FOR RHCS1 REGISTER
```



```

8424 032236 000200          RDY          ;WAIT FOR RDY BIT IN RHCS1 REGISTER
8425 032240 001614          908.        ;ALLOW 9080 MICRO SECONDS
8426 032242 001507          839.        ;RDY MUST SET BETWEEN
8427                                     ;690 AND 17470 MICRO SECONDS
8428
8429                                     ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
8430                                     ;*RO AND R5 IMMEDIATELY AFTER GO
8431
8432 032244 013746 002450    MOV    @#REFOR,-(SP)    ;SAVE COMMAND
8433 032250 052716 004101    BIS    #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
8434 032254 011637 001124    MOV    (SP),@#SGDDAT  ;SAVE FOR PRINTOUT
8435 032260 022600          CMP    (SP)+,RO        ;DURING ABOVE OPERATION ONLY IE!GO!DVA
8436                                     ;AND COMMAND SHOULD BE SET
8437 032262 001405          BEQ    67$            ;BRANCH IF GOOD
8438 032264 010037 001126    MOV    RO,@#SBDDAT    ;BAD DATA
8439 032270 010137 004600    MOV    R1,@#REGADR    ;FAILING REGISTER RHCS1
8440 032274 104021          ERROR  21            ;DURING ABOVE OPERATION ONLY
8441                                     ;COMMAND AND IE!GO!DVA SHOULD BE SET
8442 032276 012746 010500    67$: MOV    #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
8443 032302 011637 001124    MOV    (SP),@#SGDDAT  ;SAVE FOR PRINTOUT
8444 032306 022605          CMP    (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
8445                                     ;SHOULD BE SET
8446 032310 001405          BEQ    69$            ;BRANCH IF GOOD
8447 032312 010537 001126    MOV    R5,@#SBDDAT    ;BAD DATA
8448 032316 010337 004600    MOV    R3,@#REGADR    ;FAILING REGISTER RHDS1
8449 032322 104063          ERROR  63            ;DURING ABOVE OPERATION ONLY
8450                                     ;MOL!DPR!VV SHOULD BE SET
8451 032324          69$:
8452
8453                                     ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
8454 032324 004037 041426    JSR    RO,@#FILLRE    ;MOV 0 INTO SAVED RHWC
8455 032330 002272          RHWC          ;SAVED REGISTER TO CHANGE
8456 032332 000000          0            ;DATA
8457 032334 004037 041426    JSR    RO,@#FILLRE    ;MOV REINTO+<260.*2> INTO SAVED RHBA
8458 032340 002274          RHBA          ;SAVED REGISTER TO CHANGE
8459 032342 004544          REINTO+<260.*2> ;DATA
8460 032344 004037 041426    JSR    RO,@#FILLRE    ;MOV 1 INTO SAVED RHDST
8461 032350 002304          RHDST        ;SAVED REGISTER TO CHANGE
8462 032352 000001          1            ;DATA
8463
8464                                     ;*COMPARE REGISTERS BEFORE READ HEADER AND DATA
8465                                     ;*WITH REGISTERS AFTER COMMAND
8466
8467 032354 004037 042512    JSR    RO,@#COMREG    ;COMPARE SAVED REGISTERS WITH
8468                                     ;PRESENT VALUE
8469 032360 004612          SAVERE        ;GOOD DATA SAVED IN 'SAVERE'
8470 032362 002354          WC          ;TEST DATA STARTING FROM 'RHWC'
8471 032364 000022          18.         ;18. REGISTERS TO BE COMPARED
8472 032366 032372          5$          ;RETURN TO 5$ ON ERROR
8473 032370 032376          6$          ;RETURN TO 6$ ON NO ERROR
8474
8475
8476 032372 104031          5$: ERROR  31        ;READ HEADER AND DATA CAUSED
8477 032374 000207          RTS    PC         ;IMPROPER REGISTER CHANGE
8478                                     ;GOOD DATA GIVES WHAT SHOULD
8479                                     ;BE THERE

```

```
8480 ;RECEIVED DATA GIVES WHAT WAS
8481 ;THERE AFTER COMMAND
8482
8483 ;*NOW READ INTO BUFFER WILL BE CHECKED TO SEE
8484 ;*THAT READ WAS GOOD
8485
8486 032376 6$:
8487
8488 032376 004037 043542 JSR RO,@#COMPAR ;COMPARE TWO BLOCKS OF MEMORY
8489 032402 002470 WRFROM ;GOOD DATA STARTS FROM WRFROM
8490 032404 003534 REINTO ;TEST DATA STARTS FROM REINTO
8491 032406 000404 260. ;260. WORDS TO BE COMPARED
8492 032410 032414 7$ ;RETURN TO 7$ ON ERROR
8493 032412 032420 10$ ;RETURN TO 10$ ON NO ERROR
8494
8495
8496
8497 032414 104032 7$: ERROR 32 ;WRITE HEADER AND DATA
8498 032416 000207 RTS PC ;FOLLOWED BY A READ HEADER
8499 ;AND DATA GAVE A READ ERROR
8500 ;ERROR MAY BE IN READ OR WRITE
8501 032420 10$:
8502
```

```
8503
8504
8505      ;*****
8506      ;*TEST 43      SEEK & WRT TEST (CYL = 0-10)
8507
8508      ;*          THIS TEST GETS THE HEADS OUT TO CYLINDER 10 NOT BY ONE
8509      ;*          SEEK BUT BY TEN IMPLIED SEEKS ONE CYLINDER AT A TIME
8510
8511      ;*          THIS TEST STARTS WITH A (ALREADY TESTED) RECALIBRATE
8512      ;*          THAT IS CYLINDER ZERO. THEN ON CYLINDER 0 SECTOR
8513      ;*          #21 TRACK #18 IT WRITES 266 WORDS THERE BY GETTING
8514      ;*          THE HEAD TO CYLINDER 1 THEN IT WRITES 266 WORDS ON
8515      ;*          CYLINDER 1 SECTOR #21 TRACK #18 THERE BY GETTING
8516      ;*          THE HEADS TO CYLINDER 2
8517      ;*          THIS IS REPEATED 10 TIMES GETTING THE
8518      ;*          HEADS TO CYLINDER 10
8519      ;*          THEN A SEEK COMMAND IS GIVEN TO CYLINDER 0
8520      ;*          AND DATA ALREADY WRITTEN IS CHECKED
8521
8522      ;*****
8523      TST43: SCOPE
8524      MOV      #STACK,SP      ;RESET STACK
8525      MOV      #43,@#TSTNM    ;SAVE TEST NUMBER
8526      JSR      PC,@#CLDISK   ;SET R1-RHCS1, R2-RHCS2
8527      ;R3-RHDS1, R4-RHER1
8528      ;GIVE RH-11 INITIALIZE
8529      ;SETUP UNIT NUMBER
8530
8531      ;*THE FOLLOWING MOVES ARE TO INITIALIZE TEST FROM
8532      ;*CYLINDER 0
8533      ;*THESE LOCATIONS ARE CHANGED DURING TEST TO ENABLE
8534      ;*GOING TO NEXT CYLINDER
8535
8536      MOV      #10000,@#ST1+10
8537      MOV      #<<18.*40>!21.>,@#ST2+10
8538      MOV      #10001,@#ST3+10
8539      MOV      #2000,@#ST4+10
8540      MOV      #0,@#ST5+4
8541      MOV      #1,@#ST6+6
8542      MOV      #1,@#ST6+16
8543      MOV      #10001,@#ST9+10
8544      MOV      #2000,@#ST10+10
8545      MOV      #1,@#ST11+4
8546
8547      ;*THIS IS TO GET THE HEADS TO CYLINDER 0
8548
8549      MOV      @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
8550      ;TO 'TIME1' IF P-CLOCK IS PRESENT
8551      ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
8552      ;'TIME' WILL ONLY SAVE
8553      ;CURRENT CYLINDER ADDRESS
8554      ;AND LOOK AHEAD REGISTERS
8555
8556      MOV      @#RECALI,-(SP) ;GET READY TO MOVE COMMAND
8557      BIS      #GO!IE,(SP)   ;GET READY TO SET 'GO' AND
8558      ;ENABLE INTERRUPT
```

```

8559 032552 012677 147522      MOV      (SP)+, @RHCS1      ;GO WITH
8560                                ;6 IN RHCS1 FOR RECALIBRATE
8561                                ;WITH INTERRUPT ENABLED
8562 032556 011100      MOV      @R1, R0            ;SAVE RHCS1 DURING ABOVE OPERATION
8563 032560 011305      MOV      @R3, R5            ;SAVE RHDS1 DURING ABOVE OPERATION
8564
8565 032562 104413      WAT                                ;WAIT FOR DRY BIT TO SET
8566 032564 002322      RHDS1                          ;WAIT FOR RHDS1 REGISTER
8567 032566 000200      DRY                              ;WAIT FOR DRY BIT IN RHDS1 REGISTER
8568 032570 076377      31999.                          ;ALLOW 319990 MICRO SECONDS
8569 032572 056701      24001.                          ;DRY MUST SET BETWEEN
8570                                ;79980 AND 560000 MICRO SECONDS
8571 032574 012737 000012 001200  MOV      #10., @#STMP1      ;TEN COUNT TO GET TO CYLINDER 10
8572
8573 032602 004737 041524      JSR      PC, @#CLDISK        ;SET R1-RHCS1, R2-RHCS2
8574                                ;R3-RHDS1, R4-RHER1
8575                                ;GIVE RH-11 INITIALIZE
8576                                ;SETUP UNIT NUMBER
8577
8578                                ;*FILL WRITE FROM BUFFER WITH HEADER
8579 032606      ST1:
8580
8581 032606 004037 041350      JSR      R0, @#FLHEAD        ;SAVE HEADER DATA IN WRFROM
8582 032612 002470      WRFROM                          ;LOCATION WHERE SAVED
8583 032614 000004      4                                ;NUMBER OF WORDS SAVED
8584 032616 010000      10000                          ;FIRST DATA WORD
8585 032620 011025      <18.*400>!21.                  ;SECOND DATA WORD
8586 032622 000000      0                                ;THIRD DATA WORD
8587 032624 000000      0                                ;FOURTH DATA WORD
8588
8589                                ;*FILL WRITE FROM BUFFER WITH DATA
8590 032626      ST2:
8591 032626 004037 041374      JSR      R0, @#CLAREA        ;CLEAR 256. WORDS, FROM WRFROM+10
8592 032632 002500      WRFROM+10                      ;STARTING FROM WRFROM+10
8593 032634 000400      256.                          ;256. WORDS
8594 032636 001125      <0.*2000>!<18.*40>!21.        ;FILL WITH <0.*2000>!<18.*40>!21.
8595
8596
8597                                ;*FILL WRITE FROM BUFFER WITH NEXT TRACK HEADER
8598 032640      ST3:
8599
8600 032640 004037 041350      JSR      R0, @#FLHEAD        ;SAVE HEADER DATA IN WRFROM+<260.*2>
8601 032644 003500      WRFROM+<260.*2>                ;LOCATION WHERE SAVED
8602 032646 000004      4                                ;NUMBER OF WORDS SAVED
8603 032650 010001      10001                          ;FIRST DATA WORD
8604 032652 000000      0                                ;SECOND DATA WORD
8605 032654 000000      0                                ;THIRD DATA WORD
8606 032656 000000      0                                ;FOURTH DATA WORD
8607
8608                                ;*FILL WRITE FROM BUFFER WITH NEXT TRACK DATA
8609 032660      ST4:
8610 032660 004037 041374      JSR      R0, @#CLAREA        ;CLEAR 2 WORDS, FROM WRFROM+<264.*2>
8611 032664 003510      WRFROM+<264.*2>                ;STARTING FROM WRFROM+<264.*2>
8612 032666 000002      2                                ;2 WORDS
8613 032670 002000      1.*2000                       ;FILL WITH 1.*2000
8614

```

```

8615
8616
8617 032672          ST5:      ;*THE WRITE HEADER AND DATA COMMAND WILL BE FILLED
8618
8619 032672 004037 043476      JSR      RO,@#RUN      ;SETUP TO RUN FOR DATA COMMAND
8620 032676 000000              0          ;CYLINDER 0
8621 032700          025        .BYTE    21.         ;SECTOR 21.
8622 032701          022        .BYTE    18.         ;TRACK 18.
8623 032702 177366          -262.-4    ;WORD COUNT (DATA) = 262. +
8624
8625 032704 002470          WRFROM      ;4 HEADER WORDS
8626
8627
8628 032706 000000              0          ;BUS ADDRESS
8629 032710 010000          FMT22      ;STARTING ADDRESS OF DATA
8630
8631
8632 032712 002444          WRIFOR      ;BUFFER = WRFROM
8633
8634
8635
8636
8637
8638 032714 004037 041672      JSR      RO,@#SAVER    ;SAVE REGISTERS
8639 032720 002272          RHWC        ;RHWC IS THE FIRST REGISTER SAVED
8640 032722 004612          SAVERE      ;STARTING ADDRESS OF WHERE
8641
8642 032724 000022          18.         ;THE REGISTERS ARE SAVED
8643
8644
8645 032726 004737 041604      JSR      PC,@#CHECKT   ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
8646
8647 032732 104401 067033      TYPE    ,CPHALT      ;AND THAT NO STATUS BITS ARE STUCK = 1
8648
8649 032736 000000          HALT        ;CANNOT CONTINUE TESTING IF ANY OF
8650
8651 032740 013777 004606 147320  MOV      @#RP4VEC,@RPVEC ;THE FIRST SET OF BITS DON'T = 1
8652
8653
8654
8655
8656
8657
8658
8659 032746 013746 002444      MOV      @#WRIFOR,-(SP) ;SET RP04 VECTOR ADDRESS
8660 032752 052716 000101      BIS      #GO!IE,(SP)   ;TO 'TIME1' IF P-CLOCK IS PRESENT
8661
8662 032756 012677 147316      MOV      (SP)+,@RHCS1  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
8663
8664
8665 032762 011100              MOV      @R1,R0        ;'TIME' WILL ONLY SAVE
8666 032764 011305              MOV      @R3,R5        ;CURRENT CYLINDER ADDRESS
8667
8668
8669
8670

```

;WRITE HEADER AND DATA WITH 62 IN RHCS1
;*SAVE REGISTERS FOR COMPARISON AFTER WRITE HEADER AND DATA
;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
;AND THAT NO STATUS BITS ARE STUCK = 1
;CANNOT CONTINUE TESTING IF ANY OF
;THE FIRST SET OF BITS DON'T = 1
;STOP
;SET RP04 VECTOR ADDRESS
;TO 'TIME1' IF P-CLOCK IS PRESENT
;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
;'TIME' WILL ONLY SAVE
;CURRENT CYLINDER ADDRESS
;AND LOOK AHEAD REGISTERS
;GET READY TO MOVE COMMAND
;GET READY TO SET 'GO' AND
;ENABLE INTERRUPT
;GO WITH
;62 IN RHCS1 FOR WRITE HEADER AND DATA
;WITH INTERRUPT ENABLED
;SAVE RHCS1 DURING ABOVE OPERATION
;SAVE RHDS1 DURING ABOVE OPERATION
;*ONE REVOLUTION = 16670 MICRO SECONDS, ONE SECTOR - 760 MICRO SEC.
;*MAX TIME ALLOWED = ONE REVOLUTION + SEEK + 2 SECTORS
;*MIN TIME ALLOWED = 2 SECTORS + SEEK

```

8671
8672
8673 032766 104413          WAT          ;WAIT FOR RDY BIT TO SET
8674 032770 002300          RHCS1        ;WAIT FOR RHCS1 REGISTER
8675 032772 000200          RDY          ;WAIT FOR RDY BIT IN RHCS1 REGISTER
8676 032774 003237          1695.       ;ALLOW 16950 MICRO SECONDS
8677 032776 001515          845.        ;RDY MUST SET BETWEEN
8678                                     ;8500 AND 25400 MICRO SECONDS
8679
8680                                     ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
8681                                     ;*R0 AND R5 IMMEDIATELY AFTER GO
8682
8683 033000 013746 002444      MOV    @#WRIFOR,-(SP)    ;SAVE COMMAND
8684 033004 052716 004101      BIS    #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
8685 033010 011637 001124      MOV    (SP),@#SGDDAT    ;SAVE FOR PRINTOUT
8686 033014 022600             CMP    (SP)+,R0         ;DURING ABOVE OPERATION ONLY IE!GO!DVA
8687                                     ;AND COMMAND SHOULD BE SET
8688 033016 001405             BEQ    64$             ;BRANCH IF GOOD
8689 033020 010037 001126      MOV    R0,@#SBDDAT      ;BAD DATA
8690 033024 010137 004600      MOV    R1,@#REGADR      ;FAILING REGISTER RHCS1
8691 033030 104021             ERROR  21             ;DURING ABOVE OPERATION ONLY
8692                                     ;COMMAND AND IE!GO!DVA SHOULD BE SET
8693 033032 012746 010500      64$: MOV    #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
8694 033036 011637 001124      MOV    (SP),@#SGDDAT    ;SAVE FOR PRINTOUT
8695 033042 022605             CMP    (SP)+,R5         ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
8696                                     ;SHOULD BE SET
8697 033044 001405             BEQ    66$             ;BRANCH IF GOOD
8698 033046 010537 001126      MOV    R5,@#SBDDAT      ;BAD DATA
8699 033052 010337 004600      MOV    R3,@#REGADR      ;FAILING REGISTER RHDS1
8700 033056 104063             ERROR  63             ;DURING ABOVE OPERATION ONLY
8701                                     ;MOL!DPR!VV SHOULD BE SET
8702 033060             66$:
8703
8704                                     ;*NOW CHANGES SAVED REGISTERS TO EXPECTED VALUES
8705
8706 033060 004037 041426      JSR    R0,@#FILLRE      ;MOV 0 INTO SAVED RHWC
8707 033064 002272             RHWC          ;SAVED REGISTER TO CHANGE
8708 033066 000000             0            ;DATA
8709 033070 004037 041426      JSR    R0,@#FILLRE      ;MOV WRFROM+<266.*2> INTO SAVED RHBA
8710 033074 002274             RHBA          ;SAVED REGISTER TO CHANGE
8711 033076 003514             WRFROM+<266.*2> ;DATA
8712 033100             ST6:
8713 033100 004037 041426      JSR    R0,@#FILLRE      ;MOV 1 INTO SAVED RHCC
8714 033104 002334             RHCC          ;SAVED REGISTER TO CHANGE
8715 033106 000001             1            ;DATA
8716 033110 004037 041426      JSR    R0,@#FILLRE      ;MOV 1 INTO SAVED RHCA
8717 033114 002312             RHCA          ;SAVED REGISTER TO CHANGE
8718 033116 000001             1            ;DATA
8719 033120 004037 041426      JSR    R0,@#FILLRE      ;MOV 1 INTO SAVED RHDST
8720 033124 002304             RHDST        ;SAVED REGISTER TO CHANGE
8721 033126 000001             1            ;DATA
8722
8723                                     ;*COMPARE REGISTERS BEFORE WRITE HEADER AND DATA
8724                                     ;*WITH REGISTERS AFTER COMMAND
8725
8726

```

```

8727 033130 004037 042512 JSR RO,@#COMREG ;COMPARE SAVED REGISTERS WITH
8728 ;PRESENT VALUE
8729 033134 004612 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
8730 033136 002354 WC ;TEST DATA STARTING FROM 'RHWC'
8731 033140 000022 18. ;18. REGISTERS TO BE COMPARED
8732 033142 033146 ST7 ;RETURN TO ST7 ON ERROR
8733 033144 033152 ST8 ;RETURN TO ST8 ON NO ERROR
8734 033146 104027 ST7: ERROR 27 ;WRITE HEADER AND DATA CAUSED
8735 033150 000207 RTS PC ;IMPROPER REGISTER CHANGE
8736 ;GOOD DATA GIVES WHAT SHOULD BE
8737 ;THERE
8738 ;RECEIVED DATA GIVES WHAT WAS BE
8739 ;THERE AFTER COMMAND
8740
8741 ;*SETUP TO READ HEADER AND DATA FOR NEXT TRACK
8742 ;*FILL READ INTO BUFFER WITH ALL ONES
8743
8744 033152 ST8:
8745
8746 033152 004737 041524 JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
8747 ;R3-RHDS1, R4-RHER1
8748 ;GIVE RH-11 INITIALIZE
8749 ;SETUP UNIT NUMBER
8750 033156 004037 041374 JSR RO,@#CLAREA ;CLEAR 260. WORDS, FROM REINTO
8751 033162 003534 REINTO ;STARTING FROM REINTO
8752 033164 000404 260. ;260. WORDS
8753 033166 177777 -1 ;FILL WITH -1
8754
8755
8756 ;*FILL WRITE FROM BUFFER WITH EXPECTED DATA
8757
8758 033170 ST9:
8759
8760 033170 004037 041350 JSR RO,@#FLHEAD ;SAVE HEADER DATA IN WRFROM
8761 033174 002470 WRFROM ;LOCATION WHERE SAVED
8762 033176 000004 4 ;NUMBER OF WORDS SAVED
8763 033200 010001 10001 ;FIRST DATA WORD
8764 033202 000000 0 ;SECOND DATA WORD
8765 033204 000000 0 ;THIRD DATA WORD
8766 033206 000000 0 ;FOURTH DATA WORD
8767 033210 ST10:
8768 033210 004037 041374 JSR RO,@#CLAREA ;CLEAR 2 WORDS, FROM WRFROM+<4*2>
8769 033214 002500 WRFROM+<4*2> ;STARTING FROM WRFROM+<4*2>
8770 033216 000002 2 ;2 WORDS
8771 033220 002000 1*2000 ;FILL WITH 1*2000
8772
8773 033222 004037 041374 JSR RO,@#CLAREA ;CLEAR 254. WORDS, FROM WRFROM+<6*2>
8774 033226 002504 WRFROM+<6*2> ;STARTING FROM WRFROM+<6*2>
8775 033230 000376 254. ;254. WORDS
8776 033232 000000 0 ;FILL WITH 0
8777
8778
8779 ;*FILL COMMAND INTO REGISTERS
8780 033234 ST11:
8781
8782 033234 004037 043476 JSR RO,@#RUN ;SETUP TO RUN FOR DATA COMMAND
  
```



```

8839
8840      ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
8841      ;*R0 AND R5 IMMEDIATELY AFTER GO
8842
8843 033342 013746 002450      MOV    @#REFOR,-(SP)      ;SAVE COMMAND
8844 033346 052716 004101      BIS    #IE!GO!DVA,(SP)  ;INCLUDE IE!GO!DVA
8845 033352 011637 001124      MOV    (SP),@#$GDDAT    ;SAVE FOR PRINTOUT
8846 033356 022600      CMP    (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA
8847      ;AND COMMAND SHOULD BE SET
8848 033360 001405      BEQ    64$             ;BRANCH IF GOOD
8849 033362 010037 001126      MOV    R0,@#$BDDAT     ;BAD DATA
8850 033366 010137 004600      MOV    R1,@#REGADR     ;FAILING REGISTER RHCS1
8851 033372 104021      ERROR  21            ;DURING ABOVE OPERATION ONLY
8852      ;COMMAND AND IE!GO!DVA SHOULD BE SET
8853 033374 012746 010500      64$: MOV    #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
8854 033400 011637 001124      MOV    (SP),@#$GDDAT    ;SAVE FOR PRINTOUT
8855 033404 022605      CMP    (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
8856      ;SHOULD BE SET
8857 033406 001405      BEQ    66$             ;BRANCH IF GOOD
8858 033410 010537 001126      MOV    R5,@#$BDDAT     ;BAD DATA
8859 033414 010337 004600      MOV    R3,@#REGADR     ;FAILING REGISTER RHDS1
8860 033420 104063      ERROR  63            ;DURING ABOVE OPERATION ONLY
8861      ;MOL!DPR!VV SHOULD BE SET
8862 033422      66$:
8863
8864      ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
8865
8866 033422 004037 041426      JSR    R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
8867 033426 002272      RHWC                    ;SAVED REGISTER TO CHANGE
8868 033430 000000      0                        ;DATA
8869 033432 004037 041426      JSR    R0,@#FILLRE     ;MOV REINTO+<260.*2> INTO SAVED RHBA
8870 033436 002274      RHBA                    ;SAVED REGISTER TO CHANGE
8871 033440 004544      REINTO+<260.*2>         ;DATA
8872 033442 004037 041426      JSR    R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
8873 033446 002304      RHDST                   ;SAVED REGISTER TO CHANGE
8874 033450 000001      1                        ;DATA
8875
8876      ;*COMPARE REGISTERS BEFORE READ HEADER AND DATA WITH
8877      ;*REGISTERS AFTER COMMAND
8878
8879
8880 033452 004037 042512      JSR    R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
8881      ;PRESENT VALUE
8882 033456 004612      SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
8883 033460 002354      WC                      ;TEST DATA STARTING FROM 'RHWC'
8884 033462 000022      18.                     ;18. REGISTERS TO BE COMPARED
8885 033464 033470      ST12                    ;RETURN TO ST12 ON ERROR
8886 033466 033474      ST13                    ;RETURN TO ST13 ON NO ERROR
8887
8888 033470 104031      ST12: ERROR  31        ;READ HEADER AND DATA CAUSED
8889 033472 000207      RTS    PC              ;IMPROPER REGISTER CHANGE
8890      ;GOOD DATA GIVES WHAT SHOULD
8891      ;BE THERE
8892      ;RECEIVED DATA GIVES WHAT
8893      ;WAS THERE AFTER COMMAND
8894

```

```

8895                                     ;*READ INTO BUFFER IS CHECKED FOR PROPER READ
8896 033474                               ST13:
8897
8898 033474 004037 043542                 JSR    RO,@#COMPAR    ;COMPARE TWO BLOCKS OF MEMORY
8899 033500 002470                         WRFROM    ;GOOD DATA STARTS FROM WRFROM
8900 033502 003534                         REINTO    ;TEST DATA STARTS FROM REINTO
8901 033504 000404                         260.     ;260. WORDS TO BE COMPARED
8902 033506 033512                         ST14     ;RETURN TO ST14 ON ERROR
8903 033510 033516                         ST15     ;RETURN TO ST15 ON NO ERROR
8904
8905
8906 033512 104032                               ST14:  ERROR  32      ;WRITE HEADER AND DATA
8907 033514 000207                               RTS      PC          ;WITH AN IMPLIED SEEK
8908                                         ;FOLLOWED BY A READ
8909                                         ;HEADER AND DATA ON THE
8910                                         ;NEXT TRACK GAVE A
8911                                         ;READ ERROR
8912                                         ;ERROR MAY BE READ OR WRITE
8913
8914                                     ;*THE HEADS HAVE ADVANCED ONE CYLINDER BY AN IMPLIED
8915                                     ;*SEEK
8916                                     ;*CHANGES WILL BE MADE TO ENABLE GOING TO THE NEXT
8917                                     ;*CYLINDER AND THEN THE ABOVE WILL BE REPEATED
8918                                     ;*TILL CYLINDER 10 IS REACHED
8919
8920 033516 005237 032616                               ST15:  INC    @#ST1+10
8921 033522 062737 002000 032636                   ADD    #<1.*2000>,@#ST2+10
8922 033530 005237 032650                               INC    @#ST3+10
8923 033534 062737 002000 032670                   ADD    #<1.*2000>,@#ST4+10
8924 033542 005237 032676                               INC    @#ST5+4
8925 033546 005237 033106                               INC    @#ST6+6
8926 033552 005237 033116                               INC    @#ST6+16
8927 033556 005237 033200                               INC    @#ST9+10
8928 033562 062737 002000 033220                   ADD    #<1.*2000>,@#ST10+10
8929 033570 005237 033240                               INC    @#ST11+4
8930 033574 005337 001200                               DEC    @#STMP1      ;COUNT FOR TEN TIMES
8931 033600 001001                               BNE    ST16        ;BRANCH IF 10 NOT DONE
8932 033602 000402                               BR     ST17        ;10 COMPLETED SO CONTINUE
8933 033604 000137 032606                               ST16:  JMP    ST1      ;JUMP AS 10 NOT DONE
8934
8935                                     ;*THE HEADS ARE NOW AT CYLINDER 10
8936                                     ;*ALL REGISTERS WILL BE SAVED AND A SEEK WILL BE GIVEN
8937                                     ;*TO CYLINDER 0
8938                                     ;*FILL REGISTERS FOR A SEEK COMMAND
8939
8940 033610                               ST17:
8941
8942 033610 004737 041524                 JSR    PC,@#CLDISK   ;SET R1-RHCS1, R2-RHCS2
8943                                         ;R3-RHDS1, R4-RHER1
8944                                         ;GIVE RH-11 INITIALIZE
8945                                         ;SETUP UNIT NUMBER
8946
8947 033614 004037 041474                 JSR    RO,@#SEEKCY   ;SEEK FOR
8948 033620 000000                         0          ;CYLINDER 0
8949
8950                                     ;*SAVE REGISTERS FOR COMPARISON AFTER SEEK COMMAND
  
```

```

8951 033622 004037 041672 JSR R0,@#SAVER ;SAVE REGISTERS
8952 033626 002272 RHWC ;RHWC IS THE FIRST REGISTER SAVED
8953 033630 004612 SAVERE ;STARTING ADDRESS OF WHERE
8954 ;THE REGISTERS ARE SAVED
8955 033632 000022 18. ;NUMBER OF REGISTERS
8956 ;SAVED = 18.
8957
8958 033634 013777 004606 146424 MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
8959 ;TO 'TIME1' IF P-CLOCK IS PRESENT
8960 ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
8961 ;'TIME' WILL ONLY SAVE
8962 ;CURRENT CYLINDER ADDRESS
8963 ;AND LOOK AHEAD REGISTERS
8964
8965
8966 033642 013746 002452 MOV @#SEECOM,-(SP) ;GET READY TO MOVE COMMAND
8967 033646 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET 'GO' ,VD
8968 ;ENABLE INTERRUPT
8969 033652 012677 146422 MOV (SP)+,@RHCS1 ;GO WITH
8970 ;4 IN RHCS1 FOR SEEK
8971 ;WITH INTERRUPT ENABLED
8972 033656 011100 MOV @R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
8973 033660 011305 MOV @R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
8974
8975 ;*SEEK FOR ONE CYLINDER=7MILI SEC., FOR TEN=70 MILI SE
8976
8977 033662 104413 WAT ;WAIT FOR DRY BIT TO SET
8978 033664 002322 RHDS1 ;WAIT FOR RHDS1 REGISTER
8979 033666 000200 DRY ;WAIT FOR DRY BIT IN RHDS1 REGISTER
8980 033670 015530 7000. ;ALLOW 70000 MICRO SECONDS
8981 033672 000043 35. ;DRY MUST SET BETWEEN
8982 ;69650 AND 70350 MICRO SECONDS
8983
8984 ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
8985 ;*R0 AND R5 IMMEDIATELY AFTER GO
8986
8987 033674 013746 002452 MOV @#SEECOM,-(SP) ;SAVE COMMAND
8988 033700 052716 004301 BIS #DVA!GO!IE!RDY,(SP) ;INCLUDE DVA!GO!IE!RDY
8989 033704 011637 001124 MOV (SP),@#SGDDAT ;SAVE FOR PRINTOUT
8990 033710 022600 CMP (SP)+,R0 ;DURING ABOVE OPERATION ONLY DVA!GO!IE!RDY
8991 ;AND COMMAND SHOULD BE SET
8992 033712 001405 BEQ 64$ ;BRANCH IF GOOD
8993 033714 010037 001126 MOV R0,@#SBDDAT ;BAD DATA
8994 033720 010137 004600 MOV R1,@#REGADR ;FAILING REGISTER RHCS1
8995 033724 104021 ERROR 21 ;DURING ABOVE OPERATION ONLY
8996 ;COMMAND AND DVA!GO!IE!RDY SHOULD BE SET
8997 033726 012746 030500 64$: MOV #PIP!MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
8998 033732 011637 001124 MOV (SP),@#SGDDAT ;SAVE FOR PRINTOUT
8999 033736 022605 CMP (SP)+,R5 ;DURING ABOVE OPERATION ONLY PIP.MOL!DPR.VV
9000 ;SHOULD BE SET
9001 033740 001405 BEQ 66$ ;BRANCH IF GOOD
9002 033742 010537 001126 MOV R5,@#SBDDAT ;BAD DATA
9003 033746 010337 004600 MOV R3,@#REGADR ;FAILING REGISTER RHDS1
9004 033752 104063 ERROR 63 ;DURING ABOVE OPERATION ONLY
9005 ;PIP!MOL.DPR!VV SHOULD BE SET
9006 033754 66$:

```

```

9007
9008
9009
9010 033754 004037 041426 JSR RO,@#FILLRE ;MOV 0 INTO SAVED RHCC
9011 033760 002334 RHCC ;SAVED REGISTER TO CHANGE
9012 033762 000000 0 ;DATA
9013
9014 033764 053737 004740 004636 BIS @#ATTENT,@#SAVERE+24 ;SET APPROPRIATE 'ATA' BITS
9015 ;FOR WORKING DRIVE IN
9016 ;SAVED RHAS LOACTION
9017
9018 033772 004037 042404 JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
9019 033776 002322 RHDS1 ;CHANGE RHDS1 REGISTER
9020
9021 034000 000001 1 ;1 BIT/BITS TO BE CHANGED
9022 034002 000001 1 ;NEW VALUE OF ATA IS 1
9023 034004 100000 ATA ;CHANGE ATA BIT
9024
9025 034006 004037 042404 JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
9026 034012 002300 RHCS1 ;CHANGE RHCS1 REGISTER
9027
9028 034014 000001 1 ;1 BIT/BITS TO BE CHANGED
9029 034016 000001 1 ;NEW VALUE OF SC IS 1
9030 034020 100000 SC ;CHANGE SC BIT
9031
9032 ;*COMPARE REGISTERS AFTER A SEEK COMMAND
9033
9034 034022 004037 042512 JSR RO,@#COMREG ;COMPARE SAVED REGISTERS WITH
9035 ;PRESENT VALUE
9036 034026 004612 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
9037 034030 002354 WC ;TEST DATA STARTING FROM 'RHWC'
9038 034032 000022 18. ;18. REGISTERS TO BE COMPARED
9039 034034 034040 ST18 ;RETURN TO ST18 ON ERROR
9040 034036 034044 ST19 ;RETURN TO ST19 ON NO ERROR
9041
9042 034040 104037 ST18: ERROR 37 ;SEEK COMMAND CAUSED AN
9043 034042 000207 RTS PC ;ERROR
9044 ;GOOD DATA GIVES WHAT SHOULD
9045 ;BE THERE
9046 ;RECEIVED DATA GIVES WHAT WAS
9047 ;THERE AFTER A SEEK COMMAND
9048
9049 ;*AT THIS POINT THE CURRENT CYLINDER IS GOOD AND THERE ARE
9050 ;*NO ERROR BITS
9051 ;*A READ HEADER AND DATA WILL BE DONE ON CYLINDER 0
9052 ;*SECTOR 21 TRACK 18, EXPECTED DATA IS 1125
9053 ;*FOR 10 WORDS
9054 ;*CLEAR READ INTO BUFFER WITH ALL ONES
9055
9056 034044 ST19: JSR RO,@#CLAREA ;CLEAR 260. WORDS, FROM REINTO
9057 034044 004037 041374 REINTO ;STARTING FROM REINTO
9058 034050 003534 260. ;260. WORDS
9059 034052 000404 -1 ;FILL WITH -1
9060 034054 177777
9061
9062

```

```
9063 ;*FILL WRITE FROM BUFFER WITH EXPECTED DATA
9064
9065 034056 004037 041350 JSR RO,@#FLHEAD ;SAVE HEADER DATA IN WRFROM
9066 034062 002470 WRFROM ;LOCATION WHERE SAVED
9067 034064 000004 4 ;NUMBER OF WORDS SAVED
9068 034066 010000 10000 ;FIRST DATA WORD
9069 034070 011025 <18.*40>!<21.> ;SECOND DATA WORD
9070 034072 000000 0 ;THIRD DATA WORD
9071 034074 000000 0 ;FOURTH DATA WORD
9072 034076 004037 041374 JSR RO,@#CLAREA ;CLEAR 10. WORDS, FROM WRFROM+<4.*2>
9073 034102 002500 WRFROM+<4.*2> ;STARTING FROM WRFROM+<4.*2>
9074 034104 000012 10. ;10. WORDS
9075 034106 001125 <18.*40>!<21.> ;FILL WITH <18.*40>!<21.>
9076
9077 034110 004037 041374 JSR RO,@#CLAREA ;CLEAR 246. WORDS, FROM WRFROM+<14.*2>
9078 034114 002524 WRFROM+<14.*2> ;STARTING FROM WRFROM+<14.*2>
9079 034116 000366 246. ;246. WORDS
9080 034120 177777 -1 ;FILL WITH -1
9081
9082
9083 ;*FILL READ HEADER AND DATA COMMAND FOR 10 WORDS
9084
9085 034122 004037 043476 JSR RO,@#RUN ;SETUP TO RUN FOR DATA COMMAND
9086 034126 000000 0 ;CYLINDER 0
9087 034130 025 .BYTE 21. ;SECTOR 21.
9088 034131 022 .BYTE 18. ;TRACK 18.
9089 034132 177762 -10.-4 ;WORD COUNT (DATA) = 10. +
9090 ;4 HEADER WORDS
9091 034134 003534 REINTO ;BUS ADDRESS
9092 ;STARTING ADDRESS OF DATA
9093 ;BUFFER = REINTO
9094 034136 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
9095 034140 014000 ECI!FMT22 ;16 BITS PER WORD FORMAT
9096 ;INHIBIT ECC CORRECTION
9097 ;DO NOT INHIBIT HEADER COMPARE
9098 034142 002450 REFOR ;GET READY TO DO A REFOR
9099 ;READ HEADER AND DATA WITH 72 IN RHCS1
9100
9101
9102 034144 004737 041604 JSR PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
9103 ;AND THAT NO STATUS BITS ARE STUCK = 1
9104 034150 104401 067033 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF ANY OF
9105 ;THE FIRST SET OF BITS DON'T = 1
9106 034154 000000 HALT ;STOP
9107
9108 034156 013777 004606 146102 MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
9109 ;TO 'TIME1' IF P-CLOCK IS PRESENT
9110 ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
9111 ;'TIME' WILL ONLY SAVE
9112 ;CURRENT CYLINDER ADDRESS
9113 ;AND LOOK AHEAD REGISTERS
9114
9115
9116 034164 013746 002450 MOV @#REFOR,-(SP) ;GET READY TO MOVE COMMAND
9117 034170 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET 'GO' AND
9118 ;ENABLE INTERRUPT
```

```

9119 034174 012677 146100      MOV      (SP)+,@RHCS1      ;GO WITH
9120                                     ;72 IN RHCS1 FOR READ DATA
9121                                     ;WITH INTERRUPT ENABLED
9122
9123
9124 034200 104413      WAT      ;WAIT FOR RDY BIT TO SET
9125 034202 002300      RHCS1    ;WAIT FOR RHCS1 REGISTER
9126 034204 000200      RDY      ;WAIT FOR RDY BIT IN RHCS1 REGISTER
9127 034206 001614      908.    ;ALLOW 9080 MICRO SECONDS
9128 034210 001507      839.    ;RDY MUST SET BETWEEN
9129                                     ;690 AND 17470 MICRO SECONDS
9130
9131                                     ;*CHECK READ WORDS
9132
9133 034212 004037 043542      JSR      R0,@#COMPAR      ;COMPARE TWO BLOCKS OF MEMORY
9134 034216 002470      WRFROM   ;GOOD DATA STARTS FROM WRFROM
9135 034220 003534      REINTO   ;TEST DATA STARTS FROM REINTO
9136 034222 000404      260.    ;260. WORDS TO BE COMPARED
9137 034224 034230      ST26     ;RETURN TO ST26 ON ERROR
9138 034226 034234      ST27     ;RETURN TO ST27 ON NO ERROR
9139
9140
9141 034230 104032      ST26:   ERROR 32      ;READ HEADER AND DATA
9142 034232 000207      RTS     PC          ;FOLLOWING A SEEK TO CYLINDER 0
9143                                     ;FROM CYLINDER 10 GAVE AN
9144                                     ;ERROR
9145
9146 034234      ST27:
9147

```

```
9148
9149
9150
9151
9152
9153
9154
9155
9156
9157
9158
9159
9160
9161
9162 034234 000004
9163 034236 012706 001000
9164 034242 012737 000044 004604
9165
9166 034250 004737 041524
9167
9168
9169
9170
9171
9172 034254 013777 004606 146004
9173
9174
9175
9176
9177
9178
9179
9180 034262 013746 002426
9181 034266 052716 000101
9182
9183 034272 012677 146002
9184
9185
9186 034276 011100
9187 034300 011305
9188
9189
9190 034302 104413
9191 034304 002322
9192 034306 000200
9193 034310 076377
9194 034312 056701
9195
9196
9197
9198
9199 034314 004037 041474
9200 034320 000012
9201
9202
9203 034322 004037 041672
```

```
*****
*TEST 44      SEEK & READ TEST (CYL = 009)
*****
*      THIS TEST DEPENDS ON HEADER AND DATA WRITTEN BY THE
*      PREVIOUS TEST.  AT THIS POINT THE HEADS ARE ON
*      CYLINDER 0
*
*      ALL REGISTERS ARE SAVED
*      A SEEK IS GIVEN TO CYLINDER 9
*      ALL REGISTERS ARE CHECKED FOR IMPROPER CHANGE
*      DATA WRITTEN ON CYLINDER 9 IS CHECKED
*****
TST44:  SCOPE
        MOV      #STACK,SP      ;RESET STACK
        MOV      #44,@#TSTNM    ;SAVE TEST NUMBER
        JSR      PC,@#CLDISK    ;SET R1-RHCS1 R2-RHCS2
                                   ;R3-RHDS1, R4-RHER1
                                   ;GIVE RH-1' INITIALIZE
                                   ;SETUP UNIT NUMBER
*THIS GETS HEADS TO CYLINDER 0
        MOV      @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
                                   ;TO 'TIME1' IF P-CLOCK IS PRESENT
                                   ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
                                   ;'TIME' WILL ONLY SAVE
                                   ;CURRENT CYLINDER ADDRESS
                                   ;AND LOOK AHEAD REGISTERS
        MOV      @#RECALI,-(SP)  ;GET READY TO MOVE COMMAND
        BIS      #GO!IE,(SP)    ;GET READY TO SET 'GO' AND
                                   ;ENABLE INTERRUPT
        MOV      (SP)+,@RHCS1    ;GO WITH
                                   ;6 IN RHCS1 FOR RECALIBRATE
                                   ;WITH INTERRUPT ENABLED
        MOV      @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
        MOV      @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
        WAT                      ;WAIT FOR DRY BIT TO SET
        RHDS1                    ;WAIT FOR RHDS1 REGISTER
        DRY                      ;WAIT FOR DRY BIT IN RHDS1 REGISTER
        31999.                   ;ALLOW 319990 MICRO SECONDS
        24001.                   ;DRY MUST SET BETWEEN
                                   ;79980 AND 560000 MICRO SECONDS
*FILL REGISTERS FOR A SEEK COMMAND
        JSR      R0,@#SEEKCY    ;SEEK FOR
        10.                      ;CYLINDER 10.
*SAVE REGISTERS FOR COMPARISON AFTER SEEK COMMAND
        JSR      R0,@#SAVER     ;SAVE REGISTERS
```

```

CZRJID0, RP04/5/6 FCTNL CTLR1 MACY11 30A(1052) 25-MAY-79 10:30 1 15 PAGE 191
CZRJID.P11 28-MAR-79 09:03 T44 SEEK & READ TEST (CYL = 0J9) SEQ 0190

9204 034326 002272 RHWC ;RHWC IS THE FIRST REGISTER SAVED
9205 034330 004612 SAVERE ;STARTING ADDRESS OF WHERE
9206 ;THE REGISTERS ARE SAVED
9207 034332 000022 18. ;NUMBER OF REGISTERS
9208 ;SAVED = 18.
9209
9210 034334 013777 004606 145724 MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
9211 ;TO 'TIME1' IF P-CLOCK IS PRESENT
9212 ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
9213 ;'TIME' WILL ONLY SAVE
9214 ;CURRENT CYLINDER ADDRESS
9215 ;AND LOOK AHEAD REGISTERS
9216
9217
9218 034342 013746 002452 MOV @#SEECOM,-(SP) ;GET READY TO MOVE COMMAND
9219 034346 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET 'GO' AND
9220 ;ENABLE INTERRUPT
9221 034352 012677 145722 MOV (SP)+,@RHCS1 ;GO WITH
9222 ;4 IN RHCS1 FOR SEEK
9223 ;WITH INTERRUPT ENABLED
9224 034356 011100 MOV @R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
9225 034360 011305 MOV @R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
9226
9227 ;*SEEK FOR ONE CYLINDER=7 MILI SEC., FOR TEN=70 MILI SEC
9228
9229 034362 104413 WAT ;WAIT FOR DRY BIT TO SET
9230 034364 002322 RHDS1 ;WAIT FOR RHDS1 REGISTER
9231 034366 000200 DRY ;WAIT FOR DRY BIT IN RHDS1 REGISTER
9232 034370 015530 7000. ;ALLOW 7000 MICRO SECONDS
9233 034372 000043 35. ;DRY MUST SET BETWEEN
9234 ;69650 AND 70350 MICRO SECONDS
9235
9236 ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
9237 ;*R0 AND R5 IMMEDIATELY AFTER GO
9238
9239 034374 013746 002452 MOV @#SEECOM,-(SP) ;SAVE COMMAND
9240 034400 052716 004301 BIS #DVA!GO!IE!RDY,(SP) ;INCLUDE DVA!GO!IE!RDY
9241 034404 011637 001124 MOV (SP),@#SGDDAT ;SAVE FOR PRINTOUT
9242 034410 022600 CMP (SP)+,R0 ;DURING ABOVE OPERATION ONLY DVA.GO.IE.RDY
9243 ;AND COMMAND SHOULD BE SET
9244 034412 001405 BEQ 67$ ;BRANCH IF GOOD
9245 034414 010037 001126 MOV R0,@#SBDDAT ;BAD DATA
9246 034420 010137 004600 MOV R1,@#REGADR ;FAILING REGISTER RHCS1
9247 034424 104021 ERROR 21 ;DURING ABOVE OPERATION ONLY
9248 ;COMMAND AND DVA!GO!IE!RDY SHOULD BE SET
9249 034426 012746 030500 67$: MOV #PIP!MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
9250 034432 011637 001124 MOV (SP),@#SGDDAT ;SAVE FOR PRINTOUT
9251 034436 022605 CMP (SP)+,R5 ;DURING ABOVE OPERATION ONLY PIP.MOL.DPR!VV
9252 ;SHOULD BE SET
9253 034440 001405 BEQ 69$ ;BRANCH IF GOOD
9254 034442 010537 001126 MOV R5,@#SBDDAT ;BAD DATA
9255 034446 010337 004600 MOV R3,@#REGADR ;FAILING REGISTER RHDS1
9256 034452 104063 ERROR 63 ;DURING ABOVE OPERATION ONLY
9257 ;PIP!MOL!DPR!VV SHOULD BE SET
9258 034454 69$:
9259

```



```

9260 ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
9261
9262 034454 004037 041426 JSR RO,@#FILLRE ;MOV 10. INTO SAVED RHCC
9263 034460 002334 RHCC ;SAVED REGISTER TO CHANGE
9264 034462 000012 10. ;DATA
9265
9266 034464 053737 004740 004636 BIS @#ATTENT,@#SAVERE+24 ;SET APPROPRIATE 'ATA' BITS
9267 ;FOR WORKING DRIVE IN
9268 ;SAVED RHAS LOACTION
9269
9270 034472 004037 042404 JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
9271 034476 002322 RHDS1 ;CHANGE RHDS1 REGISTER
9272
9273 034500 000001 1 ;1 BIT/BITS TO BE CHANGED
9274 034502 000001 1 ;NEW VALUE OF ATA IS 1
9275 034504 100000 ATA ;CHANGE ATA BIT
9276
9277 034506 004037 042404 JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
9278 034512 002300 RHCS1 ;CHANGE RHCS1 REGISTER
9279
9280 034514 000001 1 ;1 BIT/BITS TO BE CHANGED
9281 034516 000001 1 ;NEW VALUE OF SC IS 1
9282 034520 100000 SC ;CHANGE SC BIT
9283
9284 ;*COMPARE REGISTERS AFTER A SEEK COMMAND
9285
9286 034522 004037 042512 JSR RO,@#COMREG ;COMPARE SAVED REG' STERS WITH
9287 ;PRESENT VALUE
9288 034526 004612 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
9289 034530 002354 WC ;TEST DATA STARTING FROM 'RHW'
9290 034532 000022 18. ;18. REGISTERS TO BE COMPARED
9291 034534 034540 1$ ;RETURN TO 1$ ON ERROR
9292 034536 034544 2$ ;RETURN TO 2$ ON NO ERROR
9293
9294 034540 104037 1$: ERROR 37 ;SEEK COMMAND CAUSED
9295 034542 000207 RTS PC ;ERROR
9296 ;GOOD DATA GIVES WHAT SHOULD
9297 ;BE THERE
9298 ;RECEIVED DATA GIVES WHAT WAS
9299 ;THERE AFTER A SEEK COMMAND
9300
9301 ;*AT THIS POINT THE CURRENT CYLINDER IS GOOD AND THERE ARE
9302 ;*NO ERROR BITS
9303 ;*A READ HEADER AND DATA WILL BE DONE ON CYLINDER 9
9304 ;*SECTOR 21 TRACK 18, EXPECTED DATA IS 23125
9305 ;*FOR 20 WORDS
9306 ;*CLEAR READ INTO BUFFER WITH ALL ONES
9307
9308 034544 2$:
9309
9310 034544 004737 041524 JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
9311 ;R3-RHDS1, R4-RHER1
9312 ;GIVE RH-11 INITIALIZE
9313 ;SETUP UNIT NUMBER
9314 034550 004037 041374 JSR RO,@#CLAREA ;CLEAR 260. WORDS, FROM REINTO
9315 034554 003534 REINTO ;STARTING FROM REINTO

```



```

9372
9373
9374 034670 013746 002450      MOV    @#REFOR,-(SP)      ;GET READY TO MOVE COMMAND
9375 034674 052716 000101      BIS    #GO!IE,(SP)      ;GET READY TO SET 'GO' AND
9376                                     ;ENABLE INTERRUPT
9377 034700 012677 145374      MOV    (SP)+,@RHCS1     ;GO WITH
9378                                     ;72 IN RHCS1 FOR READ DATA
9379                                     ;WITH INTERRUPT ENABLED
9380
9381
9382 034704 044413              WAT                                     ;WAIT FOR RDY BIT TO SET
9383 034706 002300              RHCS1                                ;WAIT FOR RHCS1 REGISTER
9384 034710 000200              RDY                                  ;WAIT FOR RDY BIT IN RHCS1 REGISTER
9385 034712 001614              908.                                ;ALLOW 9080 MICRO SECONDS
9386 034714 001507              839.                                ;RDY MUST SET BETWEEN
9387                                     ;690 AND 17470 MICRO SECONDS
9388
9389                                     ;*CHECK READ WORDS
9390
9391 034716 004037 043542      JSR    R0,@#COMPAR      ;COMPARE TWO BLOCKS OF MEMORY
9392 034722 002470              WRFROM                            ;GOOD DATA STARTS FROM WRFROM
9393 034724 003534              REINTO                             ;TEST DATA STARTS FROM REINTO
9394 034726 000404              260.                               ;260. WORDS TO BE COMPARED
9395 034730 034734              3$                                 ;RETURN TO 3$ ON ERROR
9396 034732 034740              4$                                 ;RETURN TO 4$ ON NO ERROR
9397
9398
9399 034734 104032              3$:  ERROR  32              ;READ HEADER AND DATA
9400 034736 000207              RTS    PC                        ;FOLLOWING A SEEK TO CYLINDER 9
9401                                     ;FROM CYLINDER 0 GAVE AN
9402                                     ;ERROR
9403 034740              4$:
9404
9405
9406
9407
  
```

CZRJDO, RP04/5/6 FCTNL CTLR1
CZRJID.P11 28-MAR-79 09:03

MACY11 30A(1052) 25-MAY-79 10:30 PAGE . .
WRITE CHECK DATA & WRITE PROTECT TESTS

M 15

SEQ 0194

9408
9409

.SBTTL WRITE CHECK DATA & WRITE PROTECT TESTS

9410
9411
9412
9413
9414
9415
9416
9417
9418
9419
9420
9421
9422
9423
9424
9425
9426
9427
9428
9429
9430
9431
9432
9433
9434
9435
9436
9437
9438
9439
9440
9441
9442
9443
9444
9445
9446
9447
9448
9449
9450
9451
9452
9453
9454
9455
9456
9457
9458
9459
9460
9461
9462
9463
9464
9465

034740 000004
034742 012706 001000
034746 012737 000045 004604
034754 004737 041524
034760 004737 041604
034764 104401 067033
034770 000000
034772 004037 041474
034776 000005
035000 013777 004606 145260
035006 013746 002452
035012 052716 000101
035016 012677 145256

```
*****  
: *TEST 45 WRITE CHECK HEADER AND DATA  
: *  
: * WRITE CHECK HEADER AND DATA CYLINDER 5 FORMAT 16 BITS PER WORD  
: * TRACK 7, SECTOR 4, KEYS 0, NUMBER OF WORDS 266  
: * CONSISTING OF  
: * 10 WORDS OF 12344 (6 BITS FOR CYL, 5 FOR TRACK, 5 FOR SECTOR)  
: * 10 WORDS OF 177777  
: * 10 WORDS OF 0  
: * 10 WORDS OF 052525  
: * 10 WORDS OF 125252  
: * 16 WORDS OF LEFT ROTATING ZERO (EG 177776,177775)  
: * 16 WORDS OF LEFT ROTATING ONE (EG 1,2,4,10)  
: * 174 WORDS OF 377  
: * 4 WORDS OF HEADER  
: * 2 WORDS OF 12345  
: *  
: * FIRST THE ABOVE DATA IS WRITTEN BY A WRITE HEADER AND  
: * DATA COMMAND  
: * CHECK FOR NO ERRORS  
: * THEN THE ABOVE DATA IS READ BY A READ HEADER AND DATA  
: * CHECK FOR NO ERRORS  
: * THEN THE ABOVE WRITE CHECK HEADER AND DATA IS GIVEN  
: *  
: *  
TST45: SCOPE  
MOV #STACK,SP ;RESET STACK  
MOV #45,@#TSTNM ;SAVE TEST NUMBER  
JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2  
;R3-RHDS1, R4-RHER1  
;GIVE RH-11 INITIALIZE  
;SETUP UNIT NUMBER  
JSR PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1  
;AND THAT NO STATUS BITS ARE STUCK = 1  
TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF ANY OF  
;THE FIRST SET OF BITS DON'T = 1  
HALT ;STOP  
: *GET HEADS TO CYLINDER 5  
JSR R0,@#SEEKCY ;SEEK FOR  
5 ;CYLINDER 5  
MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS  
;TO 'TIME1' IF P-CLOCK IS PRESENT  
;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT  
;'TIME' WILL ONLY SAVE  
;CURRENT CYLINDER ADDRESS  
;AND LOOK AHEAD REGISTERS  
MOV @#SEECOM,-(SP) ;GET READY TO MOVE COMMAND  
BIS #GO.IE,(SP) ;GET READY TO SET 'GO' AND  
;ENABLE INTERRUPT  
MOV (SP)+,@RHCS1 ;GO WITH
```



```

9522
9523      ;*FILL LEFT ROTATING ZEROS FROM WRFROM+<54.*2>
9524
9525 035146 010146      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
9526 035150 012700 177776  MOV      #177776,R0    ;:DATA
9527 035154 012705 000020  MOV      #16.,R5      ;:COUNT
9528 035160 012701 002644  MOV      #WRFROM+<54.*2>,R1 ;:WHERE DATA GOES
9529 035164 000261
9530 035166 010021      1$: MOV      RO,(R1)+    ;:STORE DATA
9531 035170 006100      ROL      RO          ;:GET ZERO ONE BIT LEFT
9532 035172 005305      DEC      R5          ;:COUNT 16
9533 035174 001374      BNE     1$          ;:BRANCH IF 16 NOT DONE
9534
9535      ;*FILL LEFT ROTATING ONE INTO WRFROM+<65.*2>
9536
9537 035176 000241      CLC
9538 035200 012700 000001  MOV      #1,R0
9539 035204 010021      2$: MOV      RO,(R1)+
9540 035206 006300      ASL     RO
9541 035210 103375      P/C    2$
9542 035212 012601      MOV      (SP)+,R1    ;;POP STACK INTO R1
9543
9544
9545      ;*FILL REST OF DATA
9546
9547 035214 004037 041374  JSR      RO,@#CLAREA ;:CLEAR 174. WORDS, FROM WRFROM+<86.*2>
9548 035220 002744      WRFROM+<86.*2>      ;:STARTING FROM WRFROM+<86.*2>
9549 035222 000256      174.              ;:174. WORDS
9550 035224 000377      377               ;:FILL WITH 377
9551
9552
9553 035226 004037 041350  JSR      RO,@#FLHEAD ;:SAVE HEADER DATA IN WRFROM+<260.*2>
9554 035232 003500      WRFROM+<260.*2>    ;:LOCATION WHERE SAVED
9555 035234 000004      4                 ;:NUMBER OF WORDS SAVED
9556 035236 010005      10005            ;:FIRST DATA WORD
9557 035240 003405      <7*400>!5       ;:SECOND DATA WORD
9558 035242 000000      0                 ;:THIRD DATA WORD
9559 035244 000000      0                 ;:FOURTH DATA WORD
9560 035246 004037 041374  JSR      RO,@#CLAREA ;:CLEAR 2 WORDS, FROM WRFROM+<264.*2>
9561 035252 003510      WRFROM+<264.*2>    ;:STARTING FROM WRFROM+<264.*2>
9562 035254 000002      2                 ;:2 WORDS
9563 035256 012345      <5*2000>!<7*40>!5 ;:FILL WITH <5*2000>!<7*40>!5
9564
9565
9566
9567      ;*READ INTO BUFFER WILL BE CLEARED
9568 035260 004037 041374  JSR      RO,@#CLAREA ;:CLEAR 266. WORDS, FROM REINTO
9569 035264 003534      REINTO           ;:STARTING FROM REINTO
9570 035266 000412      266.            ;:266. WORDS
9571 035270 177400      177400          ;:FILL WITH 177400
9572
9573
9574      ;*THE WRITE HEADER AND DATA COMMAND WILL BE LOADED
9575
9576 035272 004037 043476  JSR      RO,@#RUN    ;:SETUP TO RUN FOR DATA COMMAND
9577 035276 000005      5                ;:CYLINDER 5

```

CZRJID0, RP04/5/6 FCTNL CTLR1
CZRJID.P11 28-MAR-79 09:03

MACY11 30A(1052) 25-MAY-79 10:30 PAGE 199
T45 WRITE CHECK HEADER AND DATA

SEQ 0198

```

9578 035300 004 .BYTE 4 ;SECTOR 4
9579 035301 007 .BYTE 7 ;TRACK 7
9580 035302 177366 -262.-4 ;WORD COUNT (DATA) = 262. +
9581 ;4 HEADER WORDS
9582 035304 002470 WRFROM ;BUS ADDRESS
9583 ;STARTING ADDRESS OF DATA
9584 ;BUFFER = WRFROM
9585 035306 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
9586 035310 010000 FMT22 ;16 BITS PER WORD FORMAT
9587 ;DO NOT INHIBIT ECC CORRECTION
9588 ;DO NOT INHIBIT HEADER COMPARE
9589 035312 002444 WRIFOR ;GET READY TO DO A WRIFOR
9590 ;WRITE HEADER AND DATA WITH 62 IN RHCS1
9591
9592
9593 ;*NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE HEADER AND DATA ;
9594
9595 035314 004037 041672 JSR RO,@#SAVER ;SAVE REGISTERS
9596 035320 002272 RHWC ;RHWC IS THE FIRST REGISTER SAVED
9597 035322 004612 SAVERE ;STARTING ADDRESS OF WHERE
9598 ;THE REGISTERS ARE SAVED
9599 035324 000022 18. ;NUMBER OF REGISTERS
9600 ;SAVED = 18.
9601
9602 035326 004737 041604 JSR PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
9603 ;AND THAT NO STATUS BITS ARE STUCK = 1
9604 035332 104401 067033 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF ANY OF
9605 ;THE FIRST SET OF BITS DON'T = 1
9606 035336 000000 HALT ;STOP
9607
9608 035340 013777 004606 144720 MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
9609 ;TO 'TIME1' IF P-CLOCK IS PRESENT
9610 ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
9611 ;'TIME' WILL ONLY SAVE
9612 ;CURRENT CYLINDER ADDRESS
9613 ;AND LOOK AHEAD REGISTERS
9614
9615
9616 035346 013746 002444 MOV @#WRIFOR,-(SP) ;GET READY TO MOVE COMMAND
9617 035352 052716 0001C1 BIS #GO!IE,(SP) ;GET READY TO SET 'GO' AND
9618 ;ENABLE INTERRUPT
9619 035356 012677 144716 MOV (SP)+,@RHCS1 ;GO WITH
9620 ;62 IN RHCS1 FOR WRITE HEADER AND DATA
9621 ;WITH INTERRUPT ENABLED
9622
9623
9624 035362 104413 WAT ;WAIT FOR RDY BIT TO SET
9625 035364 002300 RHCS1 ;WAIT FOR RHCS1 REGISTER
9626 035366 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
9627 035370 001732 986. ;ALLOW 9860 MICRO SECONDS
9628 035372 001502 834. ;RDY MUST SET BETWEEN
9629 ;1520 AND 18200 MICRO SECONDS
9630
9631 ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
9632 035374 004037 041426 JSR RO,@#FILLRE ;MOV 0 INTO SAVED RHWC
9633 035400 002272 RHWC ;SAVED REGISTER TO CHANGE

```


CZRJID, RP04/5/6 FCTNL CTLR1
CZRJID.P11 28-MAR-79 09:03

MACY11 30A(1052) 25-MAY-79 10:30 PAGE 200
T45 WRITE CHECK HEADER AND DATA

SEQ 0199

```

9634 035402 000000 0 ;DATA
9635 035404 004037 041426 JSR RO,@#FILLRE ;MOV WRFROM+<266.*2> INTO SAVED RHBA
9636 035410 002274 RHBA ;SAVED REGISTER TO CHANGE
9637 035412 003514 WRFROM+<266.*2> ;DATA
9638 035414 004037 041426 JSR RO,@#FILLRE ;MOV 3406 INTO SAVED RHDST
9639 035420 002304 RHDST ;SAVED REGISTER TO CHANGE
9640 035422 003406 3406 ;DATA
9641
9642 ;*NOW COMPARE REGISTERS BEFORE WRITE HEADER AND DATA
9643 ;*WITH REGISTERS AFTER COMMAND
9644
9645
9646 035424 004037 042512 JSR RO,@#COMREG ;COMPARE SAVED REGISTERS WITH
9647 ;PRESENT VALUE
9648 035430 004612 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
9649 035432 002354 WC ;TEST DATA STARTING FROM 'RHWC'
9650 035434 000022 '8. ;18. REGISTERS TO BE COMPARED
9651 035436 035442 3$ ;RETURN TO 3$ ON ERROR
9652 035440 035446 4$ ;RETURN TO 4$ ON NO ERROR
9653
9654 035442 104027 3$: ERROR 27 ;WRITE HEADER AND DATA
9655 035444 000207 RTS PC ;CAUSED IMPROPER REGISTER
9656 ;CHANGE
9657 ;GOOD DATA GIVES WHAT SHOULD
9658 ;BE THERE
9659 ;RECEIVED DATA GIVES WHAT
9660 ;WAS THERE AFTER COMMAND
9661
9662 ;*NOW FILL COMMAND FOR READ
9663
9664 035446 4$:
9665
9666 035446 004737 041524 JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
9667 ;R3-RHDS1, R4-RHER1
9668 ;GIVE RH-11 INITIALIZE
9669 ;SETUP UNIT NUMBER
9670
9671 035452 004037 043476 JSR RO,@#RUN ;SETUP TO RUN FOR DATA COMMAND
9672 035456 000005 5 ;CYLINDER 5
9673 035460 004 .BYTE 4 ;SECTOR 4
9674 035461 007 .BYTE 7 ;TRACK 7
9675 035462 177366 -262.-4 ;WORD COUNT (DATA) = 262. +
9676 ;4 HEADER WORDS
9677 035464 003534 REINTO ;BUS ADDRESS
9678 ;STARTING ADDRESS OF DATA
9679 ;BUFFER = REINTO
9680 035466 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
9681 035470 014000 ECI!FMT22 ;16 BITS PER WORD FORMAT
9682 ;INHIBIT ECC CORRECTION
9683 ;DO NOT INHIBIT HEADER COMPARE
9684 035472 002450 REFOR ;GET READY TO DO A REFOR
9685 ;READ HEADER AND DATA WITH 72 IN RHCS1
9686
9687
9688 ;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ HEADER AND DATA
9689

```

CZRJID0, RP04/5/6 FCTNL CTLR1
CZRJID.P11 28-MAR-79 09:03

MACY11 30A(1052) 25-MAY-79 10:30 PAGE 201
T45 WRITE CHECK HEADER AND DATA

SEQ 0200

```

9690 035474 004037 041672      JSR      RO,@#SAVER      ;SAVE REGISTERS
9691 035500 002272              RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
9692 035502 004612              SAVERE                  ;STARTING ADDRESS OF WHERE
9693                                ;THE REGISTERS ARE SAVED
9694 035504 000022              18.                    ;NUMBER OF REGISTERS
9695                                ;SAVED = 18.
9696
9697 035506 004737 041604      JSR      PC,@#CHECKT    ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
9698                                ;AND THAT NO STATUS BITS ARE STUCK = 1
9699 035512 104401 067033      TYPE      ,CPHALT      ;CANNOT CONTINUE TESTING IF ANY OF
9700                                ;THE FIRST SET OF BITS DON'T = 1
9701 035516 000000              HALT                    ;STOP
9702
9703 035520 013777 004606 144540  MOV      @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
9704                                ;TO 'TIME1' IF P-CLOCK IS PRESENT
9705                                ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
9706                                ;'TIME' WILL ONLY SAVE
9707                                ;CURRENT CYLINDER ADDRESS
9708                                ;AND LOOK AHEAD REGISTERS
9709
9710
9711 035526 013746 002450      MOV      @#REFOR,-(SP)  ;GET READY TO MOVE COMMAND
9712 035532 052716 000101      BIS      #GO!IE,(SP)   ;GET READY TO SET 'GO' AND
9713                                ;ENABLE INTERRUPT
9714 035536 012677 144536      MOV      (SP)+,@RHCS1  ;GO WITH
9715                                ;72 IN RHCS1 FOR READ DATA
9716                                ;WITH INTERRUPT ENABLED
9717
9718
9719 035542 104413              WAT                      ;WAIT FOR RDY BIT TO SET
9720 035544 002300              RHCS1                    ;WAIT FOR RHCS1 REGISTER
9721 035546 000200              RDY                      ;WAIT FOR RDY BIT IN RHCS1 REGISTER
9722 035550 001732              986.                    ;ALLOW 9860 MICRO SECONDS
9723 035552 001502              834.                    ;RDY MUST SET BETWEEN
9724                                ;1520 AND 18200 MICRO SECONDS
9725
9726                                ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
9727
9728 035554 004037 041426      JSR      RO,@#FILLRE    ;MOV 0 INTO SAVED RHWC
9729 035560 002272              RHWC                    ;SAVED REGISTER TO CHANGE
9730 035562 000000              0                        ;DATA
9731 035564 004037 041426      JSR      RO,@#FILLRE    ;MOV REINTO+<266.*2> INTO SAVED RHBA
9732 035570 002274              RHBA                    ;SAVED REGISTER TO CHANGE
9733 035572 004560              REINTO+<266.*2>        ;DATA
9734 035574 004037 041426      JSR      RO,@#FILLRE    ;MOV 3406 INTO SAVED RHDST
9735 035600 002304              RHDST                    ;SAVED REGISTER TO CHANGE
9736 035602 003406              3406                    ;DATA
9737
9738                                ;*COMPARE REGISTERS BEFORE READ HEADER AND DATA
9739                                ;*WITH REGISTERS AFTER COMMAND
9740
9741
9742
9743 035604 004037 042512      JSR      RO,@#COMREG    ;COMPARE SAVED REGISTERS WITH
9744                                ;PRESENT VALUE
9745 035610 004612              SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'

```



```
9802 035674 002440 WRCHDT ;GET READY TO DO A WRCHDT
9803 ;WRITE CHECK HEADER AND DATA WITH 52 IN RHCS1
9804
9805
9806 ;*SAVE REGISTERS FOR COMPARISON AFTER WRITE CHECK
9807
9808 035676 ST25:
9809 035676 004037 041672 JSR RO,@#SAVER ;SAVE REGISTERS
9810 035702 002272 RHWC ;RHWC IS THE FIRST REGISTER SAVED
9811 035704 004612 SAVERE ;STARTING ADDRESS OF WHERE
9812 ;THE REGISTERS ARE SAVED
9813 035706 000022 18. ;NUMBER OF REGISTERS
9814 ;SAVED = 18.
9815
9816 035710 004737 041604 JSR PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
9817 ;AND THAT NO STATUS BITS ARE STUCK = 1
9818 035714 104401 067033 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF ANY OF
9819 ;THE FIRST SET OF BITS DON'T = 1
9820 035720 000000 HALT ;STOP
9821
9822 035722 013777 004606 144336 MOV @#RP4VEC,@RPVEC ;SET RPO4 VECTOR ADDRESS
9823 ;TO 'TIME1' IF P-CLOCK IS PRESENT
9824 ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
9825 ;'TIME' WILL ONLY SAVE
9826 ;CURRENT CYLINDER ADDRESS
9827 ;AND LOOK AHEAD REGISTERS
9828
9829
9830 035730 013746 002440 MOV @#WRCHDT,-(SP) ;GET READY TO MOVE COMMAND
9831 035734 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET 'GO' AND
9832 ;ENABLE INTERRUPT
9833 035740 012677 144334 MOV (SP)+,@RHCS1 ;GO WITH
9834 ;52 IN RHCS1 FOR WRITE CHECK HEADER AND DATA
9835 ;WITH INTERRUPT ENABLED
9836 035744 011100 MOV @R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
9837 035746 011305 MOV @R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
9838
9839
9840 035750 104413 WAT ;WAIT FOR RDY BIT TO SET
9841 035752 002300 RHCS1 ;WAIT FOR RHCS1 REGISTER
9842 035754 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
9843 035756 001732 986. ;ALLOW 9860 MICRO SECONDS
9844 035760 001502 834. ;RDY MUST SET BETWEEN
9845 ;1520 AND 18200 MICRO SECONDS
9846
9847 ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
9848 ;*R0 AND R5 IMMEDIATELY AFTER GO
9849
9850 035762 013746 002440 MOV @#WRCHDT,-(SP) ;SAVE COMMAND
9851 035766 052716 004101 BIS #IE!DVA!GO,(SP) ;INCLUDE IE!DVA!GO
9852 035772 011637 001124 MOV (SP),@#$GDDAT ;SAVE FOR PRINTOUT
9853 035776 022600 CMP (SP)+,R0 ;DURING ABOVE OPERATION ONLY IE.DVA!GO
9854 ;AND COMMAND SHOULD BE SET
9855 036000 001405 BEQ 64$ ;BRANCH IF GOOD
9856 036002 010037 001126 MOV R0,@#$BDDAT ;BAD DATA
9857 036006 010137 004600 MOV R1,@#REGADR ;FAILING REGISTER RHCS1
```

```
9858 036012 104021          ERROR 21          ;DURING ABOVE OPERATION ONLY
9859                                ;COMMAND AND IE!DVA!GO SHOULD BE SET
9860 036014 012746 010500    64$:  MOV    #MOL!DPR!VV,-(SP)    ;SAVE BITS SET DURING OPERATION IN RHDS1
9861 036020 011637 001124    MOV    (SP),@#SGDDAT    ;SAVE FOR PRINTOUT
9862 036024 022605          CMP    (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
9863                                ;SHOULD BE SET
9864 036026 001405          BEQ    66$            ;BRANCH IF GOOD
9865 036030 010537 001126    MOV    R5,@#SBDDAT    ;BAD DATA
9866 036034 010337 004600    MOV    R3,@#REGADR    ;FAILING REGISTER RHDS1
9867 036040 104063          ERROR 63          ;DURING ABOVE OPERATION ONLY
9868                                ;MOL!DPR!VV SHOULD BE SET
9869 036042          66$:
9870
9871                                ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
9872
9873 036042 004037 041426    JSR    R0,@#FILLRE    ;MOV 0 INTO SAVED RHWC
9874 036046 002272          RHWC                ;SAVED REGISTER TO CHANGE
9875 036050 000000          0                    ;DATA
9876 036052 004037 041426    JSR    R0,@#FILLRE    ;MOV WRFROM+<266.*2> INTO SAVED RHBA
9877 036056 002274          RHBA                ;SAVED REGISTER TO CHANGE
9878 036060 003514          WRFROM+<266.*2>    ;DATA
9879 036062 004037 041426    JSR    R0,@#FILLRE    ;MOV 3406 INTO SAVED RHDST
9880 036066 002304          RHDST                ;SAVED REGISTER TO CHANGE
9881 036070 003406          3406                ;DATA
9882
9883                                ;*COMPARE REGISTERS BEFORE WRITE CHECK HEADER AND DATA
9884                                ;*WITH REGISTERS AFTER COMMAND
9885
9886
9887 036072 004037 042512    JSR    R0,@#COMREG    ;COMPARE SAVED REGISTERS WITH
9888                                ;PRESENT VALUE
9889 036076 004612          SAVERE                ;GOOD DATA SAVED IN 'SAVERE'
9890 036100 002354          WC                    ;TEST DATA STARTING FROM 'RHWC'
9891 036102 000022          18.                  ;18. REGISTERS TO BE COMPARED
9892 036104 036110          8$                    ;RETURN TO 8$ ON ERROR
9893 036106 036114          9$                    ;RETURN TO 9$ ON NO ERROR
9894
9895 036110 104040          8$:  ERROR 40          ;WRITE CHECK CAUSED
9896 036112 000207          RTS    PC            ;AN IMPROPER REGISTER
9897                                ;CHANGE
9898                                ;GOOD DATA GIVES WHAT
9899                                ;SHOULD BE THERE
9900                                ;RECEIVED DATA GIVES WHAT
9901                                ;WAS THERE AFTER COMMAND
9902
9903
9904 036114          9$:
```

9905
9906
9907
9908
9909
9910
9911
9912
9913
9914
9915
9916
9917
9918
9919
9920
9921
9922
9923
9924
9925
9926
9927
9928
9929
9930
9931
9932
9933
9934
9935
9936
9937
9938
9939
9940
9941
9942
9943
9944
9945
9946
9947
9948
9949
9950
9951
9952
9953
9954
9955
9956
9957
9958
9959
9960

```
*****  
*TEST 46 WRITE CHECK DATA  
* THE DATA FOR THIS TEST IS WRITTEN ON DISK BY PREVIOUS TEST  
* WRITE CHECK DATA CYLINDER 5, FORMAT 16 BITS PER WORDS  
* TRACK 7, SECTOR 4, KEYS 0, NUMBER OF WORDS 258  
* CONSISTING OF  
* 10 WORDS OF 12344 (6 BITS FOR CYL, 5 FOR TRACK, 5 FOR SECTOR)  
* 10 WORDS OF 177777  
* 10 WORDS OF 0  
* 10 WORDS OF 052525  
* 10 WORD OF 125252  
* 16 WORDS OF LEFT ROTATING ZERO (EG177776,177775)  
* 16 WORDS OF LEFT ROTATING ONE (EG 1,2,4,10)  
* 174 WORDS OF 377  
* 2 WORDS OF 12345
```

```
*****  
* FIRST THE ABOVE DATA IS FILLED INTO WRITE FROM BUFFER  
* THEN THE ABOVE WRITE CHECK DATA IS GIVEN  
*****
```

```
TST46: SCOPE  
MOV #STACK,SP ;RESET STACK  
MOV #46,@#TSTNM ;SAVE TEST NUMBER  
JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2  
;R3-RHDS1, R4-RHER1  
;GIVE RH-11 INITIALIZE  
;SETUP UNIT NUMBER  
JSR PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1  
;AND THAT NO STATUS BITS ARE STUCK = 1  
TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF ANY OF  
;THE FIRST SET OF BITS DON'T = 1  
HALT ;STOP  
;*GET HEADS TO CYLINDER 5  
JSR RO,@#SEEKCY ;SEEK FOR  
5 ;CYLINDER 5  
MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS  
;TO 'TIME1' IF P-CLOCK IS PRESENT  
;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT  
;'TIME' WILL ONLY SAVE  
;CURRENT CYLINDER ADDRESS  
;AND LOOK AHEAD REGISTERS  
MOV @#SEECOM,-(SP) ;GET READY TO MOVE COMMAND  
BIS #GO.IE,(SP) ;GET READY TO SET 'GO' AND  
;ENABLE INTERRUPT  
MOV (SP)+,@RHCS1 ;GO WITH  
;4 IN RHCS1 FOR SEEK  
;WITH INTERRUPT ENABLED
```

```
996i
9962 036176 104413      WAT          ;WAIT FOR DRY BIT TO SET
9963 036200 002322      RHDS1        ;WAIT FOR RHDS1 REGISTER
9964 036202 000200      DRY          ;WAIT FOR DRY BIT IN RHDS1 REGISTER
9965 036204 004704      2500.       ;ALLOW 25000 MICRO SECONDS
9966 036206 004704      2500.       ;DRY MUST SET BETWEEN
9967                                     ;00 AND 5000 MICRO SECONDS
9968
9969
9970 036210 004737 041524 JSR    PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
9971                                     ;R3-RHDS1, R4-RHER1
9972                                     ;GIVE RH-11 INITIALIZE
9973                                     ;SETUP UNIT NUMBER
9974
9975                                     ;*10 WORDS OF EACH 12344,17777,0,52525,125252
9976
9977 036214 004037 041374 JSR    RO,@#CLAREA ;CLEAR 10. WORDS, FROM WRFROM
9978 036220 002470      WRFROM      ;STARTING FROM WRFROM
9979 036222 000012      10.        ;10. WORDS
9980 036224 012344      <5*2000>!<7*40>!4 ;FILL WITH <5*2000>!<7*40>!4
9981
9982 036226 004037 041374 JSR    RO,@#CLAREA ;CLEAR 10. WORDS, FROM WRFROM+<10.*2>
9983 036232 002514      WRFROM+<10.*2> ;STARTING FROM WRFROM+<10.*2>
9984 036234 000012      10.        ;10. WORDS
9985 036236 177777      -1         ;FILL WITH -1
9986
9987 036240 004037 041374 JSR    RO,@#CLAREA ;CLEAR 10. WORDS, FROM WRFROM+<20.*2>
9988 036244 002540      WRFROM+<20.*2> ;STARTING FROM WRFROM+<20.*2>
9989 036246 000012      10.        ;10. WORDS
9990 036250 000000      0         ;FILL WITH 0
9991
9992 036252 004037 041374 JSR    RO,@#CLAREA ;CLEAR 10. WORDS, FROM WRFROM+<30.*2>
9993 036256 002564      WRFROM+<30.*2> ;STARTING FROM WRFROM+<30.*2>
9994 036260 000012      10.        ;10. WORDS
9995 036262 052525      52525     ;FILL WITH 52525
9996
9997 036264 004037 041374 JSR    RO,@#CLAREA ;CLEAR 10. WORDS, FROM WRFROM+<40.*2>
9998 036270 002610      WRFROM+<40.*2> ;STARTING FROM WRFROM+<40.*2>
9999 036272 000012      10.        ;10. WORDS
10000 036274 125252     125252     ;FILL WITH 125252
10001
10002
10003                                     ;*FILL LEFT ROTATING ZEROS FROM WRFROM+<50.*2>
10004
10005 036276 010146      MOV    R1,-(SP) ;:PUSH R1 ON STACK
10006 036300 012700 177776 MOV    #177776,R0 ;DATA
10007 036304 012705 000020 MOV    #16.,R5 ;COUNT
10008 036310 012701 002634 MOV    #WRFROM+<50.*2>,R1 ;WHERE DATA GOES
10009 036314 000261      SEC
10010 036316 010021      1$: MOV    RO,(R1)+ ;STORE DATA
10011 036320 006100      ROL    RO ;GET ZERO ONE BIT LEFT
10012 036322 005305      DEC    R5 ;COUNT 16
10013 036324 001374      BNE    1$ ;BRANCH IF 16 NOT DONE
10014
10015                                     ;*FILL LEFT ROTATING ONE INTO WRFROM+<65.*2>
10016
```

```

10017 036326 000241          CLC
10018 036330 012700 000001  MOV      #1,R0
10019 036334 010021          MOV      R0,(R1)+
10020 036336 006300          ASL     R0
10021 036340 103375          BCC     2$
10022 036342 012601          MOV     (SP)+,R1      ;;POP STACK INTO R1
10023
10024                          ;*FILL REST OF DATA
10025
10026 036344 004037 041374  JSR     R0,@#CLAREA   ;CLEAR 174. WORDS, FROM WRFROM+<82.*2>
10027 036350 002734          WRFROM+<82.*2>        ;STARTING FROM WRFROM+<82.*2>
10028 036352 000256          174.                  ;174. WORDS
10029 036354 000377          377                    ;FILL WITH 377
10030
10031 036356 004037 041374  JSR     R0,@#CLAREA   ;CLEAR 2 WORDS, FROM WRFROM+<256.*2>
10032 036362 003470          WRFROM+<256.*2>       ;STARTING FROM WRFROM+<256.*2>
10033 036364 000002          2                       ;2 WORDS
10034 036366 012345          <5*2000>!<7*40>!5     ;FILL WITH <5*2000>!<7*40>!5
10035
10036                          ;*FILL THE WRITE CHECK HEADER AND DATA
10037
10038
10039 036370 004037 043476  JSR     R0,@#RUN      ;SETUP TO RUN FOR DATA COMMAND
10040 036374 000005          5                       ;CYLINDER 5
10041 036376          004                     ;SECTOR 4
10042 036377          007                     ;TRACK 7
10043 036400 177400          -256.                   ;WORD COUNT = 256.
10044 036402 002470          WRFROM                  ;BUS ADDRESS
10045                          ;STARTING ADDRESS OF DATA
10046                          ;BUFFER = WRFROM
10047 036404 000000          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
10048 036406 014000          ECI!FMT22              ;16 BITS PER WORD FORMAT
10049                          ;INHIBIT ECC CORRECTION
10050                          ;DO NOT INHIBIT HEADER COMPARE
10051 036410 002436          WRCHEK                  ;GET READY TO DO A WRCHEK
10052                          ;WRITE CHECK DATA WITH 50 IN RHCS1
10053
10054
10055                          ;*SAVE REGISTERS FOR COMPARISON AFTER WRITE CHECK
10056 036412 004037 041672  JSR     R0,@#SAVER     ;SAVE REGISTERS
10057 036416 002272          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
10058 036420 004612          SAVERE                  ;STARTING ADDRESS OF WHERE
10059                          ;THE REGISTERS ARE SAVED
10060 036422 000022          18.                     ;NUMBER OF REGISTERS
10061                          ;SAVED = 18.
10062
10063 036424 004737 041604  JSR     PC,@#CHECKT    ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
10064                          ;AND THAT NO STATUS BITS ARE STUCK = 1
10065 036430 104401 067033  TYPE     ,CPHALT       ;CANNOT CONTINUE TESTING IF ANY OF
10066                          ;THE FIRST SET OF BITS DON'T = 1
10067 036434 000000          HALT                    ;STOP
10068
10069 036436 013777 004606 143622 MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
10070                          ;TO 'TIME1' IF P-CLOCK IS PRESENT
10071                          ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
10072                          ;'TIME' WILL ONLY SAVE

```



```

10073                                     :CURRENT CYLINDER ADDRESS
10074                                     :AND LOOK AHEAD REGISTERS
10075
10076
10077 036444 013746 002436      MOV    @#WRCHEK,-(SP)  :GET READY TO MOVE COMMAND
10078 036450 052716 000101      BIS    #GO!IE,(SP)   :GET READY TO SET 'GO' AND
10079                                     :ENABLE INTERRUPT
10080 036454 012677 143620      MOV    (SP)+,@RHCS1  :GO WITH
10081                                     :50 IN RHCS1 FOR WRITE CHECK DATA
10082                                     :WITH INTERRUPT ENABLED
10083 036460 011100      MOV    @R1,R0        :SAVE RHCS1 DURING ABOVE OPERATION
10084 036462 011305      MOV    @R3,R5        :SAVE RHDS1 DURING ABOVE OPERATION
10085
10086
10087 036464 104413      WAT                                     :WAIT FOR RDY BIT TO SET
10088 036466 002300      RHCS1                                :WAIT FOR RHCS1 REGISTER
10089 036470 000200      RDY                                  :WAIT FOR RDY BIT IN RHCS1 REGISTER
10090 036472 001732      986.                                :ALLOW 9860 MICRO SECONDS
10091 036474 001502      834.                                :RDY MUST SET BETWEEN
10092                                     :1520 AND 18200 MICRO SECONDS
10093
10094                                     ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
10095                                     ;*R0 AND R5 IMMEDIATELY AFTER GO
10096
10097 036476 013746 002436      MOV    @#WRCHEK,-(SP)  :SAVE COMMAND
10098 036502 052716 004101      BIS    #IE!DVA!GO,(SP) :INCLUDE IE!DVA!GO
10099 036506 011637 001124      MOV    (SP),@#SGDDAT  :SAVE FOR PRINTOUT
10100 036512 022600      CMP    (SP)+,R0       :DURING ABOVE OPERATION ONLY IE!DVA!GO
10101                                     :AND COMMAND SHOULD BE SET
10102 036514 001405      BEQ    67$            :BRANCH IF GOOD
10103 036516 010037 001126      MOV    R0,@#SBDDAT    :BAD DATA
10104 036522 010137 004600      MOV    R1,@#REGADR   :FAILING REGISTER RHCS1
10105 036526 104021      ERROR  21            :DURING ABOVE OPERATION ONLY
10106                                     :COMMAND AND IE!DVA!GO SHOULD BE SET
10107 036530 012746 010500      67$: MUV    #MOL!DPR!VV,-(SP) :SAVE BITS SET DURING OPERATION IN RHDS1
10108 036534 011637 001124      MOV    (SP),@#SGDDAT  :SAVE FOR PRINTOUT
10109 036540 022605      CMP    (SP)+,R5      :DURING ABOVE OPERATION ONLY MOL!DPR.VV
10110                                     :SHOULD BE SET
10111 036542 001405      BEQ    69$            :BRANCH IF GOOD
10112 036544 010537 001126      MOV    R5,@#SBDDAT   :BAD DATA
10113 036550 010337 004600      MOV    R3,@#REGADR   :FAILING REGISTER RHDS1
10114 036554 104063      ERROR  63            :DURING ABOVE OPERATION ONLY
10115                                     :MOL!DPR!VV SHOULD BE SET
10116 036556      69$:
10117
10118                                     ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
10119
10120 036556 004037 041426      JSR    R0,@#FILLRE   :MOV 0 INTO SAVED RHWC
10121 036562 002272      RHWC                                     :SAVED REGISTER TO CHANGE
10122 036564 000000      0                                         :DATA
10123 036566 004037 041426      JSR    R0,@#FILLRE   :MOV WRFROM+<256.*2> INTO SAVED RHBA
10124 036572 002274      RHBA                                     :SAVED REGISTER TO CHANGE
10125 036574 003470      WRFROM+<256.*2> :DATA
10126 036576 004037 041426      JSR    R0,@#FILLRE   :MOV 3405 INTO SAVED RHDST
10127 036602 002304      RHDST                                     :SAVED REGISTER TO CHANGE
10128 036604 003405      3405                                     :DATA

```

```
10129
10130
10131
10132
10133
10134 036606 004037 042512 JSR RO,@#COMREG ;COMPARE SAVED REGISTERS WITH
10135 ;PRESENT VALUE
10136 036612 004612 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
10137 036614 002354 WC ;TEST DATA STARTING FROM 'RHWC'
10138 036616 000022 18. ;18. REGISTERS TO BE COMPARED
10139 036620 036624 8$ ;RETURN TO 8$ ON ERROR
10140 036622 036630 9$ ;RETURN TO 9$ ON NO ERROR
10141 036624 104040 8$: ERROR 40 ;WRITE CHECK CAUSED
10142 036626 000207 RTS PC ;AN IMPROPER REGISTER
10143 ;CHANGE
10144 ;GOOD DATA GIVES WHAT
10145 ;SHOULD BE THERE
10146 ;RECEIVED DATA GIVES WHAT
10147 ;WAS THERE AFTER COMMANDS
10148
10149 036630 9$:
10150
10151
```

10152
10153
10154
10155
10156
10157
10158
10159
10160
10161
10162
10163
10164
10165
10166
10167
10168
10169
10170
10171
10172
10173
10174
10175 036630 000004
10176 036632 012706 001000
10177 036636 012737 000047 004604
10178
10179 036644 004737 041524
10180
10181
10182
10183
10184
10185
10186 036650 005737 004750
10187 036654 001402
10188 036656 000137 037376
10189 036662
10190 036662 004737 041604
10191
10192 036666 104401 067033
10193
10194 036672 000000
10195
10196 036674 005737 004732
10197 036700 001402
10198 036702 000137 037376
10199
10200
10201
10202 036706
10203
10204 036706 004037 041474
10205 036712 000005
10206
10207 036714 013777 004606 143344

```
*****  
*TEST 47 WRITE CHECK DATA USING UNIBUS B  
* THIS TEST USES UNIBUS B IF CONNECTED TO THE RH  
* IF UNIBUS B IS NOT CONNECTED THEN THIS TEST IS NOT PERFORMED  
* THE DATA FOR THIS TEST IS WRITTEN ON DISK BY PREVIOUS TEST  
* WRITE CHECK DATA CYLINDER 5, FORMAT 16 BITS PER WORDS  
* TRACK 7, SECTOR 4, KEYS 0, NUMBER OF WORDS 258  
* CONSISTING OF  
* 10 WORDS OF 12344 (6 BITS FOR CYL, 5 FOR TRACK, 5 FOR SECTOR)  
* 10 WORDS OF 177777  
* 10 WORDS OF 0  
* 10 WORDS OF 052525  
* 10 WORD OF 125252  
* 16 WORDS OF LEFT ROTATING ZERO (EG177776,177775)  
* 16 WORDS OF LEFT ROTATING ONE (EG 1,2,4,10)  
* 174 WORDS OF 377  
* 2 WORDS OF 12345
```

```
*****  
* FIRST THE ABOVE DATA IS FILLED INTO WRITE FROM BUFFER  
* THEN THE ABOVE WRITE CHECK DATA IS GIVEN  
*****
```

```
TST47: SCOPE  
MOV #STACK,SP ;RESET STACK  
MOV #47,@TSTNM ;SAVE TEST NUMBER  
  
JSR PC,@CLDISK ;SET R1-RHCS1, R2-RHCS2  
;R3-RHDS1, R4-RHER1  
;GIVE RH-11 INITIALIZE  
;SETUP UNIT NUMBE'  
  
;*CHECK TO SEE IF THE PROGRAM IS RUNNING WITH AN RH70  
  
TST @RH70 ;TEST FOR RH70 CONTROLLER  
BEQ 30$ ;IF FLAG = 1, THIS TEST IS SKIPPED  
JMP TST50 ; JUMP TO NEXT TEST -----)  
30$: JSR PC,@CHECKT ;IF FLAG = 1, DO THIS TEST  
;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1  
;AND THAT NO STATUS BITS ARE STUCK = 1  
TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF ANY OF  
;THE FIRST SET OF BITS DON'T = 1  
HALT ;STOP  
  
TST @UBUSB ;IS UNIBUS B THERE  
BEQ 11$ ;UNIBUS B THERE SO CONTINUE  
JMP @9$ ;NO UNIBUS B, GO TO NEXT TEST
```

```
;*GET HEADS TO CYLINDER 5
```

```
11$: JSR R0,@SEEKCY ;SEEK FOR  
5 ;CYLINDER 5  
  
MOV @RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
```

```
10208 ;TO 'TIME1' IF P-CLOCK IS PRESENT
10209 ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
10210 ;'TIME' WILL ONLY SAVE
10211 ;CURRENT CYLINDER ADDRESS
10212 ;AND LOOK AHEAD REGISTERS
10213
10214
10215 036722 013746 002452 MOV @#SEECOM,-(SP) ;GET READY TO MOVE COMMAND
10216 036726 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET 'GO' AND
10217 ;ENABLE INTERRUPT
10218 036732 012677 143342 MOV (SP)+,@RHCS1 ;GO WITH
10219 ;4 IN RHCS1 FOR SEEK
10220 ;WITH INTERRUPT ENABLED
10221
10222
10223 036736 104413 WAT ;WAIT FOR DRY BIT TO SET
10224 036740 002322 RHDS1 ;WAIT FOR RHDS1 REGISTER
10225 036742 000200 DRY ;WAIT FOR DRY BIT IN RHDS1 REGISTER
10226 036744 004704 2500. ;ALLOW 25000 MICRO SECONDS
10227 036746 004704 2500. ;DRY MUST SET BETWEEN
10228 ;00 AND 50000 MICRO SECONDS
10229
10230
10231 036750 004737 041524 JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
10232 ;R3-RHDS1, R4-RHER1
10233 ;GIVE RH-11 INITIALIZE
10234 ;SETUP UNIT NUMBER
10235
10236 ;*10 WORDS OF EACH 12344,17777,0,52525,125252
10237
10238 036754 004037 041374 JSR R0,@#CLAREA ;CLEAR 10. WORDS, FROM WRFROM
10239 036760 002470 WRFROM ;STARTING FROM WRFROM
10240 036762 000012 10. ;10. WORDS
10241 036764 012344 <5*2000>!<7*40>!4 ;FILL WITH <5*2000>!<7*40>!4
10242
10243 036766 004037 041374 JSR R0,@#CLAREA ;CLEAR 10. WORDS, FROM WRFROM+<10.*2>
10244 036772 002514 WRFROM+<10.*2> ;STARTING FROM WRFROM+<10.*2>
10245 036774 000012 10. ;10. WORDS
10246 036776 177777 -1 ;FILL WITH -1
10247
10248 037000 004037 041374 JSR R0,@#CLAREA ;CLEAR 10. WORDS, FROM WRFROM+<20.*2>
10249 037004 002540 WRFROM+<20.*2> ;STARTING FROM WRFROM+<20.*2>
10250 037006 000012 10. ;10. WORDS
10251 037010 000000 0 ;FILL WITH 0
10252
10253 037012 004037 041374 JSR R0,@#CLAREA ;CLEAR 10. WORDS, FROM WRFROM+<30.*2>
10254 037016 002564 WRFROM+<30.*2> ;STARTING FROM WRFROM+<30.*2>
10255 037020 000012 10. ;10. WORDS
10256 037022 052525 52525 ;FILL WITH 52525
10257
10258 037024 004037 041374 JSR R0,@#CLAREA ;CLEAR 10. WORDS, FROM WRFROM+<40.*2>
10259 037030 002610 WRFROM+<40.*2> ;STARTING FROM WRFROM+<40.*2>
10260 037032 000012 10. ;10. WORDS
10261 037034 125252 125252 ;FILL WITH 125252
10262
10263
```

```
10264 ;*FILL LEFT ROTATING ZEROS FROM WRFROM+<50.*2>
10265
10266 037036 010146 MOV R1, -(SP) ;; PUSH R1 ON STACK
10267 037040 012700 177776 MOV #177776, R0 ;: DATA
10268 037044 012705 000020 MOV #16., R5 ;: COUNT
10269 037050 012701 002634 MOV #WRFROM+<50.*2>, R1 ;: WHERE DATA GOES
10270 037054 000261 SEC
10271 037056 010021 1$: MOV R0, (R1)+ ;: STORE DATA
10272 037060 006100 ROL R0 ;: GET ZERO ONE BIT LEFT
10273 037062 005305 DEC R5 ;: COUNT 16
10274 037064 001374 BNE 1$ ;: BRANCH IF 16 NOT DONE
10275
10276 ;*FILL LEFT ROTATING ONE INTO WRFROM+<65.*2>
10277
10278 037066 000241 CLC
10279 037070 012700 000001 MOV #1, R0
10280 037074 010021 2$: MOV R0, (R1)+
10281 037076 006300 ASL R0
10282 037100 103375 BCC 2$
10283 037102 012601 MOV (SP)+, R1 ;: POP STACK INTO R1
10284
10285 ;*FILL REST OF DATA
10286 037104 004037 041374 JSR R0, @#CLAREA ;: CLEAR 174. WORDS, FROM WRFROM+<82.*2>
10287 037110 002734 WRFROM+<82.*2> ;: STARTING FROM WRFROM+<82.*2>
10288 037112 000256 174. ;: 174. WORDS
10289 037114 000377 377 ;: FILL WITH 377
10290
10291 037116 004037 041374 JSR R0, @#CLAREA ;: CLEAR 2 WORDS, FROM WRFROM+<256.*2>
10292 037122 003470 WRFROM+<256.*2> ;: STARTING FROM WRFROM+<256.*2>
10293 037124 000002 2 ;: 2 WORDS
10294 037126 012345 <5*2000>!<7*40>!5 ;: FILL WITH <5*2000>!<7*40>!5
10295
10296
10297 ;*FILL THE WRITE CHECK HEADER AND DATA
10298
10299 037130 004037 043476 JSR R0, @#RUN ;: SETUP TO RUN FOR DATA COMMAND
10300 037134 000005 5 ;: CYLINDER 5
10301 037136 004 .BYTE 4 ;: SECTOR 4
10302 037137 007 .BYTE 7 ;: TRACK 7
10303 037140 177400 -256. ;: WORD COUNT = 256.
10304 037142 002470 WRFROM ;: BUS ADDRESS
10305 ;: STARTING ADDRESS OF DATA
10306 ;: BUFFER = WRFROM
10307 037144 000000 0 ;: DO NOT INHIBIT BUS ADDRESS INCREMENT
10308 037146 014000 EC1.FMT22 ;: 16 BITS PER WORD FORMAT
10309 ;: INHIBIT ECC CORRECTION
10310 ;: DO NOT INHIBIT HEADER COMPARE
10311 037150 002436 WRCHK ;: GET READY TO DO A WRCHK
10312 ;: WRITE CHECK DATA WITH 50 IN RHCS1
10313
10314
10315
10316 037152 052777 002000 143120 BIS #PSEL, @RHCS1 ;: SET PORT B
10317 ;: THAT IS UNIBUS B
10318
10319 ;*SAVE REGISTERS FOR COMPARISON AFTER WRITE CHECK
```

```

CZRJ1D0, RP04/5/6 FCTNL CTLR1 MACY11 30A(1052) 25-MAY-79 10:30 F 1 PAGE 213
CZRJ1D.P11 28-MAR-79 09:03 T47 WRITE CHECK DATA USING UNIBUS B SEQ 0212

10320 037160 004037 041672 JSR RO,@#SAVER ;SAVE REGISTERS
10321 037164 002272 RHWC ;RHWC IS THE FIRST REGISTER SAVED
10322 037166 004612 SAVERE ;STARTING ADDRESS OF WHERE
10323 ;THE REGISTERS ARE SAVED
10324 037170 000022 18. ;NUMBER OF REGISTERS
10325 ;SAVED = 18.
10326
10327 037172 004737 041604 JSR PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
10328 ;AND THAT NO STATUS BITS ARE STUCK = 1
10329 037176 104401 067033 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF ANY OF
10330 ;THE FIRST SET OF BITS DON'T = 1
10331 037202 000000 HALT ;STOP
10332
10333 037204 013777 004606 143054 MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
10334 ;TO 'TIME1' IF P-CLOCK IS PRESENT
10335 ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
10336 ;'TIME' WILL ONLY SAVE
10337 ;CURRENT CYLINDER ADDRESS
10338 ;AND LOOK AHEAD REGISTERS
10339
10340 ;*SET PORT SELECT
10341
10342 037212 013746 002436 MOV @#WRCHK,-(SP) ;GET READY TO MOVE COMMAND
10343 037216 052716 002101 BIS #GO!IE!PSEL,(SP) ;GET READY TO SET 'GO' AND
10344 ;ENABLE INTERRUPT
10345 037222 012677 143052 MOV (SP)+,@RHCS1 ;GO WITH
10346 ;50 IN RHCS1 FOR WRITE CHECK DATA
10347 ;WITH INTERRUPT ENABLED
10348 037226 011100 MOV @R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
10349 037230 011305 MOV @R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
10350
10351
10352 037232 104413 WAT ;WAIT FOR RDY BIT TO SET
10353 037234 002300 RHCS1 ;WAIT FOR RHCS1 REGISTER
10354 037236 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
10355 037240 001732 986. ;ALLOW 9860 MICRO SECONDS
10356 037242 001502 834. ;RDY MUST SET BETWEEN
10357 ;1520 AND 18200 MICRO SECONDS
10358
10359 ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
10360 ;*R0 AND R5 IMMEDIATELY AFTER GO
10361
10362 037244 013746 002436 MOV @#WRCHK,-(SP) ;SAVE COMMAND
10363 037250 052716 006101 BIS #IE.DVA!PSEL!GO,(SP) ;INCLUDE IE!DVA!PSEL!GO
10364 037254 011637 001124 MOV (SP),@#SGDDAT ;SAVE FOR PRINTOUT
10365 037260 022600 CMP (SP)+,R0 ;DURING ABOVE OPERATION ONLY IE.DVA!PSEL!GO
10366 ;AND COMMAND SHOULD BE SET
10367 037262 001405 BEQ 67$ ;BRANCH IF GOOD
10368 037264 010037 001126 MOV RO,@#SBDDAT ;BAD DATA
10369 037270 010137 004600 MOV R1,@#REGADR ;FAILING REGISTER RHCS1
10370 037274 104021 ERROR 21 ;DURING ABOVE OPERATION ONLY
10371 ;COMMAND AND IE!DVA!PSEL!GO SHOULD BE SET
10372 037276 012746 010500 67$: MOV #MOL.DPR.VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
10373 037302 011637 001124 MOV (SP),@#SGDDAT ;SAVE FOR PRINTOUT
10374 037306 022605 CMP (SP)+,R5 ;DURING ABOVE OPERATION ONLY MOL!DPR.VV
10375 ;SHOULD BE SET

```

CZRJ1D0, RP04/5/6 FCTNL CTLR1
CZRJID.P11 28-MAR-79 09:03

MACY11 30A(1052) 25-MAY-79 10:30 PAGE 214
T47 WRITE CHECK DATA USING UNIBUS B

SEQ 0213

```

10376 037310 001405          BEQ      69$          ;BRANCH IF GOOD
10377 037312 010537 001126    MOV      R5,@#SDDAT  ;BAD DATA
10378 037316 010337 004600    MOV      R3,@#REGADR ;FAILING REGISTER RHDS1
10379 037322 104063          ERROR    63          ;DURING ABOVE OPERATION ONLY
10380                                     ;MOL!DPR!VV SHOULD BE SET
10381 037324          69$:
10382
10383                                     ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
10384
10385 037324 004037 041426    JSR      R0,@#FILLRE ;MOV 0 INTO SAVED RHWC
10386 037330 002272          RHWC          ;SAVED REGISTER TO CHANGE
10387 037332 000000          0            ;DATA
10388 037334 004037 041426    JSR      R0,@#FILLRE ;MOV WRFROM+<256.*2> INTO SAVED RHBA
10389 037340 002274          RHBA          ;SAVED REGISTER TO CHANGE
10390 037342 003470          WRFROM+<256.*2> ;DATA
10391 037344 004037 041426    JSR      R0,@#FILLRE ;MOV 3405 INTO SAVED RHDST
10392 037350 002304          RHDST         ;SAVED REGISTER TO CHANGE
10393 037352 003405          3405         ;DATA
10394
10395                                     ;*COMPARE REGISTERS BEFORE WRITE CHECK HEADER AND DATA
10396                                     ;*WITH REGISTER AFTER COMMAND
10397
10398 037354 004037 042512    JSR      R0,@#COMREG ;COMPARE SAVED REGISTERS WITH
10399                                     ;PRESENT VALUE
10400 037360 004612          SAVERE        ;GOOD DATA SAVED IN 'SAVERE'
10401 037362 002354          WC           ;TEST DATA STARTING FROM 'RHWC'
10402 037364 000022          18.          ;18. REGISTERS TO BE COMPARED
10403 037366 037372          8$           ;RETURN TO 8$ ON ERROR
10404 037370 037376          9$           ;RETURN TO 9$ ON NO ERROR
10405
10406 037372 104076          8$:          ERROR    76          ;WHILE USING UNIBUS B
10407                                     ;WRITE CHECK CAUSED
10408 037374 000207          RTS      PC      ;AN IMPROPER REGISTER
10409                                     ;CHANGE
10410                                     ;GOOD DATA GIVES WHAT
10411                                     ;SHOULD BE THERE
10412                                     ;RECEIVED DATA GIVES WHAT
10413                                     ;WAS THERE AFTER COMMANDS
10414
10415 037376          9$:
10416
10417
10418
10419
10420

```

10421
10422
10423
10424
10425
10426
10427
10428
10429
10430
10431
10432
10433
10434
10435
10436
10437
10438
10439
10440
10441
10442
10443
10444
10445
10446
10447
10448
10449
10450
10451
10452
10453
10454
10455
10456
10457
10458
10459
10460
10461
10462
10463
10464
10465
10466
10467
10468
10469
10470
10471
10472
10473
10474
10475
10476

037376 000004

037400 005737 004724
037404 001007
037406 005737 000042
037412 001004
037414 005737 001100
037420 001001
037422 000402

037424
037424 000137 040622
037430
037430 012706 001000
037434 012737 000050 004604

037442 004737 041524

037446 004037 041374
037452 002470
037454 000400

```
*****  
*TEST 50 WRITE PROTECT OPERATION  
* IF STARTING ADDRESS 220 IS USED THIS TEST WILL NOT BE PERFORMED  
*  
* IF THE PROGRAM WORKS UNDER ACT-11 MONITOR  
* THEN THIS TEST IS NOT PERFORMED  
*  
* IF NO ACT-11 MONITOR IS PRESENT  
* THEN THIS TEST IS PERFORMED ONLY ON THE FIRST PASS  
* ON SUBSEQUENT PASSES THIS TEST IS NOT DONE  
*  
* WRITE FROM BUFFER IS FILLED WITH ALL ONES AND  
* SECTOR 0, TRACK 0, CYLINDER 0 IS FILLED WITH  
* ALL ONES  
* ALL REGISTERS ARE SAVED THEN WRITE LOCK BUTTON IS  
* PRESSED AND ALL REGISTERS ARE CHECKED.  
* WRITE FROM BUFFER IS FILLED WITH 377 AND A WRITE IS  
* ATTEMPTED TO SECTOR 0, TRACK 0, CYLINDER 0 70. WORDS  
* ALL REGISTERS ARE CHECKED  
* THE SAME SECTOR IS READ AND DATA COMPARED TO SEE  
* THAT NOTHING GOT DESTROYED (READ DATA SHOULD BE ALL  
* ONES AND NOT 377)  
* THEN WRITE LOCK BUTTON IS PRESSED TO UNLOCK  
* WRITE LOCKS AND ALL REGISTERS ARE COMPARED  
*****
```

```
TST50: SCOPE  
  
;*THIS CODE CHECKS TO SEE IF MANUAL INTERVENTION TESTS ARE OK  
  
TST @#NOPUSH ;IS THIS A 220 START ?  
BNE 1$ ;SKIP THIS TEST IF SO  
TST @#42 ;MONITOR (ACT 11) RETURN ADDRESS ?  
BNE 1$ ;SKIP THIS TEST  
TST @#SPASS ;FIRST PASS ?  
BNE 1$ ;SKIP THIS TEST IF NOT  
BR 2$ ;CONTINUE WITH THIS TEST  
  
1$:  
JMP TST51 ; JUMP TO NEXT TEST -----)  
2$:  
MOV #STACK,SP ;RESET STACK  
MOV #50,@#TSTNM ;SAVE TEST NUMBER  
  
JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2  
;R3-RHDS1, R4-RHER1  
;GIVE RH-11 INITIALIZE  
;SETUP UNIT NUMBER  
  
;*FILL SECTOR 0, TRACK 0, CYL 0 WITH ONES  
;*FILL WRITE FROM BUFFER  
JSR R0,@#CLAREA ;CLEAR 256. WORDS, FROM WRFROM  
WRFROM ;STARTING FROM WRFROM  
256. ;256. WORDS
```



```

10477 037456 177777          -1          ;FILL WITH -1
10478
10479
10480          ;*FILL WRITE DATA COMMAND
10481
10482 037460 004037 043476    JSR      RO,@#RUN      ;SETUP TO RUN FOR DATA COMMAND
10483 037464 000000          0          ;CYLINDER 0
10484 037466          000          .BYTE    0          ;SECTOR 0
10485 037467          000          .BYTE    0          ;TRACK 0
10486 037470 177400          -256.     ;WORD COUNT = 256.
10487 037472 002470          WRFROM    ;BUS ADDRESS
10488          ;STARTING ADDRESS OF DATA
10489          ;BUFFER = WRFROM
10490 037474 000000          0          ;DO NOT INHIBIT BUS ADDRESS INCREMENT
10491 037476 010000          FMT22     ;16 BITS PER WORD FORMAT
10492          ;DO NOT INHIBIT ECC CORRECTION
10493          ;DO NOT INHIBIT HEADER COMPARE
10494 037500 002442          WRIDAT    ;GET READY TO DO A WRIDAT
10495          ;WRITE DATA WITH 60 IN RHCS1
10496
10497
10498 037502 004737 041604    JSR      PC,@#CHECKT  ;CHECK DVA,RDY,MOL,DPR,DRY,VV - 1
10499          ;AND THAT NO STATUS BITS ARE STUCK - 1
10500 037506 104401 067033    TYPE     ,CPHALT     ;CANNOT CONTINUE TESTING IF ANY OF
10501          ;THE FIRST SET OF BITS DON'T = 1
10502 037512 000000          HALT      ;STOP
10503
10504 037514 013777 004606 142544 MOV      @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
10505          ;TO 'TIME1' IF P-CLOCK IS PRESENT
10506          ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
10507          ;'TIME' WILL ONLY SAVE
10508          ;CURRENT CYLINDER ADDRESS
10509          ;AND LOOK AHEAD REGISTERS
10510
10511
10512 037522 013746 002442    MOV      @#WRIDAT,-(SP) ;GET READY TO MOVE COMMAND
10513 037526 052716 000101    BIS      #GO.IE,(SP)   ;GET READY TO SET 'GO' AND
10514          ;ENABLE INTERRUPT
10515 037532 012677 142542    MOV      (SP)+,@RHCS1 ;GO WITH
10516          ;60 IN RHCS1 FOR WRITE DATA
10517          ;WITH INTERRUPT ENABLED
10518
10519          ;*TIME IS NOT CRITICAL
10520
10521 037536 104413          WAT      ;WAIT FOR RDY BIT TO SET
10522 037540 002300          RHCS1    ;WAIT FOR RHCS1 REGISTER
10523 037542 000200          RDY     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
10524 037544 004704          2500.   ;ALLOW 25000 MICRO SECONDS
10525 037546 004704          2500.   ;RDY MUST SET BETWEEN
10526          ;00 AND 50000 MICRO SECONDS
10527
10528          ;*SAVE REGISTERS FOR COMPARISON AFTER WRITE PROTECT
10529          ;*BUTTON HAS BEEN HIT
10530
10531 037550 004037 041672    JSR      RO,@#SAVER   ;SAVE REGISTERS
10532 037554 002272          RHWC     ;RHWC IS THE FIRST REGISTER SAVED

```



```

10589                                     ;SETUP UNIT NUMBER
10590 040000 004037 041374             JSR    RO,@#CLAREA ;CLEAR 256. WORDS, FROM WRFROM
10591 040004 002470                     WRFROM ;STARTING FROM WRFROM
10592 040006 000400                     256. ;256. WORDS
10593 040010 000377                     377 ;FILL WITH 377
10594
10595
10596                                     ;*TRY A ONE WORD WRITE
10597
10598 040012 004037 043476             JSR    RO,@#RUN ;SETUP TO RUN FOR DATA COMMAND
10599 040016 000000                     0 ;CYLINDER 0
10600 040020 000 ;.BYTE 0 ;SECTOR 0
10601 040021 000 ;.BYTE 0 ;TRACK 0
10602 040022 177777                     -1. ;WORD COUNT = 1.
10603 040024 002470                     WRFROM ;BUS ADDRESS
10604                                     ;STARTING ADDRESS OF DATA
10605                                     ;BUFFER = WRFROM
10606 040026 000000                     0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
10607 040030 010000                     FMT22 ;16 BITS PER WORD FORMAT
10608                                     ;DO NOT INHIBIT ECC CORRECTION
10609                                     ;DO NOT INHIBIT HEADER COMPARE
10610 040032 002442                     WRIDAT ;GET READY TO DO A WRIDAT
10611                                     ;WRITE DATA WITH 60 IN RHCS1
10612
10613
10614                                     ;*SAVE REGISTERS
10615 040034 004037 041672             JSR    RO,@#SAVER ;SAVE REGISTERS
10616 040040 002272                     RHWC ;RHWC IS THE FIRST REGISTER SAVED
10617 040042 004612                     SAVERE ;STARTING ADDRESS OF WHERE
10618                                     ;THE REGISTERS ARE SAVED
10619 040044 000022                     18. ;NUMBER OF REGISTERS
10620                                     ;SAVED = 18.
10621
10622 040046 013777 004606 142212     MOV    @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
10623                                     ;TO 'TIME1' IF P-CLOCK IS PRESENT
10624                                     ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
10625                                     ;'TIME' WILL ONLY SAVE
10626                                     ;CURRENT CYLINDER ADDRESS
10627                                     ;AND LOOK AHEAD REGISTERS
10628
10629
10630 040054 013746 002442             MOV    @#WRIDAT,-(SP) ;GET READY TO MOVE COMMAND
10631 040060 052716 000101             BIS    #GO!IE,(SP) ;GET READY TO SET 'GO' AND
10632                                     ;ENABLE INTERRUPT
10633 040064 012677 142210             MCV    (SP)+,@RHCS1 ;GO WITH
10634                                     ;60 IN RHCS1 FOR WRITE DATA
10635                                     ;WITH INTERRUPT ENABLED
10636                                     ;*TIME IS NOT CRITICAL
10637
10638 040070 104413                     WAT ;WAIT FOR RDY BIT TO SET
10639 040072 002300                     RHCS1 ;WAIT FOR RHCS1 REGISTER
10640 040074 000200                     RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
10641 040076 004704                     2500. ;ALLOW 25000 MICRO SECONDS
10642 040100 004704                     2500. ;RDY MUST SET BETWEEN
10643                                     ;00 AND 50000 MICRO SECONDS
10644
    
```



```

10757
10758 040334 013746 002446      MOV    @#READAT,-(SP)  ;GET READY TO MOVE COMMAND
10759 040340 052716 000101      BIS    #GO.IE,(SP)   ;GET READY TO SET 'GO' AND
10760                                     ;ENABLE INTERRUPT
10761 040344 012677 141730      MOV    (SP)+,@RHCS1  ;GO WITH
10762                                     ;70 IN RHCS1 FOR READ DATA
10763                                     ;WITH INTERRUPT ENABLED
10764
10765
10766 040350 104413      WAT                                     ;WAIT FOR RDY BIT TO SET
10767 040352 002300      RHCS1                                ;WAIT FOR RHCS1 REGISTER
10768 040354 000200      RDY                                  ;WAIT FOR RDY BIT IN RHCS1 REGISTER
10769 040356 001614      908.                                ;ALLOW 9080 MICRO SECONDS
10770 040360 001507      839.                                ;RDY MUST SET BETWEEN
10771                                     ;690 AND 17470 MICRO SECONDS
10772
10773                                     ;*COMPARE READ DATA
10774
10775 040362 004037 043542      JSR    RO,@#COMPAR  ;COMPARE TWO BLOCKS OF MEMORY
10776 040366 002470      WRFROM ;GOOD DATA STARTS FROM WRFROM
10777 040370 003534      REINTO ;TEST DATA STARTS FROM REINTO
10778 040372 000400      256.  ;256. WORDS TO BE COMPARED
10779 040374 040400      7$    ;RETURN TO 7$ ON ERROR
10780 040376 040402      8$    ;RETURN TO 8$ ON NO ERROR
10781
10782
10783 040400 104043      7$:  ERROR 43      ;WRITING WITH WRITE
10784                                     ;LOCKED CHANGED DISK
10785                                     ;GOOD DATA GIVES WHAT WAS
10786                                     ;ON DISK BEFORE WRITE WITH
10787                                     ;WRITE LOCK WAS ATTEPTED
10788                                     ;
10789                                     ;RECEIVED DATA GIVES WHAT
10790                                     ;WAS READ BACK AFTER WRITE
10791                                     ;WITH WRITE LOCKED WAS ATTEPTED
10792
10793                                     ;*SAVE REGISTERS FOR COMPARISON AFTER WRITE LOCK HAS BEEN
10794                                     ;*UNLOCKED
10795
10796 040402 004037 041672      8$:  JSR    RO,@#SAVER ;SAVE REGISTERS
10797 040402 004037 041672      RHWC  ;RHWC IS THE FIRST REGISTER SAVFD
10798 040406 002272      SAVERE ;STARTING ADDRESS OF WHERE
10799 040410 004612      18.   ;THE REGISTERS ARE SAVED
10800                                     ;NUMBER OF REGISTERS
10801 040412 000022      ;SAVED = 18.
10802
10803
10804 040414 004037 040422      ST20: TYPE ,65$      ;;TYPE ASCIZ STRING
10805 040414 104401 040422      BR    64$          ;;GET OVER THE ASCIZ
10806 040420 000407      ;:65$: .ASCIZ <15><12>/ON DRIVE /
10807 64$:
10808 040440
10809 040440 013746 004716      MOV    @#UNIT,-(SP) ;GET UNIT UNDER TEST
10810 040444 104405      TYPDS
10811 040446 104401 040454      TYPE  ,67$        ;;TYPE ASCIZ STRING
10812 040452 000440      BR    66$          ;;GET OVER THE ASCIZ
  
```

```

10813      ;;67$: .ASCIZ <15><12>/PUSH WRITE PROTCT BUTTON TO UNLOCK WRITES THEN HIT CONTINUE/<15
10814 040554      66$:
10815 040554 000000
10816
10817      ;*THE ONLY BIT THAT SHOULD CHANGE IS WRL-BIT #11 IN RHDS1
10818
10819 040556 004037 042404      JSR      RO,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
10820 040562 002322      RHDS1      ;CHANGE RHDS1 REGISTER
10821
10822 040564 000001      1          ;1 BIT/BITS TO BE CHANGED
10823 040566 000000      0          ;NEW VALUE OF WRL IS 0
10824 040570 004000      WRL        ;CHANGE WRL BIT
10825
10826      ;*COMPARE ALL REGISTERS BEFORE WRITE LOCK WAS UNLOCKED
10827      ;*WITH REGISTERS AFTER WRITE WAS UNLOCKED
10828
10829
10830 040572 004037 042512      JSR      RO,@#COMREG      ;COMPARE SAVED REGISTERS WITH
10831                                ;PRESENT VALUE
10832 040576 004612      SAVERE      ;GOOD DATA SAVED IN 'SAVERE'
10833 040600 002354      WC          ;TEST DATA STARTING FROM 'RHWC'
10834 040602 000022      18.        ;18. REGISTERS TO BE COMPARED
10835 040604 040610      9$         ;RETURN TO 9$ ON ERROR
10836 040606 040614      10$        ;RETURN TO 10$ ON NO ERROR
10837
10838 040610 104044      9$:      ERROR 44      ;UNLOCKING WRITES BY WRITE
10839 040612 000207      RTS      PC      ;LOCK BUTTON CAUSED AN ERROR
10840                                ;GOOD DATA GIVES WHAT SHOULD
10841                                ;BE THERE
10842                                ;RECEIVED DATA GIVES WHAT WAS
10843                                ;THERE AFTER WRITES WERE
10844                                ;UNLOCKED
10845                                ;ON THIS ERROR NO LOOPING IS RECOMMENDED
10846                                ;JUST A HALT ON ERROR WILL DO THE SAME
10847                                ;THING AS ONLY THE REGISTERS ARE READ
10848 040614 012737 177777 047342 10$:      MOV      #-1,@#PRITEM      ;CLEAR PREVIOUS ITEM NUMBER

```

```

10849
10850
10851
10852
10853
10854
10855
10856
10857
10858
10859
10860
10861
10862
10863
10864 040622 000004
10865 040624 012737 000001 001212
10866 040632 012737 000000 177776
10867
10868 040640 104401 040646
10869 040644 000424
10870
10871 040716
10872 040716 013746 004716
10873 040722 104405
10874 040724 104401 040732
10875 040730 000402
10876
10877 040736
10878 040736 013746 001112
10879 040742 104405
10880 040744 005037 001112
10881 040750 005037 001102
10882 040754 005737 004726
10883 040760 001413
10884
10885
10886 040762 005237 001100
10887 040766 104401 041157
10888 040772 013746 001100
10889 040776 104405
10890 041000 104401 041154
10891 041004 000137 010102
10892
10893 041010 012737 177777 047342 3$:
10894 041016 005337 004720
10895 041022 001413
10896 041024 013700 004716
10897 041030 012701 004676
10898 041034 022100 1$:
10899 041036 001401
10900 041040 000775
10901 041042 011137 004716 2$:
10902 041046 000137 010102
10903

```

```

*****
*****
*****
*TEST 51      END OF DRIVE

*      THIS IS THE END OF TEST FOR ONE DRIVE

*      IF THERE ARE MORE DRIVES, THEN THE PROGRAM
*      JUMPS TO TEST 4 FOR TESTING THE NEXT DRIVE

*      END PASS IS REACHED ONLY AFTER ALL DRIVES ARE TESTED

*****
TST5:  SCOPE
      MOV      #1,$TIMES      ;;DO 1 ITERATION
      MOV      #0,$PS        ;;REINSTATE PS TO 0

      TYPE     ,65$          ;;TYPE ASCIZ STRING
      BR       64$          ;;GET OVER THE ASCIZ
;;65$: .ASCIZ <15><12>/TOTAL ERRORS ON THIS PASS ON UNIT NO./
64$:   MOV      @#UNIT,-(SP)  ;GET READY TO TYPE UNIT NUMBER
      TYPDS
      TYPE     ,67$          ;;TYPE ASCIZ STRING
      BR       66$          ;;GET OVER THE ASCIZ
;;67$: .ASCIZ / =/
66$:   MOV      @#SERTTL,-(SP) ;GET READY TO TYPE NUMBER OF ERRORS
      TYPDS
      CLR      @#SERTTL      ;CLEAR TOTAL NUMBER OF ERRORS
      CLR      @#STSTNM     ;CLEAR TEST NUMBER
      TST      @#SELECT     ;STARTING FROM 200 ?
      BEQ      3$           ;CHECK NEXT DRIVE IF SO
                          ;CONTINUE WITH THIS ONE IF NOT

      INC      @#$PASS      ;INCREASE PASS COUNT
      TYPE     ,SENDMG      ;TYPE "END PASS #"
      MOV      @#$PASS,-(SP)

      TYPE     ,SENULL
      JMP      @#TST4       ;JUMP TEST 4 ----->

3$:   MOV      #-1,@#PRITEM  ;CLEAR PREVIOUS ITEM NUMBER
      DEC      @#NOUNITS    ;NO. OF UNITS PRESENT
      BEQ      $EOP        ;BRANCH IF ALL DRIVES COMPLETE
      MOV      @#UNIT,R0    ;UNIT UNDER TEST
      MOV      #UNITS,R1   ;TABLE POINTER
1$:   CMP      (R1)+,R0     ;IS THIS UNIT JUST TESTED ?
      BEQ      2$          ;BRANCH IF YES
      BR      1$           ;BRANCH IF NO
2$:   MOV      (R1),@#UNIT  ;MAKE THIS NEXT UNIT
      JMP      @#TST4       ;TEST THE NEXT DRIVE ----->

```



```

10904          .SBTTL
10905          .SBTTL  **SUBROUTINES**
10906          .SBTTL
10907
10908
10909          .SBTTL  END OF PASS ROUTINE
10910
10911          ::*****
10912          :*INCREMENT THE PASS NUMBER ($PASS)
10913          :*TYPE 'END PASS #XXXXX' (WHERE XXXXX IS A DECIMAL NUMBER)
10914          :*IF THERES A MONITOR GO TO IT
10915          :*IF THERE ISN'T JUMP TO TST1
10916
10917 041052      $EOP:
10918 041052      000004          SCOPE
10919 041054      005037 001102      CLR          $TSTNM          ::ZERO THE TEST NUMBER
10920 041060      005037 001212      CLR          $TIMES          ::ZERO THE NUMBER OF ITERATIONS
10921 041064      005237 001100      INC          $PASS          ::INCREMENT THE PASS NUMBER
10922 041070      042737 100000 001100  BIC          #100000,$PASS  ::DON'T ALLOW A NEG. NUMBER
10923 041076      005327          DEC          (PC)+          ::LOOP?
10924 041100      000001      $EOPCT: .WORD 1
10925 041102      003022      BGT          $DOAGN          ::YES
10926 041104      012737      MOV          (PC)+,@(PC)+  ::RESTORE COUNTER
10927 041106      000001      $ENDCT: .WORD 1
10928 041110      041100      $EOPCT
10929 041112      104401 041157      TYPE          ,SENDMG          ::TYPE 'END PASS #'
10930 041116      013746 001100      MOV          $PASS,-(SP)      ::SAVE $PASS FOR TYPEOUT
10931 041122      104405          TYPDS          ::GO TYPE--DECIMAL ASCII WITH SIGN
10932 041124      104401 041154      TYPE          ,$ENULL          ::TYPE A NULL CHARACTER
10933 041130      013700 000042      $GET42: MOV    @#42,R0          ::GET MONITOR ADDRESS
10934 041134      001405          BEQ          $DOAGN          ::BRANCH IF NO MONITOR
10935 041136      000005          RESET          ::CLEAR THE WORLD
10936 041140      004710      $ENDAD: JSR   PC,(R0)          ::GO TO MONITOR
10937 041142      000240          NOP          ::SAVE ROOM
10938 041144      000240          NOP          ::FOR
10939 041146      000240          NOP          ::ACT11
10940 041150      $DOAGN:
10941 041150      000137          JMP          @(PC)+          ::RETURN
10942 041152      006314      $RTNAD: .WORD TST1
10943 041154          377          377          000      $ENULL: .BYTE -1,-1,0          ::NULL CHARACTER STRING
10944 041157          015 042412 042116      $ENDMG: .ASCII <15><12>/END PASS #/
10945 041164      050040 051501 020123
10946 041172      000043
10947

```

10948
 10949
 10950
 10951
 10952
 10953
 10954
 10955
 10956
 10957
 10958
 10959
 10960
 10961
 10962
 10963
 10964
 10965
 10966
 10967
 10968
 10969
 10970
 10971
 10972
 10973
 10974
 10975
 10976
 10977
 10978
 10979
 10980
 10981
 10982
 10983
 10984
 10985
 10986
 10987
 10988
 10989
 10990
 10991
 10992
 10993
 10994
 10995
 10996
 10997
 10998
 10999

.SBTTL JAM CURRENT CYLINDER ROUTINE

;*THIS ROUTINE WILL CHANGE THE CURRENT CYLINDER REGISTER - 'RHCC'
 ;*BY GIVING A 'SEEK' COMMAND FOLLOWED BY AN INIT WHICH WILL LOAD
 ;*'RHCC' WITH THE DESIRED CYLINDER VALUE. THE ROUTINE THEN CHECKS
 ;*THAT THE LOADED VALUE IS CORRECT.

;*CALL JS:
 ;* JSR R0,@#MAKECYL ;DESIRED VALUE OF CURRENT CYLINDER
 ;* XC

MAKECYL:

MOV R5,-(SP) ;:PUSH R5 ON STACK
 MOV R0,@#PCJSR ;:PC OF JSR+4
 SUB #4,@#PCJSR ;:SAVE PC OF JSR
 MOV (R0)+,R5 ;:GETTING READY TO FILL DESIRED CYLINDER
 MOV R5,@RHCA ;:FILL DESIRED CYLINDER REGISTER
 CLR @RHDS1 ;:MAKE SURE DESIRED SECTOR TRACK IS NOT ILLEGAL
 MOV @#SEECOM,@RHCS1 ;:FILL SEEK COMMAND
 MOV #DMD,@RHMR ;:SET DIAGNOSTIC MODE

JSR PC,@#PUTREG ;:TAKE A REGISTER SNAPSHOT
 BIT #ERR,@#DS1 ;:CHECK FOR COMPOSITE ERROR
 BEQ 2\$;:NOT = 1, A-OK
 ERROR 103 ;:REGISTER CONTENTS INCORRECT BEFORE A
 ;:DIAGNOSTIC SEEK

2\$: BIS #GO,@RHCS1 ;:ISSUE 'GO' TO SEEK COMMAND
 NOP ;:ALLOW TIME FOR SEEK TO HANG UP
 NOP ;:ALLOW TIME FOR SEEK TO HANG UP
 NOP ;:ALLOW TIME FOR SEEK TO HANG UP
 NOP ;:ALLOW TIME FOR SEEK TO HANG UP

JSR PC,@#PUTREG ;:TAKE A 2ND REGISTER SNAPSHOT
 BIT #ERR,@#DS1 ;:CHECK FOR ERRORS
 BEQ 3\$;:NOT = 1, A-OK
 ERROR 104 ;:REGISTER CONTENTS INCORRECT AFTER
 ;:A DIAGNOSTIC SEEK

3\$: JSR PC,@#CLDISK ;:GIVE INIT TO FORCE THE TRANSFER
 MOV @RHCC,@#SBDDAT ;:TEST DATA
 CMP R5,@#SBDDAT ;:COMPARE CURRENT CYLINDER
 BEQ 1\$;:BRANCH IF GOOD
 MOV R5,@#SGDDAT ;:GOOD VALUE OF RHCC
 MOV @RHCC,@#REGADR ;:FAILING REGISTER ADDRESS
 ERROR 77 ;:CURRENT CYLINDER DOES NOT MATCH DESIRED CYLINDER
 ;:REGISTER AFTER A SEEK AND AN INIT

1\$: MOV (SP)+,R5 ;:POP STACK INTO R5
 RTS R0

041174
 041174 010546
 041176 010037 041560
 041202 162737 000004 041560
 041210 012005
 041212 010577 141074
 041216 005077 141062
 041222 013777 002452 141050
 041230 012777 000001 141062
 041236 004737 043436
 041242 032737 040000 002404
 041250 001401
 041252 104103
 041254 052777 000001 141016 2\$:
 041262 000240
 041264 000240
 041266 000240
 041270 000240
 041272 004737 043436
 041276 032737 040000 002404
 041304 001401
 041306 104104
 041310 004737 041524 3\$:
 041314 017737 141014 001126
 041322 020537 001126
 041326 001406
 041330 010537 001124
 041334 013737 002334 004600
 041342 104077
 041344 1\$:
 041344 012605
 041346 000200

```
11000
11001      ;*THIS FILLS MEMORY WITH GIVEN DATA
11002      ;*USED CHIEFLY FOR HEADER INFORMATION
11003      ;*CALL IS
11004      ;*      JSR      RO,@#FLHEAD      ;FILL HEADER
11005      ;*      LOC      ;LOCATION WHERE SAVED
11006      ;*      XN      ;NUMBER OF WORDS
11007      ;*      XD1     ;DATA REPEATED XN TIMES
11008      ;*      XD2     ;DATA REPEATED XN TIMES
11009      ;*
11010      ;*
11011      ;*
11012      ;*
11013      041350      FLHEAD:
11014      041350      010146      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
11015      041352      010246      MOV      R2,-(SP)      ;;PUSH R2 ON STACK
11016      041354      012001      MOV      (R0)+,R1      ;R1 HAS ADDRESS OF WHERE TO SAVE
11017      041356      012002      MOV      (R0)+,R2      ;R2 HAS NUMBER OF WORDS
11018
11019      ;*NOW FILL DATA
11020
11021      041360      012021      1$:      MOV      (R0)+,(R1)+      ;SAVE DATA
11022      041362      005302      DEC      R2      ;DECREMENT COUNT
11023      041364      001375      BNE     1$      ;BRANCH IF INCOMPLETE
11024      041366      012602      MOV      (SP)+,R2      ;;POP STACK INTO R2
11025      041370      012601      MOV      (SP)+,R1      ;;POP STACK INTO R1
11026      041372      000200      RTS      R0
11027
11028
11029
11030      ;*THIS CLEARS ANY BLOCK OF MEMORY.
11031      ;*FILLING IT WITH ANY DATA
11032      ;*CALL IS
11033      ;*      JSR      RO,@#CLAREA
11034      ;*      F      ;FROM
11035      ;*      N      ;NUMBER OF WORDS
11036      ;*      D      ;DATA TO BE FILLED
11037      ;*
11038      ;*R1 WILL HAVE STARTING ADDRESS OF BLOCK TO BE FILLED
11039      ;*R2 WILL HAVE NUMBER OF WORDS
11040      ;*R3 WILL HAVE DATA
11041
11042      041374      CLAREA:
11043      041374      010146      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
11044      041376      010246      MOV      R2,-(SP)      ;;PUSH R2 ON STACK
11045      041400      010346      MOV      R3,-(SP)      ;;PUSH R3 ON STACK
11046      041402      012001      MOV      (R0)+,R1      ;FROM
11047      041404      012002      MOV      (R0)+,R2      ;NUMBER
11048      041406      012003      MOV      (R0)+,R3      ;DATA
11049      041410      010321      1$:      MOV      R3,(R1)+      ;MOVE DATA
11050      041412      005302      DEC      R2      ;COUNT
11051      041414      001375      BNE     1$      ;BRANCH IF NOT COMPLETE
11052      041416      012603      MOV      (SP)+,R3      ;;POP STACK INTO R3
11053      041420      012602      MOV      (SP)+,R2      ;;POP STACK INTO R2
11054      041422      012601      MOV      (SP)+,R1      ;;POP STACK INTO R1
11055      041424      000200      RTS      R0      ;RETURN TO MAIN PROGRAM
```

11056
11057
11058
11059
11060
11061
11062
11063
11064
11065
11066
11067
11068
11069
11070 041426
11071 041426 010146
11072 041430 010246
11073 041432 012001
11074 041434 012002
11075 041436 162701 002272
11076 041442 010261 004612
11077 041446 012602
11078 041450 012601
11079 041452 000200
11080
11081
11082

;*THIS IS A SUBROUTINE TO FILL SAVED REGISTER LOCATION
;*WITH GIVEN VALUE
;*CALL IS
;* JSR RO,@#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 #RHWC,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 RO ;RETURN TO MAIN PROGRAM

```

11083      ;*THIS SUBROUTINE SETS UP FOR SEARCH
11084      ;*CALL IS
11085      ;*      JSR      R0,@#SRCH
11086      ;*      C          ;CYLINDER
11087      ;*.BYTE S          ;SECTOR
11088      ;*.BYTE T          ;TRACK
11089
11090 041454 012077 140632      SRCH:  MOV      (R0)+,@RHCA      ;SET DESIRED CYLINDER ADDRESS
11091 041460 012077 140620      MOV      (R0)+,@RHDST     ;SET DESIRED SECTOR/TRACK ADDRESS
11092 041464 013777 002434 140606  MOV      @#SERCH,@RHCS1  ;GET READY FOR SEARCH
11093                                     ;WITH 30 IN RHCS1
11094 041472 000200      RTS      R0
11095
11096
11097
11098
11099
11100
11101
11102      ;*THIS SUBROUTINE SETS UP FOR SEEK COMMANDS
11103      ;*CALL IS
11104      ;*      JSR      R0,@#SEEKCY
11105      ;*      C          ;CYLINDER
11106      ;*
11107
11108 041474 012077 140612      SEEKCY: MOV      (R0)+,@RHCA      ;SET DESIRED CYLINDER ADDRESS
11109 041500 013777 002452 140572  MOV      @#SEECOM,@RHCS1 ;MOV 4 INTO RHCS1
11110 041506 000200      RTS      R0          ;RETURN TO MAIN PROGRAM
  
```

```
11111
11112      ;*THIS SUBROUTINE SETS UP FOR OFFSET COMMANDS
11113      ;*CALL IS
11114      ;*   JSR   RO,@#OFSET
11115      ;*   0      ;MICRO INCHES OFSET
11116
11117 041510 052077 140574      OFFSET: BIS   (RO)+,@RHOF      ;SET OFFSET REGISTER
11118 041514 013777 002454 140556  MOV   @#OFSETC,@RHCS1 ;MOV14 INTO RHCS1
11119 041522 000200      RTS   RO      ;RETURN TO MAIN PROGRAM
11120
11121
11122 041524 013701 002300      CLDISK: MOV   @#RHCS1,      R1      ;R1 WILL BE CONTROL AND STATUS1
11123 041530 013702 002276      MOV   @#RHCS2,      R2      ;R2 WILL BE CONTROL AND STATUS2
11124 041534 013703 002322      MOV   @#RHDS1,      R3      ;R3 WILL BE DISK STATUS REGISTER1
11125 041540 013704 002302      MOV   @#RHER1,      R4      ;R4 WILL BE ERROR REGISTER #1
11126
11127 041544 012712 000040      MOV   #CLR,@R2      ;CLEAR ALL REG.
11128 041550 013712 004716      MOV   @#UNIT,@R2    ;REINSTATE UNIT NO.
11129 041554 005011      CLR   @R1      ;CLEAR FUNCTION BITS
11130 041556 000207      RTS   PC
```

```
11131
11132
11133
11134
11135 ;*THIS CHECKS THAT DEVICE AVAILABLE (DVA) AND READY (RDY) IN RHCS1 = 1
11136 ;*AND CHECKS MEDIUM ON LINE (MOL), DEVICE PRESENT (DPR), DEVICE READY
11137 ;* (DRY) IN RHDS1 = 1
11138
11139 ;*IT ALSO CHECKS THAT THERE ARE NO BITS STUCK AT 1 IN RHDS1
11140
11141
11142 041560 000000 PCJSR: 0 ;PC OF JSR
11143
11144 041562 011637 041560 CHECK: MOV (SP),@#PCJSR ;SAVE PC OF JSR+4
11145 041566 162737 000004 041560 SUB #4,@#PCJSR ;GET PC OF JSR
11146 041574 011346 MOV @R3,-(SP) ;GET RHDS1
11147 041576 052716 000100 BIS #VV,(SP) ;DONT CHECK VV BIT
11148 041602 000406 BR CHECKC ;GOTO COMMON CHECK ROUTINE
11149
11150 041604 011637 041560 CHECKT: MOV (SP),@#PCJSR ;SAVE PC OF JSR+4
11151 041610 162737 000004 041560 SUB #4,@#PCJSR ;GET PC OF JSR
11152 041616 011346 MOV @R3,-(SP) ;GET RHDS1 & DO VV CHECK AT 3$
11153
11154 041620 011146 CHECKC: MOV @R1,-(SP) ;GET CS1
11155 041622 042716 173577 BIC #173577,(SP) ;CLEAR UNWANTED BITS
11156 041626 022726 004200 CMP #DVA!RDY,(SP)+ ;RHCS1 SHOULD HAVE DEVICE AVAILABLE
11157 ;AND BE READY
11158 041632 001403 BEQ 3$ ;BRANCH IF IT DOES
11159 041634 011137 001122 MOV @R1,@#$BDADR ;BAD DATA REGISTER (RHCS1)
11160 041640 104061 ERROR 61 ;RHCS1 DID NOT HAVE DEVICE
11161 ;AVAILABLE RIGHT AT THE START
11162 ;ALL OTHER BITS SHOULD BE 0
11163
11164 041642 042716 102000 3$: BIC #ATA!LBT,(SP) ;CLEAR UNWANTED BITS
11165 041646 022726 010700 CMP #MOL!DPR!DRY!VV,(SP)+ ;RHDS1 SHOULD HAVE THESE SET
11166 041652 001404 BEQ 7$ ;BRANCH IF GOOD
11167 041654 011337 001122 MOV @R3,@#$BDADR ;BAD DATA IN REGISTER (RHDS1)
11168 041660 104062 ERROR 62 ;RHDS1 HAS SOME BITS OTHER
11169 ;THAN MOL, DRY, DPR,VV SET
11170 ;ALL OTHER BITS SHOULD BE 0
11171 041662 000207 RTS PC ;RETURN TO TEST AND HALT/CONTINUE
11172 ;DEPENDING ON WHETHER THIS IS A
11173 ;"FATAL" ERROR
11174
11175 041664 062716 000006 7$: ADD #6,(SP) ;ADJUST STACK TO JUMP OVER HALT IN TEST
11176 041670 000207 RTS PC ;RETURN TO THE TEST AND CONTINUE
11177
```

11178
 11179
 11180
 11181
 11182
 11183
 11184
 11185
 11186
 11187
 11188
 11189
 11190
 11191
 11192
 11193
 11194
 11195
 11196
 11197
 11198
 11199
 11200
 11201
 11202
 11203
 11204
 11205
 11206
 11207
 11208
 11209
 11210
 11211
 11212
 11213
 11214
 11215
 11216
 11217
 11218
 11219
 11220
 11221
 11222
 11223
 11224
 11225
 11226
 11227
 11228
 11229
 11230
 11231
 11232
 11233

```

; *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     @RHLA,@#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
  
```



```
11234      ;*      WAT
11235      ;*      A      ;ABSOLUTE REGISTER ADDRESS
11236      ;*      B      ;BIT WAITED FOR
11237      ;*      TA     ;TIME ALLOWED GIVEN IN 10 MICROSEC
11238      ;*      TO     ;TOLERANCE PLUS/MINUS IN 10 MICROSEC
11239      ;*
11240      ;*R1-WILL HAVE TIME ALLOWED IN 10 MICRO SECONDS
11241      ;*R2-WILL HAVE TOLERANCE PLUS/MINUS IN 10 MICRO SECONDS
11242      ;*MINIMUM TIME THAT CAN BE MEASURED IS ABOUT 12 MICRO SECONDS
11243      ;*FOR THE SLOWEST PROCESSOR
11244
11245 041754 000000      WAITPC: 0      ;WAT PC
11246 041756 000000      WAITRE: 0      ;WAIT ON REGISTER ADDRESS
11247 041760 000000      WAITBT: 0      ;WAIT ON BIT
11248 041762 000000      WAITTM: 0      ;WAITED TIME
11249 041764 005037 041762      WAIT.P: CLR @#WAITTM ;CLEAR WAITED TIME
11250 041770 005077 140352      CLR @PCLBUF ;CLEAR COUNT SET BUFFER
11251 041774 012777 000021 140342      MOV #GO!BIT4,@PCLCSR ;COUNT UP, 100 KHZ, START CLOCK
11252 042002 010046      MOV R0,-(SP) ;;PUSH R0 ON STACK
11253 042004 010146      MOV R1,-(SP) ;;PUSH R1 ON STACK
11254 042006 010246      MOV R2,-(SP) ;;PUSH R2 ON STACK
11255 042010 010346      MOV R3,-(SP) ;;PUSH R3 ON STACK
11256 042012 016600 000010      MOV 10(SP),R0 ;R0 HAS ADDRESS OF NEXT LOCATION
11257 042016 010037 041754      MOV R0,@#WAITPC ;NOW WAITPC HAS WAT PC + 2
11258 042022 162737 000002 041754      SUB #2,@#WAITPC ;WAT PC IS IN WAITPC
11259 042030 013037 041756      MOV @(R0)+,@#WAITRE ;WAIT ON REGISTER ADDRESS
11260 042034 012037 041760      MOV (R0)+,@#WAITBT ;WAIT ON BIT
11261 042040 012001      MOV (R0)+,R1 ;R1 HAS TIME IN 10 MSEC
11262 042042 012002      MOV (R0)+,R2 ;R2 HAS TOLERANCE IN 10 MSEC
11263 042044 010066 000010      MOV R0,10(SP) ;RESTORE RETURN ON STACK
11264
11265      ;*THIS SECTION WAITS FOR BIT, THROUGH TWO COUNT DOWNS
11266 042050 013703 042222      MOV @#TIMCNT,R3 ;R3 IS A TEMPORARY COUNTER
11267 042054 033777 041760 177674 1$:      BIT @#WAITBT,@#WAITRE ;IS REQUIRED BIT THERE
11268 042062 001025      BNE 4$ ;BRANCH IF YES
11269 042064 005303      DEC R3 ;COUNT IF REQUIRED BIT NOT THERE
11270 042066 001372      BNE 1$
11271 042070 013703 042222      MOV @#TIMCNT,R3 ;TEMPORARY COUNTER
11272 042074 033777 041760 177654 2$:      BIT @#WAITBT,@#WAITRE ;IS REQUIRED BIT THERE
11273 042102 001015      BNE 4$ ;BRANCH IF YES
11274 042104 005303      DEC R3 ;COUNT IF REQUIRED BIT NOT THERE
11275 042106 001372      BNE 2$
11276 042110 017737 177642 001126      MOV @#WAITRE,@#SBDDAT ;REGISTER CONTENTS FOR TYPEOUT
11277 042116 032777 000100 140154      BIT #IE,@RHCS1 ;DID ANY INTERRUPT OCCUR
11278 042124 001402      BEQ 3$ ;BRANCH IF YES
11279 042126 104001      ERROR 1 ;RPO4 DID NOT INTERRUPT
11280 042130 000427      BR 7$ ;OUT
11281 042132 104002      ERROR 2 ;RPO4 INTERRUPTED BUT WAITED
11282      ;ON BIT DID NOT OCCUR
11283      ;EVEN AFTER TWO COUNT DOWNS
11284      ;FROM 177777 TO 0
11285 042134 000425      BR 7$ ;OUT
11286
11287      ;*NOW TIME AND TOLERANCE WILL BE CHECKED
11288 042136 017737 177614 001126 4$:      MOV @#WAITRE,@#SBDDAT ;REGISTER CONTENTS FOR TYPEOUT
11289 042144 032777 000100 140126      BIT #IE,@RHCS1 ;DID ANY INTERRUPT OCCUR
```

```

11290 042152 001402          BEQ      5$          ;BRANCH IF YES
11291 042154 104003          ERROR    3          ;INTERRUPT DID NOT OCCUR EVEN
11292                                ;AFTER ONE BNE AND ONE MOV
11293                                ;OF THE WAITED ON BIT SETTING
11294 042156 000414          BR       7$          ;OUT
11295 042160 160201          SUB      R2,R1      ;R1 NOW HAS LOWER LIMIT OF TIME
11296 042162 023701 041762 5$:  CMP      @#WAITTM,R1 ;FOR GOOD RESULTS, WAITTM
11297                                ;MUST BE GREATER OR EQUAL
11298                                ;TORI
11299 042166 103002          BHS     6$          ;BRANCH IF GOOD
11300 042170 104004          ERROR    4          ;BIT DID OCCUR BUT TIME
11301                                ;TAKEN IS BELOW LOWER LIMIT
11302 042172 000406          BR       7$          ;OUT
11303
11304 042174 060202          6$:  ADD     R2,R2      ;DOUBLE TOLERANCE
11305 042176 060201          ADD     R2,R1      ;R1 NOW HAS UPPER LIMIT OF TIME
11306 042200 020137 041762  CMP     R1,@#WAITTM ;FOR GOOD RESULTS, WAITTM
11307                                ;MUST BE LESS OR EQUAL TO R1
11308 042204 103001          BHS     7$          ;BRANCH IF GOOD
11309 042206 104004          ERRGR   4          ;BIT DID OCCUR BUT TIME TAKEN
11310                                ;IS ABOVE UPPER LIMIT
11311 042210          7$:
11312 042210 012603          MOV     (SP)+,R3    ;;POP STACK INTO R3
11313 042212 012602          MOV     (SP)+,R2    ;;POP STACK INTO R2
11314 042214 012601          MOV     (SP)+,R1    ;;POP STACK INTO R1
11315 042216 012600          MOV     (SP)+,R0    ;;POP STACK INTO R0
11316 042220 000002          RTI                    ;RETURN TO MAIN TEST
11317
11318
11319
11320
11321
11322
11323          ;*THIS IS A WAIT LOOP WHEN NO P-CLOCK IS AVAILABLE
11324          ;*NO TIMING IS DONE
11325          ;*CALL IS
11326          ;*   WAT
11327          ;*   A           ;ABSOLUTE REGISTER ADDRESS
11328          ;*   B           ;BIT WAITE) FOR
11329          ;*   TA          ;TIME-NOT USED HERE
11330          ;*   TO          ;TIME-NOT USED HERE
11331          ;*R3-IS A TEMPORARY COUNTER
11332 042222 177777          TIMCNT: 177777          ;COUNT FOR WAIT LOOP
11333
11334          WAIT.T:
11335 042224 010046          MOV     R0,-(SP)    ;;PUSH R0 ON STACK
11336 042226 010346          MOV     R3,-(SP)    ;;PUSH R3 ON STACK
11337 042230 016600 000004          MOV     4(SP),R0    ;R0 HAS ADDRESS OF NEXT LOCATION
11338 042234 010037 041754          MOV     R0,@#WAITPC ;WAT PC +2 IS IN WAITPC
11339 042240 162737 000002 041754          SUB     #2,@#WAITPC ;WAT PC IS IN WAITPC
11340 042246 013037 041756          MOV     @ (R0)+,@#WAITRE ;WAIT ON REGISTER ADDRESS
11341 042252 012037 041760          MOV     (R0)+,@#WAITBT ;WAIT ON BIT
11342 042256 022020          CMP     (R0)+,(R0)+ ;DUMP NEXT TWO WORDS-TA, TO
11343 042260 010066 000004          MOV     R0,4(SP)    ;RESTORE RETURN ON STACK
11344
11345          ;*THIS HAS THE TWO COUNT DOWNS FROM 177777

```

```

11346 042264 0137J3 042222      MOV      @#TIMCNT,R3      ;R3 HAS TEMPORARY COUNT
11347 042270 033777 041760 177460 1$:  BIT      @#WAITBT,@WAITRE ;IS REQUIRED BIT THERE
11348 042276 001025      BNE      4$              ;BRANCH IF YES
11349 042300 005303      DEC      R3              ;COUNT IF REQUIRED BIT NOT THERE
11350 042302 001372      BNE      1$
11351 042304 013703 042222      MOV      @#TIMCNT,R3      ;SECOND COUNT DOWN FROM 177777
11352 042310 033777 041760 177440 2$:  BIT      @#WAITBT,@WAITRE ;IS REQUIRED BIT THERE
11353 042316 001015      BNE      4$              ;BRANCH IF YES
11354 042320 005303      DEC      R3              ;COUNT IF REQUIRED BIT NOT THERE
11355 042322 001372      BNE      2$
11356 042324 017737 177426 001126  MOV      @WAITRE,@#SBDDAT ;REGISTER CONTENTS FOR TYPEOUT
11357 042332 032777 000100 137740  BIT      #IE,@RHCS1      ;DID ANY INTERRUPT OCCUR
11358 042340 001402      BEQ      3$              ;BRANCH IF YES
11359 042342 104001      ERROR   1              ;RP04 DID NOT INTERRUPT
11360                                ;BIT DID NOT OCCUR
11361 042344 000414      BR       5$              ;OUT
11362 042346 104002      3$:  ERROR   2              ;RP04 INTERRUPTED BUT
11363                                ;WAITED ON BIT DID NOT OCCUR
11364                                ;EVEN AFTER TWO COUNT DOWNS
11365                                ;FROM 177777 TO 0
11366 042350 000412      BR       5$              ;OUT
11367
11368                                ;*BIT DID SET SO CHECK IF INTERRUPT OCCURED
11369 042352 000240      4$:  NOP                       ;ALLOW TIME FOR INTERRUPT
11370 042354 032777 000100 137716  BIT      #IE,@RHCS1      ;DID ANY INTERRUPT OCCUR
11371 042362 001405      BEQ      5$              ;BRANCH IF YES
11372 042364 017737 177366 001126  MOV      @WAITRE,@#SBDDAT ;REGISTER CONTENTS FOR TYPEOUT
11373 042372 104003      ERROR   3              ;INTERRUPT DID NOT OCCUR
11374                                ;EVEN AFTER ONE BNE OF
11375                                ;THE WAITED ON BIT OCCURING
11376 042374 000400      BR       5$              ;OUT
11377 042376      5$:
11378 042376 012603      MOV      (SP)+,R3        ;;POP STACK INTO R3
11379 042400 012600      MOV      (SP)+,R0        ;;POP STACK INTO R0
11380 042402 000002      RTI                       ;RETURN TO MAIN TEST
  
```

```
11381 ;*THIS CHANGES REGISTER SAVED VALUE
11382 ;*CALL IS
11383 ;* JSR RO,@#CHREG
11384 ;* R ;REGISTER TO BE CHANGED
11385 ;* N ;NUMBER OF BITS TO BE CHANGED
11386 ;* NEW ;NEW VALUE OF BIT MUST BE 0 OR 1
11387 ;* P ;POSITION OF BIT TO BE CHANGED
11388 ;*NEW AND P WILL BE REPEATED N NUMBER OF TIMES
11389 042404 CHREG:
11390 042404 010146 MOV R1,-(SP) ;;PUSH R1 ON STACK
11391 042406 010246 MOV R2,-(SP) ;;PUSH R2 ON STACK
11392 042410 012001 MOV (R0)+,R1 ;R1 HAS ADDRESS OF ADDRESS OF REGISTER
11393 042412 012002 MOV (R0)+,R2 ;R2 HAS NUMBER OF CHANGES
11394 042414 162701 002272 SUB #RHWC,R1 ;R1 HAS OFFSET OF REQUIRED REGISTER
11395 042420 005720 1$: TST (R0)+ ;IS A BIC OR A BIS TO BE DONE
11396 042422 001403 BEQ 2$ ;BRANCH IF A BIC IS REQUIRED
11397 042424 052061 004612 BIS (R0)+,SAVERE(R1) ;SET REQUIRED BIT
11398 042430 000402 BR 3$ ;BRANCH TO DECREMENT COUNT
11399 042432 042061 004612 2$: BIC (R0)+,SAVERE(R1) ;CLEAR REQUIRED BIT
11400 042436 005302 3$: DEC R2 ;DECREMENT NUMBER OF CHANGES
11401 042440 001367 BNE 1$ ;BRANCH IF NOT COMPLETE
11402 042442 012602 MOV (SP)+,R2 ;;POP STACK INTO R2
11403 042444 012601 MOV (SP)+,R1 ;;POP STACK INTO R1
11404 042446 000200 RTS R0 ;RETURN TO MAIN PROGRAM
11405
11406
11407
11408
11409
11410
```

```
11411 ;*THIS FILLS A BLOCK WITH INCREMENTAL DATA
11412 ;*CALL IS
11413 ;* JSR RO,@#FILL
11414 ;* F ;FROM
11415 ;* N ;NUMBER OF WORDS
11416 ;* S ;STARTING VALUE OF DATA
11417 ;* I ;INCREMENT DATA BY
11418
11419 042450 FILL:
11420 042450 010146 MOV R1,-(SP) ;;PUSH R1 ON STACK
11421 042452 010246 MOV R2,-(SP) ;;PUSH R2 ON STACK
11422 042454 010346 MOV R3,-(SP) ;;PUSH R3 ON STACK
11423 042456 010446 MOV R4,-(SP) ;;PUSH R4 ON STACK
11424 042460 012001 MOV (R0)+,R1 ;R1 HAS ADDRESS WHERE DATA IS TO GO
11425 042462 012002 MOV (R0)+,R2 ;R2 HAS NUMBER OF WORDS TO BE FILLED
11426 042464 012003 MOV (R0)+,R3 ;STARTING VALUE OF DATA
11427 042466 012004 MOV (R0)+,R4 ;R4 HAS INCREMENT
11428 ;*NOW DATA WILL BE FILLED
11429 042470 010321 1$: MOV R3,(R1)+ ;FILL DATA
11430 042472 060403 ADD R4,R3 ;GET NEXT VALUE OF DATA
11431 042474 005302 DEC R2 ;DECREMENT COUNT
11432 042476 001374 BNE 1$ ;BRANCH IF ALL NOT DONE
11433 042500 012604 MOV (SP)+,R4 ;;POP STACK INTO R4
11434 042502 012603 MOV (SP)+,R3 ;;POP STACK INTO R3
11435 042504 012602 MOV (SP)+,R2 ;;POP STACK INTO R2
11436 042506 012601 MOV (SP)+,R1 ;;POP STACK INTO R1
```

CZRJID0, RP04/5/6 FCTNL CTLR1
CZRJID.P11 28-MAR-79 09:03

MACY11 30A(1052) 25-MAY-79 10:30 PAGE 236
JAM CURRENT CYLINDER ROUTINE

C 3

SEQ 0235

11437 042510 000200
11438
11439
11440

RTS R0

;RETURN TO MAIN PROGRAM

```

11441
11442
11443
11444      ;*THIS IS A SUBROUTINE TO COMPARE REGISTERS
11445      ;*GOOD DATA IS ALREADY SAVED IN 'SAVERE'
11446      ;*TEST DATA IS IN THE REGISTERS
11447      ;*CALL IS
11448      ;*      JSR      RO,@#COMREG
11449      ;*      SAVERE      ;GOOD DATA
11450      ;*      RHCS1      ;ADDRESS OF ADDRESS TEST DATA
11451      ;*      N.        ;RETURN FOR ERROR
11452      ;*      RG        ;RETURN FOR GOOD COMPARISON
11453      ;*ON RETURN WITH ERROR '$GDDAT' HAS GOOD DATA, '$BDDAT' HAS BAD DATA
11454      ;*'REGADR' HAS REGISTER ADDRESS
11455
11456      042512      COMREG:
11457      042512      010146      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
11458      042514      010246      MOV      R2,-(SP)      ;;PUSH R2 ON STACK
11459      042516      010346      MOV      R3,-(SP)      ;;PUSH R3 ON STACK
11460      042520      010446      MOV      R4,-(SP)      ;;PUSH R4 ON STACK
11461      042522      010546      MOV      R5,-(SP)      ;;PUSH R5 ON STACK
11462      042524      012001      MOV      (R0)+,R1      ;R1 HAS ADDRESS OF GOOD DATA
11463      042526      012002      MOV      (R0)+,R2      ;R2 HAS ADDRESS OF ADDRESS OF TEST DATA
11464      042530      012003      MOV      (R0)+,R3      ;R3 HAS NUMBER OF WORDS
11465      042532      012004      MOV      (R0)+,R4      ;R4 HAS RETURN FOR ERROR
11466      042534      011000      MOV      (R0),R0      ;R0 HAS RETURN ON NO ERROR
11467      ;*NOW SAVE REGISTERS
11468      042536      004737      043436      JSR      PC,@#PUTREG      ;SAVE REGISTERS
11469      042542      113737      004637      002401      MOV      @#SAVERE+25,@#AS+1;MAKE UPPER BYTE OF R HAS SAME
11470      042550      012705      177776      MOV      #-2,R5      ;PRESET R5 TO -2
11471      ;*NOW COMPARES WILL MADE
11472      042554      062705      000002      1$:      ADD      #2,R5      ;INCREMENT TO INDEX
11473      042560      022122      CMP      (R1)+,(R2)+      ;COMPARE REGISTER CONTENTS
11474      042562      001420      BEQ      2$      ;BRANCH IF GOOD
11475      042564      014137      001124      MOV      -(R1),@#$GDDAT      ;SAVE GOOD DATA
11476      042570      014237      001126      MOV      -(R2),@#$BDDAT      ;SAVE BAD DATA
11477      042574      016537      002272      004600      MOV      RHC(R5),@#REGADR      ;SAVE ADDRESS OF FAILING REGISTER
11478      042602      004714      JSR      PC,@R4      ;RETURN TO MAIN PROGRAM
11479      ;TO PRINT ERROR
11480      042604      022122      CMP      (R1)+,(R2)+      ;UNDO -(R1) AND -(R2) FOR ERRORS
11481      042606      017746      136326      MOV      @SWR,-(SP)      ;GET SWITCH SETTING
11482      042612      042716      177177      BIC      #^C600,(SP)      ;KEEP ONLY SWITCH 7 AND 8
11483      042616      022726      000200      CMP      #SW07,(SP)+      ;IS 7 SET AND 8 DOWN
11484      042622      001402      BEQ      3$      ;BRANCH OUT IF YES
11485      042624      005303      2$:      DEC      R3      ;ARE ALL COMPARES DONE
11486      042626      001352      BNE      1$      ;BRANCH IF NOT COMPLETE
11487
11488      042630      3$:
11489      042630      012605      MOV      (SP)+,R5      ;;POP STACK INTO R5
11490      042632      012604      MOV      (SP)+,R4      ;;POP STACK INTO R4
11491      042634      012603      MOV      (SP)+,R3      ;;POP STACK INTO R3
11492      042636      012602      MOV      (SP)+,R2      ;;POP STACK INTO R2
11493      042640      012601      MOV      (SP)+,R1      ;;POP STACK INTO R1
11494      042642      000200      RTS      R0      ;RETURN TO MAIN PROGRAM
11495      042644      000000      4$:      .WORD      0      ;TEMP STORAGE
  
```

```
11496
11497
11498
11499      ;*HERE IS A DETAILED EXPLANATION OF HOW THE LOOP ON ERROR WORKS.
11500      ;*ON HITTING AN ERROR IF THE LOOP ON ERROR SWITCH IS SET, THE
11501      ;*PROGRAM GOES BACK - USUALLY BACK TO THE BEGINNING OF THE TEST.
11502
11503      ;*WHEN THIS OPERATOR SELECTABLE SCOPE LOOP IS USED THEN THE POINT
11504      ;*THE PROGRAM GOES BACK TO CAN BE CHANGED.
11505      ;*THE RESTRICTIONS TO THE POINT WHERE THE PROGRAM CAN GO ARE: -
11506      ;*1. IT MUST BE WITHIN THE TEST UNDER CONSIDERATION
11507      ;*2. LOOP ON ERROR SWITCH MUST BE SET
11508      ;*3. THE ERROR MUST OCCUR WITHIN THE TEST UNDER CONSIDERATION
11509      ;*IF THE ERROR DOES NOT OCCUR WITHIN THE TEST UNDER CONSIDERATION
11510      ;*THE PROGRAM WILL REVERT TO NORMAL OPERATION. HOWEVER, IF LOOP ON
11511      ;*TEST SWITCH IS SET AND THIS OPERATOR SELECTABLE SCOPE LOOP IS USED
11512      ;*THEN THE PROGRAM WILL LOOP BACK TO THE SELECTED POINT WHEN IT
11513      ;*COMES TO THE END OF THE TEST UNDER CONSIDERATION.
11514      ;*
11515      ;*AFTER LOOPING FOR SOME TIME IF THE LOOP SWITCH IS PUT DOWN THEN
11516      ;*NORMAL OPERATION WILL CONTINUE.
11517
11518 042646 000000      TESTAD: 0      ;FIRST ADDRESS OF TEST
11519 042650      OPERSEL:
11520 042650 005037 177776      CLR      PS      ;MAKE PROCESSOR STATUS ZERO
11521 042654 012737 177777      MOV      #-1,@#PRITEM ;CLEAR PREVIOUS ITEM NUMBER
11522 042662 104401 042670      TYPE     ,65$      ;;TYPE ASCIZ STRING
11523 042666 000421      BR       64$      ;;GET OVER THE ASCIZ
11524      ;;65$: .ASCIZ <15><12>/THE PROGRAM WAS IN TEST NUMBER /
11525      64$:
11526 042732 013746 004604      MOV      @#TSTNM,-(SP) ;GET READY TO TYPE TEST
11527 042736 104402      TYPOC      ;NUMBER
11528 042740 104401 042746      TYPE     ,67$      ;;TYPE ASCIZ STRING
11529 042744 000414      BR       66$      ;;GET OVER THE ASCIZ
11530      ;;67$: .ASCIZ <15><12>/THE LOOP BACK PC WAS /
11531      66$:
11532 042776 013746 001110      MOV      @#SLPERR,-(SP) ;GET READY TO TYPE LOOP BACK PC
11533 043002 104402      TYPOC      ;NUMBER
11534 043004 104401 001223      TYPE     ,SCLRF
11535 043010 104401 043016      TYPE     ,69$      ;;TYPE ASCIZ STRING
11536 043014 000430      BR       68$      ;;GET OVER THE ASCIZ
11537      ;;69$: .ASCIZ <15><12>/SET SWITCH FOR LOOP ON ERROR OR LOOP ON TEST/
11538      68$:
11539 043076 104401 043104      TYPE     ,71$      ;;TYPE ASCIZ STRING
11540 043102 000430      BR       70$      ;;GET OVER THE ASCIZ
11541      ;;71$: .ASCIZ <15><12>/TYPE THE FIRST PC OF THE TEST TO BE LOOPED ON/
11542      70$:
11543 043164 104401 043172      TYPE     ,73$      ;;TYPE ASCIZ STRING
11544 043170 000422      BR       72$      ;;GET OVER THE ASCIZ
11545      ;;73$: .ASCIZ <15><12>/ FOLLOWED BY A CARRIAGE RETURN /<15><12>
11546      72$:
11547 043236 104412      RDOCT
11548 043240 062716 000002      ADD      #2,(SP)      ;GET LPADR
11549 043244 012637 001106      MOV      (SP)+,@#LPADR
11550 043250 104401 043256      TYPE     ,75$      ;;TYPE ASCIZ STRING
11551 043254 000417      BR       74$      ;;GET OVER THE ASCIZ
```

CZRJID, RP04/5/6 FCTNL CTRL1
CZRJID.P11 28-MAR-79 09:03

MACY11 30A(1052) 25-MAY-79 10:30 PAGE 239
JAM CURRENT CYLINDER ROUTINE

F 3

SEQ 0238

11552
11553 043314
11554 043314 104401 043322
11555 043320 000440
11556
11557 043422
11558 043422 104412
11559 043424 012637 001110
11560 043430 013746 001106
11561 043434 000002
11562
11563

```
::75$: .ASCIZ <15><12>/TYPE THE PC WHERE YOU WANT/  
74$:  
TYPE ,77$ ;:TYPE ASCIZ STRING  
BR 76$ ;:GET OVER THE ASCIZ  
::77$: .ASCIZ <15><12>/ THE PROGRAM TO LOOP BACK TO FOLLOWED BY A CARRIAGE RETURN /<15  
76$:  
RDOCT  
MOV (SP)+,@#LPERR ;GET LPERR  
MOV @#LPADR,-(SP)  
RTI
```


11564
11565
11566
11567
11568
11569
11570
11571
11572
11573
11574
11575
11576
11577
11578
11579
11580
11581
11582
11583
11584
11585
11586
11587
11588
11589
11590
11591
11592
11593
11594
11595
11596
11597
11598
11599
11600
11601
11602
11603
11604
11605
11606
11607
11608
11609
11610
11611
11612
11613
11614
11615
11616
11617
11618

043436
043436 010046
043440 010146
043442 010246
043444 012700 002272
043450 012701 002354
043454 012702 000022
043460 013021
043462 005302
043464 001375
043466 012602
043470 012601
043472 012600
043474 000207

;;*THIS SAVES THE CONTENTS OF ALL HARDWARE REGISTERS
;;*IN MEMORY LOCATIONS TAGED FROM 'WC' TO 'EC2'

;;*THIS IS DONE SO THAT COMPARES ARE DONE WITH SAVED LOCATIONS
;;*AND NOT THE REGISTERS THEMSELVES. THIS WILL MAKE
;;*ERROR PRINTOUTS FOR GOOD AND BAD DATA ALWAYS DIFFRENT

```
PUTREG:  MOV    R0,-(SP)      ;;PUSH R0 ON STACK
        MOV    R1,-(SP)      ;;PUSH R1 ON STACK
        MOV    R2,-(SP)      ;;PUSH R2 ON STACK
        MOV    #RHWC,R0      ;;STARTING ADDRESS OF REG
        MOV    #WC,R1        ;;STARTING ADDRESS OF WERE SAVED
        MOV    #RHCC-RHWC+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
```

```
;;*THIS IS A DATA COMMAND SETUP SUBROUTINE
;;*THE CALL IS
;;*   JSR    R0,@#RUN
;;*   C      ;CYLINDER
;;*   .BYTE S ;SECTOR
;;*   .BYTE T ;TRACK
;;*   -W     ;WORD COUNT
;;*   B      ;BUS ADDRESS
;;*   BAI    ;BUS ADDRESS INHIBIT
;;*   FMT22!ECI!HCI ;FMT22=1 =16 BIT WORDS
;;*           ;ECI = ECC CORRECTION INHIBIT
;;*           ;HCI = HEADER COMPARE INHIBIT
;;*   COM    ;COMMAND ADDRESS
RUN:    MOV    (R0)+,@RHCA    ;CYLINDER
        MOV    (R0)+,@RHDST  ;DESIRED SECTOR/TRACK
        MOV    (R0)+,@RHWC   ;WORD COUNT
        MOV    (R0)+,@RHBA   ;BUS ADDRESS
        MOV    @#UNIT,-(SP)  ;GET UNIT NO
        BIS    (R0)+,(SP)    ;SET BUS ADDRESS INHIBIT
        MOV    (SP)+,@RHCS2  ;UNIT NO AND BAI TO RHCS2
        MOV    (R0)+,@RHOF   ;FORMAT, ECC INHIBIT, HEADER
        ;COMPARE, IF THERE
        MOV    @(R0)+,@RHCS1 ;COMMAND IN RHCS1
        RTS    R0           ;RETURN TO MAIN PROGRAM
```

```
11619
11620
11621      ;*THIS IS A SUBROUTINE TO COMPARE TWO BLOCKS IN MEMORY
11622
11623      ;*R1 HAS GOOD DATA BUFFER ADDRESS
11624      ;*R2 HAS TEST DATA BUFFER ADDRESS
11625      ;*R5 HAS ADDRESS OF RETURN ON ERROR
11626      ;*R3 HAS NUMBER OF WORDS TO BE COMPARED
11627      ;*R4 HAS ONE MORE THAN NUMBER OF WORDS TO BE COMPARED
11628
11629      ;*CALL IS:
11630      ;*      JSR      R0,@#COMPAR
11631      ;*      G          ;ADDRESS OF GOOD DATA
11632      ;*      T          ;ADDRESS OF TEST DATA
11633      ;*      N          ;NUMBER OF WORDS TO BE COMPARED
11634      ;*      RE         ;RETURN ON ERROR
11635      ;*      RG         ;RETURN ON NO ERROR
11636
11637
11638
11639      043542      COMPAR:
11640      043542      010146      MOV      R1,-(SP)          ;;PUSH R1 ON STACK
11641      043544      010246      MOV      R2,-(SP)          ;;PUSH R2 ON STACK
11642      043546      010346      MOV      R3,-(SP)          ;;PUSH R3 ON STACK
11643      043550      010446      MOV      R4,-(SP)          ;;PUSH R4 ON STACK
11644      043552      010546      MOV      R5,-(SP)          ;;PUSH R5 ON STACK
11645      043554      012001      MOV      (R0)+,R1          ;ADDRESS OF GOOD DATA BUFFER
11646      043556      012002      MOV      (R0)+,R2          ;ADDRESS OF TEST DATA BUFFER
11647      043560      012003      MOV      (R0)+,R3          ;NO OF WORDS TO BE COMPARED
11648      043562      012005      MOV      (R0)+,R5          ;RETURN ON ERROR
11649      043564      011000      MOV      (R0),R0          ;RETURN ON NO ERROR
11650      043566      010304      MOV      R3,R4            ;NO OF WORDS TO BE COMPARED
11651      043570      005204      INC      R4
11652      043572      010437      004602      1$:      MOV      R4,@#ERWORD      ;FOR ERROR WORD NO
11653      043576      022122      CMP      (R1)+,(R2)+      ;COMPARE GOOD WITH TEST DATA
11654      043600      001417      BEQ      2$              ;BRANCH IF GOOD
11655
11656      043602      014137      001124      MOV      -(R1),@#SGDDAT   ;GOOD DATA
11657      043606      014237      001126      MOV      -(R2),@#SBDDAT   ;BAD DATA
11658      043612      160337      004602      SUB      R3,@#ERWORD      ;ERROR WORD NO.
11659      043616      004715      JSR      PC,@R5           ;RETURN TO PRINT ERROR
11660      043620      022122      CMP      (R1)+,(R2)+      ;UNDO -(R1) AND -(R2) FOR ERRORS
11661      043622      017746      135312      MOV      @SWR,-(SP)        ;GET SWITCH SETTING
11662      043626      042716      177177      BIC      #^C600,(SP)       ;KEEP ONLY SWITCH 7 AND 8
11663      043632      022726      000200      CMP      #SW07,(SP)+      ;IS 7 SET AND 8 RESET
11664      043636      001402      BEQ      3$              ;BRANCH OUT IF YES
11665      043640      005303      2$:      DEC      R3                ;COUNT
11666      043642      001353      BNE      1$              ;BRANCH IF ALL NOT DEVICE
11667      043644      3$:
11668      043644      012605      MOV      (SP)+,R5          ;;POP STACK INTO R5
11669      043646      012604      MOV      (SP)+,R4          ;;POP STACK INTO R4
11670      043650      012603      MOV      (SP)+,R3          ;;POP STACK INTO R3
11671      043652      012602      MOV      (SP)+,R2          ;;POP STACK INTO R2
11672      043654      012601      MOV      (SP)+,R1          ;;POP STACK INTO R1
11673      043656      000200      RTS      R0              ;RETURN TO MAIN PROGRAM
```

```
11674
11675
11676
11677
11678
11679 043660
11680 043660 104401 043666
11681 043664 000425
11682
11683 043740
11684 043740 013746 002300
11685 043744 104402
11686 043746 104401 043754
11687 043752 000425
11688
11689 044026
11690 044026 004737 045734
11691 044032 104412
11692 044034 012700 002270
11693 044040 012701 000026
11694 044044 012737 044644 000004
11695 044052 021637 002300
11696 044056 001407
11697 044060 005776 000000
11698 044064 163716 002300
11699 044070 061620
11700 044072 005301
11701 044074 001375
11702 044076
11703 044076 104401 044104
11704 044102 000417
11705
11706 044142
11707 044142 013746 002266
11708 044146 104402
11709 044150 104401 044156
11710 044154 000437
11711
11712 044254
11713 044254 104412
11714 044256 012637 002266
11715 044262 104401 044270
11716 044266 000416
11717
11718 044324
11719 044324 013746 002300
11720 044330 104402
11721 044332 104401 044340
11722 044336 000416
11723
11724 044374
11725 044374 013746 002266
11726 044400 104402
11727 044402 104401 044410
11728 044406 000417
11729

;* THIS ROUTINE WILL ALLOW THE CHANGE OF THE BASE
;* ADDRESS FROM 176700 TO ANY TYPED VALUE

BASECH:
TYPE ,65$ ;;TYPE ASCIZ STRING
BR 64$ ;;GET OVER THE ASCIZ
;;65$: .ASCIZ <15><12>/PRESENT BASE ADDRESS OF REGISTERS IS /
64$:
MOV @#RHCS1,-(SP) ;GET READY TO TYPE OLD BASE
TYPOC
TYPE ,67$ ;;TYPE ASCIZ STRING
BR 66$ ;;GET OVER THE ASCIZ
;;67$: .ASCIZ <15><12>/TYPE NEW BASE ADDRESS FOLLOWED BY 'CR' /
66$:
JSR PC,@#STKINT ;INITIALIZE THE TTY KEYBOARD
RDOCT
MOV #RHDB,RO ;GET STARTING ADDRESS OF REGISTERS
MOV #22.,R1 ;NUMBER OF REGISTERS
MOV #ADTIMO,@#4 ;SET TRAP CATCHER TO CHECK THIS ADDRESS
CMP @SP,@#RHCS1 ;NEW ADDRESS?
BEQ 1$ ;NO, OLD ONE JUST RETYPED.
TST @0(SP) ;OK, SO ACCESS THIS NEW ADDRESS
SUB @#RHCS1,@SP ;GET THE ADDRESS OFFSET
2$: ADD @SP,(R0)+ ;AND PLUG IT IN.
DEC R1 ;ONE LESS REGISTER TO GO
BNE 2$ ;BUT WE'RE NOT DONE YET.
1$:
TYPE ,69$ ;;TYPE ASCIZ STRING
BR 68$ ;;GET OVER THE ASCIZ
;;69$: .ASCIZ <15><12>/PRESENT VECTOR ADDRESS IS /
68$:
MOV @#RPVEC,-(SP) ;GET READY TO TYPE OLD VECTOR ADDRESS
TYPOC
TYPE ,71$ ;;TYPE ASCIZ STRING
BR 70$ ;;GET OVER THE ASCIZ
;;71$: .ASCIZ <15><12>/TYPE NEW VECTOR ADDRESS OR RETYPE OLD ONE FOLLOWED BY "CR" /
70$:
RDOCT
MOV (SP)+,@#RPVEC ;SETUP VECTOR ADDRESS
TYPE ,73$ ;;TYPE ASCIZ STRING
BR 72$ ;;GET OVER THE ASCIZ
;;73$: .ASCIZ <15><12>/NEW BASE WILL REMAIN - /
72$:
MOV @#RHCS1,-(SP)
TYPOC
TYPE ,75$ ;;TYPE ASCIZ STRING
BR 74$ ;;GET OVER THE ASCIZ
;;75$: .ASCIZ <15><12>/NEW VECTOR WILL REMAIN - /
74$:
MOV @#RPVEC,-(SP)
TYPOC
TYPE ,77$ ;;TYPE ASCIZ STRING
BR 76$ ;;GET OVER THE ASCIZ
;;77$: .ASCIZ <15><12>/UNTIL PROGRAM IS RELOADED./
```

```
11730 044446          76$:
11731 044446 104401 044454      TYPE      ,79$      ;;TYPE ASCIZ STRING
11732 044452 000402          BR        78$      ;;GET OVER THE ASCIZ
11733          ;;79$: .ASCIZ <15><12>/ /
11734 044460          78$:
11735 044460 104401 044466      TYPE      ,81$      ;;TYPE ASCIZ STRING
11736 044464 000424          BR        80$      ;;GET OVER THE ASCIZ
11737          ;;81$: .ASCIZ <15><12>/UNLESS HALTED AND MANUALLY RESTARTED,/
11738 044536          80$:
11739 044536 104401 044544      TYPE      ,85$      ;;TYPE ASCIZ STRING
11740 044542 000426          BR        82$      ;;GET OVER THE ASCIZ
11741          ;;83$: .ASCIZ <15><12>/PROGRAM WILL AUTOMATICALLY RESTART FROM /
11742 044620          82$:
11743 044620 012746 000200      MOV      #RA,-(SP)
11744 044624 104402          TYPOC
11745 044626 104401 044634      TYPE      ,85$      ;;TYPE ASCIZ STRING
11746 044632 000402          BR        84$      ;;GET OVER THE ASCIZ
11747          ;;85$: .ASCIZ <15><12>/ /
11748 044640          84$:
11749 044640 000137 005012      JMP      @#BEGIN      ;RESTART, TO RUN ALL DRIVES
11750 044644          ADTIMO:
11751 044644 104401 044652      TYPE      ,65$      ;;TYPE ASCIZ STRING
11752 044650 000426          BR        64$      ;;GET OVER THE ASCIZ
11753          ;;65$: .ASCIZ <15><12><377>/THE SELECTED ADDRESS DID NOT RESPOND! /
11754 044726          64$:
11755 044726 022626          CMP      (SP)+,(SP)+ ;RESTORE THE STACK
11756 044730 000137 043660      JMP      @#BASECH    ;AND DO THE WHOLE THING AGAIN
11757
11758
```

CZRJID0, RPO4/5/6 FCTNL CTLR1
CZRJID.P11 28-MAR-79 09:03

MACY11 30A(1052) 25-MAY-79 10:30 PAGE 244
JAM CURRENT CYLINDER ROUTINE

K 3

SEQ 0243

11759
11760
11761 044734
11762 044734 104401 044742
11763 044740 000411
11764
11765 044764
11766 044764 104402
11767 044766 012777 044734 135272
11768 044774 000000
11769
11770
11771
11772
11773

```
;;*****  
RPVECT:  
    TYPE      ,65$      ;;TYPE ASCIZ STRING  
    BR        64$      ;;GET OVER THE ASCIZ  
;;65$: .ASCIZ /TRAPED FROM PC = /  
64$:  
    TYPOC      ;;TYPE FROM PC  
    MOV        #RPVECT, @RPVEC  ;;RESTORE TRAP RPO4 VECTOR  
    HALT      ;;CHANGE TO CONTINJE  
;;*****
```

```
11774 .SBTTL SCOPE HANDLER ROUTINE
11775
11776 ;*****
11777 ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
11778 ;*AND LOAD THE TEST NUMBER($STNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
11779 ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
11780 ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
11781 ;*SW14=1 LOOP ON TEST
11782 ;*SW11=1 INHIBIT ITERATIONS
11783 ;*SW09=1 LOOP ON ERROR
11784 ;*SW08=1 LOOP ON TEST IN SWR<7:0>
11785 ;*CALL
11786 ;* SCOPE ;:SCOPE=10T
11787
11788 $SCOPE:
11789 044776 104407 CKSWR ;:TEST FOR CHANGE IN SOFT-SWR
11790 045000 032777 040000 134132 1$: BIT #BIT14,@SWR ;:LOOP ON PRESENT TEST?
11791 045006 001111 BNE $OVER ;:YES IF SW14=1
11792 ;*****START OF CODE FOR THE XOR TESTER*****
11793 045010 000416 $XTSTR: BR 6$ ;:IF RUNNING ON THE "XOR" TESTER CHANGE
11794 ;:THIS INSTRUCTION TO A "NOP" (NOP=240)
11795 045012 013746 000004 MOV @#ERRVEC,-(SP) ;:SAVE THE CONTENTS OF THE ERROR VECTOR
11796 045016 012737 045036 000004 MOV #5,@#ERRVEC ;:SET FOR TIMEOUT
11797 045024 005737 177060 TST @#177060 ;:TIME OUT ON XOR?
11798 045030 012637 000004 MOV (SP)+,@#ERRVEC ;:RESTORE THE ERROR VECTOR
11799 045034 000463 BR $SVLAD ;:GO TO THE NEXT TEST
11800 045036 022626 5$: CMP (SP)+,(SP)+ ;:CLEAR THE STACK AFTER A TIME OUT
11801 045040 012637 000004 MOV (SP)+,@#ERRVEC ;:RESTORE THE ERROR VECTOR
11802 045044 000423 BR 7$ ;:LOOP ON THE PRESENT TEST
11803 045046 6$: ;*****END OF CODE FOR THE XOR TESTER*****
11804 045046 032777 000400 134064 BIT #BIT08,@SWR ;:LOOP ON SPEC. TEST?
11805 045054 001404 BEQ 2$ ;:BR IF NO
11806 045056 127737 134056 001102 CMPB @SWR,$STNM ;:ON THE RIGHT TEST? SWR<7:0>
11807 045064 001462 BEQ $OVER ;:BR IF YES
11808 045066 105737 001103 2$. TSTB $ERFLG ;:HAS AN ERROR OCCURRED?
11809 045072 001421 BEQ 3$ ;:BR IF NO
11810 045074 123737 001115 001103 CMPB $ERMAX,$ERFLG ;:MAX. ERRORS FOR THIS TEST OCCURRED?
11811 045102 101015 BHI 3$ ;:BR IF NO
11812 045104 032777 001000 134026 BIT #BIT09,@SWR ;:LOOP ON ERROR?
11813 045112 001404 BEQ 4$ ;:BR IF NO
11814 045114 013737 001110 001106 7$: MOV $LPERR,$LPADR ;:SET LOOP ADDRESS TO LAST SCOPE
11815 045122 000443 BR $OVER
11816 045124 105037 001103 4$: CLRB $ERFLG ;:ZERO THE ERROR FLAG
11817 045130 005037 001212 CLR $TIMES ;:CLEAR THE NUMBER OF ITERATIONS TO MAKE
11818 045134 000415 BR 1$ ;:ESCAPE TO THE NEXT TEST
11819 045136 032777 004000 133774 3$: BIT #BIT11,@SWR ;:INHIBIT ITERATIONS?
11820 045144 001011 BNE 1$ ;:BR IF YES
11821 045146 005737 001100 TST $PASS ;:IF FIRST PASS OF PROGRAM
11822 045152 001406 BEQ 1$ ;: INHIBIT ITERATIONS
11823 045154 005237 001104 INC $ICNT ;:INCREMENT ITERATION COUNT
11824 045160 023737 001212 001104 CMP $TIMES,$ICNT ;:CHECK THE NUMBER OF ITERATIONS MADE
11825 045166 002021 BGE $OVER ;:BR IF MORE ITERATION REQUIRED
11826 045170 012737 000001 001104 1$: MOV #1,$ICNT ;:REINITIALIZE THE ITERATION COUNTER
11827 045176 013737 045246 001212 MOV $MXCNT,$TIMES ;:SET NUMBER OF ITERATIONS TO DO
11828 045204 105237 001102 $SVLAD: INCB $STNM ;:COUNT TEST NUMBERS
11829 045210 011637 001106 MOV (SP),$LPADR ;:SAVE SCOPE LOOP ADDRESS
```

CZRJ1D0, RP04/5/6 FCTNL CTLR1
CZRJ1D.P11 28-MAR-79 09:03

MACY11 30A(1052) 25-MAY-79 10:30 M 3 PAGE 246
SCOPE HANDLER ROUTINE

SEQ 0245

11830	045214	011637	001110			MOV	(SP), \$LPERR	:: SAVE ERROR LOOP ADDRESS
11831	045220	005037	001214			CLR	\$ESCAPE	:: CLEAR THE ESCAPE FROM ERROR ADDRESS
11832	045224	112737	000001	001115		MOVB	#1, \$ERMAX	:: ONLY ALLOW ONE(1) ERROR ON NEXT TEST
11833	045232	013777	001102	133702	\$OVER:	MOV	\$ISTNM, @DISPLAY	:: DISPLAY TEST NUMBER
11834	045240	013716	001106			MOV	\$LPADR, (SP)	:: FUDGE RETURN ADDRESS
11835	045244	000002				RTI		:: FIXES PS
11836	045246	G00004			\$MXCNT:	4		:: MAX. NUMBER OF ITERATIONS

.SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
*REPLACED WITH SPACES.

*CALL:
* MOV NUM,-(SP) ;;PUT THE BINARY NUMBER ON THE STACK
* TYPDS ;;GO TO THE ROUTINE

\$TYPDS:
MOV R0,-(SP) ;;PUSH R0 ON STACK
MOV R1,-(SP) ;;PUSH R1 ON STACK
MOV R2,-(SP) ;;PUSH R2 ON STACK
MOV R3,-(SP) ;;PUSH R3 ON STACK
MOV R5,-(SP) ;;PUSH R5 ON STACK
MOV #20200,-(SP) ;;SET BLANK SWITCH AND SIGN
MOV 20(SP),R5 ;;GET THE INPUT NUMBER
BPL 1\$;;BR IF INPUT IS POS.
NEG R5 ;;MAKE THE BINARY NUMBER POS.
MOVB #'-,1(SP) ;;MAKE THE ASCII NUMBER NEG.
1\$: CLR R0 ;;ZERO THE CONSTANTS INDEX
MOV # \$DBLK,R3 ;;SETUP THE OUTPUT POINTER
MOVB #' ,(R3)+ ;;SET THE FIRST CHARACTER TO A BLANK
2\$: CLR R2 ;;CLEAR THE BCD NUMBER
MOV \$DTBL(R0),R1 ;;GET THE CONSTANT
3\$: SUB R1,R5 ;;FORM THIS BCD DIGIT
BLT 4\$;;BR IF DONE
INC R2 ;;INCREASE THE BCD DIGIT BY 1
BR 3\$
4\$: ADD R1,R5 ;;ADD BACK THE CONSTANT
TST R2 ;;CHECK IF BCD DIGIT=0
BNE 5\$;;FALL THROUGH IF 0
TSTB (SP) ;;STILL DOING LEADING 0'S?
BMI 7\$;;BR IF YES
5\$: ASLB (SP) ;;MSD"
BCC 6\$;;BR IF NO
MOVB 1(SP),-1(R3) ;;YES--SET THE SIGN
6\$: BIS #'0,R2 ;;MAKE THE BCD DIGIT ASCII
7\$: BIS #' ,R2 ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
MOVB R2,(R3)+ ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
TST (R0)+ ;;JUST INCREMENTING
CMP R0,#10 ;;CHECK THE TABLE INDEX
BLT 2\$;;GO DO THE NEXT DIGIT
BGT 8\$;;GO TO EXIT
MOV R5,R2 ;;GET THE LSD
BR 6\$;;GO CHANGE TO ASCII
8\$: TSTB (SP)+ ;;WAS THE LSD THE FIRST NON-ZERO?
BPL 9\$;;BR IF NO
MOVB -1(SP),-2(R3) ;;YES--SET THE SIGN FOR TYPING
9\$: CLRB (R3) ;;SET THE TERMINATOR
MOV (SP)+,R5 ;;POP STACK INTO R5
MOV (SP)+,R3 ;;POP STACK INTO R3
MOV (SP)+,R2 ;;POP STACK INTO R2

11837
11838
11839
11840
11841
11842
11843
11844
11845
11846
11847
11848
11849 045250
11850 045250 010046
11851 045252 010146
11852 045254 010246
11853 045256 010346
11854 045260 010546
11855 045262 012746 020200
11856 045266 016605 000020
11857 045272 100004
11858 045274 005405
11859 045276 112766 000055 000001
11860 045304 005000 1\$:
11861 045306 012703 045464
11862 045312 112723 000040
11863 045316 005002 2\$:
11864 045320 016001 045454
11865 045324 160105 3\$:
11866 045326 002402
11867 045330 005202
11868 045332 000774
11869 045334 060105 4\$:
11870 045336 005702
11871 045340 001002
11872 045342 105716
11873 045344 100407
11874 045346 106316 5\$:
11875 045350 103003
11876 045352 116663 000001 177777
11877 045360 052702 000060 6\$:
11878 045364 052702 000040 7\$:
11879 045370 110223
11880 045372 005720
11881 045374 020027 000010
11882 045400 002746
11883 045402 003002
11884 045404 010502
11885 045406 000764
11886 045410 105726 8\$:
11887 045412 100003
11888 045414 116663 177777 177776
11889 045422 105013 9\$:
11890 045424 012605
11891 045426 012603
11892 045430 012602


```
11893 045432 012601          MOV      (SP)+,R1          ;;POP STACK INTO R1
11894 045434 012600          MOV      (SP)+,R0          ;;POP STACK INTO R0
11895 045436 104401 045464    TYPE     ,SDBLK           ;;NOW TYPE THE NUMBER
11896 045442 016666 000002 000004 MOV      2(SP),4(SP)      ;;ADJUST THE STACK
11897 045450 012616          MOV      (SP)+,(SP)
11898 045452 000002          RTI                          ;;RETURN TO USER
11899 045454 023420          $DTBL: 10000.
11900 045456 001750          1000.
11901 045460 000144          100.
11902 045462 000012          10.
11903 045464 000004          $DBLK: .BLKW 4
```

```
11904 .SBTTL TYPE ROUTINE
11905
11906 *****
11907 *ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
11908 *THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
11909 *NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
11910 *NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
11911 *NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
11912
11913 *CALL:
11914 *1) USING A TRAP INSTRUCTION
11915 * TYPE ,MESADR ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
11916
11917 *OR
11918 * TYPE
11919 * MESADR
11920
11921 045474 105737 001157 $TYPE: TSTB $TIFLG ;;IS THERE A TERMINAL?
11922 045500 100002 BPL 1$ ;;BR IF YES
11923 045502 000000 HALT ;;HALT HERE IF NO TERMINAL
11924 045504 000407 BR 3$ ;;LEAVE
11925 045506 010046 1$: MOV RO,-(SP) ;;SAVE RO
11926 045510 017600 000002 MOV @2(SP),RO ;;GET ADDRESS OF ASCIZ STRING
11927 045514 112046 2$: MOVB (RO)+,-(SP) ;;PUSH CHARACTER TO BE TYPED ONTO STACK
11928 045516 001005 BNE 4$ ;;BR IF IT ISN'T THE TERMINATOR
11929 045520 005726 TST (SP)+ ;;IF TERMINATOR POP IT OFF THE STACK
11930 045522 012600 60$: MOV (SP)+,RO ;;RESTORE RO
11931 045524 062716 000002 3$: ADD #2,(SP) ;;ADJUST RETURN PC
11932 045530 000002 RTI ;;RETURN
11933 045532 122716 000011 4$: CMPB #HT,(SP) ;;BRANCH IF <HT>
11934 045536 001430 BEQ 8$
11935 045540 122716 000200 CMPB #CRLF,(SP) ;;BRANCH IF NOT <CRLF>
11936 045544 001006 BNE 5$
11937 045546 005726 TST (SP)+ ;;POP <CR><LF> EQUIV
11938 045550 104401 TYPE ;;TYPE A CR AND LF
11939 045552 001223 $CRLF
11940 045554 105037 045710 CLRB $CHARCNT ;;CLEAR CHARACTER COUNT
11941 045560 000755 BR 2$ ;;GET NEXT CHARACTER
11942 045562 004737 045644 5$: JSR PC,$TYPEC ;;GO TYPE THIS CHARACTER
11943 045566 123726 001156 6$: CMPB $FILLC,(SP)+ ;;IS IT TIME FOR FILLER CHARS.?
11944 045572 001350 BNE 2$ ;;IF NO GO GET NEXT CHAR.
11945 045574 013746 001154 MOV $NULL,-(SP) ;;GET # OF FILLER CHARS. NEEDED
11946 ;;AND THE NULL CHAR.
11947 045600 105366 000001 7$: DECB 1(SP) ;;DOES A NULL NEED TO BE TYPED?
11948 045604 002770 BLT 6$ ;;BR IF NO--GO POP THE NULL OFF OF STACK
11949 045606 004737 045644 JSR PC,$TYPEC ;;GO TYPE A NULL
11950 045612 105337 045710 DECB $CHARCNT ;;DO NOT COUNT AS A COUNT
11951 045616 000770 BR 7$ ;;LOOP
11952
11953 ;HORIZONTAL TAB PROCESSOR
11954
11955 045620 112716 000040 8$: MOVB #' ,(SP) ;;REPLACE TAB WITH SPACE
11956 045624 004737 045644 9$: JSR PC,$TYPEC ;;TYPE A SPACE
11957 045630 132737 000007 045710 BITB #7,$CHARCNT ;;BRANCH IF NOT AT
11958 045636 001372 BNE 9$ ;;TAB STOP
11959 045640 005726 TST (SP)+ ;;POP SPACE OFF STACK
```

```

11960 045642 000724
11961 045644 105777 133300
11962 045650 100375
11963 045652 116677 000002 133272
11964 045660 122766 000015 000002
11965 045666 001003
11966 045670 105037 045710
11967 045674 000406
11968 045676 122766 000012 000002 1$:
11969 045704 001402
11970 045706 105227
11971 045710 000000
11972 045712 000207
11973

          BR      2$
          $TYPEC: TSTB @STPS
          BPL     $TYPEC
          MOVB    2(SP),@STPB
          CMPB   #CR,2(SP)
          BNE    1$
          CLRB   $CHARCNT
          BR     $TYPEX
          $CHARCNT: .WORD 0
          $TYPEX: RTS   PC

          ;;GET NEXT CHARACTER
          ;;WAIT UNTIL PRINTER IS READY
          ;;LOAD CHAR TO BE TYPED INTO DATA REG.
          ;;IS CHARACTER A CARRIAGE RETURN?
          ;;BRANCH IF NO
          ;;YES--CLEAR CHARACTER COUNT
          ;;EXIT
          ;;IS CHARACTER A LINE FEED?
          ;;BRANCH IF YES
          ;;COUNT THE CHARACTER
          ;;CHARACTER COUNT STORAGE
  
```

```

11974 .SBTTL TTY INPUT ROUTINE
11975
11976 ::*****
11977 .ENABL LSB
11978 045714 000000 $TKCNT: .WORD 0 ;;NUMBER OF ITEMS IN QUEUE
11979 045716 000000 $TKQIN: .WORD 0 ;;INPUT POINTER
11980 045720 000000 $TKQOUT: .WORD 0 ;;OUTPUT POINTER
11981 045722 000011 $TKQSRT: .BLKB 9. ;;TTY KEYBOARD QUEUE
11982 045733 $TKQEND=.
11983 045734 .EVEN
11984
11985 ;*TK INITIALIZE ROUTINE
11986 ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
11987 ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
11988
11989 ;*CALL:
11990 ;* JSR PC,$TKINT
11991 ;* RETURN
11992
11993 045734 005037 045714 $TKINT: CLR $TKCNT ;;CLEAR COUNT OF ITEMS IN QUEUE
11994 045740 012737 045722 045716 MOV #TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
11995 045746 013737 045716 045720 MOV $TKQIN,$TKQOUT ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
11996 045754 012737 046004 000060 MOV #TKSRV,@TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
11997 045762 012737 000200 000062 MOV #200,@TKVEC+2 ;;'BR' LEVEL 4
11998 045770 005777 133152 TST @TKB ;;CLEAR DONE FLAG
11999 045774 012777 000100 133142 MOV #100,@TKS ;;ENABLE TTY KEYBOARD INTERRUPT
12000 046002 000207 RTS PC ;;RETURN TO CALLER
12001
12002 ;*TK SERVICE ROUTINE
12003 ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
12004 ;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
12005 ;*IT IN THE QUEUE.
12006 ;*IF THE CHARACTER IS A "CONTROL-C" (^C) $TKINT IS CALLED AND
12007 ;*UPON RETURN EXIT IS MADE TO THE "CONTROL-C" RESTART ADDRESS (OPERSEL)
12008
12009 046004 117746 133136 $TKSRV: MOVB @TKB,-(SP) ;;PICKUP THE CHARACTER
12010 046010 042716 177600 BIC #^C177,(SP) ;;STRIP THE JUNK
12011 046014 021627 000003 CMP (SP),#3 ;;IS IT A CONTROL C?
12012 046020 001007 BNE 1$ ;;BRANCH IF NO
12013 046022 104401 046773 TYPE ,SCNTLC ;;TYPE A CONTROL-C (^C)
12014 046026 004737 045734 JSR PC,$TKINT ;;INIT THE KEYBOARD
12015 046032 005726 TST (SP)+ ;;CLEAN UP STACK
12016 046034 000137 042650 JMP OPERSEL ;;CONTROL C RESTART
12017 046040 021627 000007 1$: CMP (SP),#7 ;;IS IT A CONTROL G?
12018 046044 001004 BNE 2$ ;;BRANCH IF NO
12019 046046 022737 000176 001140 CMP #SWREG,SWR ;;IS SOFT-SWR SELECTED?
12020 046054 001500 BEQ 6$ ;;GO TO SWR CHANGE
12021
12022 046056 2$:
12023 046056 022737 000011 045714 CMP #9,$TKCNT ;;IS THE QUEUE FULL?
12024 046064 001004 BNE 3$ ;;BRANCH IF NO
12025 046066 104401 001216 TYPE ,SBELL ;;RING THE TTY BELL
12026 046072 005726 TST (SP)+ ;;CLEAN CHARACTER OFF OF STACK
12027 046074 000451 BR 5$ ;;EXIT
12028 046076 021627 000023 3$: CMP (SP),#23 ;;IS IT A CONTROL-S?
12029 046102 001021 BNE 32$ ;;BRANCH IF NO

```

```

12030 046104 005077 133034      CLR      @STKS      ;;DISABLE TTY KEYBOARD INTERRUPTS
12031 046110 005726              TST      (SP)+      ;;CLEAN CHAR OFF STACK
12032 046112 105777 133026      31$:    TSTB     @STKS      ;;WAIT FOR A CHAR
12033 046116 100375              BPL      31$        ;;LOOP UNTIL ITS THERE
12034 046120 117746 133022      MOVB     @STKB,-(SP) ;;GET THE CHARACTER
12035 046124 042716 177600      BIC     #'^C177,(SP) ;;MAKE IT 7-BIT ASCII
12036 046130 022627 000021      CMP     (SP)+,#21   ;;IS IT A CONTROL-Q?
12037 046134 001366              BNE     31$        ;;BRANCH IF NO
12038 046136 012777 000100 133000      MOV     #100,@STKS  ;;REENABLE TTY KEYBOARD INTERRUPTS
12039 046144 000002              RTI                      ;;RETURN
12040 046146 005237 045714      32$:    INC     $TKCNT   ;;COUNT THIS CHARACTER
12041 046152 021627 000140      CMP     (SP),#140   ;;IS IT UPPER CASE?
12042 046156 002405              BLT     4$         ;;BRANCH IF YES
12043 046160 021627 000175      CMP     (SP),#175   ;;IS IT A SPECIAL CHAR?
12044 046164 003002              BGT     4$         ;;BRANCH IF YES
12045 046166 042716 000040      BIC     #40,(SP)    ;;MAKE IT UPPER CASE
12046 046172 112677 177520      4$:    MOVB     (SP)+,@STKQIN ;;AND PUT IT IN QUEUE
12047 046176 005237 045716      INC     $TKQIN      ;;UPDATE THE POINTER
12048 046202 023727 045716 045733      CMP     $TKQIN,#$STKQEND ;;GO OFF THE END?
12049 046210 001003              BNE     5$         ;;BRANCH IF NO
12050 046212 012737 045722 045716      MOV     #$STKQSRST,$TKQIN ;;RESET THE POINTER
12051 046220 000002      5$:    RTI                      ;;RETURN

```

```

12052
12053      ;;*****
12054      ;;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
12055      ;;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
12056      ;;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
12057      ;;*CALL WHEN OPERATING IN TTY INTERRUPT MODE.

```

```

12058 046222 022737 000176 001140 $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED
12059 046230 001124              BNE     15$        ;;EXIT IF NOT
12060 046232 105777 132706      TSTB     @STKS      ;;IS A CHAR WAITING?
12061 046236 100121              BPL     15$        ;;IF NOT, EXIT
12062 046240 117746 132702      MOVB     @STKB,-(SP) ;;YES
12063 046244 042716 177600      BIC     #'^C177,(SP) ;;MAKE IT 7-BIT ASCII
12064 046250 021627 000007      CMP     (SP),#7     ;;IS IT A CONTROL-G?
12065 046254 001300              BNE     2$         ;;IF NOT, PUT IT IN THE TTY QUEUE
12066
12067
12068
12069
12070
12071
12072
12073
12074
12075
12076
12077
12078

```

```

12068      ;;*****
12069      ;;*CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
12070      ;;*ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
12071      ;;*CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.

```

```

12072 046256 123727 001134 000001 6$:    CMPB     $AUTOB,#1   ;;ARE WE RUNNING IN AUTO-MODE?
12073 046264 001674              BEQ     2$         ;;BRANCH IF YES
12074 046266 005726              TST     (SP)+      ;;CLEAR CONTROL-G OFF STACK
12075 046270 004737 045734      JSR     PC,$TKINT  ;;FLUSH THE TTY INPUT QUEUE
12076 046274 005077 132644      CLR     @STKS      ;;DISABLE TTY KEYBOARD INTERRUPTS
12077 046300 112737 000001 001135      MOVB     #1,$INTAG  ;;SET INTERRUPT MODE INDICATOR
12078
12079 046306 104401 047005      $GTSWR: TYPE    , $CNTLG  ;;ECHO THE CONTROL-G (^G)
12080 046312 104401 047012      TYPE    , $MSWR     ;;TYPE CURRENT CONTENTS
12081 046316 013746 000176      MOV     SWREG,-(SP) ;;SAVE SWREG FOR TYPEOUT
12082 046322 104402              TYPOC             ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
12083 046324 104401 047023      TYPE    , $MNEW     ;;PROMPT FOR NEW SWR
12084 046330 005046      19$:    CLR     -(SP)      ;;CLEAR COUNTER
12085 046332 005046      CLR     -(SP)      ;;THE NEW SWR

```

```

12086 046334 105777 132604      7$:  TSTB  @%TKS      ;;CHAR THERE?
12087 046340 100375                BPL  7$          ;;IF NOT TRY AGAIN
12088
12089 046342 117746 132600      MOVB  @%TKB,-(SP)  ;;PICK UP CHAR
12090 046346 042716 177600      BIC  #^C177,(SP)  ;;MAKE IT 7-BIT ASCII
12091
12092 046352 021627 000003      CMP   (SP),#3     ;;IS IT A CONTROL-C?
12093 046356 001015                BNE  9$          ;;BRANCH IF NOT
12094 046360 104401 046773      TYPE  ,%CNTLC     ;;YES, ECHO CONTROL-C (^C)
12095 046364 062706 000006      ADD  #6,SP        ;;CLEAN UP STACK
12096 046370 123727 001135 000001  CMPB  $INTAG,#1   ;;REENABLE TTY KEYBOARD INTERRUPTS?
12097 046376 001003                BNE  8$          ;;BRANCH IF NO
12098 046400 012777 000100 132536  MOV   #100,@%TKS  ;;ALLOW TTY KEYBOARD INTERRUPTS
12099 046406 000137 042650      8$:  JMP   OPERSEL    ;;CONTROL-C RESTART
12100
12101
12102 046412 021627 000025      9$:  CMP   (SP),#25  ;;IS IT A CONTROL-U?
12103 046416 001005                BNE  10$         ;;BRANCH IF NOT
12104 046420 104401 047000      TYPE  ,%CNTLU     ;;YES, ECHO CONTROL-U (^U)
12105 046424 062706 000006      20$: ADD  #6,SP        ;;IGNORE PREVIOUS INPUT
12106 046430 000737                BR   19$         ;;LET'S TRY IT AGAIN
12107
12108
12109 046432 021627 000015      10$: CMP   (SP),#15  ;;IS IT A <CR>?
12110 046436 001022                BNE  16$         ;;BRANCH IF NO
12111 046440 005766 000004      TST  4(SP)        ;;YES, IS IT THE FIRST CHAR?
12112 046444 001403                BEQ  11$         ;;BRANCH IF YES
12113 046446 016677 000002 132464  MOV   2(SP),@SWR  ;;SAVE NEW SWR
12114 046454 062706 000006      11$: ADD  #6,SP        ;;CLEAR UP STACK
12115 046460 104401 001223      14$: TYPE  ,%CRLF   ;;ECHO <CR> AND <LF>
12116 046464 123727 001135 000001  CMPB  $INTAG,#1   ;;RE-ENABLE TTY KBD INTERRUPTS?
12117 046472 001003                BNE  15$         ;;BRANCH IF NOT
12118 046474 012777 000100 132442  MOV   #100,@%TKS  ;;RE-ENABLE TTY KBD INTERRUPTS
12119 046502 000002                RTI  ;;RETURN
12120 046504 004737 045644      15$: JSR   PC,%TYPEC  ;;ECHO CHAR
12121 046510 021627 000060      16$: CMP   (SP),#60  ;;CHAR < 0?
12122 046514 002420                BLT  18$         ;;BRANCH IF YES
12123 046516 021627 000067      CMP   (SP),#67   ;;CHAR > 7?
12124 046522 003015                BGT  18$         ;;BRANCH IF YES
12125 046524 042726 000060      BIC  #60,(SP)+   ;;STRIP-OFF ASCII
12126 046530 005766 000002      TST  2(SP)        ;;IS THIS THE FIRST CHAR
12127 046534 001403                BEQ  17$         ;;BRANCH IF YES
12128 046536 006316                ASL  (SP)         ;;NO, SHIFT PRESENT
12129 046540 006316                ASL  (SP)         ;;CHAR OVER TO MAKE
12130 046542 006316                ASL  (SP)         ;;ROOM FOR NEW ONE.
12131 046544 005266 000002      17$: INC  2(SP)        ;;KEEP COUNT OF CHAR
12132 046550 056616 177776      BIS  -2(SP),(SP)  ;;SET IN NEW CHAR
12133 046554 000667                BR   7$          ;;GET THE NEXT ONE
12134 046556 104401 001222      18$: TYPE  ,%QUES   ;;TYPE ?<CR><LF>
12135 046562 000720                BR   20$         ;;SIMULATE CONTROL-U
12136 .DSABL  LSB
12137
12138
12139
12140
12141

```

```

*****
; *THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
; *CALL:

```

```

12142      ;*      RDCHR      ;;GET A CHARACTER FROM THE QUEUE
12143      ;*      RETURN HERE  ;;CHARACTER IS ON THE STACK
12144      ;*      ;;WITH PARITY BIT STRIPPED OFF
12145      ;
12146
12147 046564 011646      $RDCHR: MOV      (SP),-(SP)  ;;PUSH DOWN THE PC AND
12148 046566 016666 000004 000002  MOV      4(SP),2(SP)  ;;THE PS
12149 046574 005066 000004      CLR      4(SP)      ;;GET READY FOR A CHARACTER
12150 046600 005046      CLR      -(SP)     ;;PUT NEW PS ON STACK
12151 046602 012746 046610      MOV      #64$,-(SP)  ;;PUT NEW PC ON STACK
12152 046606 000002      RTI      ;;POP NEW PC AND PS
12153 046610
12154 046610 005737 045714      64$: 1$: TST      $TKCNT  ;;WAIT ON A CHARACTER
12155 046614 001775      BEQ      1$
12156 046616 005337 045714      DEC      $TKCNT  ;;DECREMENT THE COUNTER
12157 046622 117766 177072 000004  MOVB    @$TKQOUT,4(SP)  ;;GET ONE CHARACTER
12158 046630 005237 045720      INC      $TKQOUT  ;;UPDATE THE POINTER
12159 046634 023727 045720 045733  CMP     $TKQOUT,#$TKQEND ;;DID IT GO OFF OF THE END?
12160 046642 001003      BNE      2$      ;;BRANCH IF NO
12161 046644 012737 045722 045720  MOV     #$TKQSRT,$TKQOUT ;;RESET THE POINTER
12162 046652 000002      RTI      ;;RETURN
12163      ;*****
12164      ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
12165      ;*CALL:
12166      ;*      RDLIN      ;;INPUT A STRING FROM THE TTY
12167      ;*      RETURN HERE  ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
12168      ;*      ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
12169
12170 046654 010346      $RDLIN: MOV     R3,-(SP)  ;;SAVE R3
12171 046656 012703 046762      1$:  MOV     #$TTYIN,R3  ;;GET ADDRESS
12172 046662 022703 046773      2$:  CMP     #$TTYIN+9.,R3  ;;BUFFER FULL?
12173 046666 101405      BLOS    4$      ;;BR IF YES
12174 046670 104410      RDCHR   ;;GO READ ONE CHARACTER FROM THE TTY
12175 046672 112613      MOVB    (SP)+,(R3)  ;;GET CHARACTER
12176 046674 122713 000177      10$:  CMPB   #177,(R3)  ;;IS IT A RUBOUT
12177 046700 001003      BNE     3$      ;;SKIP IF NOT
12178 046702 104401 001222      4$:  TYPE   , $QUES  ;;TYPE A '?'
12179 046706 000763      BR      1$      ;;CLEAR THE BUFFER AND LOOP
12180 046710 111337 046760      3$:  MOVB   (R3),9$  ;;ECHO THE CHARACTER
12181 046714 104401 046760      TYPE   ,9$
12182 046720 122723 000015      CMPB   #15,(R3)+  ;;CHECK FOR RETURN
12183 046724 001356      BNE     2$      ;;LOOP IF NOT RETURN
12184 046726 105063 177777      CLRB   -1(R3)    ;;CLEAR RETURN (THE 15)
12185 046732 104401 001224      TYPE   , $LF     ;;TYPE A LINE FEED
12186 046736 012603      MOV     (SP)+,R3  ;;RESTORE R3
12187 046740 011646      MOV     (SP),-(SP) ;;ADJUST THE STACK AND PUT ADDRESS OF THE
12188 046742 016666 000004 000002  MOV     4(SP),2(SP)  ;;FIRST ASCII CHARACTER ON IT
12189 046750 012766 046762 000004  MOV     #$TTYIN,4(SP)
12190 046756 000002      RTI      ;;RETURN
12191 046760      000      9$:  .BYTE  0      ;;STORAGE FOR ASCII CHAR. TO TYPE
12192 046761      000      .BYTE  0      ;;TERMINATOR
12193 046762 000011      $TTYIN: .BLKB  9.  ;;RESERVE 9. BYTES FOR TTY INPUT
12194 046773      136 006503 000012  $CNTLC: .ASCIIZ /^C/<15><12>  ;;CONTROL 'C'
12195 047000 052536 005015 000      $CNTLU: .ASCIIZ /^U/<15><12>  ;;CONTROL 'U'
12196 047005      136 006507 000012  $CNTLG: .ASCIIZ /^G/<15><12>  ;;CONTROL 'G'
12197 047012 005015 053523 020122  $MSWR: .ASCIIZ <15><12>/SWR = /

```

CZRJ1D0, RP04/5/6 FCTNL CTLR1 MACY11 30A(1052) 25-MAY-79 10:30 ¹ 4 PAGE 255
CZRJID.P11 28-MAR-79 09:03 TTY INPUT ROUTINE

SEQ 0254

12198 047020 020075 000
12199 047023 040 047040 053505 \$MNEW: .ASCIZ / NEW = /
12200 047030 036440 000G40
12201

;FROM THE TTY


```
12202 .SBTTL READ AN OCTAL NUMBER FROM THE TTY
12203
12204
12205 ::*****
12206 :*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
12207 :*CHANGE IT TO BINARY.
12208 :*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
12209 :*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A '?' WILL BE TYPED
12210 :*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
12211 :*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
12212 :*CALL:
12213 :*      RDOCT          ::READ AN OCTAL NUMBER
12214 :*      RETURN HERE   ::LOW ORDER BITS ARE ON TOP OF THE STACK
12215 :*                   ::HIGH ORDER BITS ARE IN $HI OCT
12216 047034 011646 $RDOCT: MOV      (SP),-(SP)      ::PROVIDE SPACE FOR THE
12217 047036 016666 000004 000002 MOV      4(SP),2(SP)      ::INPUT NUMBER
12218 047044 010046 MOV      RO,-(SP)        ::PUSH RO ON STACK
12219 047046 010146 MOV      R1,-(SP)        ::PUSH R1 ON STACK
12220 047050 010246 MOV      R2,-(SP)        ::PUSH R2 ON STACK
12221 047052 104411 1$:      RDLIN          ::READ AN ASCII LINE
12222 047054 012600 MOV      (SP)+,RO        ::GET ADDRESS OF 1ST CHARACTER
12223 047056 010037 047162 MOV      RO,$$          ::AND SAVE IT
12224 047062 005001 CLR      R1              ::CLEAR DATA WORD
12225 047064 005002 CLR      R2
12226 047066 112046 2$:      MOVB      (RO)+,-(SP)      ::PICKUP THIS CHARACTER
12227 047070 001420 BEQ      3$              ::IF ZERO GET OUT
12228 047072 122716 000060 CMPB     #'0,(SP)        ::MAKE SURE THIS CHARACTER
12229 047076 003026 BGT      4$              ::IS AN OCTAL DIGIT
12230 047100 122716 000067 CMPB     #'7,(SP)
12231 047104 002423 BLT      4$
12232 047106 006301 ASL      R1              ::*2
12233 047110 006102 ROL      R2
12234 047112 006301 ASL      R1              ::*4
12235 047114 006102 ROL      R2
12236 047116 006301 ASL      R1              ::*8
12237 047120 006102 ROL      R2
12238 047122 042716 177770 BIC     #'^C7,(SP)      ::STRIP THE ASCII JUNK
12239 047126 062601 ADD     (SP)+,R1        ::ADD IN THIS DIGIT
12240 047130 000756 BR       2$              ::LOOP
12241 047132 005726 3$:      TST      (SP)+          ::CLEAN TERMINATOR FROM STACK
12242 047134 010166 000012 MOV      R1,12(SP)      ::SAVE THE RESULT
12243 047140 010237 047172 MOV      R2,$HI OCT
12244 047144 012602 MOV      (SP)+,R2      ::POP STACK INTO R2
12245 047146 012601 MOV      (SP)+,R1      ::POP STACK INTO R1
12246 047150 012600 MOV      (SP)+,RO      ::POP STACK INTO RO
12247 047152 000002 RTI          ::RETURN
12248 047154 005726 4$:      TST      (SP)+          ::CLEAN PARTIAL FROM STACK
12249 047156 105010 CLRB     (RO)          ::SET A TERMINATOR
12250 047160 104401 TYPE          ::TYPE UP THRU THE BAD CHAR.
12251 047162 000000 5$:      .WORD     0
12252 047164 104401 001222 TYPE     $QUES        ::'"?' 'CR' & 'LF'
12253 047170 000730 BR       1$              ::TRY AGAIN
12254 047172 000000 $HI OCT: .WORD     0      ::HIGH ORDER BITS GO HERE
```

```
12255 .SBTTL ERROR HANDLER ROUTINE
12256
12257 ;*****
12258 ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
12259 ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
12260 ;*AND GO TO $ERRTYP ON ERROR
12261 ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
12262 ;*SW15=1 HALT ON ERROR
12263 ;*SW13=1 INHIBIT ERROR TYPEOUTS
12264 ;*SW10=1 BELL ON ERROR
12265 ;*SW09=1 LOOP ON ERROR
12266 ;*CALL
12267 ;* ERROR N ;;ERROR=EMT AND N=ERROR ITEM NUMBER
12268
12269 047174 $ERROR:
12270 047174 104407 CKSWR ;;TEST FOR CHANGE IN SOFT-SWR
12271
12272 047176 REGSAV:
12273 047176 012737 177777 004734 MOV #-1,@#ERFLG ;SET ERROR FLAG
12274 047204 REGSA1:
12275 047204 105237 001103 7$: INCB $ERFLG ;;SET THE ERROR FLAG
12276 047210 001775 BEQ 7$ ;;DON'T LET THE FLAG GO TO ZERO
12277 047212 013777 001102 131722 MOV $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
12278 047220 032777 002000 131712 BIT #BIT10,@SWR ;;BELL ON ERROR?
12279 047226 001402 BEQ 1$ ;;NO - SKIP
12280 047230 104401 001216 TYPE ,SBELL ;;RING BELL
12281 047234 005237 001112 1$: INC $ERTTL ;;COUNT THE NUMBER OF ERRORS
12282 047240 011637 001116 MOV (SP),$ERRPC ;;GET ADDRESS OF ERROR INSTRUCTION
12283 047244 162737 000002 001116 SUB #2,$ERRPC
12284 047252 117737 131640 001114 MOVB @ $ERRPC,$ITEMB ;;STRIP AND SAVE THE ERROR ITEM CODE
12285 047260 032777 020000 131652 BIT #BIT13,@SWR ;;SKIP TYPEOUT IF SET
12286 047266 001004 BNE 20$ ;;SKIP TYPEOUTS
12287 047270 004737 047344 JSR PC,$ERRTYP ;;GO TO USER ERROR ROUTINE
12288 047274 104401 001223 TYPE ,SCLRF
12289 047300 20$:
12290 047300 005777 131634 2$: TST @SWR ;;HALT ON ERROR
12291 047304 100002 BPL 3$ ;;SKIP IF CONTINUE
12292 047306 000000 HALT ;;HALT ON ERROR!
12293 047310 104407 CKSWR ;;TEST FOR CHANGE IN SOFT-SWR
12294 047312 032777 001000 131620 3$: BIT #BIT09,@SWR ;;LOOP ON ERROR SWITCH SET?
12295 047320 001402 BEQ 4$ ;;BR IF NO
12296 047322 013716 001110 MOV $LPERR,(SP) ;;FUDGE RETURN FOR LOOPING
12297 047326 005737 001214 4$: TST $ESCAPE ;;CHECK FOR AN ESCAPE ADDRESS
12298 047332 001402 BEQ 5$ ;;BR IF NONE
12299 047334 013716 001214 MOV $ESCAPE,(SP) ;;FUDGE RETURN ADDRESS FOR ESCAPE
12300 047340 5$:
12301 047340 000002 RTI ;;RETURN
12302
```

12303
12304
12305
12306
12307
12308
12309
12310
12311
12312
12313
12314
12315
12316
12317
12318
12319
12320
12321
12322
12323
12324
12325
12326
12327
12328
12329
12330
12331
12332
12333
12334
12335
12336
12337
12338
12339
12340
12341
12342
12343
12344
12345
12346
12347
12348
12349
12350
12351
12352
12353
12354
12355
12356
12357
12358

047342 000000
047344 017746 131570
047350 042716 177277
047354 022726 000100
047360 001001
047362 000402
047364 000137 050304
047370
047370 104401 047376
047374 000406
047412
047412 013746 002354
047416 104402
047420 104401 047426
047424 000406
047442
047442 013746 002356
047446 104402
047450 104401 047456
047454 000406
047472
047472 013746 002360
047476 104402
047500 104401 047506
047504 000406
047522
047522 013746 002362

```
;;*****  
.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  
;:65$: .ASCIZ <15><12>/RHWC = /  
64$: MOV @#WC,-(SP) ;GET READY TO TYPE RHWC CONTENTS  
;TYPOC  
;TYPE ,67$ ;:TYPE ASCIZ STRING  
;BR 66$ ;:GET OVER THE ASCIZ  
;:67$: .ASCIZ <15><12>/RHBA = /  
66$: MOV @#BA,-(SP) ;GET READY TO TYPE RHBA CONTENTS  
;TYPOC  
;TYPE ,69$ ;:TYPE ASCIZ STRING  
;BR 68$ ;:GET OVER THE ASCIZ  
;:69$: .ASCIZ <15><12>/RHCS2 = /  
68$: MOV @#CS2,-(SP) ;GET READY TO TYPE RHCS2 CONTENTS  
;TYPOC  
;TYPE ,71$ ;:TYPE ASCIZ STRING  
;BR 70$ ;:GET OVER THE ASCIZ  
;:71$: .ASCIZ <15><12>/RHCS1 = /  
70$: MOV @#CS1,-(SP) ;GET READY TO TYPE RHCS1 CONTENTS
```

```

12359 047526 104402          TYPOC
12360
12361
12362 047530 104401 047536    TYPE      ,73$          ;;TYPE ASCIZ STRING
12363 047534 000406          BR        72$          ;;GET OVER THE ASCIZ
12364          ;;73$: .ASCIZ <15><12>/RHDS1 = /
12365 047552          72$:
12366 047552 013746 002404    MOV      @#D$1,-(SP)    ;GET READY TO TYPE RHDS1 CONTENTS
12367 047556 104402          TYPOC
12368
12369
12370 047560 104401 047566    TYPE      ,75$          ;;TYPE ASCIZ STRING
12371 047564 000406          BR        74$          ;;GET OVER THE ASCIZ
12372          ;;75$: .ASCIZ <15><12>/RH$R1 = /
12373 047602          74$:
12374 047602 013746 002364    MOV      @#ER1,-(SP)   ;GET READY TO TYPE RH$R1 CONTENTS
12375 047606 104402          TYPOC
12376
12377
12378 047610 104401 047616    TYPE      ,77$          ;;TYPE ASCIZ STRING
12379 047614 000406          BR        76$          ;;GET OVER THE ASCIZ
12380          ;;77$: .ASCIZ <15><12>/RH$R2 = /
12381 047632          76$:
12382 047632 013746 002370    MOV      @#ER2,-(SP)   ;GET READY TO TYPE RH$R2 CONTENTS
12383 047636 104402          TYPOC
12384
12385
12386 047640 104401 047646    TYPE      ,79$          ;;TYPE ASCIZ STRING
12387 047644 000406          BR        78$          ;;GET OVER THE ASCIZ
12388          ;;79$: .ASCIZ <15><12>/RH$R3 = /
12389 047662          78$:
12390 047662 013746 002376    MOV      @#ER3,-(SP)   ;GET READY TO TYPE RH$R3 CONTENTS
12391 047666 104402          TYPOC
12392
12393
12394 047670 104401 047676    TYPE      ,81$          ;;TYPE ASCIZ STRING
12395 047674 000406          BR        80$          ;;GET OVER THE ASCIZ
12396          ;;81$: .ASCIZ <15><12>/RH$D$T = /
12397 047712          80$:
12398 047712 013746 002366    MOV      @#D$T,-(SP)   ;GET READY TO TYPE RH$D$T CONTENTS
12399 047716 104402          TYPOC
12400
12401
12402 047720 104401 047726    TYPE      ,83$          ;;TYPE ASCIZ STRING
12403 047724 000406          BR        82$          ;;GET OVER THE ASCIZ
12404          ;;83$: .ASCIZ <15><12>/RH$C$A = /
12405 047742          82$:
12406 047742 013746 002374    MOV      @#C$A,-(SP)   ;GET READY TO TYPE RH$C$A CONTENTS
12407 047746 104402          TYPOC
12408
12409
12410 047750 104401 047756    TYPE      ,85$          ;;TYPE ASCIZ STRING
12411 047754 000406          BR        84$          ;;GET OVER THE ASCIZ
12412          ;;85$: .ASCIZ <15><12>/RH$A$S = /
12413 047772          84$:
12414 047772 013746 002400    MOV      @#A$S,-(SP)   ;GET READY TO TYPE RH$A$S CONTENTS
  
```

```

12415 047776 104402          TYPOC
12416
12417
12418 050000 104401 050006      TYPE      ,87$          ;;TYPE ASCIZ STRING
12419 050004 000406          BR        86$          ;;GET OVER THE ASCIZ
12420          ;;87$: .ASCIZ <15><12>/RHOF = /
12421 050022 86$:
12422 050022 013746 002372      MOV      @#OF,-(SP)    ;GET READY TO TYPE RHOF CONTENTS
12423 050026 104402          TYPOC
12424
12425
12426 050030 104401 050036      TYPE      ,89$          ;;TYPE ASCIZ STRING
12427 050034 000406          BR        88$          ;;GET OVER THE ASCIZ
12428          ;;89$: .ASCIZ <15><12>/RHMR = /
12429 050052 88$:
12430 050052 013746 002402      MOV      @#MR,-(SP)    ;GET READY TO TYPE RHMR CONTENTS
12431 050056 104402          TYPOC
12432
12433
12434 050060 104401 050066      TYPE      ,91$          ;;TYPE ASCIZ STRING
12435 050064 000406          BR        90$          ;;GET OVER THE ASCIZ
12436          ;;91$: .ASCIZ <15><12>/RHLA = /
12437 050102 90$:
12438 050102 013746 002420      MOV      @#LA,-(SP)    ;GET READY TO TYPE RHLA CONTENTS
12439 050106 104402          TYPOC
12440
12441
12442 050110 104401 050116      TYPE      ,93$          ;;TYPE ASCIZ STRING
12443 050114 000406          BR        92$          ;;GET OVER THE ASCIZ
12444          ;;93$: .ASCIZ <15><12>/RHCC = /
12445 050132 92$:
12446 050132 013746 002416      MOV      @#CC,-(SP)    ;GET READY TO TYPE RHCC CONTENTS
12447 050136 104402          TYPOC
12448
12449
12450 050140 104401 050146      TYPE      ,95$          ;;TYPE ASCIZ STRING
12451 050144 000406          BR        94$          ;;GET OVER THE ASCIZ
12452          ;;95$: .ASCIZ <15><12>/RHEC1 = /
12453 050162 94$:
12454 050162 013746 002412      MOV      @#EC1,-(SP)   ;GET READY TO TYPE RHEC1 CONTENTS
12455 050166 104402          TYPOC
12456
12457
12458 050170 104401 050176      TYPE      ,97$          ;;TYPE ASCIZ STRING
12459 050174 000406          BR        96$          ;;GET OVER THE ASCIZ
12460          ;;97$: .ASCIZ <15><12>/RHEC2 = /
12461 050212 96$:
12462 050212 013746 002414      MOV      @#EC2,-(SP)   ;GET READY TO TYPE RHEC2 CONTENTS
12463 050216 104402          TYPOC
12464
12465
12466 050220 104401 050226      TYPE      ,99$          ;;TYPE ASCIZ STRING
12467 050224 000406          BR        98$          ;;GET OVER THE ASCIZ
12468          ;;99$: .ASCIZ <15><12>/RHDT = /
12469 050242 98$:
12470 050242 013746 002406      MOV      @#DT,-(SP)    ;GET READY TO TYPE RHDT CONTENTS

```

```

12471 050246 104402          TYP0C
12472
12473
12474 050250 104401 050256    TYPE      ,101$      ;;TYPE ASCIZ STRING
12475 050254 000406          BR          100$      ;;GET OVER THE ASCIZ
12476          ;;101$: .ASCIZ <15><12>/RHSN = /
12477 050272          100$:
12478 050272 013746 002410    MOV      @#SN,-(SP)   ;GET READY TO TYPE RHSN CONTENTS
12479 050276 104402          TYP0C
12480
12481 050300 005037 047342    CLR      @#PRITEM    ;CLEAR PREVIOUS ERROR ITEM
12482 050304          TYPERR:
12483 050304 104401 001223    TYPE      , $CRLF    ;'"CARRIAGE RETURN" & "LINE FEED"'
12484 050310 010046          MOV      RO,-(SP)    ;SAVE RO
12485 050312 005000          CLR      RO          ;PICKUP THE ITEM INDEX
12486 050314 153700 001114    BISB     @#$ITEMB,RO
12487 050320 001004          BNE      1$         ;IF ITEM NUMBER IS ZERO, JUST
12488          ;TYPE THE PC OF THE ERROR
12489 050322 013746 001116    MOV      $ERRPC,-(SP);SAVE $ERRPC FOR TYPEOUT
12490          ;ERROR ADDRESS
12491 050326 104402          TYP0C          ;GO TYPE--OCTAL ASCII(ALL DIGITS)
12492 050330 000454          BR          10$      ;GET OUT
12493 050332 005300          1$: DEC      RO      ;ADJUST THE INDEX SO THAT IT WILL
12494 050334 006300          ASL      RO          ;
12495 050336 006300          ASL      RO          ;
12496 050340 006300          ASL      RO          ;
12497 050342 062700 001226    ADD      #$ERRTB,RO  ;FORM TABLE POINTER
12498 050346 020037 047342    CMP      RO,@#PRITEM;WAS PREVIOUS ERROR SAME
12499 050352 001002          BNE      13$        ;BRANCH IF NOT
12500 050354 022020          CMP      (RO)+,(RO)+;POP RO OVER EM AND DH
12501 050356 000420          BR          5$
12502 050360 010037 047342          13$: MOV      RO,@#PRITEM ;SAVE NEW ERROR ITEM
12503 050364 012037 050374    MOV      (RO)+,2$   ;PICKUP "ERROR MESSAGE" POINTER
12504 050370 001404          BEQ      3$         ;SKIP TYPEOUT IF NO POINTER
12505 050372 104401          TYPE     ;TYPE THE "ERROR MESSAGE"
12506 050374 000000          2$: .WORD    0      ;"ERROR MESSAGE" POINTER GOES HERE
12507 050376 104401 001223    TYPE     , $CRLF    ;'"CARRIAGE RETURN" & "LINE FEED"'
12508 050402 012037 050412          3$: MOV      (RO)+,4$   ;PICKUP "DATA HEADER" POINTER
12509 050406 001404          BEQ      5$         ;SKIP TYPEOUT IF 0
12510 050410 104401          TYPE     ;TYPE THE "DATA HEADER"
12511 050412 000000          4$: .WORD    0      ;"DATA HEADER" POINTER GOES HERE
12512 050414 104401 001223    TYPE     , $CRLF    ;'"CARRIAGE RETURN" & "LINE FEED"'
12513 050420 010146          5$: MOV      R1,-(SP)   ;SAVE R1
12514 050422 012001          MOV      (RO)+,R1   ;PICKUP "DATA TABLE" POINTER
12515 050424 001415          BEQ      9$         ;BR IF NO DATA TO BE TYPED
12516 050426 012000          MOV      (RO)+,RO   ;PICKUP "DATA FORMAT" POINTER
12517 050430 105720          6$: TSTB     (RO)+    ;"OCTAL" OR "DECIMAL"
12518 050432 001003          BNE      7$         ;BR IF DECIMAL
12519 050434 013146          MOV      @#(R1)+,-(SP);SAVE @#(R1)+ FOR TYPEOUT
12520 050436 104402          TYP0C          ;GO TYPE--OCTAL ASCII(ALL DIGITS)
12521 050440 000402          BR          8$
12522 050442          7$:
12523 050442 013146          MOV      @#(R1)+,-(SP);SAVE @#(R1)+ FOR TYPEOUT
12524 050444 104405          TYPDS          ;GO TYPE--DECIMAL ASCII WITH SIGN
12525 050446 005711          8$: TST      (R1)     ;IS THERE ANOTHER NUMBER?
12526 050450 001403          BEQ      9$         ;BR IF NO
  
```

```
12527 050452 104401 050466          TYPE      ,11$          ;TYPE TWO(2) SPACES
12528 050456 000764                   BR          6$          ;LOOP
12529
12530 050460 012601                   9$:      MOV      (SP)+,R1      ;RESTORE R1
12531 050462 012600                   10$:     MOV      (SP)+,R0      ;"CARRIAGE RETURN" & "LINE FEED"
12532 050464 000207                   RTS        PC              ;RETURN
12533 050466 020040 000                11$:     .ASCIZ  /  /          ;TWO(2) SPACES
12534 050472
12535
```

12536
12537
12538
12539
12540
12541
12542
12543
12544
12545
12546
12547
12548
12549
12550
12551
12552
12553
12554
12555
12556
12557
12558
12559
12560
12561
12562
12563
12564
12565
12566
12567
12568
12569
12570
12571
12572
12573
12574
12575
12576
12577
12578
12579
12580
12581
12582
12583
12584
12585
12586
12587
12588
12589
12590
12591

050472 017646 000000
050476 116637 000001 050715
050504 112637 050717
050510 062716 000002
050514 000406
050516 112737 000001 050715
050524 112737 000006 050717
050532 112737 000005 050714
050540 010346
050542 010446
050544 010546
050546 113704 050717
050552 005404
050554 062704 000006
050560 110437 050716
050564 113704 050715
050570 016605 000012
050574 005003
050576 006105 1\$:
050600 000404 BR 3\$:
050602 006105 2\$:
050604 006105
050606 006105
050610 010503
050612 006103
050614 105337 050716
050620 100016
050622 042703 177770
050626 001002
050630 005704

.SBTTL BINARY TO OCTAL (ASCII) AND TYPE

```
*****
*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
*OCTAL (ASCII) NUMBER AND TYPE IT.
*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
*CALL:
*   MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
*   TYPOS    ;;CALL FOR TYPEOUT
*   .BYTE   N              ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
*   .BYTE   M              ;;M=1 OR 0
*                               ;;1=TYPE LEADING ZEROS
*                               ;;0=SUPPRESS LEADING ZEROS
*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
*$TYPOS OR $TYPOC
*CALL:
*   MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
*   TYPON    ;;CALL FOR TYPEOUT
*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
*CALL:
*   MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
*   TYPOC    ;;CALL FOR TYPEOUT
*$TYPOS: MOV      @(SP),-(SP)  ;;PICKUP THE MODE
        MOV      1(SP),%OFILL  ;;LOAD ZERO FILL SWITCH
        MOV      (SP)+,%OMODE+1 ;;NUMBER OF DIGITS TO TYPE
        ADD      #2,(SP)      ;;ADJUST RETURN ADDRESS
        BR      $TYPON
*$TYPOC: MOV      #1,%OFILL    ;;SET THE ZERO FILL SWITCH
        MOV      #6,%OMODE+1   ;;SET FOR SIX(6) DIGITS
*$TYPON: MOV      #5,%OCNT     ;;SET THE ITERATION COUNT
        MOV      R3,-(SP)      ;;SAVE R3
        MOV      R4,-(SP)      ;;SAVE R4
        MOV      R5,-(SP)      ;;SAVE R5
        MOV      %OMODE+1,R4   ;;GET THE NUMBER OF DIGITS TO TYPE
        NEG      R4
        ADD      #6,R4         ;;SUBTRACT IT FOR MAX. ALLOWED
        MOV      R4,%OMODE     ;;SAVE IT FOR USE
        MOV      %OFILL,R4    ;;GET THE ZERO FILL SWITCH
        MOV      12(SP),R5    ;;PICKUP THE INPUT NUMBER
        CLR      R3           ;;CLEAR THE OUTPUT WORD
        ROL     R5            ;;ROTATE MSB INTO "C"
        BR      3$           ;;GO DO MSB
        ROL     R5            ;;FORM THIS DIGIT
        ROL     R5
        ROL     R5
        MOV     R5,R3
        ROL     R3            ;;GET LSB OF THIS DIGIT
        DECB   %OMODE        ;;TYPE THIS DIGIT?
        BPL    7$            ;;BR IF NO
        BIC   #177770,R3    ;;GET RID OF JUNK
        BNE   4$            ;;TEST FOR 0
        TST   R4            ;;SUPPRESS THIS 0?
```


12592	050632	001403			BEQ	5\$::BR IF YES
12593	050634	005204			INC	R4	::DON'T SUPPRESS ANYMORE 0'S
12594	050636	052703	000060		BIS	#'0,R3	::MAKE THIS DIGIT ASCII
12595	050642	052703	000040		BIS	#',R3	::MAKE ASCII IF NOT ALREADY
12596	050646	110337	050712		MOVB	R3,8\$::SAVE FOR TYPING
12597	050652	104401	050712		TYPE	,8\$::GO TYPE THIS DIGIT
12598	050656	105337	050714		DECB	\$OCNT	::COUNT BY 1
12599	050662	003347			BGT	2\$::BR IF MORE TO DO
12600	050664	002402			BLT	6\$::BR IF DONE
12601	050666	005204			INC	R4	::INSURE LAST DIGIT ISN'T A BLANK
12602	050670	000744			BR	2\$::GO DO THE LAST DIGIT
12603	050672	012605			MOV	(SP)+,R5	::RESTORE R5
12604	050674	012604			MOV	(SP)+,R4	::RESTORE R4
12605	050676	012603			MOV	(SP)+,R3	::RESTORE R3
12606	050700	016666	000002	000004	MOV	2(SP),4(SP)	::SET THE STACK FOR RETURNING
12607	050706	012616			MOV	(SP)+,(SP)	
12608	050710	000002			RTI		::RETURN
12609	050712	000			.BYTE	0	::STORAGE FOR ASCII DIGIT
12610	050713	000			.BYTE	0	::TERMINATOR FOR TYPE ROUTINE
12611	050714	000			\$OCNT:	.BYTE 0	::OCTAL DIGIT COUNTER
12612	050715	000			\$OFILL:	.BYTE 0	::ZERO FILL SWITCH
12613	050716	000000			\$OMODE:	.WORD 0	::NUMBER OF DIGITS TO TYPE

```
12614 .SBTTL TRAP DECODER
12615
12616 ;*****
12617 ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
12618 ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
12619 ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
12620 ;*GO TO THAT ROUTINE.
12621
12622 050720 010046 $TRAP: MOV RO,-(SP) ;;SAVE RO
12623 050722 016600 000002 MOV 2(SP),RO ;;GET TRAP ADDRESS
12624 050726 005740 TST -(RO) ;;BACKUP BY 2
12625 050730 111000 MOVB (RO),RO ;;GET RIGHT BYTE OF TRAP
12626 050732 006300 ASL RO ;;POSITION FOR INDEXING
12627 050734 016000 050754 MOV $TRPAD(RO),RO ;;INDEX TO TABLE
12628 050740 000200 RTS RO ;;GO TO ROUTINE
12629
12630
12631 ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
12632
12633 050742 011646 $TRAP2: MOV (SP),-(SP) ;;MOVE THE PC DOWN
12634 050744 016666 000004 000002 MOV 4(SP),2(SP) ;;MOVE THE PSW DOWN
12635 050752 000002 RTI ;;RESTORE THE PSW
12636
12637 .SBTTL TRAP TABLE
12638
12639 ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
12640 ;*BY THE "TRAP" INSTRUCTION.
12641
12642 : ROUTINE
12643 : -----
12644 050754 050742 $TRPAD: .WORD $TRAP2
12645 050756 045474 $TYPE ;;CALL=TYPE TRAP+1(104401) TTY TYPEOUT ROUTINE
12646 050760 050516 $TYPOC ;;CALL=TYPOC TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
12647 050762 050472 $TYPOS ;;CALL=TYPOS TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)
12648 050764 050532 $TYPON ;;CALL=TYPON TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)
12649 050766 045250 $TYPDS ;;CALL=TYPDS TRAP+5(104405) TYPE DECIMAL NUMBER (WITH SIGN)
12650
12651 050770 046312 $GTSWR ;;CALL=GTSWR TRAP+6(104406) GET SOFT-SWR SETTING
12652
12653 050772 046222 $CKSWR ;;CALL=CKSWR TRAP+7(104407) TEST FOR CHANGE IN SOFT-SWR
12654 050774 046564 $RDCHR ;;CALL=RDCHR TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
12655 050776 046654 $RDLIN ;;CALL=RDLIN TRAP+11(104411) TTY TYPEIN STRING ROUTINE
12656 051000 047034 $RDOCT ;;CALL=RDOCT TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
12657 051002 042224 WAIT.T ;;CALL=WAT TRAP+13(104413) DONT ADD ABOVE THIS TRAP
12658
```

```
12659 .SBTTL POWER DOWN AND UP ROUTINES
12660
12661 ::*****
12662 :POWER DOWN ROUTINE
12663 051004 012737 051150 000024 $PWRDN: MOV #SILLUP,@#PWRVEC ;;SET FOR FAST UP
12664 051012 012737 000340 000026 MOV #340,@#PWRVEC+2 ;;PRIO:7
12665 051020 010046 MOV R0,-(SP) ;;PUSH R0 ON STACK
12666 051022 010146 MOV R1,-(SP) ;;PUSH R1 ON STACK
12667 051024 010246 MOV R2,-(SP) ;;PUSH R2 ON STACK
12668 051026 010346 MOV R3,-(SP) ;;PUSH R3 ON STACK
12669 051030 010446 MOV R4,-(SP) ;;PUSH R4 ON STACK
12670 051032 010546 MOV R5,-(SP) ;;PUSH R5 ON STACK
12671 051034 017746 130100 MOV @SWR,-(SP) ;;PUSH @SWR ON STACK
12672 051040 010637 051154 MOV SP,$SAVR6 ;;SAVE SP
12673 051044 012737 051056 000024 MOV #SPWRUP,@#PWRVEC ;;SET UP VECTOR
12674 051052 000000 HALT
12675 051054 000776 BR .-2 ;;HANG UP
12676
12677 ::*****
12678 :POWER UP ROUTINE
12679 051056 012737 051150 000024 $PWRUP: MOV #SILLUP,@#PWRVEC ;;SET FOR FAST DOWN
12680 051064 013706 051154 MOV $SAVR6,SP ;;GET SP
12681 051070 005037 051154 CLR $SAVR6 ;;WAIT LOOP FOR THE TTY
12682 051074 005237 051154 1$: INC $SAVR6 ;;WAIT FOR THE INC
12683 051100 001375 BNE 1$ ;;OF WORD
12684 051102 012677 130032 MOV (SP)+,@SWR ;;POP STACK INTO @SWR
12685 051106 012605 MOV (SP)+,R5 ;;POP STACK INTO R5
12686 051110 012604 MOV (SP)+,R4 ;;POP STACK INTO R4
12687 051112 012603 MOV (SP)+,R3 ;;POP STACK INTO R3
12688 051114 012602 MOV (SP)+,R2 ;;POP STACK INTO R2
12689 051116 012601 MOV (SP)+,R1 ;;POP STACK INTO R1
12690 051120 012600 MOV (SP)+,R0 ;;POP STACK INTO R0
12691 051122 012737 051004 000024 MOV #SPWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
12692 051130 012737 000340 000026 MOV #340,@#PWRVEC+2 ;;PRIO:7
12693 051136 104401 TYPE ;;REPORT THE POWER FAILURE
12694 051140 051156 $PWRMG: .WORD $POWER ;;POWER FAIL MESSAGE POINTER
12695 051142 012716 MOV (PC)+,(SP) ;;RESTART AT BEGIN
12696 051144 005012 $PWRAD: .WORD BEGIN ;;RESTART ADDRESS
12697 051146 000002 RTI
12698 051150 000000 $ILLUP: HALT ;;THE POWER UP SEQUENCE WAS STARTED
12699 051152 000776 BR .-2 ;; BEFORE THE POWER DOWN WAS COMPLETE
12700 051154 000000 $SAVR6: 0 ;;PUT THE SP HERE
12701 051156 005015 047520 042527 $POWER: .ASCIZ <15><12>'POWER'
12702 051164 000122
12703 .EVEN
```

```

12704
12705
12706
12707
12708
12709
12710
12711
12712
12713 051166 050122 032060 042040 EM1: .ASCIZ /RPO4 DID NOT INTERRUPT/
12714 051174 042111 047040 052117
12715 051202 044440 052116 051105
12716 051210 052522 052120 000
12717 051215 111 052116 051105 EM2: .ASCIZ /INTERRUPT ENABLE BIT DOWN BUT EXPECTED BIT DID NOT SET/
12718 051222 052522 052120 042440
12719 051230 040516 046102 020105
12720 051236 044502 020124 047504
12721 051244 047127 041040 052125
12722 051252 042440 050130 041505
12723 051260 042524 020104 044502
12724 051266 020124 044504 020104
12725 051274 047516 020124 042523
12726 051302 000124
12727 051304 050122 032060 042040 EM3: .ASCIZ /RPO4 DID NOT INTERRUPT WHEN EXPECTED BIT DID SET/
12728 051312 042111 047040 052117
12729 051320 044440 052116 051105
12730 051326 052522 052120 053440
12731 051334 042510 020116 054105
12732 051342 042520 052103 042105
12733 051350 041040 052111 042040
12734 051356 042111 051440 052105
12735 051364 000
12736 051365 105 050130 041505 EM4: .ASCIZ /EXPECTED BIT DID SET BUT TIME IS IN ERROR (TIME IN 10 MICROSEC, DECIMAL
12737 051372 042524 020104 044502
12738 051400 020124 044504 020104
12739 051406 042523 020124 052502
12740 051414 020124 044524 042515
12741 051422 044440 020123 047111
12742 051430 042440 051122 051117
12743 051436 024040 044524 042515
12744 051444 044440 020116 030061
12745 051452 046440 041511 047522
12746 051460 042523 026103 042040
12747 051466 041505 046511 046101
12748 051474 000051
12749 051476 044122 051501 042040 EM5: .ASCIZ /RHAS DOES NOT CLEAR BY MOVING IN ALL ONES/
12750 051504 042517 020123 047516
12751 051512 020124 046103 040505
12752 051520 020122 054502 046440
12753 051526 053117 047111 020107
12754 051534 047111 040440 046114
12755 051542 047440 042516 000123
12756 051550 047514 042101 047111 EM6: .ASCIZ /LOADING RHER1 FOR ALL UNITS DID NOT SET ANY RHAS BITS/
12757 051556 020107 044122 051105
12758 051564 020061 047506 020122
12759 051572 046101 020114 047125
  
```

12760	051600	052111	020123	044504	
12761	051606	020104	047516	020124	
12762	051614	042523	020124	047101	
12763	051622	020131	044122	051501	
12764	051630	041040	052111	000123	
12765	051636	047516	020116	054105	EM7: .ASCIZ /NON EXISTENT REGISTER, PROGRAM ABORTED./
12766	051644	051511	042524	052116	
12767	051652	051040	043505	051511	
12768	051660	042524	026122	050040	
12769	051666	047522	051107	046501	
12770	051674	040440	047502	052122	
12771	051702	042105	000056		
12772					
12773	051706	052123	050117	042520	EM10: .ASCIZ /STOPPED DRIVE HAS MOL BIT IN RHDS1 SET/
12774	051714	020104	051104	053111	
12775	051722	020105	040510	020123	
12776	051730	047515	020114	044502	
12777	051736	020124	047111	051040	
12778	051744	042110	030523	051440	
12779	051752	052105	000		
12780	051755	127	052111	020110	EM11: .ASCIZ /WITH SPINDLE POWERED DOWN RHCS2 SHOULD ONLY HAVE UNIT NO. AND IR SET/
12781	051762	050123	047111	046104	
12782	051770	020105	047520	042527	
12783	051776	042522	020104	047504	
12784	052004	047127	051040	041510	
12785	052012	031123	051440	047510	
12786	052020	046125	020104	047117	
12787	052026	054514	044040	053101	
12788	052034	020105	047125	052111	
12789	052042	047040	027117	040440	
12790	052050	042116	044440	020122	
12791	052056	042523	000124		
12792	052062	043101	042524	020122	EM12: .ASCIZ /AFTER SPINDLE POWERED UP, NO PACK ACKN. RHDS1 SHOULD HAVE MOL=1, VV=0/
12793	052070	050123	047111	046104	
12794	052076	020105	047520	042527	
12795	052104	042522	020104	050125	
12796	052112	020054	047516	050040	
12797	052120	041501	020113	041501	
12798	052126	047113	020056	044122	
12799	052134	051504	020061	044123	
12800	052142	052517	042114	044040	
12801	052150	053101	020105	047515	
12802	052156	036514	026061	053040	
12803	052164	036526	000060		
12804	052170	044527	044124	051440	EM13: .ASCIZ /WITH SPINDLE POWERED UP, NO INTIALIZE, RHCS1 SHOULD HAVE GO=0, DVA=1, R
12805	052176	044520	042116	042514	
12806	052204	050040	053517	051105	
12807	052212	042105	052440	026120	
12808	052220	047040	020117	047111	
12809	052226	044524	046101	055111	
12810	052234	026105	051040	041510	
12811	052242	030523	051440	047510	
12812	052250	046125	020104	040510	
12813	052256	042526	043440	036517	
12814	052264	026060	042040	040526	
12815	052272	030475	020054	042122	

12816	052300	036531	026061	044440	
12817	052306	036505	000060		
12818	052312	043101	042524	020122	EM14: .ASCIZ /AFTER SPINDLE POWERED UP, RMCC SHOULD BE=0/
12819	052320	050123	047111	046104	
12820	052326	020105	047520	042527	
12821	052334	042522	020104	050125	
12822	052342	020054	044122	041503	
12823	052350	051440	047510	046125	
12824	052356	020104	042502	030075	
12825	052364	000			
12826	052365	120	041501	020113	EM15: .ASCII /PACK ACKNOWLEDGE COMMAND CAUSED AN ERROR/<15><12>
12827	052372	041501	047113	053517	
12828	052400	042514	043504	020105	
12829	052406	047503	046515	047101	
12830	052414	020104	040503	051525	
12831	052422	042105	040440	020116	
12832	052430	051105	047522	006522	
12833	052436	012			
12834	052437	107	047517	020104	.ASCIZ /GOOD DATA IS BEFORE COMMAND, REC DATA IS AFTER COMMAND/
12835	052444	040504	040524	044440	
12836	052452	020123	042502	047506	
12837	052460	042522	041440	046517	
12838	052466	040515	042116	020054	
12839	052474	042522	020103	040504	
12840	052502	040524	044440	020123	
12841	052510	043101	042524	020122	
12842	052516	047503	046515	047101	
12843	052524	000104			
12844	052526	047516	047455	020120	EM16: .ASCII /NO-OP COMMAND CAUSED AN ERROR/<15><12>
12845	052534	047503	046515	047101	
12846	052542	020104	040503	051525	
12847	052550	042105	040440	020116	
12848	052556	051105	047522	006522	
12849	052564	012			
12850	052565	107	047517	020104	.ASCIZ /GOOD DATA IS BEFORE COMMAND, REC DATA IS AFTER COMMAND/
12851	052572	040504	040524	044440	
12852	052600	020123	042502	047506	
12853	052606	042522	041440	046517	
12854	052614	040515	042116	020054	
12855	052622	042522	020103	040504	
12856	052630	040524	044440	020123	
12857	052636	043101	042524	020122	
12858	052644	047503	046515	047101	
12859	052652	000104			
12860	052654	051104	053111	020105	EM17: .ASCII /DRIVE CLEAR COMMAND CAUSED AN ERROR/<15><12>
12861	052662	046103	040505	020122	
12862	052670	047503	046515	047101	
12863	052676	020104	040503	051525	
12864	052704	042105	040440	020116	
12865	052712	051105	047522	006522	
12866	052720	012			
12867	052721	107	047517	020104	.ASCIZ /GOOD DATA GIVES SHOULD BE, REC DATA GIVES AFTER COMMAND/
12868	052726	040504	040524	043440	
12869	052734	053111	051505	051440	
12870	052742	047510	046125	020104	
12871	052750	042502	020054	042522	

12872	052756	020103	040504	040524	
12873	052764	043440	053111	051505	
12874	052772	040440	052106	051105	
12875	053000	041440	046517	040515	
12876	053006	042116	000		
12877					
12878	053011	122	040505	026504	EM20: .ASCII /READ-IN COMMAND CAUSED AN ERROR/<15><12>
12879	053016	047111	041440	046517	
12880	053024	040515	042116	041440	
12881	053032	052501	042523	020104	
12882	053040	047101	042440	051122	
12883	053046	051117	005015		
12884	053052	047507	042117	042040	.ASCIZ /GOOD DATA GIVES SHOULD BE, REC DATA GIVES RFG. CONT. AFTER COMMAND/
12885	053060	052101	020101	044507	
12886	053066	042526	020123	044123	
12887	053074	052517	042114	041040	
12888	053102	026105	051040	041505	
12889	053110	042040	052101	020101	
12890	053116	044507	042526	020123	
12891	053124	042522	027107	041440	
12892	053132	047117	027124	040440	
12893	053140	052106	051105	041440	
12894	053146	046517	040515	042116	
12895	053154	000			
12896	053155	122	041510	030523	EM21: .ASCIZ /RHCS1 CONTENTS DURING COMMAND WAS IN ERROR/
12897	053162	041440	047117	042524	
12898	053170	052116	020123	052504	
12899	053176	044522	043516	041440	
12900	053204	046517	040515	042116	
12901	053212	053440	051501	044440	
12902	053220	020116	051105	047522	
12903	053226	000122			
12904	053230	044122	051504	020061	EM22: .ASCIZ /RHDS1 CONTENTS DURING COMMAND WAS IN ERROR/
12905	053236	047503	052116	047105	
12906	053244	051524	042040	051125	
12907	053252	047111	020107	047503	
12908	053260	046515	047101	020104	
12909	053266	040527	020123	047111	
12910	053274	042440	051122	051117	
12911	053302	000			
12912	053303	125	046116	040517	EM23: .ASCII /UNLOAD COMMAND CAUSED AN ERROR/<15><12>
12913	053310	020104	047503	046515	
12914	053316	047101	020104	040503	
12915	053324	051525	042105	040440	
12916	053332	020116	051105	047522	
12917	053340	006522	012		
12918	053343	107	047517	020104	.ASCIZ /GOOD DATA GIVES SHOULD BE, REC DATA GIVES REG. CONT. AFTER COMMAND/
12919	053350	040504	040524	043440	
12920	053356	053111	051505	051440	
12921	053364	047510	046125	020104	
12922	053372	042502	020054	042522	
12923	053400	020103	040504	040524	
12924	053406	043440	053111	051505	
12925	053414	051040	043505	020056	
12926	053422	047503	052116	020056	
12927	053430	043101	042524	020122	

12928	053436	047503	046515	047101	
12929	053444	000104			
12930	053446	043117	051506	052105	EM24: .ASCII /OFFSET COMMAND CAUSED AN ERROR/<15><12>
12931	053454	041440	046517	040515	
12932	053462	042116	041440	052501	
12933	053470	042523	020104	047101	
12934	053476	042440	051122	051117	
12935	053504	005015			
12936	053506	047507	042117	042040	.ASCIZ /GOOD DATA GIVES SHOULD BE, REC DATA GIVES REG. CONT. AFTER COMMAND/
12937	053514	052101	020101	044507	
12938	053522	042526	020123	044123	
12939	053530	052517	042114	041040	
12940	053536	026105	051040	041505	
12941	053544	042040	052101	020101	
12942	053552	044507	042526	020123	
12943	053560	042522	027107	041440	
12944	053566	047117	027124	040440	
12945	053574	052106	051105	041440	
12946	053602	046517	040515	042116	
12947	053610	000			
12948	053611	122	052105	051125	EM25: .ASCII /RETURN TO CENTER LINE COMMAND CAUSED AN ERROR/<15><12>
12949	053616	020116	047524	041440	
12950	053624	047105	042524	020122	
12951	053632	044514	042516	041440	
12952	053640	046517	040515	042116	
12953	053646	041440	052501	042523	
12954	053654	020104	047101	042440	
12955	053662	051122	051117	005015	
12956	053670	047507	042117	042040	.ASCIZ /GOOD DATA GIVES SHOULD BE, REC DATA GIVES REG. CONT. AFTER COMMAND/
12957	053676	052101	020101	044507	
12958	053704	042526	020123	044123	
12959	053712	052517	042114	041040	
12960	053720	026105	051040	041505	
12961	053726	042040	052101	020101	
12962	053734	044507	042526	020123	
12963	053742	042522	027107	041440	
12964	053750	047117	027124	040440	
12965	053756	052106	051105	041440	
12966	053764	046517	040515	042116	
12967	053772	000			
12968	053773	065	030060	047440	EM26: .ASCIZ /500 OFFSET COMMANDS ONE AFTER THE OTHER CAUSED AN ERROR/
12969	054000	043106	042523	020124	
12970	054006	047503	046515	047101	
12971	054014	051504	047440	042516	
12972	054022	040440	052106	051105	
12973	054030	052040	042510	047440	
12974	054036	044124	051105	041440	
12975	054044	052501	042523	020104	
12976	054052	047101	042440	051122	
12977	054060	051117	000		
12978	054063	127	044522	042524	EM27: .ASCII /WRITE HEADER AND DATA CAUSED IMPROPER REGISTER CHANGE/<15><12>
12979	054070	044040	040505	042504	
12980	054076	020122	047101	020104	
12981	054104	040504	040524	041440	
12982	054112	052501	042523	020104	
12983	054120	046511	051120	050117	

12984	054126	051105	051040	043505	
12985	054134	051511	042524	020122	
12986	054142	044103	047101	042507	
12987	054150	005015			
12988	054152	047507	042117	042040	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
12989	054160	052101	020101	044507	
12990	054166	042526	020123	044127	
12991	054174	052101	051440	047510	
12992	054202	046125	020104	042502	
12993	054210	052040	042510	042522	
12994	054216	005015			
12995	054220	042522	042503	053111	.ASCIZ /RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND/
12996	054226	042105	042040	052101	
12997	054234	020101	044507	042526	
12998	054242	020123	044127	052101	
12999	054250	053440	051501	052040	
13000	054256	042510	042522	040440	
13001	054264	052106	051105	041440	
13002	054272	046517	040515	042116	
13003	054300	000			
13004					
13005	054301	127	044522	042524	EM30: .ASCIZ /WRITE HEADER AND DATA CHANGED WRITE FROM BUFFER/
13006	054306	044040	040505	042504	
13007	054314	020122	047101	020104	
13008	054322	040504	040524	041440	
13009	054330	040510	043516	042105	
13010	054336	053440	044522	042524	
13011	054344	043040	047522	020115	
13012	054352	052502	043106	051105	
13013	054360	000			
13014	054361	122	040505	020104	EM31: .ASCII /READ HEADER AND DATA CAUSED IMPROPER REGISTER CHANGE/<15><12>
13015	054366	042510	042101	051105	
13016	054374	040440	042116	042040	
13017	054402	052101	020101	040503	
13018	054410	051525	042105	044440	
13019	054416	050115	047522	042520	
13020	054424	020122	042522	044507	
13021	054432	052123	051105	041440	
13022	054440	040510	043516	006505	
13023	054446	012			
13024	054447	107	047517	020104	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
13025	054454	040504	040524	043440	
13026	054462	053111	051505	053440	
13027	054470	040510	020124	044123	
13028	054476	052517	042114	041040	
13029	054504	020105	044124	051105	
13030	054512	006505	012		
13031	054515	122	041505	044505	.ASCIZ /RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND/
13032	054522	042526	020104	040504	
13033	054530	040524	043440	053111	
13034	054536	051505	053440	040510	
13035	054544	020124	040527	020123	
13036	054552	044124	051105	020105	
13037	054560	043101	042524	020122	
13038	054566	047503	046515	047101	
13039	054574	000104			

13040	054576	051127	052111	020105	EM32: .ASCIZ /WRITE HEADER DATA, FOLLOWED BY READ HEADER AND DATA, CAUSED DATA ERROR/
13041	054604	042510	042101	051105	
13042	054612	042040	052101	026101	
13043	054620	043040	046117	047514	
13044	054626	042527	020104	054502	
13045	054634	051040	040505	020104	
13046	054642	042510	042101	051105	
13047	054650	040440	042116	042040	
13048	054656	052101	026101	041440	
13049	054664	052501	042523	020104	
13050	054672	040504	040524	042440	
13051	054700	051122	051117	000	
13052	054705	122	040505	070104	EM33: .ASCII /READ DATA CAUSED IMPROPER REGISTER CHANGE/<15><12>
13053	054712	040504	040524	041440	
13054	054720	052501	042523	020104	
13055	054726	046511	051120	050117	
13056	054734	051105	051040	043505	
13057	054742	051511	042524	020122	
13058	054750	044103	047101	042507	
13059	054756	005015			
13060	054760	047507	042117	042040	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
13061	054766	052101	020101	044507	
13062	054774	042526	020123	044127	
13063	055002	052101	051440	047510	
13064	055010	046125	020104	042502	
13065	055016	052040	042510	042522	
13066	055024	005015			
13067	055026	042522	042503	053111	.ASCIZ /RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND/
13068	055034	042105	042040	052101	
13069	055042	020101	044507	042526	
13070	055050	020123	044127	052101	
13071	055056	053440	051501	052040	
13072	055064	042510	042522	040440	
13073	055072	052106	051105	041440	
13074	055100	046517	040515	042116	
13075	055106	000			
13076	055107	122	040505	020104	EM34: .ASCIZ /READ DATA INCORRECT/
13077	055114	040504	040524	044440	
13078	055122	041516	051117	042522	
13079	055130	052103	000		
13080	055133	127	044522	042524	EM35: .ASCII /WRITE DATA COMMAND CAUSED IMPROPER REGISTER CHANGE/<15><12>
13081	055140	042040	052101	020101	
13082	055146	047503	046515	047101	
13083	055154	020104	040503	051525	
13084	055162	042105	044440	050115	
13085	055170	047522	042520	020122	
13086	055176	042522	044507	052123	
13087	055204	051105	041440	040510	
13088	055212	043516	006505	012	
13089	055217	107	047517	020104	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
13090	055224	040504	040524	043440	
13091	055232	053111	051505	053440	
13092	055240	040510	020124	044123	
13093	055246	052517	042114	041040	
13094	055254	020105	044124	051105	
13095	055262	006505	012		

13096	055265	122	041505	044505		.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER COMMAND/
13097	055272	042526	020104	040504		
13098	055300	040524	043440	053111		
13099	055306	051505	051040	043505		
13100	055314	051511	042524	020122		
13101	055322	047503	052116	047105		
13102	055330	051524	040440	052106		
13103	055336	051105	041440	046517		
13104	055344	040515	042116	000		
13105	055351	127	044522	042524	EM36:	.ASCIZ /WRITE DATA COMMAND CHANGED WRITE FROM BUFFER/
13106	055356	042040	052101	020101		
13107	055364	047503	046515	047101		
13108	055372	020104	044103	047101		
13109	055400	042507	020104	051127		
13110	055406	052111	020105	051106		
13111	055414	046517	041040	043125		
13112	055422	042506	000122			
13113	055426	042523	045505	041440	EM37:	.ASCII /SEEK COMMAND CAUSED IMPROPER REGISTER CHANGE/<15><12>
13114	055434	046517	040515	042116		
13115	055442	041440	052501	042523		
13116	055450	020104	046511	051120		
13117	055456	050117	051105	051040		
13118	055464	043505	051511	042524		
13119	055472	020122	044103	047101		
13120	055500	042507	005015			
13121	055504	047507	042117	042040		.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
13122	055512	052101	020101	044507		
13123	055520	042526	020123	044127		
13124	055526	052101	051440	047510		
13125	055534	046125	020104	042502		
13126	055542	052040	042510	042522		
13127	055550	005015				
13128	055552	042522	042503	053111		.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER SEEK COMMAND/
13129	055560	042105	042040	052101		
13130	055566	020101	044507	042526		
13131	055574	020123	042522	044507		
13132	055602	052123	051105	041440		
13133	055610	047117	042524	052116		
13134	055616	020123	043101	042524		
13135	055624	020122	042523	045505		
13136	055632	041440	046517	040515		
13137	055640	042116	000			
13138						
13139	055643	127	044522	042524	EM40:	.ASCII /WRITE CHECK CAUSED IMPROPER REGISTER CHANGE/<15><12>
13140	055650	041440	042510	045503		
13141	055656	041440	052501	042523		
13142	055664	020104	046511	051120		
13143	055672	050117	051105	051040		
13144	055700	043505	051511	042524		
13145	055706	020122	044103	047101		
13146	055714	042507	005015			
13147	055720	047507	042117	042040		.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
13148	055726	052101	020101	044507		
13149	055734	042526	020123	044127		
13150	055742	052101	051440	047510		
13151	055750	046125	020104	042502		

13152	055756	052040	042510	042522	
13153	055764	005015			
13154	055766	042522	042503	053111	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER COMMAND/
13155	055774	042105	042040	052101	
13156	056002	020101	044507	042526	
13157	056010	020123	042522	044507	
13158	056016	052123	051105	041440	
13159	056024	047117	042524	052116	
13160	056032	020123	043101	042524	
13161	056040	020122	047503	046515	
13162	056046	047101	000104		
13163	056052	047514	045503	047111	EM41: .ASCII /LOCKING OUT WRITE BY WRITE LOCK BUTTON CAUSED IMPROPER REGISTER CHANGE/
13164	056060	020107	052517	020124	
13165	056066	051127	052111	020105	
13166	056074	054502	053440	044522	
13167	056102	042524	046040	041517	
13168	056110	020113	052502	052124	
13169	056116	047117	041440	052501	
13170	056124	042523	020104	046511	
13171	056132	051120	050117	051105	
13172	056140	051040	043505	051511	
13173	056146	042524	020122	044103	
13174	056154	047101	042507	005015	
13175	056162	047507	042117	042040	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
13176	056170	052101	020101	044507	
13177	056176	042526	020123	044127	
13178	056204	052101	051440	047510	
13179	056212	046125	020104	042502	
13180	056220	052040	042510	042522	
13181	056226	005015			
13182	056230	042522	042503	053111	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER WRITES WERE LOCKED OUT/
13183	056236	042105	042040	052101	
13184	056244	020101	044507	042526	
13185	056252	020123	042522	044507	
13186	056260	052123	051105	041440	
13187	056266	047117	042524	052116	
13188	056274	020123	043101	042524	
13189	056302	020122	051127	052111	
13190	056310	051505	053440	051105	
13191	056316	020105	047514	045503	
13192	056324	042105	047440	052125	
13193	056332	000			
13194	056333	101	052124	046505	EM42: .ASCII /ATTEMPTING TO WRITE WITH WRITES LOCKED OUT CAUSED IMPROPER REGISTER CHA
13195	056340	052120	047111	020107	
13196	056346	047524	053440	044522	
13197	056354	042524	053440	052111	
13198	056362	020110	051127	052111	
13199	056370	051505	046040	041517	
13200	056376	042513	020104	052517	
13201	056404	020124	040503	051525	
13202	056412	042105	044440	050115	
13203	056420	047522	042520	020122	
13204	056426	042522	044507	052123	
13205	056434	051105	041440	040510	
13206	056442	043516	006505	012	
13207	056447	107	047517	020104	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>

13208	056454	040504	040524	043440
13209	056462	053111	051505	053440
13210	056470	040510	020124	044123
13211	056476	052517	042114	041040
13212	056504	020105	044124	051105
13213	056512	006505	012	
13214	056515	122	041505	044505
13215	056522	042526	020104	040504
13216	056530	040524	043440	053111
13217	056536	051505	051040	043505
13218	056544	051511	042524	020122
13219	056552	047503	052116	047105
13220	056560	051524	040440	052106
13221	056566	051105	040440	052124
13222	056574	046505	052120	042105
13223	056602	053440	044522	042524
13224	056610	000		
13225	056611	127	044522	044524
13226	056616	043516	053440	052111
13227	056624	020110	051127	052111
13228	056632	051505	046040	041517
13229	056640	042513	020104	052517
13230	056646	020124	044103	047101
13231	056654	042507	020104	044504
13232	056662	045523	042040	052101
13233	056670	006501	012	
13234	056673	107	047517	020104
13235	056700	040504	040524	043440
13236	056706	053111	051505	053440
13237	056714	040510	020124	040527
13238	056722	020123	047117	042040
13239	056730	051511	020113	042502
13240	056736	047506	042522	053440
13241	056744	044522	042524	024040
13242	056752	044527	044124	053440
13243	056760	044522	042524	046040
13244	056766	041517	042513	020104
13245	056774	052517	024524	005015
13246	057002	040527	020123	052101
13247	057010	042524	050115	042524
13248	057016	006504	012	
13249	057021	122	041505	044505
13250	057026	042526	020104	040504
13251	057034	040524	043440	053111
13252	057042	051505	053440	040510
13253	057050	020124	040527	020123
13254	057056	042522	042101	041040
13255	057064	041501	020113	043101
13256	057072	042524	020122	051127
13257	057100	052111	006505	012
13258	057105	050	044527	044124
13259	057112	053440	044522	042524
13260	057120	046040	041517	042513
13261	057126	020104	052517	024524
13262	057134	053440	051501	040440
13263	057142	052124	046505	052120

.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ATTEMPTED WRITE/

EM43: .ASCII /WRITING WITH WRITES LOCKED OUT CHANGED DISK DATA/<15><12>

.ASCII /GOOD DATA GIVES WHAT WAS ON DISK BEFORE WRITE (WITH WRITE LOCKED OUT)/<

.ASCII /WAS ATTEMPTED/<15><12>

.ASCII /RECEIVED DATA GIVES WHAT WAS READ BACK AFTER WRITE/<15><12>

.ASCIZ /(WITH WRITE LOCKED OUT) WAS ATTEMPTED/

13264	057150	042105	000		
13265	057153	105	040516	046102	EM44: .ASCII /ENABLING WRITES BY WRITE LOCK BUTTON CAUSED IMPROPER REGISTER CHANGE/<1
13266	057160	047111	020107	051127	
13267	057166	052111	051505	041040	
13268	057174	020131	051127	052111	
13269	057202	020105	047514	045503	
13270	057210	041040	052125	047524	
13271	057216	020116	040503	051525	
13272	057224	042105	044440	050115	
13273	057232	047522	042520	020122	
13274	057240	042522	044507	052123	
13275	057246	051105	041440	040510	
13276	057254	043516	006505	012	
13277	057261	107	047517	020104	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
13278	057266	040504	040524	043440	
13279	057274	053111	051505	053440	
13280	057302	040510	020124	044123	
13281	057310	052517	042114	041040	
13282	057316	020105	044124	051105	
13283	057324	006505	012		
13284	057327	122	041505	044505	.ASCII /RECEIVED DATA GIVES REGISTER CONTENTS AFTER WRITE LOCK BUTTON/<15><12>
13285	057334	042526	020104	040504	
13286	057342	040524	043440	053111	
13287	057350	051505	051040	043505	
13288	057356	051511	042524	020122	
13289	057364	047503	052116	047105	
13290	057372	051524	040440	052106	
13291	057400	051105	053440	044522	
13292	057406	042524	046040	041517	
13293	057414	020113	052502	052124	
13294	057422	047117	005015		
13295	057426	047105	041101	042514	.ASCII /ENABLED WRITES/
13296	057434	020104	051127	052111	
13297	057442	051505	000		
13298	057445	124	040522	051516	EM45: .ASCII /TRANSFERRING ON LAST BLOCK - CYLINDER 410, SECTOR 21, TRACK 18/<15><12>
13299	057452	042506	051122	047111	
13300	057460	020107	047117	046040	
13301	057466	051501	020124	046102	
13302	057474	041517	020113	020055	
13303	057502	054503	044514	042116	
13304	057510	051105	032040	030061	
13305	057516	020054	042523	052103	
13306	057524	051117	031040	026061	
13307	057532	052040	040522	045503	
13308	057540	030440	006470	012	
13309	057545	103	052501	042523	.ASCII /CAUSED IMPROPER REGISTER CHANGE/<15><12>
13310	057552	020104	046511	051120	
13311	057560	050117	051105	051040	
13312	057566	043505	051511	042524	
13313	057574	020122	044103	047101	
13314	057602	042507	005015		
13315	057606	047507	042117	042040	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
13316	057614	052101	020101	044507	
13317	057622	042526	020123	044127	
13318	057630	052101	051440	047510	
13319	057636	046125	020104	042502	

13320 057644 052040 042510 042522
 13321 057652 005015
 13322 057654 042522 042503 053111
 13323 057662 042105 042040 052101
 13324 057670 020101 044507 042526
 13325 057676 020123 042522 044507
 13326 057704 052123 051105 041440
 13327 057712 047117 042524 052116
 13328 057720 020123 043101 042524
 13329 057726 020122 051124 047101
 13330 057734 043123 051105 000
 13331 057741 104 052101 020101
 13332 057746 042522 042101 043040
 13333 057754 047522 020115 040514
 13334 057762 052123 041040 047514
 13335 057770 045503 026440 041440
 13336 057776 046131 047111 042504
 13337 060004 020122 030464 026060
 13338 060012 051440 041505 047524
 13339 060020 020122 030462 020054
 13340 060026 051124 041501 020113
 13341 060034 034061 005015
 13342 060040 051511 044440 020116
 13343 060046 051105 047522 000122
 13344 060054 051124 047101 043123
 13345 060062 051105 044522 043516
 13346 060070 042040 052101 020101
 13347 060076 051106 046517 047040
 13348 060104 047117 054105 051511
 13349 060112 040524 052116 051440
 13350 060120 041505 047524 020122
 13351 060126 040503 051525 042105
 13352 060134 044440 050115 047522
 13353 060142 042520 020122 005015
 13354 060150 042522 044507 052123
 13355 060156 051105 041440 040510
 13356 060164 043516 026105 043440
 13357 060172 047517 020104 040504
 13358 060200 040524 043440 053111
 13359 060206 051505 053440 040510
 13360 060214 020124 044123 052517
 13361 060222 042114 041040 020105
 13362 060230 04412. 051105 006505
 13363 060236 012
 13364 060237 122 041505 044505
 13365 060244 042526 020104 040504
 13366 060252 040524 043440 053111
 13367 060260 051505 051040 043505
 13368 060266 051511 042524 020122
 13369 060274 047503 052116 047105
 13370 060302 051524 040440 052106
 13371 060310 051105 040440 052124
 13372 060316 046505 052120 042105
 13373 060324 052040 040522 051516
 13374 060332 042506 000122
 13375

.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER TRANSFER/

EM46: .ASCII /DATA READ FROM LAST BLOCK - CYLINDER 410, SECTOR 21, TRACK 18/<15><12>

.ASCIZ /IS IN ERROR/

EM47: .ASCII /TRANSFERRING DATA FROM NONEXISTANT SECTOR CAUSED IMPROPER /<15><12>

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

.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ATTEMPTED TRANSFER/

13376	060336	051124	047101	043123	EM50: .ASCII /TRANSFERRING FROM NONEXISTANT SECTOR CAUSED DATA ERROR/<15><12>
13377	060344	051105	044522	043516	
13378	060352	043040	047522	020115	
13379	060360	047516	042516	044530	
13380	060366	052123	047101	020124	
13381	060374	042523	052103	051117	
13382	060402	041440	052501	042523	
13383	060410	020104	040504	040524	
13384	060416	042440	051122	051117	
13385	060424	005015			
13386	060426	047507	042117	042040	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
13387	060434	052101	020101	044507	
13388	060442	042526	020123	044127	
13389	060450	052101	051440	047510	
13390	060456	046125	020104	042502	
13391	060464	052040	042510	042522	
13392	060472	005015			
13393	060474	040502	020104	040504	.ASCIIZ /BAD DATA GIVES WHAT WAS IN BUFFER AFTER TRANSFER/
13394	060502	040524	043440	053111	
13395	060510	051505	053440	040510	
13396	060516	020124	040527	020123	
13397	060524	047111	041040	043125	
13398	060532	042506	020122	043101	
13399	060540	042524	020122	051124	
13400	060546	047101	043123	051105	
13401	060554	000			
13402	060555	107	053111	047111	EM51: .ASCII /GIVING ILLEGAL FUNCTION CAUSED IMPROPER REGISTER CHANGE/<15><12>
13403	060562	020107	046111	042514	
13404	060570	040507	020114	052506	
13405	060576	041516	044524	047117	
13406	060604	041440	052501	042523	
13407	060612	020104	046511	051120	
13408	060620	050117	051105	051040	
13409	060626	043505	051511	042524	
13410	060634	020122	044103	047101	
13411	060642	042507	005015		
13412	060646	047507	042117	042040	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
13413	060654	052101	020101	044507	
13414	060662	042526	020123	044127	
13415	060670	052101	051440	047510	
13416	060676	046125	020104	042502	
13417	060704	052040	042510	042522	
13418	060712	005015			
13419	060714	042522	042503	053111	.ASCIIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ILLEGAL FUNCTION IS GIVEN/
13420	060722	042105	042040	052101	
13421	060730	020101	044507	042526	
13422	060736	020123	042522	044507	
13423	060744	052123	051105	041440	
13424	060752	047117	042524	052116	
13425	060760	020123	043101	042524	
13426	060766	020122	046111	042514	
13427	060774	040507	020114	052506	
13428	061002	041516	044524	047117	
13429	061010	044440	020123	044507	
13430	061016	042526	000116		
13431	061022	051127	052111	020105	EM52: .ASCII /WRITE DATA COMMAND ON NONEXISTANT SECTOR CAUSED IMPROPER REG. CHANGE/<1

13432	061030	040504	040524	041440
13433	061036	046517	040515	042116
13434	061044	047440	020116	047516
13435	061052	042516	044530	052123
13436	061060	047101	020124	042523
13437	061066	052103	051117	041440
13438	061074	052501	042523	020104
13439	061102	046511	051120	050117
13440	061110	051105	051040	043505
13441	061116	020056	044103	047101
13442	061124	042507	005015	
13443	061130	047507	042117	042040
13444	061136	052101	020101	044507
13445	061144	042526	020123	044127
13446	061152	052101	051440	047510
13447	061160	046125	020104	042502
13448	061166	052040	042510	042522
13449	061174	005015		
13450	061176	042522	042503	053111
13451	061204	042105	042040	052101
13452	061212	020101	044507	042526
13453	061220	020123	042522	044507
13454	061226	052123	051105	041440
13455	061234	047117	042524	052116
13456	061242	020123	043101	042524
13457	061250	020122	052101	042524
13458	061256	050115	042524	020104
13459	061264	051127	052111	020105
13460	061272	040504	040524	000
13461	061277	122	040505	020104
13462	061304	042510	042101	051105
13463	061312	040440	042116	042040
13464	061320	052101	020101	043101
13465	061326	042524	020122	020101
13466	061334	042523	051101	044103
13467	061342	041440	052501	042523
13468	061350	020104	040504	040524
13469	061356	042440	051122	051117
13470	061364	000		
13471	061365	101	052124	046505
13472	061372	052120	047111	020107
13473	061400	047503	046515	047101
13474	061406	020104	044527	044124
13475	061414	044440	053116	046101
13476	061422	042111	040440	042104
13477	061430	042522	051523	041440
13478	061436	052501	042523	020104
13479	061444	046511	051120	050117
13480	061452	051105	051040	043505
13481	061460	051511	042524	020122
13482	061466	044103	047101	042507
13483	061474	005015		
13484	061476	047507	042117	042040
13485	061504	052101	020101	044507
13486	061512	042526	020123	044127
13487	061520	052101	051440	047510

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

.ASCIIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ATTEMPTED WRITE DATA/

EM53: .ASCIIZ /READ HEADER AND DATA AFTER A SEARCH CAUSED DATA ERROR/

EM54: .ASCII /ATTEMPTING COMMAND WITH INVALID ADDRESS CAUSED IMPROPER REGISTER CHANGE

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

13488	061526	046125	020104	042502	
13489	061534	052040	042510	042522	
13490	061542	005015			
13491	061544	042522	042503	053111	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER OPERATION/
13492	061552	042105	042040	052101	
13493	061560	020101	044507	042526	
13494	061566	020123	042522	044507	
13495	061574	052123	051105	041440	
13496	061602	047117	042524	052116	
13497	061610	020123	043101	042524	
13498	061616	020122	050117	051105	
13499	061624	052101	047511	000116	
13500	061632	051127	052111	047111	EM55: .ASCII /WRITING OR READING WITH EXPECTED ADDRESS OVERFLOW ERROR/<15><12>
13501	061640	020107	051117	051040	
13502	061646	040505	044504	043516	
13503	061654	053440	052111	020110	
13504	061662	054105	042520	052103	
13505	061670	042105	040440	042104	
13506	061676	042522	051523	047440	
13507	061704	042526	043122	047514	
13508	061712	020127	051105	047522	
13509	061720	006522	012		
13510	061723	103	052501	042523	.ASCII /CAUSED IMPROPER REGISTER CHANGE/<15><12>
13511	061730	020104	046511	051120	
13512	061736	050117	051105	051040	
13513	061744	043505	051511	042524	
13514	061752	020122	044103	047101	
13515	061760	042507	005015		
13516	061764	047507	042117	042040	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
13517	061772	052101	020101	044507	
13518	062000	042526	020123	044127	
13519	062006	052101	051440	047510	
13520	062014	046125	020104	042502	
13521	062022	052040	042510	042522	
13522	062030	005015			
13523	062032	042522	042503	053111	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER OPERATION/
13524	062040	042105	042040	052101	
13525	062046	020101	044507	042526	
13526	062054	020123	042522	044507	
13527	062062	052123	051105	041440	
13528	062070	047117	042524	052116	
13529	062076	020123	043101	042524	
13530	062104	020122	050117	051105	
13531	062112	052101	047511	000116	
13532	062120	040504	040524	051040	EM56: .ASCII /DATA READ WITH AN EXPECTED ADDRESS OVERFLOW ERROR IS INCORRECT/<15><12>
13533	062126	040505	020104	044527	
13534	062134	044124	040440	020116	
13535	062142	054105	042520	052103	
13536	062150	042105	040440	042104	
13537	062156	042522	051523	047440	
13538	062164	042526	043122	047514	
13539	062172	020127	051105	047522	
13540	062200	020122	051511	044440	
13541	062206	041516	051117	042522	
13542	062214	052103	005015		
13543	062220	047527	042122	047040	.ASCII /WORD NO. 1 TO 260 SHOULD BE READ, WORD NO 261 TO 266 SHOULD/<15><12>

13544	062226	027117	030440	052040
13545	062234	020117	033062	020060
13546	062242	044123	052517	042114
13547	062250	041040	020105	042522
13548	062256	042101	020054	047527
13549	062264	042122	047040	020117
13550	062272	033062	020061	047524
13551	062300	031040	033066	051440
13552	062306	047510	046125	006504
13553	062314	012		
13554	062315	102	020105	044103
13555	062322	047101	042507	000104
13556	062330	052101	042524	050115
13557	062336	044524	043516	042040
13558	062344	052101	020101	047503
13559	062352	046515	047101	020104
13560	062360	044527	044124	053440
13561	062366	047522	043516	043040
13562	062374	051117	040515	020124
13563	062402	044502	020124	040503
13564	062410	051525	042105	005015
13565	062416	046511	051120	050117
13566	062424	051105	051040	043505
13567	062432	051511	042524	020122
13568	062440	044103	047101	042507
13569	062446	005015		
13570	062450	047507	042117	042040
13571	062456	052101	020101	044507
13572	062464	042526	020123	044127
13573	062472	052101	051440	047510
13574	062500	046125	020104	042502
13575	062506	052040	042510	042522
13576	062514	005015		
13577	062516	042522	042503	053111
13578	062524	042105	042040	052101
13579	062532	020101	044507	042526
13580	062540	020123	042522	044507
13581	062546	052123	051105	041440
13582	062554	047117	042524	052116
13583	062562	020123	043101	042524
13584	062570	020122	052101	042524
13585	062576	050115	042524	020104
13586	062604	040504	040524	052040
13587	062612	040522	051516	042506
13588	062620	000122		
13589				
13590	062622	052101	042524	050115
13591	062630	044524	043516	052040
13592	062636	020117	047515	044504
13593	062644	054506	051040	043505
13594	062652	051511	042524	020122
13595	062660	052504	044522	043516
13596	062666	040440	020116	050117
13597	062674	051105	052101	047511
13598	062702	020116	040503	051525
13599	062710	042105	044440	050115

.ASCIZ /BE CHANGED/
 EM57: .ASCII /ATTEMPTING DATA COMMAND WITH WRONG FORMAT BIT CAUSED/<15><12>
 .ASCII /IMPROPER REGISTER CHANGE/<15><12>
 .ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
 .ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ATTEMPTED DATA TRANSFER/
 EM60: .ASCII /ATTEMPTING TO MODIFY REGISTER DURING AN OPERATION CAUSED IMPROPER/<15><

13600	062716	047522	042520	006522	
13601	062724	012			
13602	062725	122	043505	051511	.ASCII /REGISTER CHANGE. GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
13603	062732	042524	020122	044103	
13604	062740	047101	042507	020056	
13605	062746	047507	042117	042040	
13606	062754	052101	020101	044507	
13607	062762	042526	020123	044127	
13608	062770	052101	051440	047510	
13609	062776	046125	020104	042502	
13610	063004	052040	042510	042522	
13611	063012	005015			
13612	063014	042522	042503	053111	.ASCII /RECEIVED DATA GIVES REGISTER CONTENTS AFTER OPERATION WAS ATTEMPTED/<15
13613	063022	042105	042040	052101	
13614	063030	020101	044507	042526	
13615	063036	020123	042522	044507	
13616	063044	052123	051105	041440	
13617	063052	047117	042524	052116	
13618	063060	020123	043101	042524	
13619	063066	020122	050117	051105	
13620	063074	052101	047511	020116	
13621	063102	040527	020123	052101	
13622	063110	042524	050115	042524	
13623	063116	006504	012		
13624	063121	115	042117	043111	.ASCIIZ /MODIFYING REG GIVES ADDRESS OF REGISTER BEING MODIFIED WHICH CAUSED ERR
13625	063126	044531	043516	051040	
13626	063134	043505	043440	053111	
13627	063142	051505	040440	042104	
13628	063150	042522	051523	047440	
13629	063156	020106	042522	044507	
13630	063164	052123	051105	041040	
13631	063172	044505	043516	046440	
13632	063200	042117	043111	042511	
13633	063206	020104	044127	041511	
13634	063214	020110	040503	051525	
13635	063222	042105	042440	051122	
13636	063230	051117	000		
13637	063233	122	041510	030523	EM61: .ASCIIZ /RHCS1 HAS SOME INCORRECT STATUS BITS = 1, OR - 0/
13638	063240	044040	051501	051440	
13639	063246	046517	020105	047111	
13640	063254	047503	051122	041505	
13641	063262	020124	052123	052101	
13642	063270	051525	041040	052111	
13643	063276	020123	020075	026061	
13644	063304	047440	020122	020075	
13645	063312	000060			
13646	063314	044122	051504	020061	EM62: .ASCIIZ /RHDS1 HAS SOME INCORRECT STATUS BITS = 1, OR - 0/
13647	063322	040510	020123	047523	
13648	063330	042515	044440	041516	
13649	063336	051117	042522	052103	
13650	063344	051440	040524	052524	
13651	063352	020123	044502	051524	
13652	063360	036440	030440	020054	
13653	063366	051117	036440	030040	
13654	063374	000			
13655	063375	122	042110	030523	EM63: .ASCIIZ /RHDS1 CONTENTS DURING COMMAND WERE IN ERROR/

13656 063402 041440 047117 042524
13657 063410 052116 020123 052504
13658 063416 044522 043516 041440
13659 063424 046517 040515 042116
13660 063432 053440 051105 020105
13661 063440 047111 042440 051122
13662 063446 051117 000
13663 063451 122 041505 046101
13664 063456 041111 040522 042524
13665 063464 041440 046517 040515
13666 063472 042116 041440 052501
13667 063500 042523 020104 046511
13668 063506 051120 050117 051105
13669 063514 051040 043505 051511
13670 063522 042524 020122 044103
13671 063530 047101 042507 005015
13672 063536 047507 042117 042040
13673 063544 052101 020101 044507
13674 063552 042526 020123 044127
13675 063560 052101 051440 047510
13676 063566 046125 020104 042502
13677 063574 052040 042510 042522
13678 063602 005015
13679 063604 042522 042503 053111
13680 063612 042105 042040 052101
13681 063620 020101 044507 042526
13682 063626 020123 042522 044507
13683 063634 052123 051105 041440
13684 063642 047117 042524 052116
13685 063650 020123 043101 042524
13686 063656 020122 047503 046515
13687 063664 047101 000104
13688 063670 047111 042524 051122
13689 063676 050125 020124 040506
13690 063704 046111 047111 000107
13691 063712 042510 042101 051105
13692 063720 040440 042116 042040
13693 063726 052101 020101 047503
13694 063734 046515 047101 020104
13695 063742 047506 020122 042510
13696 063750 042101 051440 046105
13697 063756 041505 044524 047117
13698 063764 052040 051505 020124
13699 063772 040503 051525 042105
13700 064000 005015
13701 064002 051105 047522 020122
13702 064010 020055 044122 051504
13703 064016 020124 044507 042526
13704 064024 020123 051124 041501
13705 064032 020113 042502 047111
13706 064040 020107 051127 052111
13707 064046 042524 020116 051117
13708 064054 051040 040505 020104
13709 064062 047117 041440 046131
13710 064070 030040 020054 041523
13711 064076 051124 030040 005015

EM64: .ASCII /RECALIBRATE COMMAND CAUSED IMPROPER REGISTER CHANGE/<15><12>

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

.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER COMMAND/

EM65: .ASCIZ /INTERRUPT FAILING/

EM66: .ASCII /HEADER AND DATA COMMAND FOR HEAD SELECTION TEST CAUSED/<15><12>

.ASCII /ERROR - RHDST GIVES TRACK BEING WRITTEN OR READ ON CYL 0, SCTR 0/<15><1

13712	064104	042522	042101	044040	EM67: .ASCII /READ HEADER AND DATA ERROR IN HEAD SELECTION TEST./<12><15>
13713	064112	040505	042504	020122	
13714	064120	047101	020104	040504	
13715	064126	040524	042440	051122	
13716	064134	051117	044440	020116	
13717	064142	042510	042101	051440	
13718	064150	046105	041505	044524	
13719	064156	047117	052040	051505	
13720	064164	027124	006412		
13721	064170	044506	051522	020124	.ASCII /FIRST FOUR WORD NUMBERS ARE THE HEADER./<12><15>
13722	064176	047506	051125	053440	
13723	064204	051117	020104	052516	
13724	064212	041115	051105	020123	
13725	064220	051101	020105	044124	
13726	064226	020105	042510	042101	
13727	064234	051105	005056	015	
13728	064241	127	051117	020104	.ASCII /WORD NUMBERS 5 TO 260 ARE DATA WORDS./<12><15>
13729	064246	052516	041115	051105	
13730	064254	020123	020065	047524	
13731	064262	031040	030066	040440	
13732	064270	042522	042040	052101	
13733	064276	020101	047527	042122	
13734	064304	026123	006412		
13735	064310	047101	020104	047111	.ASCIZ /AND IN DATA WORDS BITS 4,5,6,7,8 GIVE TRACK NUMBER./
13736	064316	042040	052101	020101	
13737	064324	047527	042122	020123	
13738	064332	044502	051524	032040	
13739	064340	032454	033054	033454	
13740	064346	034054	043440	053111	
13741	064354	020105	051124	041501	
13742	064362	020113	052516	041115	
13743	064370	051105	000056		
13744					
13745	064374	042522	042101	044040	EM70: .ASCII /READ HEADER AND DATA ERROR IN/<15><12>
13746	064402	040505	042504	020122	
13747	064410	047101	020104	040504	
13748	064416	040524	042440	051122	
13749	064424	051117	044440	006516	
13750	064432	012			
13751	064433	104	043111	042506	.ASCII /DIFFERENCE LINE TEST/<15><12>
13752	064440	042522	041516	020105	
13753	064446	044514	042516	052040	
13754	064454	051505	006524	012	
13755	064461	127	051117	020104	.ASCII /WORD NOS 1-4 GIVE HEADER/<15><12>
13756	064466	047516	020123	026461	
13757	064474	020064	044507	042526	
13758	064502	044040	040505	042504	
13759	064510	006522	012		
13760	064513	127	051117	020104	.ASCIZ /WORD NOS 5-260 GIVE DATA WHICH IS THE CYLINDER ADDRESS/
13761	064520	047516	020123	026465	
13762	064526	033062	020060	044507	
13763	064534	042526	042040	052101	
13764	064542	020101	044127	041511	
13765	064550	020110	051511	052040	
13766	064556	042510	041440	046131	
13767	064564	047111	042504	020122	

13768	064572	042101	051104	051505	
13769	064600	000123			
13770	064602	047506	041522	047111	EM71: .ASCII /FORCING OPI BY 3 INDEX PULSES/<15><12>
13771	064610	020107	050117	020111	
13772	064616	054502	031440	044440	
13773	064624	042116	054105	050040	
13774	064632	046125	042523	006523	
13775	064640	012			
13776	064641	103	052501	042523	.ASCII /CAUSED IMPROPER REGISTER CHANGE/<15><12>
13777	064646	020104	046511	051120	
13778	064654	050117	051105	051040	
13779	064662	043505	051511	042524	
13780	064670	020122	044103	047101	
13781	064676	042507	005015		
13782	064702	047507	042117	042040	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
13783	064710	052101	020101	044507	
13784	064716	042526	020123	044127	
13785	064724	052101	051440	047510	
13786	064732	046125	020104	042502	
13787	064740	052040	042510	042522	
13788	064746	005015			
13789	064750	042522	042503	053111	.ASCIIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER 3 INDEX PULSES/
13790	064756	042105	042040	052101	
13791	064764	020101	044507	042526	
13792	064772	020123	042522	044507	
13793	065000	052123	051105	041440	
13794	065006	047117	042524	052116	
13795	065014	020123	043101	042524	
13796	065022	020122	020063	047111	
13797	065030	042504	020130	052520	
13798	065036	051514	051505	000	
13799	065043	127	044510	042514	EM72: .ASCII /WHILE USING UNIBUS B/<15><12>
13800	065050	052440	044523	043516	
13801	065056	052440	044516	052502	
13802	065064	020123	006502	012	
13803	065071	122	040505	020104	.ASCII /READ DATA CAUSED IMPROPER REGISTER CHANGE/<15><12>
13804	065076	040504	040524	041440	
13805	065104	052501	042523	020104	
13806	065112	046511	051120	050117	
13807	065120	051105	051040	043505	
13808	065126	051511	042524	020122	
13809	065134	044103	047101	042507	
13810	065142	005015			
13811	065144	047507	042117	042040	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
13812	065152	052101	020101	044507	
13813	065160	042526	020123	044127	
13814	065166	052101	051440	047510	
13815	065174	046125	020104	042502	
13816	065202	052040	042510	042522	
13817	065210	005015			
13818	065212	042522	042503	053111	.ASCIIZ /RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND/
13819	065220	042105	042040	052101	
13820	065226	020101	044507	042526	
13821	065234	020123	044127	052101	
13822	065242	053440	051501	052040	
13823	065250	042510	042522	040440	

13824	065256	072106	051105	04440	
13825	065264	0651	040515	042116	
13826	065272	07			
13827	065273	27	044510	042514	EM73: .ASCII /WHILE USING UNIBUS B/<15><12>
13828	065300	440	044523	043516	
13829	065306	052440	044516	052502	
13830	065314	020123	006502	012	
13831	065321	122	040505	020104	.ASCIZ /READ DATA INCORRECT/
13832	065326	040504	040524	044440	
13833	065334	041516	051117	042522	
13834	065342	052103	000		
13835	065345	127	044510	042514	EM74: .ASCII /WHILE USING UNIBUS B/<15><12>
13836	065352	052440	044523	043516	
13837	065360	052440	044516	052502	
13838	065366	020123	006502	012	
13839	065373	127	044522	042524	.ASCII /WRITE DATA COMMAND CAUSED IMPROPER REGISTER CHANGE/<15><12>
13840	065400	042040	052101	020101	
13841	065406	047503	046515	047101	
13842	065414	020104	040503	051525	
13843	065422	042105	044440	050115	
13844	065430	047522	042520	020122	
13845	065436	042522	044507	052123	
13846	065444	051105	041440	040510	
13847	065452	043516	006505	012	
13848	065457	107	047517	020104	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
13849	065464	040504	040524	043440	
13850	065472	053111	051505	053440	
13851	065500	040510	020124	044123	
13852	065506	052517	042114	041040	
13853	065514	020105	044124	051105	
13854	065522	006505	012		
13855	065525	122	041505	044505	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER COMMAND/
13856	065532	042526	020104	040504	
13857	065540	040524	043440	053111	
13858	065546	051505	051040	043505	
13859	065554	051511	042524	020122	
13860	065562	047503	052116	047105	
13861	065570	051524	040440	052106	
13862	065576	051105	041440	046517	
13863	065604	040515	042116	000	
13864	065611	127	044510	042514	EM75: .ASCII /WHILE USING UNIBUS B/<15><12>
13865	065616	052440	044523	043516	
13866	065624	052440	044516	052502	
13867	065632	020123	006502	012	
13868	065637	127	044522	042524	.ASCIZ /WRITE DATA COMMAND CHANGED WRITE FROM BUFFER/
13869	065644	042040	052101	020101	
13870	065652	047503	046515	047101	
13871	065660	020104	044103	047101	
13872	065666	042507	020104	051127	
13873	065674	052111	020105	051106	
13874	065702	046517	041040	043125	
13875	065710	042506	000122		
13876	065714	044127	046111	020105	EM76: .ASCII /WHILE USING UNIBUS B/<15><12>
13877	065722	051525	047111	020107	
13878	065730	047125	041111	051525	
13879	065736	041040	005015		

13880	065742	051127	052111	020105		.ASCII /WRITE CHECK CAUSED IMPROPER REGISTER CHANGE/<15><12>
13881	065750	044103	041505	020113		
13882	065756	040503	051525	042105		
13883	065764	044440	050115	047522		
13884	065772	J42520	020122	042522		
13885	066000	044507	052123	051105		
13886	066006	041440	040510	043516		
13887	066014	006505	012			
13888	066017	107	047517	020104		.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
13889	066024	040504	040524	043440		
13890	066032	053111	051505	053440		
13891	066040	040510	020124	044123		
13892	066046	052517	042114	041040		
13893	066054	020105	044124	051105		
13894	066062	006505	012			
13895	066065	122	041505	044505		.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER COMMAND/
13896	066072	042526	020104	040504		
13897	066100	040524	043440	053111		
13898	066106	051505	051040	043505		
13899	066114	051511	042524	020122		
13900	066122	047503	052116	047105		
13901	066130	051524	040440	052106		
13902	066136	051105	041440	046517		
13903	066144	040515	042116	000		
13904	066151	120	042522	047514	EM77:	.ASCII /PRELOADING 'RHCC' PRIOR TO DOING NEXT TEST DOES NOT PRODUCE CORRECT RES
13905	066156	042101	047111	020107		
13906	066164	051047	041510	023503		
13907	066172	050040	044522	051117		
13908	066200	052040	020117	047504		
13909	066206	047111	020107	042516		
13910	066214	052130	052040	051505		
13911	066222	020124	047504	051505		
13912	066230	047040	052117	050040		
13913	066236	047522	052504	042503		
13914	066244	041440	051117	042522		
13915	066252	052103	051040	051505		
13916	066260	046125	006524	012		
13917	066265	124	042510	042522		.ASCIZ /THEREFORE NEXT TEST RESULTS ARE SUSPECT WITH REGARD TO 'RHCC' CONTENTS/
13918	066272	047506	042522	047040		
13919	066300	054105	020124	042524		
13920	066306	052123	051040	051505		
13921	066314	046125	051524	040440		
13922	066322	042522	051440	051525		
13923	066330	042520	052103	053440		
13924	066336	052111	020110	042522		
13925	066344	040507	042122	052040		
13926	066352	020117	051047	041510		
13927	066360	023503	041440	047117		
13928	066366	042524	052116	000123		
13929						
13930	066374	042101	051104	051505	EM100:	.ASCIZ /ADDRESS PLUG CHANGE RESULTED IN BAD REGISTER DATA/
13931	066402	020123	046120	043525		
13932	066410	041440	040510	043516		
13933	066416	020105	042522	052523		
13934	066424	052114	042105	044440		
13935	066432	020116	040502	020104		

13936	066440	042522	044507	052123	
13937	066446	051105	042040	052101	
13938	066454	000101			
13939	066456	047125	052111	042040	EM101: .ASCIZ /UNIT DID NOT GO OFFLINE WHEN ADDRESS PLUG REMOVED/
13940	066464	042111	047040	052117	
13941	066472	043440	020117	043117	
13942	066500	046106	047111	020105	
13943	066506	044127	047105	040440	
13944	066514	042104	042522	051523	
13945	066522	050040	052514	020107	
13946	066530	042522	047515	042526	
13947	066536	000104			
13948	066540	047125	052111	047040	EM102: .ASCIZ /UNIT NOT AVAILABLE AFTER ADDRESS PLUG REPLACED/
13949	066546	052117	040440	040526	
13950	066554	046111	041101	042514	
13951	066562	040440	052106	051105	
13952	066570	040440	042104	042522	
13953	066576	051523	050040	052514	
13954	066604	020107	042522	046120	
13955	066612	041501	042105	000	
13956	066617	122	043505	051511	EM103: .ASCIZ /REGISTER CONTENTS INCORRECT BEFORE A DIAG MODE SEEK/
13957	066624	042524	020122	047503	
13958	066632	052116	047105	051524	
13959	066640	044440	041516	051117	
13960	066646	042522	052103	041040	
13961	066654	043105	051117	020105	
13962	066662	020101	044504	043501	
13963	066670	046440	042117	020105	
13964	066676	042523	045505	000	
13965	066703	122	043505	051511	EM104: .ASCIZ /REGISTER CONTENTS INCORRECT AFTER A DIAG MODE SEEK/
13966	066710	042524	020122	047503	
13967	066716	052116	047105	051524	
13968	066724	044440	041516	051117	
13969	066732	042522	052103	040440	
13970	066740	052106	051105	040440	
13971	066746	042040	040511	020107	
13972	066754	047515	042504	051440	
13973	066762	042505	000113		
13974	066766	050040	047522	051107	NOUSE: .ASCIZ / PROGRAMMABLE-DRIVE WILL NOT BE USED/
13975	066774	046501	040515	046102	
13976	067002	026505	051104	053111	
13977	067010	020105	044527	046114	
13978	067016	047040	052117	041040	
13979	067024	020105	051525	042105	
13980	067032	000			

13981										
13982										
13983	067033	106	052101	046101	CPHALT: .ASCII /FATAL ERROR - SEE DOCUMENT LISTING/<15><12><15><12>					
13984	067040	042440	051122	051117						
13985	067046	026440	051440	042505						
13986	067054	042040	041517	046525						
13987	067062	047105	020124	044514						
13988	067070	052123	047111	006507						
13989	067076	006412	012							
13990	067101	124	042510	041440	.ASCII /THE CONTROLLER OR DEVICE HAS GONE OFFLINE, LOST/<15><12>					
13991	067106	047117	051124	046117						
13992	067114	042514	020122	051117						
13993	067122	042040	053105	041511						
13994	067130	020105	040510	020123						
13995	067136	047507	042516	047440						
13996	067144	043106	044514	042516						
13997	067152	020054	047514	052123						
13998	067160	005015								
13999	067162	051047	040505	054504	.ASCII /'READY', BECOME UNAVAILABLE, OR HAS STATUS BITS/<15><12>					
14000	067170	026047	041040	041505						
14001	067176	046517	020105	047125						
14002	067204	053101	044501	040514						
14003	067212	046102	026105	047440						
14004	067220	020122	040510	020123						
14005	067226	052123	052101	051525						
14006	067234	041040	052111	006523						
14007	067242	012								
14008	067243	127	044510	044103	.ASCIIZ /WHICH CANNOT BE CLEARED/					
14009	067250	041440	047101	047516						
14010	067256	020124	042502	041440						
14011	067264	042514	051101	042105						
14012	067272	000								
14013										
14014										
14015										
14016										
14017	067273	120	020103	020040	DH1: .ASCII /PC TEST WAT BIT REG REG RHCS1/<15><12>					
14018	067300	020040	052040	051505						
14019	067306	020124	020040	053440						
14020	067314	052101	020040	020040						
14021	067322	041040	052111	020040						
14022	067330	020040	051040	043505						
14023	067336	020040	020040	051040						
14024	067344	043505	020040	020040						
14025	067352	051040	041510	030523						
14026	067360	005015								
14027	067362	020040	020040	020040	.ASCIIZ / NO PC EXPECT ADDRESS CONTENT CONTENT/					
14028	067370	020040	047516	020040						
14029	067376	020040	020040	041520						
14030	067404	020040	020040	020040						
14031	067412	054105	042520	052103						
14032	067420	020040	042101	051104						
14033	067426	051505	020123	047503						
14034	067434	052116	047105	020124						
14035	067442	047503	052116	047105						
14036	067450	000124								

14149	070620	020040	051040	041505						
14150	070626	042126	020040	044440						
14151	070634	046114	043505	006514						
14152	070642	012								
14153	070643	040	020040	020040		.ASCIZ /	NO	ADDRESS DATA	DATA	FUNCTN/
14154	070650	020040	047040	020117						
14155	070656	020040	020040	040440						
14156	070664	042104	042522	051523						
14157	070672	042040	052101	020101						
14158	070700	020040	042040	052101						
14159	070706	020101	020040	043040						
14160	070714	047125	052103	000116						
14161										
14162	070722	041520	020040	020040	DH60:	.ASCII /PC	TEST	REG	GOOD	RECVD
14163	070730	020040	042524	052123						MODFING/<15><12>
14164	070736	020040	020040	042522						
14165	070744	020107	020040	020040						
14166	070752	047507	042117	020040						
14167	070760	020040	042522	053103						
14168	070766	020104	020040	047515						
14169	070774	043104	047111	006507						
14170	071002	012								
14171	071003	040	020040	020040		.ASCIZ /	NO	ADDRESS DATA	DATA	REG/
14172	071010	020040	047040	020117						
14173	071016	020040	020040	040440						
14174	071024	042104	042522	051523						
14175	071032	042040	052101	020101						
14176	071040	020040	042040	052101						
14177	071046	020101	020040	051040						
14178	071054	043505	000							
14179	071057	120	020103	020040	DH61:	.ASCII /PC	TEST	PC OF	RHCS1/<15><12>	
14180	071064	020040	052040	051505						
14181	071072	020124	020040	050040						
14182	071100	020103	043117	020040						
14183	071106	051040	041510	030523						
14184	071114	005015								
14185	071116	020040	020040	020040		.ASCIZ /	NO	JSR	WAS/	
14186	071124	020040	047516	020040						
14187	071132	020040	020040	051512						
14188	071140	020122	020040	020040						
14189	071146	040527	000123							
14190	071152	041520	020040	020040	DH62:	.ASCII /PC	TEST	PC OF	RHDS1/<15><12>	
14191	071160	020040	042524	052123						
14192	071166	020040	020040	041520						
14193	071174	047440	020106	020040						
14194	071202	044122	051504	006461						
14195	071210	012								
14196	071211	040	020040	020040		.ASCIZ /	NO	JSR	WAS/	
14197	071216	020040	047040	020117						
14198	071224	020040	020040	045040						
14199	071232	051123	020040	020040						
14200	071240	053440	051501	000						
14201	071245	120	020103	020040	DH65:	.ASCII /PC	TEST	CONT	CONT	CONT
14202	071252	020040	052040	051505						/<15><12>
14203	071260	020124	020040	041440						
14204	071266	047117	020124	020040						

14261 071724 002362 002360 002404
14262 071732 002364 000000
14263 071736 001116 004604 002362 DT26: .WORD \$ERRPC,TSTNM,CS1,CS2,DS1,ER1,ER2,ER3,0
14264 071744 002360 002404 002364
14265 071752 002370 002376 000000
14266 071760 001116 004604 004602 DT30: .WORD \$ERRPC,TSTNM,ERWORD,\$GDDAT,\$BDDAT,0
14267 071766 001124 001126 000000
14268
14269 071774 001116 004604 004600 DT51: .WORD \$ERRPC,TSTNM,REGADR,\$GDDAT,\$BDDAT,ILLEGL,0
14270 072002 001124 001126 002464
14271 072010 000000
14272
14273 072012 001116 004604 004600 DT60: .WORD \$ERRPC,TSTNM,REGADR,\$GDDAT,\$BDDAT,\$BDADR,0
14274 072020 001124 001126 001122
14275 072026 000000
14276 072030 001116 004604 041560 DT61: .WORD \$ERRPC,TSTNM,PCJSR,\$BDADR,0
14277 072036 001122 000000
14278 072042 001116 004604 041560 DT62: .WORD \$ERRPC,TSTNM,PCJSR,\$BDADR,0
14279 072050 001122 000000
14280 072054 001116 004604 002362 DT65: .WORD \$ERRPC,TSTNM,CS1,AS,DS1,0
14281 072062 002400 002404 000000
14282 072070 001116 004604 002364 DT66: .WORD \$ERRPC,TSTNM,ER1,ER2,ER3,CS1,CS2,0
14283 072076 002370 002376 002362
14284 072104 002360 000000
14285
14286 07211C 001116 004604 041560 DT77: .WORD \$ERRPC,TSTNM,PCJSR,REGADR,\$GDDAT,\$BDDAT,0
14287 072116 004600 001124 001126
14288 072124 000000
14289
14290 072126 000 000 000 DF1: .BYTE 0,0,0,0,0,0,0
14291 072131 000 000 000
14292 072134 000
14293 072135 000 000 000 DF4: .BYTE 0,0,0,0,0,1,0
14294 072140 000 000 001
14295 072143 000
14296 072144 000 000 000 DF5: .BYTE 0,0,0,0,0
14297 072147 000 000
14298 072151 000 000 000 DF6: .BYTE 0,0,0,0
14299 072154 000
14300 072155 000 000 000 DF7: .BYTE 0,0,0
14301
14302 072160 000 000 000 DF10: .BYTE 0,0,0,0,0,0,0
14303 072163 000 000 000
14304 072166 000
14305
14306 072167 000 000 000 DF26: .BYTE 0,0,0,0,0,0,0,0
14307 072172 000 000 000
14308 072175 000 000
14309
14310 072177 000 000 000 DF30: .BYTE 0,0,0,0,0
14311 072202 000 000
14312
14313 072204 000 000 000 DF51: .BYTE 0,0,0,0,0,0,0
14314 072207 000 000 000
14315
14316 072212 000 000 000 DF60: .BYTE 0,0,0,0,0,0,0

14317	072215	000	000	000			
14318	072220	000	000	000	DF61:	.BYTE	0,0,0,0
14319	072223	000					
14320	072224	000	000	000	DF62:	.BYTE	0,0,0,0
14321	072227	000					
14322	072230	000	000	000	DF65:	.BYTE	0,0,0,0,0
14323	072233	000	000				
14324	072235	000	000	000	DF66:	.BYTE	0,0,0,0,0,0,0,0
14325	072240	000	000	000			
14326	072243	000	000				
14327							
14328	072245	000	000	000	DF77:	.BYTE	0,0,0,0,0,0
14329	072250	000	000	000			
14330							
14331	072254					.EVEN	
14332							
14333							
14334	000001					.END	

CZRJID, RPO4/5/6 FCTNL CTLR1		MACY11		30A(1052)		25-MAY-79		10:30		PAGE 300		N 7		SEQ 0298	
CZRJID.P11 28-MAR-79 09:03		CROSS REFERENCE TABLE -- USER SYMBOLS													
DF5	072144	989	1074	1086	1100	1114	1126	1134	1141	1151	1160	1171	1194	1215	
		1233	1249	1267	1278	1292	1304	1329	1342	1364	1404	1423	1435	1460	
		1503	1515	1581	1591	1609	1629	1648	14296#						
DF51	072204	1390	14313#												
DF6	072151	999	14298#												
DF60	072212	1477	14316#												
DF61	072220	1486	14318#												
DF62	072224	1495	14320#												
DF65	072230	1527	14322#												
DF66	072235	1545	14324#												
DF7	072155	1007	14300#												
DF77	072245	1639	14328#												
DH1	067273	936	948	960	14017#										
DH10	070071	1012	1026	1040	1056	14088#									
DH26	070250	1175	1653	1665	1677	1689	14107#								
DH30	070447	1199	1221	1237	1254	1315	1349	1373	1409	1445	1554	1565	1596	1615	
		14129#													
DH4	067452	973	14037#												
DH5	067613	984	1069	1081	1095	1109	1121	1132	1139	1149	1158	1169	1192	1213	
		1231	1247	1265	1276	1290	1302	1327	1340	1362	1402	1421	1433	1458	
		1501	1513	1576	1589	1607	1627	1643	14055#						
DH51	070563	1384	14144#												
DH6	067732	995	14070#												
DH60	070722	1471	14162#												
DH61	071057	1481	14179#												
DH62	071152	1490	14190#												
DH65	071245	1521	14201#												
DH66	071370	1537	14216#												
DH7	070031	1004	14081#												
DH77	071466	1633	14228#												
DIGB =	000004	1764#													
DISPLA	001142	883#	2161*	2169*	11833*	12277*									
DISPRE	000174	782#	2169												
DLT =	100000	1735#													
DL64 =	000020	1766#													
DMD =	000001	1799#	3162	3181	3252	3440	4528	4547	4750	4769	10969				
DOG	017420	4513	4537#												
DPR =	000400	1770#	2898	3095	3315	4681	4873	5160	5191	5441	5561	5843	5990	6152	
		6348	6511	6700	6852	7049	7212	7426	7614	7859	8026	8273	8442	8693	
		8853	8997	9249	9860	10107	10372	11165							
DRY =	000200	1769#	2900	3095	3319	4655	4855	5197	5415	5535	5663	5686	8567	8979	
		9192	9231	9474	9964	10225	11165								
DST	002366	2000#	12398												
DSWR =	177570	680#	882	2160											
DS1	002404	2007#	3315	3319	10972	10984	12366	14260	14263	14280					
DT	002406	2008#	2684*	4055	12470										
DTE =	010000	1792#													
DTSY =	001000	1805#													
DT1	071620	942	954	965	14247#										
DT10	071716	1019	1033	1047	1063	14260#									
DT26	071736	1182	1659	1671	1683	1695	14263#								
DT30	071760	1203	1222	1238	1255	1316	1350	1374	1410	1446	1555	1566	1597	1616	
		14266#													
DT4	071640	978	14250#												
DT5	071660	988	1073	1085	1099	1113	1125	1133	1140	1150	1159	1170	1193	1214	
		1232	1248	1266	1277	1291	1303	1328	1341	1363	1403	1422	1434	1459	

CZRJIDO, RP04/5/6 FCTNL CTLR1
CZRJID.P11 28-MAR-79 09:03

MACY11 30A(1052) 25-MAY-79 10:30 PAGE 303
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0301

		9262	9632	9635	9638	9728	9731	9734	9873	9876	9879	10120	10123	10126
		10385	10388	10391	10662	11070#								
FINACC	004664	2068#	11226*											
FINALA	004662	2067#	11227*											
FIRST	004736	2095#	2183	2220*										
FISH	017454	4534	4557#											
FLHEAD	041350	5733	5755	7721	7738	7759	7772	7936	8134	8151	8172	8185	8355	8581
		8600	8760	9065	9323	9486	9553	11013#						
FMT22 =	010000	1881#	4295	4467	5594	5781	5782	5783	5930	5931	5932	6092	6093	6094
		6286	6287	6288	6451	6452	6453	6638	6639	6640	6792	6793	6794	6987
		6988	6989	7152	7153	7154	7363	7364	7365	7553	7554	7555	7798	7799
		7800	7964	7965	7966	8211	8212	8213	8382	8383	8384	8629	8630	8631
		8792	8793	8794	9095	9096	9097	9353	9354	9355	9586	9587	9588	9681
		9682	9683	9799	9800	9801	10048	10049	10050	10308	10309	10310	10491	10492
		10493	10607	10608	10609	10743	10744	10745						
FUTABL	002422	2020#												
GNS =	***** U	781	2191	2195	2200	2204	2208	2213	2217	2272	2314	2329	2336	2390
		2417	2421	2425	2429	2486	2492	2498	2575	2579	2617	2623	2629	2662
		2668	2674	2704	2708	2712	2718	2855	2851	2887	2893	3276	3282	3302
		3308	3552	3556	3570	3576	3580	3607	3613	3649	3653	4150	4161	4984
		4988	5179	5245	5251	7447	7455	10540	10546	10807	10813	10870	10876	11524
		11530	11537	11541	11545	11552	11556	11682	11688	11705	11711	11717	11723	11729
		11733	11737	11741	11747	11753	11764	12332	12340	12348	12356	12364	12372	12380
		12388	12396	12404	12412	12420	12428	12436	12444	12452	12460	12468	12476	12645
		12646	12647	12648	12649	12651	12653	12654	12655	12656	12657			
GO =	000001	1749#	2960	3065	3473	3631	3710	3825	3961	4253	4266	4425	4438	4645
		4664	4845	4864	5006	5121	5141	5207	5267	5404	5424	5524	5544	5655
		5678	5812	5834	5961	5981	6123	6143	6317	6339	6482	6502	6669	6691
		6823	6843	7018	7040	7183	7203	7395	7417	7585	7605	7828	7850	7995
		8017	8242	8264	8413	8433	8557	8660	8684	8824	8844	8967	8988	9117
		9181	9219	9240	9375	9463	9617	9712	9831	9851	9955	10078	10098	10216
		10343	10363	10513	10631	10759	10977	11251						
GRV =	000010	1765#												
GTSWR =	104406	2227	12651#											
HCE =	000200	1787#												
HCI =	002000	1879#	4299	4471	5594	5783	5932	6094	6288	6453	6640	6794	6989	7154
		7365	7555	7800	7966	8213	8384	8631	8794	9097	9355	9588	9683	9801
		10050	10310	10493	10609	10745								
HCRC =	000400	1788#												
HT =	000011	672#	11933	11974										
HT. =	003323	2968#	2973	3075#	3080	3482#	3487	3640#	3645	3718#	3723	3833#	3838	3969#
		3974	4274#	4279	4446#	4451	4653#	4658	4853#	4858	5014#	5019	5276#	5281
		5413#	5418	5533#	5538	5661#	5666	5684#	5689	5823#	5828	5970#	5975	6132#
		6137	6328#	6333	6491#	6496	6680#	6685	6832#	6837	7029#	7034	7192#	7197
		7406#	7411	7594#	7599	7839#	7844	8006#	8011	8253#	8258	8422#	8427	8565#
		8570	8673#	8678	8833#	8838	8977#	8982	9124#	9129	9190#	9195	9229#	9234
		9382#	9387	9472#	9477	9624#	9629	9719#	9724	9840#	9845	9962#	9967	10087#
		10092	10223#	10228	10352#	10357	10521#	10526	10638#	10643	10766#	10771		
IAE =	002000	1790#												
IE =	000100	1750#	2744	2783	4266	4438	4645	4664	4668	4669	4845	4864	5145	5146
		5404	5424	5428	5429	5524	5544	5548	5549	5655	5678	5812	5834	5961
		5981	6123	6143	6317	6339	6482	6502	6669	6691	6823	6843	7018	7040
		7183	7203	7395	7417	7585	7605	7828	7850	7995	8017	8242	8264	8413
		8433	8557	8660	8684	8824	8844	8967	8988	9117	9181	9219	9240	9375
		9463	9617	9712	9831	9851	9955	10078	10098	10216	10343	10363	10513	10631
		10759	11277	11289	11357	11370								

CZRJIDO, RPO4/5/6 FCTNL CTLR1		MACY11	30A(1052)	25-MAY-79	10:30	F 8	PAGE 305							SEQ 0303
CZRJID.P11 28-MAR-79 09:03		CROSS REFERENCE TABLE -- USER SYMBOLS												
NOUNIT	004720	2084#	2446*	2506*	2512	2518	2585*	10894*						
NOUSE	066766	2475	13974#											
NUNIT	004722	2086#	2518*	2519*	3844	4020	4665	5142	5209	5425	5545			
OCYL	= 100000	1911#												
OF	002372	2002#	12422											
OFREV	= 000200	1878#												
OFSET	041510	5649	11117#											
OFSETC	002454	2034#	5383	5403	5423	5654	11118							
OFSTVL	004610	2060#	5315*	5316*	5353	5618*	5619	5621*	5622					
OF100	= 000004	1873#												
OF200	= 000010	1874#												
OF25	= 000001	1871#												
OF400	= 000020	1875#												
OF50	= 000002	1872#												
OF800	= 000040	1876#												
OPERSE	042650	11519#	12016	12099										
OPI	= 020000	1793#												
OR	= 000200	1727#	10650	10651										
PAR	= 000010	1783#	4072											
PCJSR	041560	10963*	10964*	11142#	11144*	11145*	11150*	11151*	14276	14278	14286			
PCLBUF	002346	1972#	11250*											
PCLCSR	002344	1971#	11224*	11251*										
PCLCTR	002350	1973#	11225											
PGE	= 002000	1730#												
PIP	= 020000	1775#	4681	4873	5160	5191	5441	5561	8997	9249				
PIRQ	= 177772	679#												
PIRQVE	= 000240	773#												
PKACK	002460	2036#	2938	2959	3043	3064	3085	3472	3630	5005	5266			
PLU	= 020000	1866#												
PRE	= 000020	1906#												
PRITEM	047342	2263*	3010*	3586*	4336*	5254*	10848*	10893*	11521*	12319#	12481*	12498	12502*	
PROG	= 001000	1771#												
PRO	= 000000	696#												
PR1	= 000040	697#												
PR2	= 000100	698#												
PR3	= 000140	699#												
PR4	= 000200	700#												
PR5	= 000240	701#												
PR6	= 000300	702#												
PR7	= 000340	703#												
PS	= 177776	676#	677	2175*	2743*	2782*	2837*	10866*	11520*					
PSEL	= 002000	1754#	7369	7395	7417	7559	7585	7605	10316	10343	10363			
PSU	= 000001	1903#												
PSW	= 177776	677#												
PUTREG	043436	3289	3314	3599	10971	10983	11468	11576#						
PWRVEC	= 000024	768#	2149*	2150*	12663*	12664*	12673*	12679*	12691*	12692*				
RA	000200	799#	11743											
RDCHR	= 104410	12174	12654#											
RDLIN	= 104411	12221	12655#											
RDOCT	= 104412	2274	11547	11558	11691	11713	12656#							
RDY	= 000200	1751#	2744	2755	2783	2911	3086	3335	3720	3835	3971	4448	4664	4864
		5132	5141	5424	5544	5825	5972	6134	6330	6493	6682	6834	7031	7194
		7408	7596	7841	8008	8255	8424	8675	8835	8988	9126	9240	9384	9626
		9721	9842	10089	10354	10523	10640	10768	11156					
READAT	002446	2031#	6095	6122	6142	6454	6481	6501	6795	6822	6842	7155	7182	7202
		7556	7584	7604	10746	10758								

CZRJID0, RP04/5/6 FCTNL CTLR1
CZRJID.P11 28-MAR-79 09:03

MACY11 30A(1052) 25-MAY-79 10:30 PAGE 311
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0309

UNITS	004676	2082#	2440	2444	2517	2566	10897							
UNITSL	004730	2091#	2277*	2282	2524									
UNLOAD	002424	2022#	5099	5120	5140									
UNS	= 040000	1794#												
UPE	= 020000	1733#												
US1	= 000001	1720#												
US2	= 000002	1721#												
US4	= 000004	1722#												
UWR	= 000010	1905#												
VUF	= 000002	1904#												
VU30	= 010000	1865#												
VV	= 000100	1768#	2902	2970	2981	3077	3095	3111	3352	3484	3642	4276	4306	4478
		4681	4873	4980	5016	5162	5163	5193	5194	5278	5441	5561	5843	5990
		6152	6348	6511	6700	6852	7049	7212	7426	7614	7859	8026	8273	8442
		8693	8853	8997	9249	9860	10107	10372	11147	11165				
WAITBT	041760	11247#	11260*	11267	11272	11341*	11347	11352	14247	14250				
WAITPC	041754	11245#	11257*	11258*	11338*	11339*	14247	14250						
WAITRE	041756	11246#	11259*	11267	11272	11276	11288	11340*	11347	11352	11356	11372	14247	14250
WAITTM	041762	11225*	11248#	11249*	11296	11306	14250							
WAIT.P	041764	11249#												
WAIT.T	042224	11334#	12657											
WAT	= 104413	2968	3075	3482	3640	3718	3833	3969	4274	4446	4653	4853	5014	5130
		5276	5413	5533	5661	5684	5823	5970	6132	6328	6491	6680	6832	7029
		7192	7406	7594	7839	8006	8253	8422	8565	8673	8833	8977	9124	9190
		9229	9382	9472	9624	9719	9840	9962	10087	10223	10352	10521	10638	10766
		12657#												
WC	002354	1993#	2990	3121	3378	3452	3731	3868	4082	4316	4488	4718	4913	5230
		5479	5601	5873	6020	6182	6378	6541	6730	6882	7079	7242	7479	7644
		7889	8056	8309	8470	8730	8883	9037	9289	9649	9746	9890	10137	10401
		10565	10693	10833	11581	12334								
WCE	= 040000	1734#												
WCF	= 000040	1785#												
WCU	= 000001	1853#												
WLE	= 004000	1791#	10671											
WRCHDT	002440	2028#	9802	9830	9850									
WRCHK	002436	2027#	10051	10077	10097	10311	10342	10362						
WRFROM	002470	2045#	5734	5744	5777	5861	5893	6038	6075	6197	6249	6254	6282	6366
		6396	6429	6434	6557	6611	6634	6718	6748	6775	6898	6950	6955	6983
		7067	7097	7130	7135	7258	7326	7331	7359	7467	7498	7531	7536	7661
		7722	7731	7739	7748	7794	7877	7909	7937	7946	8075	8135	8144	8152
		8161	8207	8291	8329	8356	8365	8489	8582	8592	8601	8611	8625	8711
		8761	8769	8774	8899	9066	9073	9078	9134	9324	9331	9336	9392	9487
		9498	9503	9508	9513	9518	9528	9548	9554	9561	9582	9637	9765	9795
		9878	9978	9983	9988	9993	9998	10008	10027	10032	10044	10125	10239	10244
		10249	10254	10259	10269	10287	10292	10304	10390	10475	10487	10591	10603	10727
		10776												
WRIDAT	002442	2029#	6289	6316	6338	6641	6668	6690	6990	7017	7039	7366	7394	7416
		10494	10512	10610	10630									
WRIFOR	002444	2030#	5784	5811	5833	7801	7827	7849	8214	8241	8263	8632	8659	8683
		9589	9616											
WRL	= 004000	1773#	10557	10824										
WRU	= 000400	1861#												
WSU	= 000004	1855#												
XE2	007464	2407	2438#											
\$AUTOB	001134	879#	2229*	12072	12201									
\$BDADR	001122	874#	2866*	2874*	2904*	2914*	11159*	11167*	14260	14273	14276	14278		

CZRJ10, RP04/5/6 FCTNL CTLR1
CZRJID.P11 28-MAR-79 09:03

MACY11 30A(1052) 25-MAY-79 10:30 PAGE 312
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0310

\$BDDAT	001126	876#	2359*	2924*	3091*	3100*	4677*	4686*	4869*	4878*	5154*	5170*	5437*	5446*
		5557*	5566*	5839*	5848*	5986*	5995*	6148*	6157*	6344*	6353*	6507*	6516*	6696*
		6705*	6848*	6857*	7045*	7054*	7208*	7217*	7422*	7431*	7610*	7619*	7855*	7864*
		8022*	8031*	8269*	8278*	8438*	8447*	8689*	8698*	8849*	8858*	8993*	9002*	9245*
		9254*	9856*	9865*	10103*	10112*	10368*	10377*	10990*	10991	11276*	11288*	11356*	11372*
		11476*	11657*	14247	14250	14253	14255	14266	14269	14273	14286			
\$BELL	001216	908#	12025	12194	12280	12302								
\$CHARC	045710	11940*	11950*	11957	11966*	11971#								
\$CKSWR	046222	12058#	12653											
\$CMTAG	001100	862#	2136	2137	2145	2151	2152							
\$CM1 =	000006	894#	895#	896#	897#	898#	899#	900#						
\$CM2 =	000014	894#	895#	896#	897#	898#	899#	900#						
\$CM3 =	000006	892#	894											
\$CM4 =	000006	900#	901#	902#	903#	904#	905#	906#						
\$CNTLC	046773	12013	12094	12194#										
\$CNTLG	047005	12079	12196#											
\$CNTLU	047000	12104	12195#											
\$CRLF	001223	910#	2472	2503	4155	4166	7449	7450	7457	7458	11534	11939	11974	12115
		12194	12255	12288	12302	12483	12507	12512						
\$DBLK	045464	11861	11895	11903#										
\$DOAGN	041150	10925	10934	10940#										
\$DTBL	045454	11864	11899#											
\$ENDAD	041140	791	10936#											
\$ENDCT	041106	10927#												
\$ENDMG	041157	10887	10929	10944#										
\$ENULL	041154	10890	10932	10943#										
\$EOP	041052	2321	2432	2514	10895	10917#								
\$EOPCT	041100	10924#	10928											
\$ERFLG	001103	865#	11779	11808	11810	11816*	11837	12275*	12302					
\$ERMAX	001115	871#	2153*	11810	11832*	11837								
\$ERROR	047174	2145	2338	12269#										
\$ERRPC	001116	872#	12282*	12283*	12284	12302	12489	14247	14250	14253	14255	14257	14260	14263
		14266	14269	14273	14276	14278	14280	14282	14286					
\$ERRTB	001226	927#	12497											
\$ERRTY	047344	12287	12321#											
\$ERTTL	001112	869#	10878	10880*	12281*	12302								
\$ESCAP	001214	907#	2152*	11831*	12297	12299	12302							
\$FILLC	001156	890#	11943	11974										
\$FILLS	001155	889#	11974											
\$GDADR	001120	873#												
\$GDDAT	001124	875#	2360*	2923*	3087*	3096*	4673*	4682*	4865*	4874*	5150*	5166*	5433*	5442*
		5553*	5562*	5835*	5844*	5982*	5991*	6144*	6153*	6340*	6349*	6503*	6512*	6692*
		6701*	6844*	6853*	7041*	7050*	7204*	7213*	7418*	7427*	7606*	7615*	7851*	7860*
		8018*	8027*	8265*	8274*	8434*	8443*	8685*	8694*	8845*	8854*	9989*	8998*	9241*
		9250*	9852*	9861*	10099*	10108*	10364*	10373*	10993*	11475*	11656*	14253	14266	14269
		14273	14286											
\$GET42	041130	10933#												
\$GTSWR	046312	12080#	12651											
\$HD =	000000	619												
\$HIOCT	047172	12243*	12254#											
\$ICNT	001104	866#	11823*	11824	11826*	11836								
\$ILLUP	051150	12663	12679	12698#										
\$INTAG	001135	880#	12077*	12096	12116	12201								
\$ITEMB	001114	870#	12284*	12302	12486									
\$LF	001224	911#	11974	12185	12194	12255	12302							
\$LPADR	001106	867#	2154*	2543*	5303*	11549*	11560	11814*	11829*	11834	11836			

CZRJIDO, RP04/5/6 FCTNL CTLR1
CZRJID.P11 28-MAR-79 09:03

MACY11 30A(1052) 25-MAY-79 10:30 PAGE 313
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0311

\$LPERR	001110	868#	2155*	10583*	11532	11559*	11814	11830*	11836	12296				
\$MAIL =	***** U	2172	2225	11829	11927	12290								
\$LW	047023	12083	12199#											
\$MSWR	047012	12080	12197#											
\$MXCNT	045246	11827	11836#											
\$NULL	001154	888#	11945	11974										
\$NWTST=	000001	2285#	2287	2344#	2346	2367#	2369	2528#	2530	2689#	2691	2721#	2723	2762#
		2764	2802#	2804	3015#	3017	3154#	3173#	3190#	3192	3492#	3494	3666#	3668
		3883#	3885	4098#	4100	4339#	4341	4520#	4539#	4558#	4560	4742#	4761#	4779#
		4781	4925#	4927	5283#	5285	5628#	5630	5695#	5697	6052#	6054	6209#	6211
		6571#	6573	6910#	6912	7273#	7275	7680#	7682	8089#	8091	8504#	8506	9150#
		9152	9410#	9412	9906#	9908	10153#	10155	10422#	10424	10853#	10855		
\$OCNT	050714	12569*	12598*	12611#										
\$OMODE	050716	12564*	12568*	12573	12576*	12587*	12613#							
\$OVER	045232	11791	11807	11815	11825	11833#								
\$PASS	001100	863#	2846	3241	3529	4141	4955	10457	10886*	10888	10921*	10922*	10930	10943
		11821	11837											
\$POWER	051156	12694	12701#											
\$PWRAD	051144	12696#												
\$PWRDN	051004	2149	12663#	12691										
\$PWRMG	051140	12694#												
\$PWRUP	051056	12673	12679#											
\$QUES	001222	909#	11974	12134	12178	12194	12252	12255	12302					
\$RDCHR	046564	12147#	12654											
\$RDDEC=	***** U	12657												
\$RDLIN	046654	12170#	12655											
\$RDOCT	047034	12216#	12656											
\$RDSZ =	000011	12163#												
\$REGAD	001160	892#												
\$REG0	001162	894#												
\$REG1	001164	895#												
\$REG2	001166	896#												
\$REG3	001170	897#												
\$REG4	001172	898#												
\$REG5	001174	899#												
\$RTNAD	041152	10942#												
\$R2A =	***** U	12657												
\$SAVRE=	***** U	12657												
\$SAVR6	051154	12672*	12680	12681*	12682*	12700#								
\$SCOPE	044776	2143	11788#											
\$SETUP=	000117	2134#	2142	2143	2145	2147	2149	2151	2152	2154	2222	10919	11789	12017
		12022	12023	12053	12201	12270	12293	12301						
\$SS1 =	000000	2174#												
\$STUP =	177777	2134#												
\$SVLAD	045204	11799	11828#											
\$SVPC =	000200	789#	794											
\$SWR =	167770	609#	619	651	652	653	654	655	656	657	658	906	907	908
		2151	2152	2154	2155	2292	2351	2381	2542	2698	2730	2769	2830	3025
		3158	3177	3209	3502	3675	3891	4127	4361	4524	4543	4568	4746	4765
		4789	4947	5303	5635	5722	6064	6238	6600	6939	7305	7711	8124	8523
		9163	9436	9927	10176	10449	10865	10914	10920	10935	10941	10943	11780	11781
		11782	11783	11784	11790	11802	11804	11805	11808	11809	11810	11817	11818	11819
		11830	11833	11836	12261	12262	12263	12264	12265	12278	12285	12290	12294	12302
		12697												
\$SWRMK=	000000	658	659	11784	11785	11806								
\$TIMES	001212	906#	2151*	2292*	2351*	2381*	2542*	3209*	3502*	7305*	10865*	10920*	11817*	11824

ALLREG	647#	12328	12337	12345	12353	12361	12369	12377	12385	12393	12401	12409	12417	12425	12433
	12441	12449	12457	12465	12473										
CHANGR	647#	2975	3105	3329	3342	3353	3360	3400	3411	3420	3427	3434	3848	3852	3990
	4001	4010	4066	4289	4300	4461	4472	4700	4888	4892	4899	5201	5211	5218	5454
	5461	5468	5574	5581	5588	9013	9017	9024	9265	9269	9276	10551	10653	10665	10672
	10681	10818													
CHECKD	647#	3229	3543	3682	3798	3898	4575	4796	4969	5644	5797	5946	6108	6302	6467
	6654	6808	7003	7168	7381	7571	7813	7980	8227	8398	8645	8809	9102	9360	9443
	9602	9697	9816	9934	10063	10190	10327	10498							
CHECKV	647#	2934	3039	4175	4368	4997	5311								
CHKCNT	647#	5021	5325	5497											
CKCNTV	647#	5256													
CLEARA	647#	5743	5762	5911	6074	6248	6253	6263	6268	6415	6420	6428	6433	6610	6620
	6766	6774	6949	6954	6964	6969	7116	7121	7129	7134	7325	7330	7340	7345	7517
	7522	7530	7535	7730	7747	7766	7779	7927	7945	8143	8160	8179	8192	8347	8364
	8590	8609	8750	8767	8773	9056	9072	9077	9314	9330	9335	9497	9502	9507	9512
	9517	9547	9560	9568	9977	9982	9987	9992	9997	10026	10031	10238	10243	10248	10253
	10258	10286	10291	10474	10590	10719	10726								
CMPBLK	647#	5889	6035	6194	6392	6554	6744	6895	7093	7255	7494	7658	7905	8072	8325
	8486	8896	9132	9390	9762	10774									
CMREGI	647#	2986	3117	3374	3448	3727	3864	4078	4312	4484	4714	4909	5226	5475	5597
	5869	6016	6178	6374	6537	6726	6878	7075	7238	7475	7640	7885	8052	8305	8466
	8726	8879	9033	9285	9645	9742	9886	10133	10397	10561	10689	10829			
COMMEN	1#	620	774#												
DATA CO	647#	5770	5919	6082	6276	6441	6628	6782	6977	7142	7353	7543	7787	7953	8200
	8371	8617	8780	9084	9342	9575	9670	9788	10038	10298	10481	10597	10733		
DISREG	647#	2938	3043	3688	3803	4229	4401	4625	4825	5099	5383	5502			
DUM	647#	2938	3043	3688	3803	4229	4401	4625	4825	5099	5383	5502			
ENDCOM	1#	629	774#												
ERROR	668#	2308	2363	2750	2757	2794	2867	2875	2905	2915	2925	2995	3093	3102	3126
	3293	3322	3383	3457	3603	3736	3873	4087	4321	4493	4679	4688	4723	4871	4880
	4918	5156	5172	5235	5439	5448	5484	5559	5568	5605	5669	5841	5850	5878	5899
	5988	5997	6025	6045	6150	6159	6187	6204	6346	6355	6383	6402	6509	6518	6546
	6564	6698	6707	6735	6754	6850	6859	6887	6905	7047	7056	7084	7103	7210	7219
	7247	7265	7424	7433	7484	7504	7612	7621	7649	7668	7857	7866	7894	7915	8024
	8033	8062	8083	8271	8280	8314	8335	8440	8449	8476	8497	8691	8700	8734	8851
	8860	8888	8906	8995	9004	9042	9141	9247	9256	9294	9399	9654	9752	9773	9858
	9867	9895	10105	10114	10141	10370	10379	10406	10570	10698	10783	10838	10974	10986	10995
	11160	11168	11279	11281	11291	11300	11309	11359	11362	11373					
ESCAPE	1#	774#													
F I HEAD	647#	5732	5754	7720	7737	7758	7771	7935	8133	8150	8171	8184	8354	8579	8598
	8758	9064	9322	9485	9552										
F I L L B L	647#	3978	3981	4024	4027	4030	4033	4036	4039	4046	4050	4060	4063	4283	4286
F L S V R E	647#	3978	3981	4024	4027	4030	4033	4036	4039	4046	4050	4060	4063	4283	4286
	4455	4458	4694	4697	4704	4707	4885	5592	5856	5859	5862	6003	6006	6009	6165
	6168	6171	6361	6364	6367	6524	6527	6530	6713	6716	6719	6865	6868	6871	7062
	7065	7068	7225	7228	7231	7462	7465	7468	7627	7630	7633	7872	7875	7878	8039
	8042	8045	8286	8289	8292	8295	8298	8454	8457	8460	8706	8709	8712	8716	8719
	8866	8869	8872	9010	9262	9632	9635	9638	9728	9731	9734	9873	9876	9879	10120
	10123	10126	10385	10388	10391	10662									
GETPRI	1#	774#													
GETSWR	1#	609#	774#	2222											
GOO	647#	2958	3063	3471	3629	3708	3823	3959	4251	4263	4423	4435	4643	4843	5004
	5119	5265	5402	5522	5653	5676	5810	5959	6121	6315	6480	6667	6821	7016	7181
	7393	7583	7826	7993	8240	8411	8555	8658	8822	8965	9115	9179	9217	9373	9461
	9615	970	9829	9953	10076	10214	10341	10511	10629	10757					

CZRJIDO, RPO4/5/6 FCTNL CTLR1
CZRJID.P11 28-MAR-79 09:03

MACY11 30A(1052) 25-MAY-79 10:30 PAGE 317
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0314

LOAD	647#	3754	3756	3758	3759	3761	3763	3765	3767	3769	3771	3773	3795	3907	3909
	3911	3912	3914	3916	3918	3920	3922	3924	3926	3948	4184	4186	4188	4189	4191
	4193	4195	4197	4199	4201	4203	4225	4377	4379	4381	4382	4384	4386	4388	4390
	4392	4394	4396	4397	4581	4583	4585	4586	4588	4590	4592	4594	4596	4598	4600
	4622	4803	4805	4807	4808	4810	4812	4814	4816	4818	4820	4822	4823	5033	5037
	5041	5044	5048	5052	5056	5060	5064	5068	5072	5096	5332	5335	5338	5340	5343
	5346	5349	5355	5357	5359	5381									
MAKECL	647#	3153	3172	4519	4537	4741	4759								
MANUAL	647#	2839	3234	3522	4134	4948	10450								
MSG	2284#	2287	2343#	2346	2366#	2369	2527#	2530	2688#	2691	2720#	2723	2761#	2764	2802#
	2804	3014#	3017	3189#	3192	3491#	3494	3665#	3668	3883#	3885	4097#	4100	4337#	4341
	4556#	4560	4777#	4781	4925#	4927	5283#	5285	5628#	5630	5695#	5697	6051#	6054	6208#
	6211	6571#	6573	6910#	6912	7273#	7275	7679#	7682	8088#	8091	8503#	8506	9149#	9152
	9410#	9412	9905#	9908	10152#	10155	10421#	10424	10853#	10855					
MULT	1#	774#													
NEWTST	1#	774#	2285	2344	2367	2528	2689	2721	2762	2802	3015	3154	3173	3190	3492
	3666	3883	4098	4339	4520	4539	4557	4742	4761	4778	4925	5283	5628	5695	6052
	6209	6571	6910	7273	7680	8089	8504	9150	9410	9906	10153	10422	10853		
OFFST	647#	5649													
POP	1#	774#	3791	3944	4221	4618	5090	5377	9542	10022	10283	10997	11024	11052	11077
	11199	11311	11377	11402	11433	11488	11586	11667	11890	12244	12684	12685			
PUSH	1#	774#	3775	3928	4205	4602	5074	5361	9525	10005	10266	10961	11013	11042	11070
	11189	11252	11334	11389	11419	11456	11576	11639	11849	12218	12665	12671			
REPORT	1#	774#													
RFORGC	647#														
RMCLEA	647#	2546	2733	2772	2832	2879	2928	3034	3224	3391	3538	3589	3677	3747	3893
	4129	4170	4363	4570	4791	4964	4972	5306	5319	5612	5639	5671	5724	5905	6066
	6240	6409	6602	6760	6941	7110	7308	7511	7713	7921	8126	8341	8525	8572	8744
	8940	9165	9308	9438	9478	9664	9782	9929	9969	10178	10230	10466	10585	10711	
RH7OCK	647#	7313	10183												
SAVE	647#	12271													
SAVTST	647#	2293	2353	2382	2698	2731	2770	3159	3178	4525	4544	4747	4766		
SCH	647#														
SCOPE	669#	2291	2350	2380	2541	2697	2729	2768	2829	3024	3157	3176	3208	3501	3674
	3890	4126	4360	4523	4542	4567	4745	4764	4788	4946	5302	5634	5670	5721	6063
	6237	6599	6938	7304	7710	8123	8522	9162	9435	9926	10175	10448	10864	10918	
SEEKCO	647#	8946	9198	9450	9942	10202									
SETPRI	1#	774#	12150												
SETTRA	12637#	12646	12647	12648	12649	12651	12653	12654	12655	12656	12657				
SETUP	1#	774#	2135												
SKIP	1#	647#	774#	2269	2281	2358	2523	2701	2752	2756	2791	5625	5691		
SLASH	1#	774#													
SPACE	774#														
SREGIS	647#	2944	3049	3256	3694	3809	4235	4407	4631	4831	5105	5388	5508	5790	5939
	6101	6295	6460	6647	6801	6996	7161	7374	7564	7806	7973	8220	8391	8638	8802
	8951	9203	9595	9690	9808	10056	10320	10531	10615	10796					
STARS	1#	639	645	774#	787	853	857	912	1699	1701	1741	1743	1984	1989	2236
	2258	2285	2290	2344	2349	2367	2379	2458	2478	2528	2540	2591	2610	2637	2676
	2689	2696	2721	2728	2762	2767	2802	2828	3015	3023	3140	3150	3154	3156	3173
	3175	3190	3207	3211	3220	3492	3500	3504	3513	3666	3673	3883	3889	4098	4125
	4339	4359	4509	4515	4520	4522	4539	4541	4558	4566	4732	4738	4742	4744	4761
	4763	4779	4787	4925	4945	5283	5301	5495	5628	5633	5695	5720	6052	6062	6209
	6236	6571	6598	6910	6937	7273	7303	7680	7709	8089	8122	8504	8521	9150	9161
	9410	9434	9906	9925	10153	10174	10422	10447	10851	10852	10853	10863	10911	11759	11769
STARTT	11776	11839	11906	11976	12053	12068	12139	12163	12204	12257	12539	12616	12661	12677	
	647#	2544	2830	3032	3222	3536	3675	3891	4127	4361	4568	4789	4962	5304	5636

.\$DB20	1#		
.\$DIV	i#		
.\$EOP	1#	609#	10909
.\$ERRO	1#	609#	12255
.\$ERRT	1#	609#	
.\$MULT	1#		
.\$POWE	1#	609#	12659
.\$RAND	1#		
.\$RDDE	1#		
.\$RDOC	1#	609#	12202
.\$READ	1#	609#	11974
.\$R2AZ	1#		
.\$SAVE	1#		
.\$SB2D	1#		
.\$SB2O	1#		
.\$SCOP	1#	609#	11774
.\$SIZE	1#		
.\$SUPR	1#		
.\$TRAP	1#	609#	12614
.\$TYPB	1#		
.\$TYPD	1#	609#	11837
.\$TYPE	1#	609#	11904
.\$TYPO	1#	609#	12537
.\$4OCA	1#		
.1170	1#		

. ABS. 072254 000

ERRORS DETECTED: 0

DSKZ:CZRJID.BIN,DSKZ:CZRJID.LST/CRF/SOL/NL:TOC:MD:MC:CND/LI:ME=CZRJID.SML,CZRJID.P11
RUN-TIME: 110 161 8 SECONDS
RUN-TIME RATIO: 569/281=2.0
CORE USED: 38K (75 PAGES)