

RP04/5/6

FUNCTIONAL CONTROLLER 1
CZRJICO

AH-9220C-MC

COPYRIGHT © 74-78

FICHE 2 OF 2

DEC 1978

digital

MADE IN USA

.REM @

IDENTIFICATION

PRODUCT CODE: AC-9218C-MC
PRODUCT NAME: CZRJICO RP04/5/6 FUNCTIONAL CONTROLLER TEST PART I
DATE CREATED: MAY 1976
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 OHN A SINGLE COMPUTER SYSTEMM AND CAN BE COPIED (WITH INCLUSION OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE SBE PROVIDED IN WRITING BY DIGITAL.

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

COPYRIGHT (C) 1974,1978 DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION

DIGITAL	PDP	UNIBUS	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 RPO4/5/6S 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 RP04/5/6 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 2 - 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

CZRJICO, RPO4/5/6 FCTNL CTLR1
CZRJIC.P11 27-JUL-78 13:05

MACY11 30A(1052) 27-JUL-78 13:06 PAGE 15
BASIC DEFINITIONS

B 2

SEQ 0014

629

CZ
CZ

```
630 000174 000000      DISPREG: .WORD 0          ;;SOFTWARE DISPLAY REGISTER
631 000176 000000      SWREG:   .WORD 0          ;;SOFTWARE SWITCH REGISTER
632
633 000046 041072      $ENDAD          ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
634 000052 002000      .WORD 2000      ;;2)SET LOC.52 TO 2000
635
636 .SBTTL STARTING ADDRESSES
637
638                .=200
639 000200 000137 005012    RA:   JMP @MBEGIN        ;NORMAL START
640 000204 000137 043612  ADDMOD: JMP @MBASECH     ;START FOR ADDRESS-MODIFICATION
641 000210 000137 004776    JMP @MBEGIN2         ;JUMP TO SELECT DRIVE START
642                .=220
643 000220 000137 004762    JMP @MBEGIN1        ;JUMP TO NO OPERATOR TESTS START
644
645
646
647                ;*STARTING ADDRESS 200 FOR NORMAL STARTS
648                ;*THIS WILL TEST ALL DRIVES ON THE SYSTEM A SINGLE DRIVE AT A TIME
649
650                ;*STARTING ADDRESS 204 FOR NON-DEFAULT ADDRESS PARAMETERS
651                ;*AUTO RESTART AT ADDRESS 200 AFTER LOADING PARAMETERS
652
653                ;*STARTING ADDRESS 210 WILL TEST ONLY ONE SPECIFIED DRIVE
654
655                ;*STARTING ADDRESS 220 WILL JUMP OVER THE TESTS REQUIRING AN OPERATOR
656                ;*AT THE DRIVE
657
```

CZRJICO, RP04/5/6 FCTNL CTLR1
CZRJIC.P11 27-JUL-78 13:05

MACY11 30A(1052) 27-JUL-78 13:06 PAGE 17
MEMORY MANAGEMENT DEFINITIONS

D 2

SEG 0016

658

001110

.-1110

CZ
CZ

659	001100	000000	\$PASS:	.WORD	0	::CONTAINS	PASS COUNT
660	001102	000	\$TSTNM:	.BYTE	0	::CONTAINS	THE TEST NUMBER
661	001103	000	\$ERFLG:	.BYTE	0	::CONTAINS	ERROR FLAG
662	001104	000000	\$ICNT:	.WORD	0	::CONTAINS	SUBTEST ITERATION COUNT
663	001106	000000	\$LPADR:	.WORD	0	::CONTAINS	SCOPE LOOP ADDRESS
664	001110	000000	\$LPERR:	.WORD	0	::CONTAINS	SCOPE RETURN FOR ERRORS
665	001112	000000	\$ERTTL:	.WORD	0	::CONTAINS	TOTAL ERRORS DETECTED
666	001114	000	\$ITEMB:	.BYTE	0	::CONTAINS	ITEM CONTROL BYTE
667	001115	001	\$ERMAX:	.BYTE	1	::CONTAINS	MAX. ERRORS PER TEST
668	001116	000000	\$ERRPC:	.WORD	0	::CONTAINS	PC OF LAST ERROR INSTRUCTION
669	001120	000000	\$GDADR:	.WORD	0	::CONTAINS	ADDRESS OF 'GOOD' DATA
670	001122	000000	\$BDADR:	.WORD	0	::CONTAINS	ADDRESS OF 'BAD' DATA
671	001124	000000	\$GDDAT:	.WORD	0	::CONTAINS	'GOOD' DATA
672	001126	000000	\$BDDAT:	.WORD	0	::CONTAINS	'BAD' DATA
673	001130	000000		.WORD	0	::RESERVED--NOT TO BE USED	
674	001132	000000		.WORD	0		
675	001134	000	\$AUTOB:	.BYTE	0	::AUTOMATIC	MODE INDICATOR
676	001135	000	\$INTAG:	.BYTE	0	::INTERRUPT	MODE INDICATOR
677	001136	000000		.WORD	0		
678	001140	177570	\$SWR:	.WORD	DSWR	::ADDRESS	OF SWITCH REGISTER
679	001142	177570	\$DISPLAY:	.WORD	DDISP	::ADDRESS	OF DISPLAY REGISTER
680	001144	177560	\$TKS:	177560		::TTY KBD	STATUS
681	001146	177562	\$TKB:	177562		::TTY KBD	BUFFER
682	001150	177564	\$TPS:	177564		::TTY PRINTER	STATUS REG. ADDRESS
683	001152	177566	\$TPB:	177566		::TTY PRINTER	BUFFER REG. ADDRESS
684	001154	000	\$NULL:	.BYTE	0	::CONTAINS	NULL CHARACTER FOR FILLS
685	001155	002	\$FILLS:	.BYTE	2	::CONTAINS	# OF FILLER CHARACTERS REQUIRED
686	001156	012	\$FILLC:	.BYTE	12	::INSERT	FILL CHARS. AFTER A 'LINE FEED'
687	001157	000	\$TPFLG:	.BYTE	0	::'TERMINAL	AVAILABLE' FLAG (BIT<07>=0=YES)
688	001160	000000	\$REGAD:	.WORD	0	::CONTAINS	THE ADDRESS FROM
689	001162	000000	\$REG0:	.WORD	0	::CONTAINS	((REGAD)+0)
690	001164	000000	\$REG1:	.WORD	0	::CONTAINS	((REGAD)+2)
691	001166	000000	\$REG2:	.WORD	0	::CONTAINS	((REGAD)+4)
692	001170	000000	\$REG3:	.WORD	0	::CONTAINS	((REGAD)+6)
693	001172	000000	\$REG4:	.WORD	0	::CONTAINS	((REGAD)+10)
694	001174	000000	\$REG5:	.WORD	0	::CONTAINS	((REGAD)+12)
695	001176	000000	\$TMP0:	.WORD	0	::USER	DEFINED
696	001200	000000	\$TMP1:	.WORD	0	::USER	DEFINED
697	001202	000000	\$TMP2:	.WORD	0	::USER	DEFINED
698	001204	000000	\$TMP3:	.WORD	0	::USER	DEFINED
699	001206	000000	\$TMP4:	.WORD	0	::USER	DEFINED
700	001210	000000	\$TMP5:	.WORD	0	::USER	DEFINED
701	001212	000000	\$TIMES:	0		::MAX.	NUMBER OF ITERATIONS
702	001214	000000	\$ESCAPE:	0		::ESCAPE	ON ERROR ADDRESS
703	001216	177607	\$BELL:	.ASCIZ	<207><377><377>	::CODE	FOR BELL
704	001222	077	\$QUES:	.ASCII	/?/	::QUESTION	MARK
705	001223	015	\$CRLF:	.ASCII	<15>	::CARRIAGE	RETURN
706	001224	000012	\$LF:	.ASCIZ	<12>	::LINE	FEED

000377

707					
708					
709					
710					
711					
712			:*ITEM1		
713	001226	051120	EM1		:RP04 DID NOT INTERRUPT
714					:WAITED ON BIT DID NOT OCCUR
715	001230	067160	DH1		:PC
716					:WAT PC
717					:BIT WAITED
718					:REG ADDRESS
719					:REG CONTENTS
720					:RHCS1 CONTENTS
721	001232	071506	DT1		:\$ERRPC, WAITPC, WAITBT, WAITRE, \$BDDAT, CS1
722	001234	072014	DF1		:0,0,0,0,0,0
723					
724			:*ITEM2		
725	001236	051147	EM2		: INTERRUPT ENABLE BIT DOWN BUT
726					: WAITED ON BIT DID NOT OCCUR
727	001240	067160	DH1		:PC
728					:WAT PC
729					:BIT WAITED
730					:REG ADDRESS
731					:REG CONTENTS
732					:RHCS1 CONTENTS
733	001242	071506	DT1		:\$ERRPC, WAITPC, WAITBT, WAITRE, \$BDDAT, CS1
734	001244	072014	DF1		:0,0,0,0,0,0
735					
736			:*ITEM3		
737	001246	051236	EM3		:RP04 DID NOT INTERRUPT WHEN
738					:WAITED ON BIT DID SET
739	001250	067160	DH1		:PC
740					:WAT PC
741					:BIT WAITED
742					:REG ADDRESS
743					:RHCS1 CONTENTS
744	001252	071506	DT1		:\$ERRPC, WAITPC, WAITBT, WAITRE, \$BDDAT, CS1
745	001254	072014	DF1		:0,0,0,0,0,0
746					
747			:*ITEM4		
748	001256	051317	EM4		:WAITED ON BIT DID SET BUT
749					:TIME IS IN ERROR
750					:TIME IS GIVEN IN 10 MICRO SEC.
751					:(DECIMAL)
752	001260	067337	DH4		:PC
753					:WAT PC
754					:BIT WAITED
755					:REG ADDRESS
756					:TIME IN 10 MSEC
757	001262	071526	DT4		:\$ERRPC, WAITPC, WAITBT, WAITRE, \$BDDAT, WAITIM
758	001264	072023	DF4		:0,0,0,0,0,1
759					
760			:*ITEM5		
761	001266	051430	EM5		:RHAS DOES NOT CLEAR BY
762					:MOVING IN ALL ONES

763	001270	067500	DH5	:PC
764				:REG. ADDR.
765				:GOOD DATA
766				:RECEIVED DATA
767	001272	071546	DT5	:\$ERRPC,REGADR,\$GDDAT,\$BDDAT
768	001274	072032	DF5	:0,0,0,0
769				
770			:*ITEM6	
771	001276	051502	EM6	:LOADING RHER1 FOR ALL
772				:UNITS DID NOT SET ANY BITS
773				:IN RHAS-NO UNITS PRESENT
774	001300	067617	DH6	:PC
775				:REG ADDR
776				:RECEIVED DATA
777	001302	071562	DT6	:\$ERRPC,REGADR,\$BDDAT
778	001304	072037	DF6	:0,0,0
779				
780			:*ITEM7	
781	001306	051570	EM7	:SPECIFIED REGISTER NONEXISTANT
782				:SO ABORT PROGRAM
783	001310	067716	DH7	:PC
784				:ADDR. OF REG.
785	001312	071574	DT7	:\$ERRPC,TEMP1
786	001314	072043	DF7	:0,0
787				
788			:*ITEM10	
789	001316	051640	EM10	:STOPED DRIVE HAS MOL BIT
790				:IN RHDS1 = 1
791	001320	067756	DH10	:PC
792				:TEST NO
793				:FAILING REG ADDR
794				:CONTENTS OF RHCS1
795				:CONTENTS OF RHCS2
796				:CONTENTS OF RHDS1
797				:CONTENTS OF RHER1
798	001322	071604	DT10	:\$ERRPC,\$TSTNM,\$BDADR,CS1,CS2,DS1,ER1
799	001324	072046	DF10	:0,0,0,0,0,0,0
800				
801			:*ITEM11	
802	001326	051707	EM11	:WITH SPINDLE POWERED DOWN
803				:RHCS2 SHOULD HAVE ONLY
804				:UNIT NUMBER AND IR HIGH
805	001330	067756	DH10	:PC
806				:TEST NO
807				:FAILING REG. ADR
808				:CONTENTS OF RHCS1
809				:CONTENTS OF RHCS2
810				:CONTENTS OF RHDS1
811				:CONTENTS OF RHER1
812	001332	071604	DT10	:\$ERRPC,\$TSTNM,\$BDADR,CS1,CS2,DS1,ER1
813	001334	072046	DF10	:0,0,0,0,0,0,0
814				
815			:*ITEM12	
816	001336	052014	EM12	:AFTER A POWER UP WITH
817				:NO PACK ACKNOWLEDGE COMMAND
818				:RHDS1 SHOULD HAVE MOL-1, VV-0

875				:REG. ADDR.
876				:GOOD DATA
877				:RECEIVED DATA
878	001402	071546	DT5	:\$ERRPC,REGADR,\$GDDAT,\$BDDAT
879	001404	072032	DF5	:0,0,0,0
880				
881			:*ITEM17	
882	001406	052606	EM17	:DRIVE CLEAR COMMAND
883				:CAUSED AN ERROR
884				:GOOD DATA GIVES WHAT SHOULD
885				:BE THERE
886				:RECEIVED DATA GIVES WHAT WAS
887				:THERE AFTER COMMAND
888	001410	067500	DH5	:PC
889				:REG. ADDR.
890				:GOOD DATA
891				:RECEIVED DATA
892	001412	071546	DT5	:\$ERRPC,REGADR,\$GDDAT,\$BDDAT
893	001414	072032	DF5	:0,0,0,0
894				
895			:*ITEM20	
896	001416	052743	EM20	:READ-IN COMMAND GAVE AN ERROR
897				:GOOD DATA HAS WHAT SHOULD BE THERE
898				:RECEIVED DATA HAS WHAT WAS
899				:AFTER COMMAND
900	001420	067500	DH5	:PC
901				:REG. ADDR.
902				:GOOD DATA
903				:RECEIVED DATA
904	001422	071546	DT5	:\$ERRPC,REGADR,\$GDDAT,\$BDDAT
905	001424	072032	DF5	:0,0,0,0
906				
907				
908			:*ITEM 21	
909	001426	053107	EM21	:RHCS1 CONTENTS DURING
910				:COMMAND WAS IN ERROR
911	001430	067500	DH5	
912	001432	071546	DT5	
913	001434	072032	DF5	
914				
915			:*ITEM 22	
916	001436	053162	EM22	:RHDS1 CONTENTS DURING
917				:COMMANDS WAS IN ERROR
918	001440	067500	DH5	
919	001442	071546	DT5	
920	001444	072032	DF5	
921				
922			:*ITEM 23	
923	001446	053235	EM23	:UNLOAD COMMAND GAVE AN ERROR
924				:GOOD DATA GIVES WHAT SHOULD
925				:BE THERE
926				:RECEIVED DATA GIVES WHAT WAS
927				:THERE AFTER COMMAND
928	001450	067500	DH5	
929	001452	071546	DT5	
930	001454	072032	DF5	

931					
932			:*ITEM 24		
933	001456	053400	EM24		:OFFSET COMMAND CAUSED AN ERROR
934					:GOOD DATA IS WHAT SHOULD BE THERE
935					:RECEIVED DATA GIVES WHAT WAS THERE
936					:AFTER AN OFFSET COMMAND
937	001460	067500	DH5		
938	001462	071546	DT5		
939	001464	072032	DF5		
940					
941			:*ITEM 25		
942	001466	053543	EM25		:RETURN TO CENTER LINE COMMAND
943					:CAUSED AN ERROR
944					:GOOD DATA GIVES WHAT SHOULD BE
945					:THERE
946					:RECEIVED DATA GIVES WHAT WAS
947					:THERE AFTER COMMAND
948	001470	067500	DH5		
949	001472	071546	DT5		
950	001474	072032	DF5		
951					
952			:*ITEM 26		
953	001476	053725	EM26		:500 OFFSETS CAUSED AN ERROR
954	001500	070135	DH26		:PC
955					:CONT. OF RHCS1
956					:CONT. OF RHCS2
957					:CONT. OF RHDS1
958					:CONT. OF RHER1
959					:CONT. OF RHER2
960					:CONT. OF RHER3
961	001502	071624	DT26		:\$ERRPC,CS1,CS2,DS1,ER1,ER2,ER3
962	001504	072055	DF26		:0,0,0,0,0,0,0
963					
964			:*ITEM 27		
965	001506	054015	EM27		:WRITE HEADER AND DATA
966					:CAUSED IMPROPER REGISTER CHANGE
967					:GOOD DATA GIVES WHAT
968					:SHOULD BE THERE
969					:RECEIVED DATA GIVES WHAT
970					:WAS THERE AFTER COMMAND
971	001510	067500	DH5		
972	001512	071546	DT5		
973	001514	072032	DF5		
974					
975			:*ITEM 30		
976	001516	054233	EM30		:WRITE HEADER AND DATA
977					:CHANGED WRITE FROM BUFFER
978	001520	070334	DH30		:PC
979					:WORD NO
980					:GOOD DATA
981					:BAD DATA
982	001522	071646	DT30		:\$ERRPC,ERWORD,\$GDDAT,\$BDDAT
983	001524	072065	DF30		:0,0,0,0
984					
985			:*ITEM 31		
986	001526	054313	EM31		:READ HEADER AND DATA CAUSED

Line	Code	Address	Pointer	Description
1211				; WAS THERE AFTER OPERATION
1212	001770	067500	DH5	
1213	001772	071546	DT5	
1214	001774	072032	DF5	
1215				
1216				; *ITEM 56
1217	001776	062052	EM56	; DATA READ WITH AN EXPECTED
1218				; ADDRESS OVERFLOW ERROR IS
1219				; INCORRECT
1220				; WORD NO 1 TO 260 SHOULD
1221				; BE READ
1222				; WORD NOS 261 TO 266 SHOULD
1223				; NOT CHANGE DUE TO READ
1224	002000	070334	DH30	
1225	002002	071646	DT30	
1226	002004	072065	DF30	
1227				
1228				; *ITEM 57
1229	002006	062262	EM57	; ATTEMPTING DATA COMMAND
1230				; WITH WRONG FORMAT BIT CAUSED
1231				; IMPROPER REGISTER CHANGE
1232				; GOOD DATA GIVES WHAT SHOULD BE
1233				; THERE
1234				; RECEIVED DATA GIVES WHAT WAS
1235				; THERE AFTER ATTEMPTED DATA
1236				; TRANSFER
1237	002010	067500	DH5	
1238	002012	071546	DT5	
1239	002014	072032	DF5	
1240				
1241				; *ITEM 60
1242	002016	062554	EM60	; ATTEMPTING TO MODIFY REGISTER
1243				; DURING AN OPERATION CAUSED
1244				; IMPROPER REGISTER CHANGE
1245				; GOOD DATA GIVES WHAT SHOULD
1246				; BE THERE
1247				; RECEIVED DATA GIVES WHAT WAS
1248				; THERE AFTER OPERATION
1249				; WAS COMPLETE
1250	002020	070607	DH60	; PC
1251				; REG. ADDR.
1252				; GOOD DATA
1253				; RECEIVED DATA
1254				; MODFING REGISTER
1255	002022	071700	DT60	; \$ERRPC, REGADR, \$GDDAT, \$BDDAT, \$BDADR
1256	002024	072100	DF60	; 0,0,0,0,0
1257				
1258				; *ITEM 61
1259	002026	063165	EM61	; DEVICE NOT AVAILBLE BEFORE COMMAND WAS TO BE GIVEN
1260	002030	070744	DH61	; PC
1261				; TEST NO.
1262				; PC OF JSR
1263				; RHCS1 CONTENTS
1264	002032	071716	DT61	; \$ERRPC, \$STNM, PCJSR, \$BDADR
1265	002034	072106	DF61	; 0,0,0,0
1266				

Line	Code	Address	Item	Device	Description
1267			:*ITEM 62		
1268	002036	063246		EM62	:RHDS1 HAS STATUS BITS STUCK AT ONE
1269	002040	071037		DH62	:PC
1270					:TEST NO.
1271					:PC OF JSR
1272					:RHDS1 CONTENTS
1273	002042	071730		DT62	:\$ERRPC,\$TSTNM,PCJSR,\$BDADR
1274	002044	072112		DF62	:0,0,0,0
1275					
1276					
1277			:*ITEM 63		
1278	002046	063327		EM63	:RHDS1 CONTENTS DURING
1279					:COMMAND WAS IN ERROR
1280	002050	067500		DH5	
1281	002052	071546		DT5	
1282	002054	072032		DF5	
1283					
1284					
1285			:*ITEM 64		
1286	002056	063403		EM64	:RECALIBRATE COMMAND CAUSED
1287					:IMPROPER REGISTER CHANGE.
1288					:GOOD DATA GIVES WHAT SHOULD BE
1289					:THERE.
1290					:RECEIVED DATA GIVES WHAT WAS THERE
1291					:AFTER COMMAND
1292	002060	067500		DH5	
1293	002062	071546		DT5	
1294	002064	072032		DF5	
1295					
1296					
1297			:*ITEM65		
1298					
1299	002066	063622		EM65	: INTERRUPT FAILING
1300	002070	071132		DH65	:PC
1301					:TEST NO
1302					:CONTENTS OF RHCS1
1303					:CONTENTS OF RHAS
1304					:CONTENTS OF RHDS1
1305	002072	071742		DT65	:\$ERRPC,TSTNM,CS1,AS,DS1
1306	002074	072116		DF65	:0,0,0,0,0
1307					
1308					
1309			:*ITEM66		
1310	002076	063644		EM66	:HEADER AND DATA COMMAND
1311					:FOR HEAD SELECTION TEST
1312					:CAUSED AN ERROR
1313					:RHDST GIVES WHAT TRACK
1314					:WAS BEING WRITTEN ON CYLINDER 0
1315					:SECTOR 0
1316	002100	071255		DH66	:PC
1317					:RHDST
1318					:RHER1
1319					:RHER2
1320					:RHER3
1321					:RHCS1
1322					:RHCS2

1323	002102	071756		DT66	:SERRPC,DST,ER1,ER2,ER3,CS1,CS2
1324	002104	072123		DF66	:0,0,0,0,0,0,0
1325			:*ITEM67		
1326	002106	064036		EM67	:READ HEADER AND DATA ERROR
1327					:IN HEAD SELECTION TEST
1328					:FIRST FOUR WORDS GIVE HEADER
1329					:NEXT WORDS ARE DATA
1330					:GOOD DATA WORDS GIVE
1331					:THE TRACK NUMBER IN
1332					:BITS 4,5,6,7,8
1333	002110	070334		DH30	
1334	002112	071646		DT30	
1335	002114	072065		DF30	
1336			:*ITEM70		
1337	002116	064326		EM70	:READ HEADER AND DATA ERROR
1338					:IN DIFFERENCE LINE TEST
1339					:WORD NOS. 1-4 GIVE
1340					:HEADER
1341					:WORD NOS. 5-260 GIVE DATA
1342					:WHICH IS THE CYLINDER
1343					:ADDRESS
1344	002120	070334		DH30	
1345	002122	071646		DT30	
1346	002124	072065		DF30	
1347					
1348			:*ITEM 71		
1349	002126	064534		EM71	:FORCING OPI CAUSED IMPROPER REGISTER
1350					: CHANGE
1351					:GOOD DATA GIVES WHAT SHOULD
1352					:BE THERE
1353					:RECEIVED DATA GIVES WHAT WAS
1354					:THERE AFTER 3 INDEX PULSES
1355	002130	067500		DH5	:PC
1356					:REG. ADDR.
1357					:GOOD DATA
1358					:RECEIVED DATA
1359	002132	071546		DT5	:SERRPC,REGADR,\$GDDAT,\$BDDAT
1360	002134	072032		DF5	:0,0,0,0
1361			:*ITEM 74		
1362	002136	065277		EM74	:WHILE USING UNIBUS B
1363					:READ DATA CAUSED IMPROPER REGISTER
1364					:CHANGE
1365					:GOOD DATA GIVES WHAT SHOULD BE THERE
1366					:RECEIVED DATA GIVES WHAT WAS THERE AFTER
1367					:COMMAND
1368	002140	067500		DH5	
1369	002142	071546		DT5	
1370	002144	072032		DF5	
1371					
1372			:*ITEM 73		
1373	002146	065225		EM73	:WHILE USING UNIBUS B
1374					:READ DATA INCORRECT
1375	002150	070334		DH30	
1376	002152	071646		DT30	
1377	002154	072065		DF30	
1378					

Line	Code	Address	Item	Register	Description
1379			*ITEM 74		
1380	002156	065277		EM74	:WHILE USING UNIBUS B
1381					:WRITE DATA COMMAND CAUSED
1382					:IMPROPER REGISTER CHANGE
1383					:GOOD DATA GIVES WHAT SHOULD BE THERE
1384					:RECEIVED DATA GIVES REGISTER
1385					:CONTENTS AFTER WRITE DATA
1386	002160	067500		DH5	
1387	002162	071546		DT5	
1388	002164	072032		DF5	
1389					
1390			*ITEM 75		
1391	002166	065543		EM75	:WHILE USING UNIBUS B
1392					:WRITE DATA COMMAND CHANGED
1393					:WRITE FROM BUFFER
1394	002170	070334		DH30	
1395	002172	071646		DT30	
1396	002174	072065		DF30	
1397					
1398			*ITEM 76		
1399	002176	065646		EM76	:WHILE USING UNIBUS B
1400					:WRITE CHECK CAUSED AN
1401					:IMPROPER REGISTER CHANGE
1402					:GOOD DATA GIVES WHAT SHOULD
1403					:BE THERE
1404					:RECEIVED DATA GIVES WHAT WAS
1405					:THERE AFTER COMMAND
1406	002200	067500		DH5	
1407	002202	071546		DT5	
1408	002204	072032		DF5	
1409					
1410			*ITEM 77		
1411	002206	066103		EM77	:CURRENT CYLINDER DOES NOT REFLECT DESIRED 'RHCC'
1412	002210	071353		DH77	:PC
1413					:PC OF JSR
1414					:REGISTER ADDRESS
1415					:GOOD DATA
1416					:BAD DATA
1417	002212	071776		DT77	:SERRPC,PCJSR,REGADR,\$GDDAT,\$BDDAT
1418	002214	072133		DF77	:0,0,0,0,0
1419					
1420			*ITEM 100		
1421	002216	066326		EM100	:ERROR AFTER ADDRESS PLUG CHANGE
1422	002220	067500		DH5	:PC
1423					:REGISTER ADDRESS
1424					:GOOD DATA
1425					:RECEIVED DATA
1426	002222	071546		DT5	:SERRPC,REGADR,\$GDDAT,\$BDDAT
1427	002224	072032		DF5	:0,0,0,0
1428					
1429			*ITEM 101		
1430	002226	066410		EM101	:UNIT DID NOT GO OFFLINE WHEN ADDR
1431					:PLUG WAS REMOVED
1432	002230	070135		DH26	:PC
1433					:CONT OF RHCS1
1434					:CONT OF RHCS2

1435				:CONT OF RHDS1
1436				:CONT OF RHER2
1437				:CONT OF RHER3
1438	002232	071624	DT26	:\$ERRPC,CS1,CS2,DS1,ER2,ER3
1439	002234	072055	DF26	:0,0,0,0,0,0,0
1440				
1441			:*ITEM 102	
1442	002236	066472	EM102	:UNIT DID NOT COME BACK ONLINE WHEN
1443				:ADDR PLUG WAS REPLACED
1444	002240	070135	DH26	:PC
1445				:CONT OF RHCS1
1446				:CONT OF RHCS2
1447				:CONT OF RHDS1
1448				:CONT OF RHER2
1449				:CONT OF RHER3
1450	002242	071624	DT26	:\$ERRPC,CS1,CS2,DS1,ER2,ER3
1451	002244	072055	DF26	:0,0,0,0,0,0,0
1452				
1453			:*ITEM 103	
1454	002246	066551	EM103	:REGISTER CONTENTS INCORRECT BEFORE A
1455				:DIAGNOSTIC SEEK
1456	002250	070135	DH26	:PC
1457				:CONT OF RHCS1
1458				:CONT OF RHCS2
1459				:CONT OF RHDS1
1460				:CONT OF RHER2
1461				:CONT OF RHER3
1462	002252	071624	DT26	:\$ERRPC,CS1,CS2,DS1,ER2,ER3
1463	002254	072055	DF26	:0,0,0,0,0,0,0
1464				
1465			:*ITEM 104	
1466	002256	066635	EM104	:REGISTER CONTENTS INCORRECT AFTER A
1467				:DIAGNOSTIC SEEK
1468	002260	070135	DH26	:PC
1469				:CONT OF RHCS1
1470				:CONT OF RHCS2
1471				:CONT OF RHDS1
1472				:CONT OF RHER2
1473				:CONT OF RHER3
1474	002262	071624	DT26	:\$ERRPC,CS1,CS2,DS1,ER2,ER3
1475	002264	072055	DF26	:0,0,0,0,0,0,0
1476				


```
1477
1478           ;*RH11 REGISTERS
1479
1480
1481 002266 000254      RPVEC: 254           ;RP04 VECTOR ADDRESS
1482
1483
1484
1485           ;*WORD COUNT REGISTER (RHWC)
1486           ;*EACH BIT IS CALLED BY BIT NUMBER
1487
1488
1489
1490           ;*BUS ADDRESS REGISTER (RHBA)
1491           ;*EACH BIT IS CALLED BY BIT NUMBER
1492
1493
1494
1495           ;*CONTROL AND STATUS REGISTER 2 (RHCS2)
1496
1497           000001      US1= 1           ;UNIT SELECT (BIT #0)
1498           000002      US2= 2           ;UNIT SELECT (BIT #1)
1499           000004      US4= 4           ;UNIT SELECT (BIT #2)
1500           000010      BAI= 10          ;BUS ADDRESS INCREMENT INHIBIT (BIT #3)
1501           000020      UNIB= 20         ;UNIBUS B DC LO (BIT #4)
1502           000040      CLR= 40          ;CLEAR (BIT #5)
1503           000100      IR= 100          ;INPUT READY (BIT #6)
1504           000200      CR= 200          ;OUTPUT READY (BIT #7)
1505           000400      MPE= 400         ;MASS BUS PARITY ERROR (BIT #8)
1506           001000      MXF= 1000        ;MISSED TRANSFER ERROR (BIT #9)
1507           002000      PGE= 2000       ;PROGRAM ERROR (BIT #10)
1508           004000      NEM= 4000        ;NON EXISTANT MEMORY (BIT #11)
1509           010000      NED= 10000       ;NON EXISTANT DRIVE (BIT #12)
1510           020000      UPE= 20000      ;UNIBUS PARITY ERROR (BIT #13)
1511           040000      WCE= 40000      ;WRITE CHECK ERROR (BIT #14)
1512           100000      DLT= 100000     ;DATA LATE (BIT #15)
1513
1514           ;*DATA BUFFER REGISTER (RHDB)
1515           ;*EACH BIT IS CALLED BY BIT NUMBER
1516
1517
```

```

1518 ;*RP04 REGISTERS
1519
1520
1521
1522 ;*CONTROL AND STATUS 1 REGISTER. (#00)
1523
1524 000001 GO= 1 ;GO (BIT #0)
1525 000100 IE= 100 ;INTERRUPT ENABLE (BIT #6)
1526 000200 RDY= 200 ;READY (BIT #7)
1527 000400 A16= 400 ;HIGH ORDER UNIBUS BITS (BIT #8)
1528 001000 A17= 1000 ;HIGH ORDER UNIBUS BITS (BIT #9)
1529 002000 PSEL= 2000 ;PORT SELECT (BIT #10)
1530 004000 DVA= 4000 ;DEVICE AVAILABLE (BIT #11)
1531 020000 MCPE= 20000 ;MASSBUSS PARITY ERROR (BIT #13)
1532 040000 TRE= 40000 ;TRANSFER ERROR (BIT #14)
1533 100000 SC= 100000 ;SPECIAL CONDITION (BIT #15)
1534
1535 ;*STATUS REGISTER (RHDS1) (#01)
1536
1537 000001 DFF5= 1 ;DRIVE FORWARD 5'/SEC. (BIT #0)
1538 000002 DFF20= 2 ;DRIVE FORWARD 20'/SEC. (BIT #1)
1539 000004 DIGB= 4 ;DRIVE TO INNER GAVRD BAND (BIT #2)
1540 000010 GRV= 10 ;GO REVERSE (BIT #3)
1541 000020 DL64= 20 ;DIFFERENCE LESS THAN 64 (BIT #4)
1542 000040 DE1= 40 ;DIFFERENCE EQUALS 1 (BIT #5)
1543 000100 VV= 100 ;VOLUME VALID (BIT #6)
1544 000200 DRY= 200 ;DRIVE READY (BIT #7)
1545 000400 DPR= 400 ;DRIVE PRESENT (BIT #8)
1546 001000 PROG= 1000 ;PROGRAMABLE (BIT #9)
1547 002000 LBT= 2000 ;LAST SECTOR TRANSFERRED (BIT #10)
1548 004000 WRL= 4000 ;WRITE LOCK (BIT #11)
1549 010000 MOL= 10000 ;MEDIUM ON-LINE (BIT #12)
1550 020000 PIP= 20000 ;POSITIONING OPERATION IN PROGRESS (BIT #13)
1551 040000 ERR= 40000 ;COMPOSIT ERROR. (BIT #14)
1552 100000 ATA= 100000 ;ATTENTION ACTIVE (BIT #15)
1553
1554 ;*ERROR REGISTER #01 (RHER1) (#02)
1555 000001 ILF 1 ;ILLEGAL FUNCTION (BIT #0)
1556 000002 ILR= 2 ;ILLEGAL REGISTER (BIT #1)
1557 000004 RMR= 4 ;REGISTER MODIFICATION REFUSED (BIT #2)
1558 000010 PAR= 10 ;PARITY ERROR (BIT #3)
1559 000020 FER= 20 ;FORMAT ERROR (BIT #4)
1560 000040 WCF= 40 ;WRITE CLOCK FAIL (BIT #5)
1561 000100 ECH= 100 ;ECC HARD ERROR (BIT #6)
1562 000200 HCE= 200 ;HEADER COMPARE ERROR (BIT #7)
1563 000400 HCRC= 400 ;HEADER CRC ERROR (BIT #8)
1564 001000 AOE= 1000 ;ADDRESS OVERFLOW ERROR (BIT #9)
1565 002000 IAE= 2000 ;INVALID ADDRESS ERROR (BIT #10)
1566 004000 WLE= 4000 ;WRITE LOCK ERROR (BIT #11)
1567 010000 DTE= 10000 ;DRIVE TIMING ERROR (BIT #12)
1568 020000 OPI= 20000 ;OPERATION INCOMPLETE (BIT #13)
1569 040000 UNS= 40000 ;DRIVE UNSAFE (BIT #14)
1570 100000 DCK= 100000 ;DATA CHECK ERROR (BIT 15)
1571
1572 ;*MAINTAINABILITY REGISTER (RHMR) (#03)
1573

```

```
1574      000001      DMD= 1      ;DIAGINOSTIC MODE (BIT #0)
1575      000002      MCLK= 2     ;MAINTAINABILITY CLOCK (BIT #1)
1576      000004      MINX= 4     ;MAINTAINABILITY INDEX (BIT #2)
1577      000010      MSTCK= 10   ;MAINTAINABILITY SECTOR CLOCK (BIT #3)
1578      000020      MRD= 20    ;MAINTAINABILITY READ (BIT #4)
1579      000040      MWR= 40    ;MAINTAINABILITY WRITE (BIT #5)
1580      001000      DTSY= 1000 ;MAINTAINABILITY SYNC DETECTED (BIT #9)
1581
1582      ;*ATTENTION SUMMARY PSEUDO-REGISTER (RHAS) (#04)
1583
1584      000001      AT0= 1      ;DEVICE 0 (BIT #0)
1585      000002      AT1= 2      ;DEVICE 1 (BIT #1)
1586      000004      AT2= 4      ;DEVICE 2 (BIT #2)
1587      000010      AT3= 10     ;DEVICE 3 (BIT #3)
1588      000020      AT4= 20     ;DEVICE 4 (BIT #4)
1589      000040      AT5= 40     ;DEVICE 5 (BIT #5)
1590      000100      AT6= 100    ;DEVICE 6 (BIT #6)
1591      000200      AT7= 200    ;DEVICE 7 (BIT #7)
1592
1593      ;*DESIRED SECTOR/TRACK ADDRESS REGISTER (RHDST) (#1)
1594      ;*EACH BIT IS CALLED BY BIT NUMBER
1595
1596
1597
1598
1599
1600      ;*DRIVE TYPE REGISTER (RHDT) (#06)
1601      ;*EACH BIT IS CALLED BY BIT NUMBER
1602
1603
1604
1605
1606
1607      ;*LOOK-AHEAD REGISTER (RHLA) (#07)
1608
1609      000001      EXT1= 1     ;EXTENSION 1 (BIT #0)
1610      000002      EXT2= 2     ;EXTENSION 2 (BIT #1)
1611      000004      EXT4= 4     ;EXTENSION 3 (BIT #2)
1612      000010      EXT10= 10    ;EXTENSION 4 (BIT #3)
1613      000020      EXT20= 20    ;EXTENSION 5 (BIT #4)
1614      000040      EXT40= 40    ;EXTENSION 6 (BIT #5)
1615      000100      SC1= 100    ;SECTOR COUNT FIELD 0 (BIT #6)
1616      000200      SC2= 200    ;SECTOR COUNT FIELD 1 (BIT #7)
1617      000400      SC4= 400    ;SECTOR COUNT FIELD 2 (BIT #8)
1618      001000      SC10= 1000 ;SECTOR COUNT FIELD 3 (BIT #9)
1619      002000      SC20= 2000 ;SECTOR COUNT FIELD 4 (BIT #10)
1620      004000      TRK1= 4000 ;TRACK FIELD 1 (BIT #11)
1621      010000      TRK2= 10000 ;TRACK FIELD 2 (BIT #12)
1622      020000      TRK4= 20000 ;TRACK FIELD 3 (BIT #13)
1623      040000      TRK10= 40000 ;TRACK FIFLD 4 (BIT #14)
1624      100000      TRK20= 100000 ;TRACK FIELD 5 (BIT #15)
1625
1626      ;*ERROR REGISTER #2 (RHER2) (#10)
1627
1628      000001      WCU= 1     ;WRITE CURRENT UNSAFE (BIT #0)
1629      000002      CSF= 2     ;CURRENT SINK FAILURE (BIT #1)
```

1630	000004	WSU= 4	:WRITE SELECT UNSAFE (BIT #2)
1631	000010	CSU= 10	:CURRENT SWITCH UNSAFE (BIT #3)
1632	000020	MSE= 20	:MOTOR SEQUENCE ERROR (BIT #4)
1633	000040	TDF= 40	:TRANSITIONS DETECTOR FAILURE (BIT #5)
1634	000100	TUF= 100	:TRANSITIONS UNSAFE (BIT #6)
1635	000200	FEN= 200	:FAILSAFE ENABLED (BIT #7)
1636	000400	WRU= 400	:WRITE READY UNSAFE (BIT #8)
1637	001000	MHS= 1000	:MULTIPLE HEAD SELECT (BIT #9)
1638	002000	NHS= 2000	:NO HEAD SELECTION (BIT #10)
1639	004000	IXE= 4000	:INDEX ERROR (BIT #11)
1640	010000	VU30= 10000	:30VOLT UNSAFE (BIT #12)
1641	020000	PLU= 20000	:PLO UNSAFE (BIT #13)
1642	100000	ACU= 100000	:ACUNSAFE (BIT #15)
1643			
1644		;*OFFSET REGISTER (RHOF) (#11)	
1645			
1646	000001	OF25= 1	:OFFSET 25 MICRO INCHES (BIT #0)
1647	000002	OF50= 2	:OFFSET 50 MICRO INCHES (BIT #1)
1648	000004	OF100= 4	:OFFSET 100 MICRO INCHES (BIT #2)
1649	000010	OF200= 10	:OFFSET 200 MICRO INCHES (BIT #3)
1650	000020	OF400= 20	:OFFSET 400 MICRO INCHES (BIT #4)
1651	000040	OF800= 40	:OFFSET 800 MICRO INCHES (BIT #5)
1652			
1653	000200	OFREV= 200	:OFFSET NEGATIVE (REVERSE) (BIT #5)
1654	002000	HCI= 2000	:HEADER COMPARE INHIBIT (BIT #10)
1655	004000	ECI= 4000	:ERROR CORRECTION CODE INHIBIT (BIT #11)
1656	010000	FMT22= 10000	:FORMAT BIT (BIT #12)
1657			
1658		;*DESIRED CYLINDER ADDRESS (RHCA) (#12)	
1659		;*EACH BIT IS CALLED BY BIT NUMBER.	
1660			
1661			
1662			
1663			
1664		;*CURRENT CYLINDER ADDRESS (RHCC) (#13)	
1665		;*EACH BIT IS CALLED BY BIT NUMBER	
1666			
1667			
1668			
1669			
1670		;*SERIAL NUMBER REGISTER (RHSN) (#14)	
1671		;*EACH IS CALLED BY BIT NUMBER	
1672			
1673			
1674			
1675			
1676		;*ERROR REGISTER #03 (RHER3) (#15)	
1677			
1678	000001	PSU= 1	:PACK SPEED UNSAFE (BIT #0)
1679	000002	VUF= 2	:VELOCITY UNSAFE (BIT #1)
1680	000010	UWR= 10	:ANY UNSAFE EXCEPT READ/WRITE (BIT #3)
1681	000020	PRE= 20	:DISK PACK ROTATION ERROR (BIT #4)
1682	000040	ACL= 40	:AC LOW (BIT #5)
1683	000100	DCL= 100	:DC LOW (BIT #6)
1684	020000	ACE= 20000	:ADDRESS CHANGE ERROR (BIT #13)
1685	040000	SKI= 40000	:SEEK INCOMPLETE (BIT #14)

1686 100000 OCYL= 100000 ;OFF CYLINDER (BIT #15)

1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706

;*ECC POSITION REGISTER (RHEC1) (#16)
;*EACH BIT IS CALLED BY BIT NUMBER

;*ECC PATTERN REGISTER (RHEC2) (#17)
;*EACH BIT IS CALLED BY BIT NUMBER


```
1821
1822
1823          ;*RESERVED CORE LOCATIONS
1824
1825 004600 000000      REGADR: 0          ;SAVE REGISTER ADDRESS HERE
1826 004602 000000      ERWORD: 0         ;SAVE ERROR WORD NUMBER HERE
1827 004604 000000      TSTNM: 0          ;TEST NUMBER
1828 004606 000000      RP4VEC: 0         ;CONTAINS ADDRESS OF LOCATION
1829                                     ;WHERE AN RP04 INTERRUPT IS TO VECTOR TO
1830                                     ;THIS MUST BE MOVED INTO 'RPVEC' TO BE
1831                                     ;EFFECTIVE.
1832
1833 004610 000000      OFSTVL: 0          ;OFFSET VALUE USED IN OFFSET TEST
1834
1835
1836 004612 000024      SAVERE: .BLKW 20.   ;BLOCK TO SAVE REGISTERS FOR PRETEST
1837                                     ;HARDWARE REGISTER SNAPSHOTS - THESE
1838                                     ;ARE USUALLY THEN CHANGED TO REFLECT
1839                                     ;EXPECTED CONDITIONS AFTER THE TEST
1840 004662 000000      FINALA: 0          ;SAVE LOOK AHEAD REGISTER AT END OF OPERATION
1841 004664 000000      FINACC: 0         ;SAVE CURRENT CYLINDER REGISTER AT END OF OPERATION
1842
1843
1844          ;*TABLE FOR ATTENTION BITS
1845          ;*ATTENTION TABLE
1846
1847 004666      001      002      004      ATABLE: .BYTE 1,2,4,10,20,40,100,200
1848 004671      010      020      040
1849 004674      100      200
1850
```

```
1851
1852
1853
1854
1855 004676 000010 UNITS: .BLKW 8. ;THIS IS FILLED WITH -1
1856 004716 000000 UNIT: .WORD 0 ;UNIT UNDER TEST
1857 004720 000000 NOUNIT: .WORD 0 ;NUMBER OF UNITS PRESENT
1858 ;USED TO KEEP TRACK OF UNIT UNDER TEST
1859 004722 000000 NUNIT: .WORD 0 ;USED TO DETERMIN IF THERE ARE MORE
1860 ;THAN ONE UNIT
1861 004724 000000 NOPUSH: 0 ;ALL ONES INDICATE NONE OF THE OPERATOR
1862 ;INTERVENTION TESTS WILL BE PERFORMED
1863 004726 000000 SELECT: .WORD 0 ;ALL ONES INDICATE UNIT TO BE SELECTED
1864 004730 000000 UNITSL: .WORD 0 ;UNIT NO. SELECTED
1865 004732 000000 UBUSB: 0 ;IF ZERO UNIBUS PRESENT
1866 ;IF ONES NO UNIBUS B
1867 004734 000000 ERFLG$: 0 ;ERROR FLAG
1868 004736 000000 FIRST: 0 ;IF ZERO WILL TYPE HEADER
1869 ;IF ONES WILL NOT TYPE HEADER
1870
1871 004740 000000 ATTENT: 0 ;ATTENTION BIT FOR PRESENT UNIT
1872 004742 000000 TOTALAT: 0 ;TOTAL ATTENTION BITS
1873
1874 004744 000000 RP06: 0 ;RP06 DEVICE TYPE FLAG LOCATION
1875 004746 000000 RP05: 0 ;MEMOREX RP04 DEVICE TYPE FLAG
1876 004750 000000 RH70: 0 ;IF = 1, PROGRAM IS RUNNING ON RWPO4 SYSTEM
1877 ;IF = 0, PROGRAM IS RUNNING ON RJPO4
1878 004752 000000 INUNIT: 0 ;INITIAL UNIT NO. - USED DURING
1879 ;CHECKING ALL ADDRESS PLUG ADDRESSES
1880
1881
1882
1883
1884
1885 004754 000000 TMP0: .WORD 0 ;TEMP STORAGE
1886 004756 000000 TMP1: .WORD 0
1887 004760 000000 TMP4: .WORD 0 ;TEMP STORAGE
```

```

1888
1889      .SBTTL
1890      .SBTTL  **DIAGNOSTIC CODE**
1891      .SBTTL
1892
1893      .SBTTL  SETUP TESTS
1894
1895
1896 004762 012737 177777 004724 BEGIN1: MOV    #-1,@#NOPUSH ;JUMP OVER OPERATOR REQUIRED TESTS
1897 004770 005037 004726          CLR    @#SELECT ;DO NOT SELECT UNIT
1898 004774 000412          BR     START
1899 004776 012737 177777 004726 BEGIN2: MOV    #-1,@#SELECT ;SELECT UNIT
1900 005004 005037 004724          CLR    @#NOPUSH ;DO NOT JUMP OVER ANY TEST
1901 005010 000404          BR     START
1902 005012 005037 004726          BEGIN: CLR    @#SELECT ;DO NOT SELECT UNIT
1903 005016 005037 004724          CLR    @#NOPUSH ;DO NOT JUMP OVER ANY OPERATOR
1904                                     ;INTERVENTION TESTS - NORMAL RUN
1905
1906 005022          START:
1907 005022 000005          RESET
1908 005024 012706 001100          MOV    #SCMTAG,R6 ;:FIRST LOCATION TO BE CLEARED
1909 005030 005026          CLR    (R6)+ ;:CLEAR MEMORY LOCATION
1910 005032 022706 001140          CMP    #SWR,R6 ;:DONE?
1911 005036 001374          BNE    -6 ;:LOOP BACK IF NO
1912 005040 012706 001000          MOV    #STACK,SP ;:SETUP THE STACK POINTER
1913 005044 012737 044730 000020          MOV    #SCOPE,@#IOTVEC ;:IOT VECTOR FOR SCOPE ROUTINE
1914 005052 012737 000340 000022          MOV    #'0,@#IOTVEC+2 ;:LEVEL 7
1915 005060 012737 047126 000030          MOV    #ERROR,@#EMTVEC ;:EMT VECTOR FOR ERROR ROUTINE
1916 005066 012737 000340 000032          MOV    #340,@#EMTVEC+2 ;:LEVEL 7
1917 005074 012737 050652 000034          MOV    #STRAP,@#TRAPVEC ;:TRAP VECTOR FOR TRAP CALLS
1918 005102 012737 000340 000036          MOV    #340,@#TRAPVEC+2 ;:LEVEL 7
1919 005110 012737 050736 000024          MOV    #SPWRDN,@#PWRVEC ;:POWER FAILURE VECTOR
1920 005116 012737 000340 000026          MOV    #340,@#PWRVEC+2 ;:LEVEL 7
1921 005124 005037 001212          CLR    $TIMES ;:INITIALIZE NUMBER OF ITERATIONS
1922 005130 005037 001214          CLR    $ESCAPE ;:CLEAR THE ESCAPE ON ERROR ADDRESS
1923 005134 112737 000001 001115          MOV    #1,$ERMAX ;:ALLOW ONE ERROR PER TEST
1924 005142 012737 005142 001106          MOV    #,$LPADR ;:INITIALIZE THE LOOP ADDRESS FOR SCOPE
1925 005150 012737 005150 001110          MOV    #,$LPERR ;:SETUP THE ERROR LOOP ADDRESS
1926 005156 013746 000004          MOV    @#ERRVEC,-(SP) ;:SAVE ERROR VECTOR
1927 005162 012737 005216 000004          MOV    #64$,@#ERRVEC ;:SET UP ERROR VECTOR
1928 005170 012737 177570 001140          MOV    #DSWR,SWR ;:SETUP FOR A HARDWARE SWICH REGISTER
1929 005176 012737 177570 001142          MOV    #DDISP,DISPLAY ;:AND A HARDWARE DISPLAY REGISTER
1930 005204 022777 177777 173726          CMP    #-1,@SWR ;:TRY TO REFERENCE HARDWARE SWR
1931 005212 001012          BNE    66$ ;:BRANCH IF NO TIMEOUT TRAP OCCURRED
1932 005214 000403          BR     65$ ;:BRANCH IF NO TIMEOUT
1933 005216 012716 005224          64$: MOV    #65$,(SP) ;:SET UP FOR TRAP RETURN
1934 005222 000002          RTI
1935 005224 012737 000176 001140          65$: MOV    #SWREG,SWR ;:POINT TO SOFTWARE SWR
1936 005232 012737 000174 001142          MOV    #DISPREG,DISPLAY
1937 005240 012637 000004          66$: MOV    (SP)+,@#ERRVEC ;:RESTORE ERROR VECTOR
1938
1939
1940
1941 005244 012737 000000 177776          MOV    #0,PS ;:SET PROCESSOR STATUS TO 0
1942 005252 012737 000200 000036          MOV    #200,@#TRAPVEC+2 ;:TRAP PRIORITY = 4
1943 005260 013700 002266          MOV    @#RPVEC,R0 ;:GET RP VECTOR ADDRESS
    
```

```

1944 005264 012720 044666      MOV      #RPVECT,(R0)+      ;THIS IS FOR UNTIMELY INTERRUPTS
1945 005270 012710 000340      MOV      #340,(R0)         ;DRIVE INTERRUPT SERVICE ROUTINE
1946                                     ;PRIORITY = 7
1947
1948 005274 004737 045666      JSR      PC,@#STKINT       ;INITIALIZE THE TTY KEYBOARD
1949 005300 005737 004736      TST      @#FIRST          ;IS THIS FIRST TIME ROUND ?
1950 005304 001001                BNE      1$                ;DON'T GIVE HEADER IF NOT
1951 005306 000402                BR       2$                ;HEADER 1ST TIME THROUGH
1952 005310 000137 006110      1$:     JMP      @#SND1        ;NO HEADER
1953
1954                                     2$:
1955 005314 104401 005322      TYPE    ,68$              ;;TYPE ASCIZ STRING
1956 005320 000435                BR       67$              ;;GET OVER THE ASCIZ
1957 005414 104401 005422      TYPE    ,70$              ;;TYPE ASCIZ STRING
1958 005420 000417                BR       69$              ;;GET OVER THE ASCIZ
1959
1960 005460 104401 005466      TYPE    ,72$              ;;TYPE ASCIZ STRING
1961 005464 000435                BR       71$              ;;GET OVER THE ASCIZ
1962 005560 104401 005566      TYPE    ,74$              ;;TYPE ASCIZ STRING
1963 005564 000433                BR       73$              ;;GET OVER THE ASCIZ
1964 005654 104401 005662      TYPE    ,76$              ;;TYPE ASCIZ STRING
1965 005660 000435                BR       75$              ;;GET OVER THE ASCIZ
1966
1967 005754 104401 005762      TYPE    ,78$              ;;TYPE ASCIZ STRING
1968 005760 000426                BR       77$              ;;GET OVER THE ASCIZ
1969 006036 104401 006044      TYPE    ,80$              ;;TYPE ASCIZ STRING
1970 006042 000422                BR       79$              ;;GET OVER THE ASCIZ
1971
1972 006110 012737 177777 004736  SND1:  MOV      #-1,@#FIRST      ;NEXT TIME DO NOT GIVE HEADER
1973
1974 006116 005737 000042                TST      @#42              ;;ARE WE RUNNING UNDER XXDP/ACT?
1975 006122 001006                BNE      64$              ;;BRANCH IF YES
1976 006124 023727 001140 000176      CMP      SWR,#SWREG        ;SOFTWARE SWITCH REG SELECTED?
1977 006132 001005                BNE      65$              ;;BRANCH IF NO
1978 006134 104406                GTSWR                    ;;GET SOFT-SWR SETTINGS
1979 006136 000403                BR       65$
1980 006140 112737 000001 001134 64$:  MOV      #1,$AUTOB        ;;SET AUTO-MODE INDICATOR
1981
1982 006146 032777 010000 172764  RH70CK: BIT      #SW12,@SWR      ;LOOK TO SEE IF USING RH70
1983 006154 001403                BEQ      3$                ;IF SW12 = 0, SKIP NEXT
1984 006156 012737 000001 004750      MOV      #1,@#RH70        ;IF SW12 = 1, CU IS AN RH70
1985 006164
1986                                     3$:
1987                                     ;*IS THERE A P-CLOCK (KW11-P) ON THE SYSTEM
1988                                     ;*IF SO MAKE 'WAT' TRAPS GO TO 'WAIT.P'
1989                                     ;*IF SO MAKE RP04 INTERRUPTS GO TO 'TIME 1'
1990                                     ;*IF NOT MAKE 'WAT' TRAPS GO TO 'WAIT.T'
1991                                     ;*IF NOT MAKE RP04 INTERRUPTS GO TO 'TIME 2'
1992
1993                                     ;*THE NEXT LINE IS TO BE ADDED LATER
1994                                     ;*AND THE JUMP AND NOP REMOVED
1995                                     ;*FOR NOW NO CLOCK WILL BE USED
1996
1997                                     :
1998                                     MOV      @#1$,@#ERRVEC    ;SET TIME-OUT VECTOR
1999                                     :
2000                                     JMP      @#1$            ;DO NOT USE CLOCK
2001                                     :
2002                                     NOP
  
```

```
2000      :      TST      @#PCLCSR      ;REFERENCE P-CLOCK STATUS REGISTER
2001      :      :      :      :      :ADDRESS = 172540
2002      :      :      :      :      :THERE IS A P-CLOCK
2003      :      :      :      :      :THERE IS A P CLOCK SO
2004      :      :      :      :      :VECTOR TO TIME1
2005      :      :      :      :      :
2006      :1$:      :      :      :      :
2007      :      :      :      :      :
2008      :      :      :      :      :
2009      :      :      :      :      :
2010 006164 012737 041670 004606      :      :      :      :      :MOV      #TIME2,@#RP4VEC ;RP04/5/6 INTERRUPTS GO TO 'TIME 2'
2011 006172 012737 177777 047274 2$:      :      :      :      :MOV      #-1,@#PRITEM ;CLEAR PREVIOUS ITEM NUMBER
2012
2013
2014
2015
2016 006200 005737 004726      :      :      :      :      :TST      @#SELECT      ;WAS IT A 200 START
2017 006204 001442      :      :      :      :      :BEQ      TST1 ;      ;BRANCH IF STARTING FROM 200
2018 006206 104401 006214      :      :      :      :      :TYPE      ,65$      ;      ;TYPE ASCIZ STRING
2019 006212 000424      :      :      :      :      :BR      64$      ;      ;GET OVER THE ASCIZ
2020 006264 104412      :      :      :      :      :RDOCT
2021 006266 042716 177770      :      :      :      :      :BIC      #177770,(SP) ;ONLY KEEP LAST 3 BITS
2022 006272 011637 004716      :      :      :      :      :MOV      (SP),@#UNIT ;SAVE UNIT TO BE TESTED
2023 006276 012637 004730      :      :      :      :      :MOV      (SP)+,@#UNITSL ;SAVE UNIT TO BE TESTED
2024
2025
2026
2027 006302 001403      :      :      :      :      :BEQ      TST1 ;      ;BRANCH IF STARTING FROM 200
2028 006304 013737 004730 004716      :      :      :      :      :MOV      @#UNITSL,@#UNIT ;SET UNIT NUMBER
2029
```

```

2030
2031 006312 000004          TST1:  SCOPE
2032 006314 012737 000001 001212  MOV    #1,$TIMES          ;;DO 1 ITERATION
2033 006322 012737 000001 004604  MOV    #2-1,@#TSTNM      ;;THIS SAVES TEST NUMBER
2034 006330 012706 001000          MOV    #STACK, SP        ;;SET UP STACK POINTER
2035 006334 012737 047136 000030  MOV    #REGSA1,@#EMTVEC  ;;ERROR VECTOR SO THAT
2036                                ;;NO REGISTERS ARE SAVED
2037 006342 012737 006370 000004  MOV    #2$,@#ERRVEC      ;;SET UP FOR BUS TIMEOUT
2038
2039 006350 012700 000024          MOV    #24,R0            ;THERE ARE 24 REG TO TEST
2040 006354 012701 002270          MOV    #RHDB,R1         ;R1 NOW HAS ADDR OF ADDR OF FIRST REG.
2041 006360 013102          1$:  MOV    @(R1)+,R2        ;READ HARDWARE REG.
2042 006362 005300          DEC    R0                ;COUNT DOWN
2043 006364 001375          BNE    1$                ;BRANCH IF 24 NOT DONE
2044 006366 000454          BR     3$                ;BRANCH IF 24 DONE
2045 006370 012737 000006 000004  2$:  MOV    #ERRVEC+2,@#ERRVEC ;RESTORE TRAP CATCHER
2046 006376 022626          CMP    (SP)+,(SP)+      ;CLEAN STACK
2047 006400 016137 177776 001200  MOV    -2(R1),$TMP1     ;STORE FAILING REG ADDR
2048 006406 104007          ERROR 7                ;REGISTER NON EXISTANT
2049 006410 032777 020000 172522  BIT    #SW13,@SWR       ;INHIBIT ERROR PRINTOUT ?
2050 006416 001036          BNE    4$                ;BRANCH IF YES
2051
2052 006420 104401 006426          TYPE   ,65$             ;;TYPE ASCIZ STRING
2053 006424 000427          BR     64$             ;;GET OVER THE ASCIZ
2054
2055 006504 012746 000204          MOV    #ADDMOD,-(SP)    ;GET READY TO TYPE STARTING ADDRESS
2056                                ;OF 'CHANGE OF BASE ADDRESS' ROUTINE
2057 006510 104402          TYPOC
2058 006512 000000          HALT
2059 006514 000137 041004          4$:  JMP    @#$EOP          ;STOP TO FORCE THE CORRECT RESTART
2060                                ;GO TO END OF PROGRAM ----->
2061 006520 012737 006600 000004  3$:  MOV    #TRP,@#4        ;INITIALIZE VECTOR
2062 006526 005037 004750          CLR    RH70             ;INIT RH INDICATOR ++ C.W
2063 006532 005777 173602          TST    @RHBAE          ;ADDRESS RPBAE(RH11/RH70?)
2064 006536 005237 004750          INC    RH70            ;FOUND AN RH70-SET MASK
2065 006542 104401 006550          TYPE   ,67$             ;;TYPE ASCIZ STRING
2066 006546 000413          BR     66$             ;;GET OVER THE ASCIZ
2067 006576 000420          BR     RTN            ;AND GET OUT
2068 006600 005726          TRP:  TST    (SP)+      ;RESTORE THE STACK
2069 006602 005726          TST    (SP)+          ;RESTORE THE STACK
2070 006604 104401 006612          TYPE   ,65$             ;;TYPE ASCIZ STRING
2071 006610 000413          BR     64$             ;;GET OVER THE ASCIZ
2072 006640 012737 047126 000030  RTN:  MOV    #ERROR,@#EMTVEC ;RESTORE ERROR VECTOR
2073                                ;SO THAT REGISTERS ARE SAVED
2074 006646 012737 000006 000004  MOV    #ERRVEC+2,@#ERRVEC ;RESTORE TRAP CATCHER
2075

```



```
2094
2095 006730 000004
2096 006732 012737 000001 001212 TST3: SCOPE
2097 006740 012737 000003 004604 MOV #1,$TIMES ;;DO 1 ITERATION
2098 006746 000005 RESET #4-1,@#TSTNM ;;THIS SAVES TEST NUMBER
2099 006750 004737 045666 JSR PC,@#STKINT ;;START WITH AN INIT
2100 ;;INITIALIZE TTY KEYBOARD
2101 006754 032777 020000 172156 BIT #SW13,@SWR ;;INHIBIT ERROR TYPEOUT?
2102 006762 001026 BNE 4$ ;;BRANCH IF YES
2103 006764 104401 006772 TYPE ,65$ ;;TYPE ASCIZ STRING
2104 006770 000423 BR 64$ ;;GET OVER THE ASCIZ
2105 007040 013701 002316 4$: MOV @#RHAS,R1 ;;R1 HAS ADDR. OF RHAS
2106 007044 013702 002276 MOV @#RHCS2,R2 ;;R2 HAS ADDR. OF RHCS2
2107 007050 005012 CLR @R2 ;;CLEAR RHCS2
2108 007052 012700 000010 MOV #8.,R0 ;;COUNT
2109 007056 013704 002302 MOV @#RHER1,R4 ;;R4 HAS ADDR. OF RHER1
2110 007062 012714 177777 1$: MOV #-1,@R4 ;;MOVE ERRORS INTO RHER1
2111 007066 005212 INC @R2 ;;INCREMENT UNIT NO.
2112 007070 005300 DEC R0 ;;COUNT DOWN DRIVE COUNTER
2113 007072 001373 BNE 1$ ;;DO NEXT UNIT IF 8 NOT DONE
2114
2115 007074 111137 004742 MOVB @R1,@#TOTALAT ;;SAVE TOTAL ATTENTION
2116 ;;USED IN DRIVE CLEAR TEST
2117 007100 105037 004743 CLRB @#TOTALAT+1 ;;CLEAR UPPER BYTE
2118 007104 105711 TSTB @R1 ;;TEST 'RHAS' FOR ANY DRIVES PRESENT
2119 007106 001402 BEQ 2$ ;;NONE RESPONDING - TYPE THE MESSAGE
2120 007110 000137 007462 JMP XE2 ;;SOME THERE - GO FILL 'UNITS' TABLE
2121
2122 007114 032777 020000 172016 2$: BIT #SW13,@SWR ;;INHIBIT ERROR TYPE OUT?
2123 007122 001402 BEQ 3$ ;;'NO DRIVES' MESSAGE IF NOT
2124 007124 000137 010020 JMP SELTST ;;CHECK FOR SELECTED UNIT START AND LOAD
2125 ;;'UNITS' TABLE WITH SELECTED ONE IF SO
2126
2127 007130 3$:
2128 007130 104401 007136 TYPE ,67$ ;;TYPE ASCIZ STRING
2129 007134 000420 BR 66$ ;;GET OVER THE ASCIZ
2130 007176 104401 007204 TYPE ,69$ ;;TYPE ASCIZ STRING
2131 007202 000430 BR 68$ ;;GET OVER THE ASCIZ
2132 007264 104401 007272 TYPE ,71$ ;;TYPE ASCIZ STRING
2133 007270 000430 BR 70$ ;;GET OVER THE ASCIZ
2134 007352 104401 007360 TYPE ,73$ ;;TYPE ASCIZ STRING
2135 007356 000437 BR 72$ ;;GET OVER THE ASCIZ
2136
2137 007456 000137 041004 JMP @#SEOP ;;GO OUT----->
2138
2139
2140
2141 ;*SET UP UNITS TABLE
2142
2143 007462 XE2:
2144 007462 012700 000010 2$: MOV #8.,R0 ;;COUNTER
2145 007466 012703 004676 MOV #UNITS,R3 ;;POINTER
2146 007472 012723 177777 3$: MOV #-1,(R3)+ ;;PRESET BLOCK TO ALL ONES
2147 007476 005300 DEC R0 ;;COUNT
2148 007500 001374 BNE 3$ ;;BRANCH IF 8 NOT DONE
2149 007502 012703 004676 MOV #UNITS,R3 ;;POINTER
```

```

2150 007506 005005          CLR      R5          ;INITIALIZE UNIT NO. TO 0
2151 007510 005037 004720  CLR      @#NOUNIT   ;NO. OF UNITS PRESENT
2152 007514 012700 000010  MOV      #8,R0      ;COUNTER
2153 007520 011137 001176  MOV      @R1,@#STMP0 ;TEMPORARY STORAGE
2154 007524 006037 001176  ROR      @#STMP0    ;SET CARRY IF ONE IN 0 BIT
2155 007530 103120          BCC      5$         ;CHECK NEXT UNIT IF ONE NOT IN BIT 0
2156
2157 007532 010577 172540  MOV      R5,@RHCS2  ;INSERT UNIT NUMBER INTO RHCS2 UA BITS
2158 007536 022777 024020 172560  CMP      #24020,@RHDT ;IS THIS A DUAL PORT RP04 ?
2159 007544 001503          BEQ      6$         ;TYPE THE UNIT NO. IF YES
2160 007546 022777 020020 172550  CMP      #20020,@RHDT ;IS THIS A SINGLE PORT RP04 ?
2161 007554 001477          BEQ      6$         ;TYPE UNIT NO. IF YES
2162
2163 007556 022777 024021 172540  CMP      #24021,@RHDT ;DUAL PORT RP05 ?
2164 007564 001473          BEQ      6$         ;TYPE UNIT NO. IF SO
2165 007566 022777 020021 172530  CMP      #20021,@RHDT ;SINGLE PORT RP05 ?
2166 007574 001467          BEQ      6$         ;TYPE UNIT NO. IF SO
2167
2168 007576 022777 024022 172520  CMP      #24022,@RHDT ;IS THIS A DUAL PORT RP06 ?
2169 007604 001463          BEQ      6$         ;TYPE THE UNIT NO. IF SO
2170 007606 022777 020022 172510  CMP      #20022,@RHDT ;IS THIS A SINGLE PORT RP06 ?
2171 007614 001457          BEQ      6$         ;TYPE UNIT NO. IF SO
2172
2173
2174          ;*NO...IT'S NOT AN RP04/RP05/RP06 DEVICE SO TYPE
2175          ;*OUT THE DEVICE TYPE
2176
2177 007616 104401 007624  TYPE      ,65$      ;;TYPE ASCIZ STRING
2178 007622 000410  BR        64$       ;;GET OVER THE ASCIZ
2179 007644 010546  MOV      R5,-(SP)   ;GET READY TO TYPE UNIT NUMBER
2180 007646 104405  TYPDS
2181 007650 104401 007656  TYPE      ,67$      ;;TYPE ASCIZ STRING
2182 007654 000406  BR        66$       ;;GET OVER THE ASCIZ
2183 007672 017746 172426  MOV      @RHDT,-(SP) ;GET READY TO TYPE RHDT
2184 007676 104402  TYPDC
2185 007700 104401 007706  TYPE      ,69$      ;;TYPE ASCIZ STRING
2186 007704 000422  BR        68$       ;;GET OVER THE ASCIZ
2187 007752 000407  BR        5$        ;NO RP04/RP05/RP06 FOUND SO TEST NEXT UNIT
2188
2189 007754 010523          6$:  MOV      R5,(R3)+  ;LOAD TABLE POSITION AND INCR IT
2190 007756 104401 001223  TYPE      ,%CRLF
2191 007762 010546  MOV      R5,-(SP)   ;PUT DRIVE NO. ON STACK
2192 007764 104405  TYPDS
2193 007766 005237 004720  INC      @#NOUNIT   ;INCR THE TOTAL NO. OF UNITS
2194
2195 007772 005205          5$:  INC      R5          ;'RHCS2' UNIT ADDRESS
2196 007774 005300  DEC      R0          ;DRIVE COUNTER DOWN ONE
2197 007776 001252  BNE     4$          ;TEST AND DO NEXT UNIT IF 8 NOT DONE
2198
2199 010000 013737 004676 004716  MOV      @#UNITS,@#UNIT ;SET UNIT NO. TO FIRST ONE FOUND OR 0
2200 010006 013737 004720 004722  MOV      @#NOUNIT,@#NUNIT ;SAVE NO. OF UNITS
2201 010014 005337 004722  DEC      @#NUNIT    ;IF NUNIT = 0 THEN ONLY ONE UNIT
2202          ;IF NUNIT > 0 THEN MORE THAN ONE UNIT
2203
2204 010020 005737 004726  SELTST: TST      @#SELECT ;STARTING ADDRESS 200
2205 010024 001403  BEQ     TST4      ;SKIP NEXT IF STARTING FROM 200
  
```

CZRJICO, RP04/5/6 FCTNL CTLR1 MACY11 30A(1052) 27-JUL-78 13:06 PAGE 50
CZRJIC.P11 27-JUL-78 13.05 T3 TEST FOR DRIVES PRESENT USING RHAS AND RHCS2

SEQ 0049

2206 010026 013737 004730 004716 MOV @#UNITSL,@#UNIT ;CHANGE UNIT NUMBER TO SELECTED ONE

CZF
CZF

```

2207
2208
2209
2210 010034 000004          TST4:  SCOPE
2211 010036 012737 000001 001212  MOV    #1,$TIMES          ;;DO 1 ITERATION
2212 010044 012737 010776 001106  MOV    #1$, $LPADR        ;;SET SCOPE LOOP ADDRESS
2213 010052 012706 001000          MOV    #STACK,SP         ;;RESET STACK
2214 010056 012737 000004 004604  MOV    #4,@TSTNM         ;;SAVE TEST NUMBER
2215 010064 004737 041456          JSR    PC,@CLDISK        ;;SET R1-RHCS1, R2-RHCS2
2216 010070 005037 004740          CLR    @ATTENT           ;;CLEAR
2217
2218 010074 005737 004716          TST    @#UNIT            ;;IS 'UNIT 0' NEXT IN THE UNITS TABLE ?
2219 010100 001107          BNE    9$                ;;IF NOT TEST THIS UNIT
2220 010102 012700 000041          MOV    #41,R0            ;;IF SO, CHECK THE LOAD MEDIA LOCATION
2221 010106 122710 000011          CMPB   #11,(R0)          ;;WAS IT AN RPO4/5/6 ?
2222 010112 001102          BNE    9$                ;;NO - GO AHEAD WITH TESTING UNIT #0
2223
2224 010114 005737 004726          TST    @#SELECT          ;;WAS UNIT #0 SELECTED ?
2225                                ;;(IE. WAS IT A 210 START ?)
2226 010120 001006          BNE    12$              ;;IF SO, CHANGE PACKS
2227
2228                                ;*INCREMENT THE UNITS TABLE TO NEXT DRIVE (IF ANY)
2229                                ;*& DECREMENT THE 'NOUNITS' PRESENT (TO BE TESTED)
2230
2231 010122 012700 004676          MOV    #UNITS,R0         ;;LOAD UNITS TABLE POINTER
2232 010126 005720          TST    (R0)+             ;;SELECT THE NEXT UNIT IN THE TABLE
2233                                ;;(DOUBLE INCREMENT THE POINTER, R0)
2234 010130 022710 177777          CMP    #-1,(R0)          ;;IS THERE ANOTHER TABLE ENTRY PRESENT ?
2235 010134 001065          BNE    11$              ;;IF SO, USE THE NEXT DRIVE & DEC 'NOUNITS'
2236                                ;;IF NOT, MUST USE DRIVE #0 & CHANGE PACK
2237
2237 010136          12$:
2238 010136 104401 010144          TYPE   ,65$              ;;TYPE ASCIZ STRING
2239 010142 000434          BR     64$               ;;GET OVER THE ASCIZ
2240 010234 104401 010242          TYPE   ,67$              ;;TYPE ASCIZ STRING
2241 010240 000421          BR     66$               ;;GET OVER THE ASCIZ
2242 010304 000000          HALT
2243 010306 000404          BR     9$                ;;CONTINUE, USING SCRATCH PACK ON UNIT #0
2244
2245 010310 011037 004716          11$:  MOV    (R0),@#UNIT      ;;SET UP TO BE THE UNIT UNDER TEST
2246 010314 005337 004720          DEC    @#NOUNITS         ;;DECREMENT BECAUSE UNIT #0 WON'T BE TESTED
2247
2248 010320 013700 004716          9$:   MOV    @#UNIT,R0        ;;R0 CONTAINS UNIT UNDER TEST
2249
2250
2251
2252 010324 005037 004744          CLR    @#RPO6            ;;CLEAR RPO6 DEVICE TYPE FLAG
2253 010330 010077 171742          MOV    R0,@RHCS2        ;;SET UP UNIT ADDRESSING
2254 010334 022777 024022 171762  CMP    #24022,@RHDT     ;;IS IT A DUAL PORT RPO6 ?
2255 010342 001405          BEQ    2$                ;;YES...SET THE FLAG
2256 010344 022777 020022 171752  CMP    #20022,@RHDT     ;;IS IT A SINGLE PORT RPO6 ?
2257 010352 001401          BEQ    2$                ;;YES...SET FLAG
2258 010354 000404          BR     3$                ;;DON'T SET FLAG - CHECK FOR RPO4
2259 010356 012737 177777 004744  2$:  MOV    #-1,@#RPO6       ;;SET THE FLAG
2260 010364 000416          BR     8$                ;;DON'T CHECK FOR RPO4, IT WAS RPO6
2261
2262 010366 005037 004746          3$:  CLR    @#RPO5            ;;CLEAR MEMOREX RPO4 DEVICE FLAG
  
```



```
2330
2331
2332
2333
2334 011022 000004          TST5:  SCOPE
2335 011024 012737 000005 004604  MOV    #6-1,@TSTNM ;THIS SAVES TEST NUMBER
2336 011032 004737 041456          JSR    PC,@CLDISK ;GIVE INITIALIZE
2337 011036 032713 010000          BIT    #MOL,@R3 ;CHECK MOL IN RHDS1
2338 011042 001151          BNE    TST6 ; BRANCH IF MOL HIGH
2339 011044 104401 011052          TYPE  ,65$ ;:TYPE ASCIZ STRING
2340 011050 000420          BR     64$ ;:GET OVER THE ASCIZ
2341 011112 104401 011120          TYPE  ,67$ ;:TYPE ASCIZ STRING
2342 011116 000424          BR     66$ ;:GET OVER THE ASCIZ
2343 011170 104401 011176          TYPE  ,69$ ;:TYPE ASCIZ STRING
2344 011174 000431          BR     68$ ;:GET OVER THE ASCIZ
2345 011260 032713 010000          1$:  BIT    #MOL,@R3 ;CHECK MOL IN RHDS1
2346 011264 001775          BEQ    1$ ;BRANCH IF MOL IS HIGH
2347 011266 104401 011274          TYPE  ,71$ ;:TYPE ASCIZ STRING
2348 011272 000435          BR     70$ ;:GET OVER THE ASCIZ
```



```

2349
2350 011366 000004          TST6: SCOPE
2351
2352 011370 012737 000006 004604      MOV    #7-1,@TSTNM      ;THIS SAVES TEST NUMBER
2353 011376 012706 001000              MOV    #STACK,SP       ;RESET STACK
2354 011402 004737 041456              JSR    PC,@CLDISK      ;SET R1-RHCS1, R2-RHCS2
2355
2356 011406 013700 002266              MOV    @RPVEC,R0       ;GET RP VECTOR ADDRESS
2357 011412 012720 011460              MOV    #RPTRP1,(R0)+   ;THIS IS FOR TIMELY INTERRUPTS
2358 011416 012710 000340              MOV    #340,(R0)      ;RP04 INTERRUPT SERVICE ROUTINE
2359                                ;PRIORITY = 7
2360 011422 012737 000200 177776      MOV    #200,PS         ;SET PROCESSOR PRIORITY @ 4 (DISK @ 5)
2361 011430 012711 000300              MOV    #RDY!IE,@R1    ;RDY, IE IN RHSC1 SHOULD CAUSE INTERRUPT
2362
2363 011434 013737 042154 001200      MOV    @TIMCNT,@$TMP1 ;COUNTER
2364 011442 005337 001200      1$:  DEC    @$TMP1       ;WAIT FOR INTERRUPT
2365 011446 001375              BNE    1$             ;BRANCH IF NOT ZERO
2366                                ;BEFORE THIS IS ZERO INTERRUPT SHOULD OCCUR
2367 011450 104065              ERROR  65            ;INTERRUPT DID NOT OCCUR
2368 011452 012712 000040              MOV    #40,@R2        ;CLEAR CONTROLLER VIA CS2 CLR BIT
2369 011456 000407              BR     TST7           ;BRANCH TO NEXT TEST -----)
2370
2371 011460 022626          RPTRP1: CMP    (SP)+,(SP)+   ;RESTORE STACK
2372 011462 022711 004200      CMP    #DVA!RDY,@R1   ;IE SHOULD BE LOW AS RUPT OCCURRED
2373 011466 001403              BEQ    TST7           ;CONTINUE IF GOOD -----)
2374 011470 104065              ERROR  65            ;INTERRUPT OCCURED BUT
2375                                ;IE FAILED TO RESET
2376 011472 012712 000040              MOV    #40,@R2        ;CLEAR CONTROLLER VIA CS2 CLR BIT
  
```

```

2377
2378
2379 011476 000004          TST7:  SCOPE
2380
2381 011500 012737 000007 004604      MOV    #10-1,@#TSTNM  ;THIS SAVES TEST NUMBER
2382 011506 012706 001000              MOV    #STACK,SP      ;RESET STACK
2383 011512 004737 041456              JSR    PC,@#CLDISK    ;SET R1-RHCS1, R2-RHCS2
2384
2385 011516 013700 002266              MOV    @#RPVEC,R0     ;GET RP VECTOR ADDRESS
2386 011522 012720 011566              MOV    #RPTRP2,(R0)+ ;THIS IS FOR UNTIMELY INTERRUPTS
2387 011526 012710 000340              MOV    #340,(R0)      ;RP04 INTERRUPT SERVICE ROUTINE
2388                                     ;PRIORITY = 7
2389 011532 012737 000240 177776      MOV    #240,PS        ;SET PROCESSOR PRIORITY @ 5
2390 011540 012711 000300              MOV    #RDY!IE,@R1    ;RDY, IE IN RHSC1 WHOULD CAUSE INTERRUPT
2391
2392 011544 013737 042154 001200      MOV    @#TIMCNT,@#STMP1;COUNTER
2393 011552 005337 001200      1$:  DEC    @#STMP1      ;WAIT FOR INTERRUPT
2394 011556 001375              BNE    1$             ;BRANCH IF NOT ZERO
2395                                     ;BEFORE THIS IS ZERO INTERRUPT WHOULD
2396                                     ;OCCUR
2397 011560 012712 000040              MOV    #40,@R2        ;CLEAR THE CONTROLLER VIA CS2 CLR BIT
2398 011564 000404              BR     TST10 ;        ;NO INTERRUPT SO CONTINUE -----)
2399
2400 011566 022626          RPTRP2: CMP    (SP)+,(SP)+ ;RESTORE STACK
2401 011570 104065          ERROR  65           ;INTERRUPT OCCURRED WITH PROCESSOR
2402                                     ;PROCESSOR STATUS SAME AS DISK
2403 011572 012712 000040              MOV    #40,@R2        ;CLEAR THE CONTROLLER VIA CS2 CLR BIT
2404
2405
2406
2407
2408
  
```



```

2465
2466 012146 011100          6$:  MOV    @R1,R0          ;GET RHCS1 CONTENTS
2467 012150 042700 160076  BIC    #SC.TRE!MCPE!76,R0 ;CLEAR SC,TRE,MCPE AND
2468                                ;ALL FUNCTION BITS
2469 012154 022700 004200  .MP    #DVA!RDY,R0      ;RHCS1 SHOULD HAVE
2470                                ;GO = 0, DVA = 1, RDY = 1
2471 012160 001403          BEQ    7$              ;BRANCH IF RHCS1 IS GOOD
2472 012162 010137 001122  MOV    R1,@#SBDADR      ;FAILING REGISTER RHCS1
2473 012166 104013          ERROR  13              ;AFTER A POWER UP WITHOUT ANY
2474                                ;INIT RHCS1 SHOULD HAVE
2475                                ;GO = 0, DVA = 1, RDY = 1, IE  0
2476                                ;DISREGARD ALL OTHER BITS
2477
2478 012170 005777 170140          7$:  TST    @RHCC           ;TEST RHCC, IT SHOULD - 0
2479 012174 001411          BEQ    10$            ;BRANCH IF RHCC = 0
2480 012176 013737 002334 004600  MOV    @#RHCC,@#REGADR  ;FAILING REGISTER RHCC
2481 012204 005037 001124          CLR    @#SGDDAT        ;RHCC SHOULD BE = 0
2482 012210 017737 170120 001126  MOV    @RHCC,@#SBDAT    ;BAD DATA
2483 012216 104014          ERROR  14              ;AFTER POWER UP RHCC
2484                                ;SHOULD BE 0
2485
2486 012220          10$:
2487 012220 004737 041456  JSR    PC,@#CLDISK      ;SET R1-RHCS1, R2-RHCS2
2488 012224 004737 041514  JSR    PC,@#CHECK       ;CHECK THAT DVA,RDY,MOL,DPR,DRY - 1
2489 012230 104401 066720  TYPE    ,CPHALT        ;CANNOT CONTINUE TESTS IF THEY AREN'T
2490 012234 000000          HALT                    ;STOP
2491 012236 013777 002460 170034  MOV    @#PKACK,@RHCS1  ;GET READY FOR PKACK
2492
2493                                ;*NOW SAVE REGISTERS FOR COMPARISON AFTER PACK ACKNOWLEDGE
2494
2495 012244 004037 041624  JSR    R0,@#SAVER       ;SAVE REGISTERS
2496 012250 002272          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
2497 012252 004612          SAVERE                 ;STARTING ADDRESS OF WHERE
2498 012254 000022          18.                  ;NUMBER OF REGISTERS
2499
2500 012256 013777 004606 170002  MOV    @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
2501
2502 012264 013746 002460  MOV    @#PKACK,-(SP)    ;GET READY TO MOVE COMMAND
2503 012270 052716 000001  BIS    #GO,(SP)         ;GET READY TO SET GO
2504 012274 012677 170000  MOV    (SP)+,@RHCS1    ;GO WITH
2505
2506 012300 104413          WAT                    ;WAIT FOR VV BIT TO SET
2507 012302 002322          RHDS1                 ;WAIT FOR RHDS1 REGISTER
2508 012304 000100          VV                    ;WAIT FOR VV BIT IN RHDS1 REGISTER
2509 012306 000001          1.                    ;ALLOW 10 MICRO SECONDS
2510 012310 000001          1.                    ;VV MUST SET BETWEEN
2511
2512 012312 004037 042336  JSR    R0,@#CHREG       ;CHANGE BITS IN SAVED REGISTER
2513 012316 002322          RHDS1                 ;CHANGE RHDS1 REGISTER
2514 012320 000001          1                      ;1 BIT/BITS TO BE CHANGED
2515 012322 000001          1                      ;NEW VALUE OF VV IS 1
2516 012324 000100          VV                    ;CHANGE VV BIT
2517
2518                                ;*NOW COMPARE REGISTERS SO THAT NO REGISTERS
2519                                ;*CHANGED EXCEPT VV IN RHDS1 AND IE IN RHCS1
2520

```

```
2521 012326 004037 042444 JSR R0,@COMREG ;COMPARE SAVED REGISTERS WITH
2522 012332 004612 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
2523 012334 002354 WC ;TEST DATA STARTING FROM 'RHWC'
2524 012336 000022 18. ;18. REGISTERS TO BE COMPARED
2525 012340 012344 11$ ;RETURN TO 11$ ON ERROR
2526 012342 012350 12$ ;RETURN TO 12$ ON NO ERROR
2527
2528 012344 104015 11$: ERROR 15 ;GIVING A PACK ACKNOWLEDGE
2529 012346 000207 RTS PC ;CAUSED AN ERROR
2530 ;PACK ACKNOWLEDGE SHOULD
2531 ;SET VV IN RHDS1
2532 ;INTERRUPT SHOULD MAKE
2533 ;IE = 0
2534 ;NO OTHER REGISTERS SHOULD
2535 ;CHANGE
2536 ;GOOD DATA GIVES
2537 ;CONTENTS OF REGISTER BEFORE
2538 ;PACK ACKNOWLEDGE
2539 ;RECEIVED DATA GIVES
2540 ;CONTENTS OF REGISTER
2541 ;AFTER PACK ACKNOWLEDGE
2542
2543 012350 012737 177777 047274 12$: MOV #-1,@PRITEM ;CLEAR PREVIOUS ITEM NUMBER
2544
```

```

2545
2546
2547
2548 012356 000004          TST11: SCOPE
2549
2550                      ;*IN CASE THERE IS ANY DRIVE ERRORS DURING POWER UP
2551                      ;*OR POWER DOWN OR ANY PARITY ERRORS A RESET IS GIVEN
2552 012360 000005          RESET
2553 012362 004737 045666  JSR    PC,@#STKINT      ;INITILIZE TK
2554
2555
2556 012366 012706 001000  MOV    #STACK,SP        ;RESET STACK
2557 012372 012737 000011 004604  MOV    #11,@#TSTNM     ;SAVE TEST NUMBER
2558 012400 004737 041456  JSR    PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
2559 012404 004737 041514  JSR    PC,@#CHECK      ;CHECK THAT DVA,RDY,MOL,DPR,DRY 1
2560 012410 104401 066720  TYPE   ,CPHALT         ;CANNOT CONTINUE TESTS IF THEY AREN'T
2561 012414 000000          HALT                    ;STOP
2562 012416 013777 002460 167654  MOV    @#PKACK,@#RHCS1 ;GET READY FOR PKACK
2563
2564                      ;*NOW SAVE REGISTERS FOR COMPARISON AFTER PACK ACKNOWLEDGE
2565
2566 012424 004037 041624  JSR    R0,@#SAVER      ;SAVE REGISTERS
2567 012430 002272          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
2568 012432 004612          SAVERE                   ;STARTING ADDRESS OF WHERE
2569 012434 000022          18.                     ;NUMBER OF REGISTERS
2570
2571 012436 013777 004606 167622  MOV    @#RP4VEC,@#RVEC ;SET RP04 VECTOR ADDRESS
2572
2573 012444 013746 002460  MOV    @#PKACK,-(SP)   ;GET READY TO MOVE COMMAND
2574 012450 052716 000001  BIS    #GO,(SP)        ;GET READY TO SET GO
2575 012454 012677 167620  MOV    (SP)+,@#RHCS1   ;GO WITH
2576 012460 011100          MOV    @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
2577 012462 011305          MOV    @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
2578
2579 012464 104413          WAT                      ;WAIT FOR VV BIT TO SET
2580 012466 002322          RHDS1                   ;WAIT FOR RHDS1 REGISTER
2581 012470 000100          VV                      ;WAIT FOR VV BIT IN RHDS1 REGISTER
2582 012472 000001          1.                      ;ALLOW 10 MICRO SECONDS
2583 012474 000001          1.                      ;VV MUST SET BETWEEN
2584 012476 013746 002460  MOV    @#PKACK,-(SP)   ;SAVE COMMAND
2585 012502 052716 004200  BIS    #DVA!RDY,(SP)   ;INCLUDE DVA!RDY
2586 012506 011637 001124  MOV    (SP),@#SGDDAT   ;SAVE FOR PRINTOUT
2587 012512 022600          CMP    (SP)+,R0        ;DURING ABOVE OPERATION ONLY DVA!RDY
2588 012514 001405          BEQ    64$             ;BRANCH IF GOOD
2589 012516 010037 001126  MOV    R0,@#SBDDAT     ;BAD DATA
2590 012522 010137 004600  MOV    R1,@#REGADR     ;FAILING REGISTER RHCS1
2591 012526 104021          ERROR 21                ;DURING ABOVE OPERATION ONLY
2592 012530 012746 010700 64$: MOV    #MOL!DPR!DRY!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
2593 012534 011637 001124  MOV    (SP),@#SGDDAT   ;SAVE FOR PRINTOUT
2594 012540 022605          CMP    (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR.DRY.VV
2595 012542 001405          BEQ    66$             ;BRANCH IF GOOD
2596 012544 010537 001126  MOV    R5,@#SBDDAT     ;BAD DATA
2597 012550 010337 004600  MOV    R3,@#REGADR     ;FAILING REGISTER RHDS1
2598 012554 104063          ERROR 63                ;DURING ABOVE OPERATION ONLY
2599 012556 004037 042336  JSR    R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
2600 012562 002322          RHDS1                   ;CHANGE RHDS1 REGISTER
  
```



```

2629
2630 012614 005737 004744      TST    @#RPO6      ;TEST FOR RPO6 DRIVE
2631 012620 001005              BNE    3$          ;IF = 1 DO 'MAKECYL' 777
2632 012622 005737 004746      TST    @#RPO5      ;TEST FOR RPO5 DRIVE
2633 012626 001004              BNE    4$          ;IF = 1 DO 'MAKECYL' 377
2634 012630 000137 014754      JMP    TST16      ;
2635                                ;OR THE ADDRESS PLUG TESTS
2636 012634              3$:
2637 012634 000137 012700      JMP    TST13      ; DO 'MAKECYL' 777 -----)
2638 012640              4$:
2639
2640
2641 012640 000004              TST12: SCOPE
2642 012642 012706 001000      MOV    #STACK,SP  ;RESET STACK
2643 012646 012737 000012 004604  MOV    #13-1,@#TSTNM ;THIS SAVES TEST NUMBER
2644 012654 004737 041456      JSR    PC,@#CLDISK ;INIT DRIVE
2645 012660 012777 000001 167432  MOV    #DMD,@RHMR  ;SET DIAGNOSTIC MODE
2646 012666 004037 041126      JSR    R0,@#MAKECYL ;SUBROUTINE TO GIVE A SEEK
2647 012672 000377
2648
2649 012674 000137 012734      JMP    TST14      ; SKIP NEXT 'MAKECYL' -----)
2650
2651
2652 012700 000004              TST13: SCOPE
2653 012702 012706 001000      MOV    #STACK,SP  ;RESET STACK
2654 012706 012737 000013 004604  MOV    #14-1,@#TSTNM ;THIS SAVES TEST NUMBER
2655 012714 004737 041456      JSR    PC,@#CLDISK ;INIT DRIVE
2656 012720 012777 000001 167372  MOV    #DMD,@RHMR  ;SET DIAGNOSTIC MODE
2657 012726 004037 041126      JSR    R0,@#MAKECYL ;SUBROUTINE TO GIVE A SEEK
2658 012732 000777
2659
  
```



```

2660
2661
2662 012734 000004          TST14: SCOPE
2663 012736 012737 000001 001212 MOV      #1,$TIMES      ;;DO 1 ITERATION
2664
2665 012744 005737 004744          TST      @#RP06        ;TEST FOR RP06 DRIVE
2666 012750 001005          BNE      4$           ;IF = 1, DO THIS TEST
2667 012752 005737 004746          TST      @#RP05        ;TEST FOR MEMOREX RP04
2668 012756 001002          BNE      4$           ;IF = 1, DO THIS TEST
2669                                ;IF NEITHER FLAG IS UP, ASSUME THE
2670                                ;DRIVE IS AN ISS RP04 AND SKIP TEST
2671 012760 000137 013746          JMP      TST15      ; JUMP TO NEXT TEST -----)
2672 012764          4$:
2673
2674 012764 012706 001000          MOV      #STACK,SP    ;RESET STACK
2675 012770 012737 000014 004604 MOV      #14,@#TSTNM  ;SAVE TEST NUMBER
2676 012776 004737 041456          JSR      PC,@#CLDISK  ;SET R1-RHCS1, R2-RHCS2
2677 013002 004737 041536          JSR      PC,@#CHECKT  ;CHECK DVA,RDY,MOL,DPR,DRY,VV - 1
2678 013006 104401 066720          TYPE    ,CPHALT      ;CANNOT CONTINUE TESTING IF ANY OF
2679 013012 000000          HALT                    ;STOP
2680 013014 005737 004724          TST      @#NOPUSH     ;IS THIS A 220 START ?
2681 013020 001007          BNE      1$           ;SKIP THIS TEST IF SO
2682 013022 005737 000042          TST      @#42        ;MONITOR (ACT 11) RETURN ADDRESS ?
2683 013026 001004          BNE      1$           ;SKIP THIS TEST
2684 013030 005737 001100          TST      @#$PASS     ;FIRST PASS ?
2685 013034 001001          BNE      1$           ;SKIP THIS TEST IF NOT
2686 013036 000402          BR       2$           ;CONTINUE WITH THIS TEST
2687 013040 000137 013746          JMP      TST15      ; JUMP TO NEXT TEST -----)
2688
2689                                ;*SET DIAGNOSTIC MODE TO ENABLE A COMMAND ACTIVE WHILE
2690                                ;*THE PLUG IS PULLED
2691
2692 013044 052777 000001 167246 BIS      #DMD,@#RHMR  ;SET UP DIAGNOSTIC MODE
2693
2694                                ;*TAKE AN INITIAL REGISTER SNAPSHOT
2695
2696 013052 004037 041624          JSR      R0,@#SAVER   ;SAVE REGISTERS
2697 013056 002272          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
2698 013060 004612          SAVERE                  ;STARTING ADDRESS OF WHERE
2699 013062 000022          18.                    ;NUMBER OF REGISTERS
2700
2701
2702                                ;*ISSUE A COMMAND AND THE 'GO' BIT (NOT POSITIONING COMMAND)
2703                                ;*TO VERIFY COMMAND ABORT IF PLUG IS PULLED
2704
2705
2706                                ;*ISSUE SOME CLOCKS TO GET THE COMMAND STARTED
2707                                ;*(USE 'SEARCH' WITH 'DTETST' FLAG UP TO STOP CLOCKING ?)
2708
2709
2710
2711
2712 013064 104401 013072          TYPE    ,65$         ;;TYPE ASCIZ STRING
2713 013070 000420          BR       64$         ;;GET OVER THE ASCIZ
2714 013132 013746 004716          MOV      @#UNIT,-(SP) ;GET THE UNIT NO. UNDER TEST
2715 013136 104405          TYPDS                    ;TYPE IT OUT

```

```
2716 013140 104401 013146 TYPE 67$ ;:TYPE ASCIZ STRING
2717 013144 000415 BR 66$ ;:GET OVER THE ASCIZ
2718 013200 000000 HALT ;:WAIT FOR OPERATOR PLUG CHANGE
2719
2720 ;*CHECK THAT THE UNIT NO. UNDER TEST HAS GONE OFFLINE
2721
2722 013202 017700 167076 MOV @RHDST,R0 ;:ATTEMPT TO ADDRESS THE DRIVE
2723 013206 004737 043370 JSR PC,@PUTREG ;:TAKE REGISTER SNAPSHOTS
2724 013212 032737 010000 002360 BIT #NED,@#CS2 ;:TEST FOR NON EXISTENT DRIVE
2725 013220 001001 BNE 7$ ;:CONTINUE IF 'NED' BIT SET (UNIT
2726 ;:IS OFFLINE AS IT SHOULD BE)
2727 013222 104101 ERROR 101 ;:UNIT DID NOT GO OFFLINE WHEN ADDRESS
2728 ;:PLUG WAS REMOVED
2729 013224 7$:
```

```

2730
2731
2732
2733
2734 013224 104401 013232
2735 013230 000420
2736 013272 013746 004716
2737 013276 104405
2738 013300 104401 013306
2739 013304 000416
2740 013342 000000
2741
2742
2743
2744 013344 004737 043370
2745 013350 032737 000400 002404
2746 013356 001411
2747 013360 032737 004000 002362
2748 013366 001405
2749 013370 032737 000200 002404
2750 013376 001401
2751 013400 000403
2752 013402 104102 9$:
2753
2754 013404 000137 013746
2755
2756 013410 8$:
2757
2758
2759 013410 004037 042336
2760 013414 002300
2761 013416 000004
2762 013420 000001
2763 013422 000200
2764 013424 000001
2765 013426 100000
2766 013430 000001
2767 013432 040000
2768 013434 000001
2769 013436 020000
2770 013440 004037 042336
2771 013444 002322
2772 013446 000003
2773 013450 000001
2774 013452 100000
2775 013454 000001
2776 013456 040000
2777 013460 000000
2778 013462 000100
2779 013464 004037 042336
2780 013470 002276
2781 013472 000001
2782 013474 000001
2783 013476 010000
2784 013500 004037 042336
2785 013504 002314
  
```

```

;*NOW REPLACE THE ADDRESS PLUG

TYPE ,69$      ;;TYPE ASCIZ STRING
BR 68$        ;;GET OVER THE ASCIZ
MOV @#UNIT,-(SP) ;;GET THE UNIT UNDER TEST
TYPDS        ;;TYPE IT OUT
TYPE ,71$      ;;TYPE ASCIZ STRING
BR 70$        ;;GET OVER THE ASCIZ
HALT         ;;WAIT FOR OPERATOR PLUG REPLACEMENT
  
```

```

;*CHECK THAT THE ORIGINAL UNIT HAS COME BACK ON LINE

JSR PC,@#PUTREG ;;TAKE NEW REGISTER SNAPSHOTS
BIT #DPR,@#DS1  ;;TEST THAT 'DPR' = 1
BEQ 9$          ;;ERROR - DRIVE SHOULD BE PRESENT
BIT #DVA,@#CS1  ;;TEST THAT DEVICE IS NOW AVAILABLE
BEQ 9$          ;;ERROR - 'DVA' SHOULD = 1
BIT #DRY,@#DS1  ;;TEST THAT DRIVE READY IS - 1
BEQ 9$          ;;ERROR - 'DRY' SHOULD = 1
BR 8$          ;;A-OK: 'DPR' = 1, 'DVA' = 1, & 'DRY' 1
9$: ERROR 102   ;;UNIT NOT AVAILABLE AFTER
                ;;ADDRESS PLUG WAS REPLACED
                ;;JUMP TO NEXT TEST -----)
JMP TST15      ;
  
```

```

8$: ;CHANGE THE INITIAL REGISTER SNAPSHOT TO EXPECTED VALUES
;*AFTER THE PLUG CHANGE

JSR R0,@#CHREG ;;CHANGE BITS IN SAVED REGISTER
RHCS1          ;;CHANGE RHCS1 REGISTER
4             ;;4 BIT/BITS TO BE CHANGED
1            ;;NEW VALUE OF RDY IS 1
RDY          ;;CHANGE RDY BIT
1            ;;NEW VALUE OF SC IS 1
SC           ;;CHANGE SC BIT
1            ;;NEW VALUE OF TRE IS 1
TRE         ;;CHANGE TRE BIT
1            ;;NEW VALUE OF MCPE IS 1
MCPE       ;;CHANGE MCPE BIT
JSR R0,@#CHREG ;;CHANGE BITS IN SAVED REGISTER
RHDS1     ;;CHANGE RHDS1 REGISTER
3         ;;3 BIT/BITS TO BE CHANGED
1        ;;NEW VALUE OF ATA IS 1
ATA     ;;CHANGE ATA BIT
1        ;;NEW VALUE OF ERR IS 1
ERR    ;;CHANGE ERR BIT
0      ;;NEW VALUE OF VV IS 0
VV     ;;CHANGE VV BIT
JSR R0,@#CHREG ;;CHANGE BITS IN SAVED REGISTER
RHCS2  ;;CHANGE RHCS2 REGISTER
1      ;;1 BIT/BITS TO BE CHANGED
1      ;;NEW VALUE OF NED IS 1
NED   ;;CHANGE NED BIT
JSR R0,@#CHREG ;;CHANGE BITS IN SAVED REGISTER
RHER3 ;;CHANGE RHER3 REGISTER
  
```



```

2807
2808
2809                ;*NOW CLEAR OUT THE CONTROLLER AND DRIVE
2810
2811 013550          6$:
2812 013550 004737 041456 JSR    PC,@#CLDISK    ;SET R1-RHCS1, R2-RHCS2
2813
2814                ;*CHANGE THE REGISTER SNAPSHOT TO EXPECTED VALUES AFTER THE CLEAR
2815
2816 013554 004037 042336 JSR    RO,@#CHREG    ;CHANGE BITS IN SAVED REGISTER
2817 013560 002300 RHCS1    ;CHANGE RHCS1 REGISTER
2818 013562 000003 3        ;3 BIT/BITS TO BE CHANGED
2819 013564 000000 0        ;NEW VALUE OF SC IS 0
2820 013566 100000 SC        ;CHANGE SC BIT
2821 013570 000000 0        ;NEW VALUE OF TRE IS 0
2822 013572 040000 TRE        ;CHANGE TRE BIT
2823 013574 000000 0        ;NEW VALUE OF MCPE IS 0
2824 013576 020000 MCPE        ;CHANGE MCPE BIT
2825 013600 004037 042336 JSR    RO,@#CHREG    ;CHANGE BITS IN SAVED REGISTER
2826 013604 002322 RHDS1    ;CHANGE RHDS1 REGISTER
2827 013606 000002 2        ;2 BIT/BITS TO BE CHANGED
2828 013610 000000 0        ;NEW VALUE OF ATA IS 0
2829 013612 100000 ATA        ;CHANGE ATA BIT
2830 013614 000000 0        ;NEW VALUE OF ERR IS 0
2831 013616 040000 ERR        ;CHANGE ERR BIT
2832 013620 004037 042336 JSR    RO,@#CHREG    ;CHANGE BITS IN SAVED REGISTER
2833 013624 002276 RHCS2    ;CHANGE RHCS2 REGISTER
2834 013626 000001 1        ;1 BIT/BITS TO BE CHANGED
2835 013630 000000 0        ;NEW VALUE OF NED IS 0
2836 013632 010000 NED        ;CHANGE NED BIT
2837 013634 004037 042336 JSR    RO,@#CHREG    ;CHANGE BITS IN SAVED REGISTER
2838 013640 002314 RHER3    ;CHANGE RHER3 REGISTER
2839 013642 000001 1        ;1 BIT/BITS TO BE CHANGED
2840 013644 000000 0        ;NEW VALUE OF ACE IS 0
2841 013646 020000 ACE        ;CHANGE ACE BIT
2842 013650 004037 042336 JSR    RO,@#CHREG    ;CHANGE BITS IN SAVED REGISTER
2843 013654 002320 RHMR        ;CHANGE RHMR REGISTER
2844 013656 000001 1        ;1 BIT/BITS TO BE CHANGED
2845 013660 000000 0        ;NEW VALUE OF DMD IS 0
2846 013662 000001 DMD        ;CHANGE DMD BIT
2847
2848 013664 043737 004740 004636 BIC    @#ATTENT,@#SAVERE+24 ;UNIT UNDER TEST ATTENTION BIT
2849
2850
2851                ;*TAKE ANOTHER REGISTER SNAPSHOT AND COMPARE RESULTS
2852                ;*WITH THE EXPECTED VALUES
2853
2854 013672 004037 042444 JSR    RO,@#COMREG    ;COMPARE SAVED REGISTERS WITH
2855 013676 004612 SAVERE    ;GOOD DATA SAVED IN 'SAVERE'
2856 013700 002354 WC        ;TEST DATA STARTING FROM 'RHWC'
2857 013702 000022 18       ;18. REGISTERS TO BE COMPARED
2858 013704 013710 10$      ;RETURN TO 10$ ON ERROR
2859 013706 013714 11$      ;RETURN TO 11$ ON NO ERROR
2860
2861 013710 104100 10$:      ERROR 100    ;ADDRESS PLUG CHANGE CAUSED SOME
2862 013712 000207 RTS      PC        ;INCORRECT REGISTER RESULT
  
```

2863
2864 013714

11S:

;(USE NED METHOD TO VERIFY
;*THAT ATTENTION BIT COMES UP IN THE PROPER LOCATION ??)

2865
2866
2867
2868
2869
2870
2871
2872

;*SET 'VV' IN RHDS1 AFTER RESET FROM THE RECALIBRATE
;*CAUSED BY PULLING THE PLUG

2874
2875 013714 013746 002460
2876 013720 052716 000001
2877 013724 012677 166350
2878 013730 011100
2879 013732 011305
2880 013734 104413
2881 013736 002322
2882 013740 000100
2883 013742 000001
2884 013744 000001
2885

MOV @#PKACK,-(SP) ;GET READY TO MOVE COMMAND
BIS #GO,(SP) ;GET READY TO SET GO
MOV (SP)+,@RHCS1 ;GO WITH
MOV @R,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
MOV @R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
WAT ;WAIT FOR VV BIT TO SET
RHDS1 ;WAIT FOR RHDS1 REGISTER
VV ;WAIT FOR VV BIT IN RHDS1 REGISTER
1. ;ALLOW 10 MICRO SECONDS
1. ;VV MUST SET BETWEEN

```
2886
2887
2888
2889 013746 000004          TST15: SCOPE
2890 013750 012737 000001 001212 MOV      #1,$TIMES      ;;DO 1 ITERATION
2891
2892 013756 005737 004744          TST      @#RP06        ;TEST FOR RP06 DRIVE
2893 013762 001005          BNE      4$            ;IF = 1, OK TO DO THIS TEST
2894 013764 005737 004746          TST      @#RP05        ;TEST FOR MEMOREX RP04
2895 013770 001002          BNE      4$            ;IF = 1, OK TO DO THIS TEST
2896                                ;NOT AN RP06 OR MEMOREX RP04 SO
2897                                ;ASSUME AN ISS RP04 AND SKIP TEST
2898 013772 000137 014754          JMP      TST16 ;      JUMP TO NEXT TEST -----)
2899 013776                                4$:
2900
2901                                ;*CHECK TO SEE IF THIS TEST HAS BEEN SELECTED WITH SW5
2902
2903 013776 032777 000040 165134    BIT      #SW5,@SWR     ;TEST THE SWITCH
2904 014004 001002          BNE      5$            ;IF 0, TEST HAS NOT BEEN SELECTED
2905 014006 000137 014754          JMP      TST16 ;      JUMP TO NEXT TEST -----)
2906 014012                                5$:
2907                                ;TEST SELECTED, CONTINUE IT
2908 014012 005737 004724          TST      @#NOPUSH     ;IS THIS A 220 START ?
2909 014016 001007          BNE      1$            ;SKIP THIS TEST IF SO
2910 014020 005737 000042          TST      @#42         ;MONITOR (ACT 11) RETURN ADDRESS ?
2911 014024 001004          BNE      1$            ;SKIP THIS TEST
2912 014026 005737 001100          TST      @#$PASS     ;FIRST PASS ?
2913 014032 001001          BNE      1$            ;SKIP THIS TEST IF NOT
2914 014034 000402          BR       2$            ;CONTINUE WITH THIS TEST
2915 014036 000137 014754          JMP      TST16 ;      JUMP TO NEXT TEST -----)
2916 014042 012706 001000          MOV      #STACK,SP    ;RESET STACK
2917 014046 012737 000015 004604    MOV      #15,@#TSTNM  ;SAVE TEST NUMBER
2918 014054 004737 041456          JSR      PC,@#CLDISK  ;SET R1-RHCS1, R2-RHCS2
2919 014060 004737 041536          JSR      PC,@#CHECKT  ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
2920 014064 104401 066720          TYPE    ,CPHALT      ;CANNOT CONTINUE TESTING IF ANY OF
2921 014070 000000          HALT                ;STOP
2922
2923
2924 014072 104401 014100          TYPE    ,65$         ;;TYPE ASCIZ STRING
2925 014076 000426          BR       64$         ;;GET OVER THE ASCIZ
2926 014154 104401 014162          TYPE    ,67$         ;;TYPE ASCIZ STRING
2927 014160 000417          BR       66$         ;;GET OVER THE ASCIZ
2928
2929
2930
2931 014220 013737 004716 004752    MOV      @#UNIT,@#INUNIT ;MAKE THE INITIAL UNIT NO. 'UNIT'
2932
```

```

2933
2934
2935                ;*CHANGE ADDRESS PLUG ON THE UNIT UNDER TEST
2936
2937 014226          6$:
2938 014226 104401 014234      TYPE      ,69$      ;;TYPE ASCIZ STRING
2939 014232 000420          BR      68$      ;;GET OVER THE ASCIZ
2940 014274 013746 004716     MOV      @#UNIT,-(SP)  ;;GET THE UNIT UNDER TEST
2941 014300 104405          TYPDS           ;;TYPE IT OUT
2942 014302 104401 014310     TYPE      ,71$      ;;TYPE ASCIZ STRING
2943 014306 000421          BR      70$      ;;GET OVER THE ASCIZ
2944 014352 104401 014360     TYPE      ,73$      ;;TYPE ASCIZ STRING
2945 014356 000417          BR      72$      ;;GET OVER THE ASCIZ
2946 014416 000000          HALT
2947
2948                ;*HOUSEKEEPING
2949
2950 014420 005037 047274      CLR      @#PRITEM      ;CLEAR THE PREVIOUS ERROR NUMBER
2951 014424 005237 004716     INC      @#UNIT      ;ADD ONE TO THE UNIT NO.
2952 014430 042737 177770 004716 BIC      #'C7,@#UNIT  ;TRUNCATE TO LOW ORDER 3 BITS
2953 014436 004737 041456     JSR      PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
2954
2955
2956                ;*ATTEMPT TO ADDRESS THE NEW UNIT NUMBER
2957
2958 014442 017700 165636      MOV      @#RHDST,RO   ;ATTEMPT TO ADDRESS THE NEW DRIVE NO.
2959 014446 004737 043370     JSR      PC,@#PUTREG  ;TAKE REG. SNAPSHOT IN CASE OF ERROR
2960 014452 032737 010000 002360 BIT      #'NED,@#CS2  ;TEST FOR NON EXISTENT DRIVE
2961 014460 001423          BEQ      7$          ;CONTINUE IF 'NED' IS NOT SET - DRIVE
2962                                ;SHOULD BE EXISTENT ON THE BUSS
2963 014462 104102          ERROR 102      ;UNIT NOT AVAILABLE AFTER ADDRESS
2964                                ;PLUG REPLACED
2965 014464 104401 014472     TYPE      ,75$      ;;TYPE ASCIZ STRING
2966 014470 000407          BR      74$      ;;GET OVER THE ASCIZ
2967 014510 013746 004716     MOV      @#UNIT,-(SP)  ;;GET THE BAD UNIT NUMBER
2968 014514 104405          TYPDS           ;;TYPE IT OUT
2969 014516 104401 014524     TYPE      ,77$      ;;TYPE ASCIZ STRING
2970 014522 000402          BR      76$      ;;GET OVER THE ASCIZ
2971
  
```



```
2972
2973
2974
2975 014530 023737 004716 004752 7$: CMP @#UNIT,@#INUNIT ;HAVE WE INCREMENTED BACK TO THE
2976 ;ORIGINAL UNIT NO. YET ?
2977 014536 001233 BNE 6$ ;NO..DO NEXT ADDRESS PLUG
2978 ;YES..CONTINUE WITH TESTS
2979
2980
2981
2982 ;*SET 'VV' IN RHDS1 AFTER RESET FROM THE RECALIBRATE
2983 ;*CAUSED BY PULLING THE ADDRESS PLUGS OUT
2984
2985 014540 013746 002460 MOV @#PKACK,-(SP) ;GET READY TO MOVE COMMAND
2986 014544 052716 000001 BIS #GO,(SP) ;GET READY TO SET GO
2987 014550 012677 165524 MOV (SP)+,@RHCS1 ;GO WITH
2988 014554 011100 MOV @R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
2989 014556 011305 MOV @R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
2990 014560 104413 WAT ;WAIT FOR VV BIT TO SET
2991 014562 002322 RHDS1 ;WAIT FOR RHDS1 REGISTER
2992 014564 000100 VV ;WAIT FOR VV BIT IN RHDS1 REGISTER
2993 014566 000001 1. ;ALLOW 10 MICRO SECONDS
2994 014570 000001 1. ;VV MUST SET BETWEEN
2995
2996 014572 104401 014600 TYPE ,82$ ;:TYPE ASCIZ STRING
2997 014576 000436 BR 81$ ;:GET OVER THE ASCIZ
2998 014674 104401 014702 TYPE ,84$ ;:TYPE ASCIZ STRING
2999 014700 000425 BR 83$ ;:GET OVER THE ASCIZ
```

```
3000
3001
3002          .SBTTL
3003          .SBTTL  **DRIVE COMMAND TESTS**
3004          .SBTTL
3005
3006
3007
3008
3009
3010
3011 014754 000004          TST16: SCOPE
3012 014756 012706 001000  MOV      #STACK,SP          ;RESET STACK
3013 014762 012737 000016 004604  MOV      #16,@TSTNM        ;SAVE TEST NUMBER
3014 014770 004737 041456  JSR      PC,@CLDISK        ;SET R1-RHCS1, R2-RHCS2
3015 014774 004737 041536  JSR      PC,@CHECKT        ;CHECK DVA,RDY,MOL,DPR,DRY,VV 1
3016 015000 104401 066720  TYPE     .CPHALT           ;CANNOT CONTINUE TESTING IF ANY OF
3017 015004 000000          HALT                       ;STOP
3018
3019 015006 013777 002422 165264  MOV      @NOPERA,@RHCS1    ;GET READY FOR NOPERA
3020
3021          ;*NOW SAVE REGISTERS FOR COMPARISON AFTER NO OPERATION
3022
3023 015014 004037 041624  JSR      RO,@SAVER          ;SAVE REGISTERS
3024 015020 002272          RHWC                       ;RHWC IS THE FIRST REGISTER SAVED
3025 015022 004612          SAVERE                     ;STARTING ADDRESS OF WHERE
3026 015024 000022          18.                       ;NUMBER OF REGISTERS
3027
3028 015026 013777 004606 165232  MOV      @RP4VEC,@RPVEC    ;SET RP04 VECTOR ADDRESS
3029
3030 015034 013746 002422  MOV      @NOPERA,-(SP)     ;GET READY TO MOVE COMMAND
3031 015040 052716 000001  BIS      #GO,(SP)         ;GET READY TO SET GO
3032 015044 012677 165230  MOV      (SP)+,@RHCS1     ;GO WITH
3033
3034 015050 104413          WAT                       ;WAIT FOR RDY BIT TO SET
3035 015052 002300          RHCS1                     ;WAIT FOR RHCS1 REGISTER
3036 015054 000200          RDY                       ;WAIT FOR RDY BIT IN RHCS1 REGISTER
3037 015056 000001          1.                         ;ALLOW 10 MICRO SECONDS
3038 015060 000001          1.                         ;RDY MUST SET BETWEEN
3039
3040          ;*NOW COMPARE REGISTERS BEFORE NO-OP COMMAND
3041          ;*WITH AFTER NO-OP COMMAND
3042 015062 004037 042444  JSR      RO,@COMREG        ;COMPARE SAVED REGISTERS WITH
3043 015066 004612          SAVERE                     ;GOOD DATA SAVED IN 'SAVERE'
3044 015070 002354          WC                       ;TEST DATA STARTING FROM 'RHWC'
3045 015072 000022          18.                       ;18. REGISTERS TO BE COMPARED
3046 015074 015100          1$                        ;RETURN TO 1$ ON ERROR
3047 015076 015104          2$                        ;RETURN TO 2$ ON NO ERROR
3048
3049 015100 104016          1$: ERROR 16              ;GIVING A NO-OP COMMAND
3050 015102 000207          RTS PC                    ;CAUSED AN ERROR
3051          ;NO REGISTERS SHOULD CHANGE
3052          ;GOOD DATA GIVES REGISTER
3053          ;CONTENTS BEFORE COMMAND
3054          ;RECEIVED DATA GIVES REGISTER
3055          ;CONTENTS AFTER COMMAND
```

CZRJICO, RP04/5/6 FCTNL CTLR1
CZRJIC.P11 27-JUL-78 13:05

MACY11 30A(1052) 27-JUL-78 13:06 PAGE 73
T16 NO OPERATION FUNCTION TEST

H 6

SEQ 0072

3056

CZ
CZ

3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112

015104
015104 004737 041456
015110 012700 002272
015114 012730 177777
015120 012730 177777
015124 052730 043010
015130 012730 001400
015134 012730 000000
015140 012730 177777
015144 012730 000000
015150 012730 177777
015154 012730 177777
015160 012730 000000
015164 010046
015166 011446
015170 011246
015172 013700 004742
015176 005012
015200 012705 000010
015204 006000
015206 103002
015210 012714 177777
015214 005212
015216 005305
015220 001371
015222 012612
015224 012614
015226 012600
015230 005720
015232 012730 177776
015236 004737 041536
015242 104401 066720
015246 000000
015250 013777 002422 165022
015256 004037 041624
015262 002272
015264 004612
015266 000022
015270 013777 004606 164770
015276 013746 002422
015302 052716 000001
015306 012677 164766
015312 104413
015314 002300
015316 000200
015320 000001

2\$:

87\$:

88\$:

.*NOW REPEAT TEST BY MOVING IN ALL POSSIBLE ONES

```
JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
MOV #RHWC,R0 ;ADDR. OF ADDR OF RHWC IN R0
MOV #177777,@(R0)+ ;LOAD 177777 INTO RHWC
MOV #177777,@(R0)+ ;LOAD 177777 INTO RHBA
BIS #43010,@(R0)+ ;LOAD 43010 INTO RHCS2
MOV #1400,@(R0)+ ;LOAD 1400 INTO RHCS1
MOV #0,@(R0)+ ;LOAD 0 INTO RHER1
MOV #177777,@(R0)+ ;LOAD 177777 INTO RHDST
MOV #0,@(R0)+ ;LOAD 0 INTO RHER2
MOV #177777,@(R0)+ ;LOAD 177777 INTO RHOF
MOV #177777,@(R0)+ ;LOAD 177777 INTO RHCA
MOV #0,@(R0)+ ;LOAD 0 INTO RHER3
MOV R0,-(SP) ;;PUSH R0 ON STACK
MOV @R4,-(SP) ;SAVE RHER1 TO REINSTATE LATER
MOV @R2,-(SP) ;SAVE RHCS2 TO BE REINSTATED
MOV @#TOTALAT,R0 ;GET DRIVES PRESENT
CLR @R2 ;CLEAR RHCS2 AND CARRY
MOV #8.,R5 ;COUNTER
ROR R0 ;GET BIT INTO CARRY
BCC 88$ ;BRANCH IF NO UNIT ON THIS BIT
MOV #-1,@R4 ;MOVE INTO ERROR REGISTER
INC @R2 ;INCREMENT RHCS2 TO NEXT UNIT
DEC R5 ;COUNT
BNE 87$ ;BRANCH IF 8 NOT DONE
MOV (SP)+,@R2 ;REINSTATE RHCS2
MOV (SP)+,@R4 ;REINSTATE RHER1
MOV (SP)+,R0 ;;POP STACK INTO R0
TST (R0)+ ;GET OVER PHAS IN R0
MOV #177776,@(R0)+ ;LOAD 177776 INTO RHMR

JSR PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV - 1
TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF ANY OF
HALT ;STOP
MOV @#NOPERA,@RHCS1 ;GET READY FOR NOPERA

.*NOW SAVE REGISTERS FOR COMPARISON AFTER A NO-OP
JSR R0,@#SAVER ;SAVE REGISTERS
RHWC ;RHWC IS THE FIRST REGISTER SAVED
SAVERE ;STARTING ADDRESS OF WHERE
18. ;NUMBER OF REGISTERS

MOV @#RP4VEC,@RPVEC ;SET RPO4 VECTOR ADDRESS

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

WAT ;WAIT FOR RDY BIT TO SET
RHCS1 ;WAIT FOR RHCS1 REGISTER
RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
1. ;ALLOW 10 MICRO SECONDS
```

```

3113 015322 000001      1.          ;RDY MUST SET BETWEEN
3114
3115                    ;*CHANGE REGISTERS TO EXPECTED VALUES
3116
3117 015324 005737 004750  TST      @#RH70      ;RUNNING ON AN RH70 ?
3118 015330 001406      BEQ      5$          ;IF NOT, SKIP NEXT
3119 015332 005737 004722  TST      @#NUNIT     ;TESTING MORE THAN ONE DRIVE ?
3120 015336 001003      BNE      5$          ;SKIP NEXT IF SO
3121 015340 042737 100000 004620  BIC      #SC,@#SAVERE+6 ;CLEAR 'SC' IN RHCS1
3122
3123 015346                    5$:
3124 015346 043737 004740 004636  BIC      @#ATTENT,@#SAVERE+24 ;CLEAR APPROPRIATE ATA BITS
3125 015354 004037 042336      JSR      RO,@#CHREG   ;CHANGE BITS IN SAVED REGISTER
3126 015360 002322      RHDS1     ;CHANGE RHDS1 REGISTER
3127 015362 000001      1          ;1 BIT/BITS TO BE CHANGED
3128 015364 000000      0          ;NEW VALUE OF ATA IS 0
3129 015366 100000      ATA        ;CHANGE ATA BIT
3130
3131
3132                    ;*NOW COMPARE REGISTERS BEFORE NO-OP WITH
3133                    ;*AFTER NO-OP COMMAND
3134
3135 015370 004037 042444      JSR      RO,@#COMREG   ;COMPARE SAVED REGISTERS WITH
3136 015374 004612      SAVERE     ;GOOD DATA SAVED IN 'SAVERE'
3137 015376 002354      WC          ;TEST DATA STARTING FROM 'RHWC'
3138 015400 000022      18.         ;18. REGISTERS TO BE COMPARED
3139 015402 015406      3$         ;RETURN TO 3$ ON ERROR
3140 015404 015412      4$         ;RETURN TO 4$ ON NO ERROR
3141
3142 015406 104016      3$:      ERROR    '6      ;GIVING A NO-OP COMMAND
3143 015410 000207      RTS      PC        ;CAUSED AN ERROR
3144                    ;NO REGISTERS SHOULD CHANGE
3145                    ;GOOD DATA GIVES REGISTER
3146                    ;CONTENTS BEFORE COMMAND
3147                    ;RECEIVED DATA GIVES REGISTER
3148                    ;CONTENTS AFTER COMMAND
3149 015412                    4$:
3150
3151

```

```

3152 015412 000004          TST17: SCOPE
3153 015414 012706 001000      MOV #STACK,SP          ;RESET STACK
3154 015420 012737 000017 004604  MOV #17,@TSTNM        ;SAVE TEST NUMBER
3155 015426 004737 041456      JSR PC,@CLDISK        ;SET R1-RHCS1, R2-RHCS2
3156 015432 004737 041536      JSR PC,@CHECKT        ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
3157 015436 104401 066720      TYPE ,CPHALT          ;CANNOT CONTINUE TESTING IF ANY OF
3158 015442 000000          HALT                   ;STOP
3159
3160                               ;*WRITE ALL WRITABLE REGISTER BITS
3161
3162 015444 012700 002272      MOV #RHC,R0           ;ADDR. OF ADDR. OF RHC IN R0
3163 015450 012730 177777      MOV #177777,@(R0)+   ;LOAD 177777 INTO RHC
3164 015454 012730 177776      MOV #177776,@(R0)+   ;LOAD 177776 INTO RHBA
3165 015460 052730 043010      BIS #43010,@(R0)+    ;LOAD 43010 INTO RHCS2
3166 015464 012730 001400      MOV #1400,@(R0)+     ;LOAD 1400 INTO RHCS1
3167 015470 012730 000000      MOV #0,@(R0)+        ;LOAD 0 INTO RHER1
3168 015474 012730 177777      MOV #177777,@(R0)+   ;LOAD 177777 INTO RHDST
3169 015500 012730 000000      MOV #0,@(R0)+        ;LOAD 0 INTO RHER2
3170 015504 012730 177777      MOV #177777,@(R0)+   ;LOAD 177777 INTO RHOF
3171 015510 012730 177777      MOV #177777,@(R0)+   ;LOAD 177777 INTO RHCA
3172 015514 012730 000000      MOV #0,@(R0)+        ;LOAD 0 INTO RHER3
3173 015520 010046          MOV R0,-(SP)          ;:PUSH R0 ON STACK
3174 015522 011446          MOV @R4,-(SP)         ;:SAVE RHER1 TO REINSTATE LATER
3175 015524 011246          MOV @R2,-(SP)         ;:SAVE RHCS2 TO BE REINSTATED
3176 015526 013700 004742      MOV @TOTALAT,R0       ;:GET DRIVES PRESENT
3177 015532 005012          CLR @R2               ;:CLEAR RHCS2 AND CARRY
3178 015534 012705 000010      MOV #8.,R5            ;:COUNTER
3179 015540 006000          ROR R0                ;:GET BIT INTO CARRY
3180 015542 103002          BCC 85$               ;:BRANCH IF NO UNIT ON THIS BIT
3181 015544 012714 177777      MOV #-1,@R4           ;:MOVE INTO ERROR REGISTER
3182 015550 005212          INC @R2               ;:INCREMENT RHCS2 TO NEXT UNIT
3183 015552 005305          DEC R5                ;:COUNT
3184 015554 001371          BNE 84$               ;:BRANCH IF 8 NOT DONE
3185 015556 012612          MOV (SP)+,@R2         ;:REINSTATE RHCS2
3186 015560 012614          MOV (SP)+,@R4         ;:REINSTATE RHER1
3187 015562 012600          MOV (SP)+,R0          ;:POP STACK INTO R0
3188 015564 005720          TST (R0)+             ;:GET OVER PHAS IN R0
3189 015566 012730 177776      MOV #177776,@(R0)+   ;LOAD 177776 INTO RHMR
3190
3191 015572 017737 164536 004654  MOV @RHCC,@SAVERE+42  ;:SAVE RHCC IN SAVERE TABLE
3192 015600 013777 004606 164460  MOV @RP4VEC,@RPVEC   ;:SET RP04 VECTOR ADDRESS
3193
3194 015606 013746 002430          MOV @DCLEAR,-(SP)    ;:GET READY TO MOVE COMMAND
3195 015612 052716 000001          BIS #GO,(SP)         ;:GET READY TO SET GO
3196 015616 012677 164456          MOV (SP)+,@RHCS1     ;:GO WITH
3197
3198 015622 104413          WAT                   ;:WAIT FOR RDY BIT TO SET
3199 015624 002300          RHCS1                ;:WAIT FOR RHCS1 REGISTER
3200 015626 000200          RDY                   ;:WAIT FOR RDY BIT IN RHCS1 REGISTER
3201 015630 000001          1.                   ;:ALLOW 10 MICRO SECONDS
3202 015632 000001          1.                   ;:RDY MUST SET BETWEEN
3203
3204                               ;*NOW LOAD 'SAVERE' REGISTER SNAPSHOT WITH EXPECTED VALUES
3205
3206 015634 004037 041360          JSR R0,@FILLRE        ;:MOV 177777 INTO SAVED RHC
3207 015640 002272          RHC                   ;:SAVED REGISTER TO CHANGE
  
```

```

3208 015642 177777          177777          ;DATA
3209 015644 004037 041360  JSR      RO,@#FILLRE  ;MOV 177776 INTO SAVED RHBA
3210 015650 002274          RHBA          ;SAVED REGISTER TO CHANGE
3211 015652 177776          177776          ;DATA
3212 015654 005037 004616  CLR      @#SAVERE+4    ;CLEAR LOCATION FOR RHCS2
3213 015660 053737 004716 004616  BIS      @#UNIT,@#SAVERE+4;PUT UNIT # BACK IN THE SAVED RHCS2
3214
3215 015666 005737 004750  TST      @#RH70        ;RUNNING ON AN RH70 CONTROLLER ?
3216 015672 001021          BNE      8$           ;IF SO SKIP NEXT RH11 CODE
3217
3218 015674 004037 042336  JSR      RO,@#CHREG    ;CHANGE BITS IN SAVED REGISTER
3219 015700 002276          RHCS2        ;CHANGE RHCS2 REGISTER
3220 015702 000003          3           ;3 BIT/BITS TO BE CHANGED
3221 015704 000001          1           ;NEW VALUE OF IR IS 1
3222 015706 000100          IR          ;CHANGE IR BIT
3223 015710 000001          1           ;NEW VALUE OF BAI IS 1
3224 015712 000010          BAI         ;CHANGE BAI BIT
3225 015714 000001          1           ;NEW VALUE OF MXF IS 1
3226 015716 001000          MXF        ;CHANGE MXF BIT
3227 015720 004037 042336  JSR      RO,@#CHREG    ;CHANGE BITS IN SAVED REGISTER
3228 015724 002300          RHCS1       ;CHANGE RHCS1 REGISTER
3229 015726 000001          1           ;1 BIT/BITS TO BE CHANGED
3230 015730 000001          1           ;NEW VALUE OF SC IS 1
3231 015732 100000          SC         ;CHANGE SC BIT
3232 015734 000416          BR         9$        ;SKIP NEXT RH70 CODE
3233
3234 015736          8$:
3235 015736 004037 042336  JSR      RO,@#CHREG    ;CHANGE BITS IN SAVED REGISTER
3236 015742 002276          RHCS2       ;CHANGE RHCS2 REGISTER
3237 015744 000002          2           ;2 BIT/BITS TO BE CHANGED
3238 015746 000001          1           ;NEW VALUE OF IR IS 1
3239 015750 000100          IR         ;CHANGE IR BIT
3240 015752 000001          1           ;NEW VALUE OF BAI IS 1
3241 015754 000010          BAI        ;CHANGE BAI BIT
3242 015756 005737 004722  TST      @#NUNIT      ;TESTING MORE THAN ONE DRIVE ?
3243 015762 001003          BNE      9$         ;SKIP NEXT IF SO
3244 015764 042737 100000 004620  BIC      #SC,@#SAVERE+6;CLEAR 'SC' IF NOT
3245 015772          9$:
3246 015772 004037 041360  JSR      RO,@#FILLRE  ;MOV 0 INTO SAVED RHER1
3247 015776 002302          RHER1      ;SAVED REGISTER TO CHANGE
3248 016000 000000          0         ;DATA
3249 016002 004037 041360  JSR      RO,@#FILLRE  ;MOV 17437 INTO SAVED RHDST
3250 016006 002304          RHDST      ;SAVED REGISTER TO CHANGE
3251 016010 017437          17437     ;DATA
3252 016012 004037 041360  JSR      RO,@#FILLRE  ;MOV 0 INTO SAVED RHER2
3253 016016 002306          RHER2      ;SAVED REGISTER TO CHANGE
3254 016020 000000          0         ;DATA
3255 016022 004037 041360  JSR      RO,@#FILLRE  ;MOV 116000 INTO SAVED RHOF
3256 016026 002310          RHOF       ;SAVED REGISTER TO CHANGE
3257 016030 116000          116000    ;DATA
3258 016032 004037 041360  JSR      RO,@#FILLRE  ;MOV 1777 INTO SAVED RHCA
3259 016036 002312          RHCA      ;SAVED REGISTER TO CHANGE
3260 016040 001777          1777     ;DATA
3261 016042 004037 041360  JSR      RO,@#FILLRE  ;MOV 0 INTO SAVED RHER3
3262 016046 002314          RHER3      ;SAVED REGISTER TO CHANGE
3263 016050 000000          0         ;DATA
  
```

```

3264 016052 013746 004742      MOV      @#TOTALAT,-(SP) ;GET ALL BITS OF DRIVE & PRESENT
3265                               ;IN RHAS
3266 016056 043716 004740      BIC      @#ATTENT,(SP) ;CLEAR WORKING DRIVE BIT
3267 016062 012637 004636      MOV      (SP)+,@#SAVERE+24 ;MOVE THIS INTO RHAS POSITION
3268 016066 004037 041360      JSR      RO,@#FILLRE ;MOV 400 INTO SAVED RHMR
3269 016072 002320                RHMR    ;SAVED REGISTER TO CHANGE
3270 016074 000400                400    ;DATA
3271
3272 016076                3$:
3273 016076 004037 041360      JSR      RO,@#FILLRE ;MOV 10700 INTO SAVED RHDS1
3274 016102 002322                RHDS1  ;SAVED REGISTER TO CHANGE
3275 016104 010700                10700 ;DATA
3276
3277 016106 013737 002406 004644 4$: MOV      @#DT,@#SAVERE+32 ;MOVE DRIVE TYPE INTO RMDT
3278                               ;POSITION OF SAVRE TABLE
3279 016114 013737 002410 004646  MOV      @#SN,@#SAVERE+34 ;MOVE SERIAL NUMBER INTO RHSN
3280                               ;POSITION OF SAVERE TABLE
3281
3282 016122 004037 041360      JSR      RO,@#FILLRE ;MOV 0 INTO SAVED RHEC1
3283 016126 002330                RHEC1  ;SAVED REGISTER TO CHANGE
3284 016130 000000                0      ;DATA
3285 016132 004037 041360      JSR      RO,@#FILLRE ;MOV 0 INTO SAVED RHEC2
3286 016136 002332                RHEC2  ;SAVED REGISTER TO CHANGE
3287 016140 000000                0      ;DATA
3288 016142 004037 042336      JSR      RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
3289 016146 002300                RHCS1  ;CHANGE RHCS1 REGISTER
3290 016150 000001                1      ;1 BIT/BITS TO BE CHANGED
3291 016152 000001                1      ;NEW VALUE OF PAR IS 1
3292 016154 000010                PAR    ;CHANGE PAR BIT
3293
3294                               ;*NOW THAT SAVERE TABLE HAS BEEN LOADED WITH
3295                               ;*EXPECTED VALUES, THE REGISTERS WILL BE COMPARED
3296                               ;*WITH SAVERE TABLE
3297
3298 016156 004037 042444      JSR      RO,@#COMREG ;COMPARE SAVED REGISTERS WITH
3299 016162 004612                SAVERE ;GOOD DATA SAVED IN 'SAVERE'
3300 016164 002354                WC     ;TEST DATA STARTING FROM 'RHWC'
3301 016166 000022                18.   ;18. REGISTERS TO BE COMPARED
3302 016170 016174                5$    ;RETURN TO 5$ ON ERROR
3303 016172 016200                6$    ;RETURN TO 6$ ON NO ERROR
3304
3305 016174 104017                5$:   ERROR 17 ;DRIVE CLEAR COMMAND
3306 016176 000207                RTS   PC ;GAVE AN ERROR
3307                               ;GOOD DATA HAS WHAT SHOULD
3308                               ;BE IN REGISTER AFTER A
3309                               ;DRIVE CLEAR
3310                               ;RECEIVED DATA HAS WHAT
3311                               ;THE REGISTER ACTUALLY
3312                               ;CONTAINED
3313 016200                6$:
3314

```



```

3315
3316 016200 000004          TST20: SCOPE
3317 016202 012706 001000 MOV      #STACK,SP      ;RESET STACK
3318 016206 012737 000020 004604 MOV      #20,@TSTNM    ;SAVE TEST NUMBER
3319 016214 004737 041456 JSR      PC,@CLDISK   ;SET R1-RHCS1, R2-RHCS2
3320 016220 005737 004724 TST      @NOPUSH     ;IS THIS A 220 START ?
3321 016224 001007          BNE      1$          ;SKIP THIS TEST IF SO
3322 016226 005737 000042 TST      @42         ;MONITOR (ACT 11) RETURN ADDRESS ?
3323 016232 001004          BNE      1$          ;SKIP THIS TEST
3324 016234 005737 001100 TST      @SPASS      ;FIRST PASS ?
3325 016240 001001          BNE      1$          ;SKIP THIS TEST IF NOT
3326 016242 000402          BR       2$          ;CONTINUE WITH THIS TEST
3327 016244 000137 016744 JMP      TST21 ;     JUMP TO NEXT TEST -----)
3328 016250 104401 016256 TYPE     ,65$        ;:TYPE ASCIZ STRING
3329 016254 000407          BR       64$        ;:GET OVER THE ASCIZ
3330
3331 016274 013746 004716          MOV      @UNIT,-(SP) ;GET UNIT UNDER TEST
3332 016300 104405          TYPDS   ;           ;TYPE IT OUT
3333 016302 104401 001223 TYPE     ,$CRLF      ;
3334 016306 032713 010000 3$: BIT      #MOL,@R3 ;MOL WILL BE HIGH TILL STOP IS HIT
3335 016312 001375          BNE     3$          ;WAIT TILL STOP IS HIT
3336
3337 016314 104401 016322          TYPE     ,67$        ;:TYPE ASCIZ STRING
3338 016320 000406          BR       66$        ;:GET OVER THE ASCIZ
3339
3340 016336 013746 004716          MOV      @UNIT,-(SP) ;GET UNIT UNDER TEST
3341 016342 104405          TYPDS   ;           ;TYPE IT
3342 016344 104401 001223 TYPE     ,$CRLF      ;
3343
3344 016350 032713 010000 4$: BIT      #MOL,@R3 ;MOL WILL BE LOW TILL FILE READY
3345 016354 001775          BEQ     4$          ;WAIT TILL FILE READY
3346 016356 004737 041456 JSR      PC,@CLDISK   ;SET R1-RHCS1, R2-RHCS2
3347 016362 004737 041514 JSR      PC,@CHECK    ;CHECK THAT DVA,RDY,MOL,DPR,DRY = 1
3348 016366 104401 066720 TYPE     ,CPHALT     ;CANNOT CONTINUE TESTS IF THEY AREN'T
3349 016372 000000          HALT                    ;STOP
3350 016374 012700 002272 MOV      #RHWC,R0    ;ADDR. OF ADDR. OF RHWC IN R0
3351
3352
3353          ;*NOW INITIALIZE ALL THE REGISTERS
3354
3355 016400 012730 177777 MOV      #177777,@(R0)+ ;LOAD 177777 INTO RHWC
3356 016404 012730 177777 MOV      #177777,@(R0)+ ;LOAD 177777 INTO RHBA
3357 016410 052730 043010 BIS      #43010,@(R0)+ ;LOAD 43010 INTO RHCS2
3358 016414 012730 001400 MOV      #1400,@(R0)+ ;LOAD 1400 INTO RHCS1
3359 016420 012730 000000 MOV      #0,@(R0)+ ;LOAD 0 INTO RHER1
3360 016424 012730 177777 MOV      #177777,@(R0)+ ;LOAD 177777 INTO RHDST
3361 016430 012730 000000 MOV      #0,@(R0)+ ;LOAD 0 INTO RHER2
3362 016434 012730 177777 MOV      #177777,@(R0)+ ;LOAD 177777 INTO RHOF
3363 016440 012730 177777 MOV      #177777,@(R0)+ ;LOAD 177777 INTO RHCA
3364 016444 012730 000000 MOV      #0,@(R0)+ ;LOAD 0 INTO RHER3
3365 016450 010046 MOV      R0,-(SP) ;:PUSH R0 ON STACK
3366 016452 011446 MOV      @R4,-(SP) ;SAVE RHER1 TO REINSTATE LATER
3367 016454 011246 MOV      @R2,-(SP) ;SAVE RHCS2 TO BE REINSTATED
3368 016456 013700 004742 MOV      @TOTALAT,R0 ;GET DRIVES PRESENT
3369 016462 005012 CLR      @R2 ;CLEAR RHCS2 AND CARRY
3370 016464 012705 000010 MOV      #8.,R5 ;COUNTER
  
```

```
3371 016470 006000      88$: ROR    R0          ;GET BIT INTO CARRY
3372 016472 103002      BCC    89$          ;BRANCH IF NO UNIT ON THIS BIT
3373 016474 012714 177777  MOV    #-1,@R4      ;MOVE INTO ERROR REGISTER
3374 016500 005212      89$: INC    @R2          ;INCREMENT RHCS2 TO NEXT UNIT
3375 016502 005305      DEC    R5           ;COUNT
3376 016504 001371      BNE    88$          ;BRANCH IF 8 NOT DONE
3377 016506 012612      MOV    (SP)+,@R2    ;REINSTATE RHCS2
3378 016510 012614      MOV    (SP)+,@R4    ;REINSTATE RHER1
3379 016512 012600      MOV    (SP)+,R0     ;POP STACK INTO R0
3380 016514 005720      TST    (R0)+        ;GET OVER PHAS IN R0
3381 016516 012730 177776  MOV    #177776,@(R0)+ ;LOAD 177776 INTO RHMR
3382
3383 016522 013777 004740 163566  MOV    @#ATTENT,@RHAS ;CLEAR WORKING DRIVE 'ATA'
3384 016530 013777 002462 163542  MOV    @#READIN,@RHCS1 ;GET READY FOR READIN
3385
3386 ;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ-IN COMMAND
3387
3388 016536 004037 041624  JSR    R0,@#SAVER   ;SAVE REGISTERS
3389 016542 002272      RHWC          ;RHWC IS THE FIRST REGISTER SAVED
3390 016544 004612      SAVERE       ;STARTING ADDRESS OF WHERE
3391 016546 000022      18.         ;NUMBER OF REGISTERS
3392
3393 016550 013777 004606 163510  MOV    @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
3394
3395 016556 005737 004750      TST    @#RH70      ;RUNNING ON AN RH70 CONTROLLER ?
3396 016562 001411      BEQ    7$          ;SKIP NEXT FOR RH70 IF NOT
3397 016564 013746 002462  MOV    @#READIN,-(SP) ;GET READY TO MOVE COMMAND
3398 016570 052716 000001  BIS    #GO,(SP)     ;GET READY TO SET GO
3399 016574 012677 163500  MOV    (SP)+,@RHCS1 ;GO WITH
3400 016600 011100      MOV    @R1,R0      ;SAVE RHCS1 DURING ABOVE OPERATION
3401 016602 011305      MOV    @R3,R5      ;SAVE RHDS1 DURING ABOVE OPERATION
3402 016604 000406      BR     8$          ;SKIP NEXT FOR RH11
3403
3404 016606      7$:
3405 016606 013746 002462  MOV    @#READIN,-(SP) ;GET READY TO MOVE COMMAND
3406 016612 052716 000101  BIS    #GO!IE,(SP)  ;GET READY TO SET 'GO' AND
3407 016616 012677 163456  MOV    (SP)+,@RHCS1 ;GO WITH
3408
3409 016622      8$:
3410 016622 104413      WAT          ;WAIT FOR VV BIT TO SET
3411 016624 002322      RHDS1       ;WAIT FOR RHDS1 REGISTER
3412 016626 000100      VV          ;WAIT FOR VV BIT IN RHDS1 REGISTER
3413 016630 000001      1.         ;ALLOW 10 MICRO SECONDS
3414 016632 000001      1.         ;VV MUST SET BETWEEN
3415
3416 ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
3417
3418 016634 004037 041360  JSR    R0,@#FILLRE  ;MOV 0 INTO SAVED RHCA
3419 016640 002312      RHCA        ;SAVED REGISTER TO CHANGE
3420 016642 000000      0           ;DATA
3421 016644 004037 041360  JSR    R0,@#FILLRE  ;MOV 0 INTO SAVED RHDST
3422 016650 002304      RHDST       ;SAVED REGISTER TO CHANGE
3423 016652 000000      0           ;DATA
3424 016654 004037 042336  JSR    R0,@#CHREG   ;CHANGE BITS IN SAVED REGISTER
3425 016660 002310      RHOF        ;CHANGE RHOF REGISTER
3426 016662 000003      3           ;3 BIT/BITS TO BE CHANGED
```

```
3427 016664 000000 0 ;NEW VALUE OF FMT22 IS 0
3428 016666 010000 FMT22 ;CHANGE FMT22 BIT
3429 016670 000000 0 ;NEW VALUE OF ECI IS 0
3430 016672 004000 ECI ;CHANGE ECI BIT
3431 016674 000000 0 ;NEW VALUE OF HCI IS 0
3432 016676 002000 HCI ;CHANGE HCI BIT
3433 016700 004037 042336 JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
3434 016704 002322 RHDS1 ;CHANGE RHDS1 REGISTER
3435 016706 000001 1 ;1 BIT/BITS TO BE CHANGED
3436 016710 000001 1 ;NEW VALUE OF VV IS 1
3437 016712 000100 VV ;CHANGE VV BIT
3438
3439 ;*NOW THAT SAVERE TABLE WITH SAVED REGISTERS HAVE
3440 ;*THE EXPECTED VALUE AFTER A READ-IN COMMAND
3441 ;*COMPARISONS ARE MADE
3442
3443 016714 004037 042444 JSR RO,@#COMREG ;COMPARE SAVED REGISTERS WITH
3444 016720 004612 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
3445 016722 002354 WC ;TEST DATA STARTING FROM 'RHWC'
3446 016724 000022 18. ;18. REGISIERS TO BE COMPARED
3447 016726 016732 5$ ;RETURN TO 5$ ON ERROR
3448 016730 016736 6$ ;RETURN TO 6$ ON NO ERROR
3449
3450 016732 104020 5$: ERROR 20 ;READ IN COMMAND GAVE AN
3451 016734 000207 RTS PC ;ERROR
3452 ;GOOD DATA HAS WHAT SHOULD
3453 ;BE IN REGISTER AFTER A
3454 ;READ-IN COMMAND
3455 ;RECEIVED DATA HAS WHAT
3456 ;THE REGISTER ACTUALLY CONTAINED
3457 ;THE FOLLOWING SHOULD
3458 ;BE THE REGISTER CONTENTS
3459 ;RHCA=0, RHDST = 0
3460 ;RHOF SHOULD HAVE FMT22 = 0,
3461 ;HCI = 0, ECI = 0,
3462 ;RHDS1 SHOULD HAVE VV = 1
3463 ;ALL OTHER BITS SHOULD
3464 ;BE UNCHANGED
3465 016736 012737 177777 047274 6$: MOV #-1,@#PRITEM ;CLEAR PREVIOUS ITEM NUMBER
```

```

3466
3467
3468 016744 000004          TST21: SCOPE
3469 016746 012706 001000  MOV      #STACK,SP          ;RESET STACK
3470 016752 012737 000021 004604  MOV      #21,@TSTNM        ;SAVE TEST NUMBER
3471 016760 004737 041456  JSR      PC,@CLDISK        ;SET R1-RHCS1, R2-RHCS2
3472 016764 004737 041514  JSR      PC,@CHECK         ;CHECK THAT DVA,RDY,MOL,DPR,DRY = 1
3473 016770 104401 066720  TYPE     ,CPHALT          ;CANNOT CONTINUE TESTS IF THEY AREN'T
3474 016774 000000          HALT
3475 016776 012700 002272  MOV      #RHWC,R0         ;ADDR. OF ADDR. OF RHWC IN R0
3476
3477
3478          ;*INITIALIZE ALL THE REGISTERS
3479
3480 017002 012730 177777  MOV      #177777,@(R0)+    ;LOAD 177777 INTO RHWC
3481 017006 012730 177777  MOV      #177777,@(R0)+    ;LOAD 177777 INTO RHBA
3482 017012 052730 043010  BIS      #43010,@(R0)+     ;LOAD 43010 INTO RHCS2
3483 017016 012730 001400  MOV      #1400,@(R0)+      ;LOAD 1400 INTO RHCS1
3484 017022 012730 000000  MOV      #0,@(R0)+         ;LOAD 0 INTO RHER1
3485 017026 012730 177777  MOV      #177777,@(R0)+    ;LOAD 177777 INTO RHDST
3486 017032 012730 000000  MOV      #0,@(R0)+         ;LOAD 0 INTO RHER2
3487 017036 012730 177777  MOV      #177777,@(R0)+    ;LOAD 177777 INTO RHOF
3488 017042 012730 177777  MOV      #177777,@(R0)+    ;LOAD 177777 INTO RHCA
3489 017046 012730 000000  MOV      #0,@(R0)+         ;LOAD 0 INTO RHER3
3490 017052 012730 177777  MOV      #-1,@(R0)+        ;CLEAR ALL BITS OF RHAS
3491 017056 012730 177776  MOV      #177776,@(R0)+    ;LOAD 177776 INTO RHMR
3492
3493 017062 013777 004740 163226  MOV      @ATTENT,@RHAS     ;CLEAR WORKING DRIVE 'ATA'
3494 017070 013777 002462 163202  MOV      @READIN,@RHCS1   ;GET READY FOR READIN
3495
3496          ;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ-IN COMMAND
3497
3498 017076 004037 041624  JSR      R0,@SAVER         ;SAVE REGISTERS
3499 017102 002272          RHWC     ;RHWC IS THE FIRST REGISTER SAVED
3500 017104 004612          SAVERE   ;STARTING ADDRESS OF WHERE
3501 017106 000022          18.     ;NUMBER OF REGISTERS
3502
3503 017110 013777 004606 163150  MOV      @RP4VEC,@RPVEC   ;SET RPO4 VECTOR ADDRESS
3504
3505 017116 005737 004750  TST      @RH70            ;RUNNING ON AN RH70 CONTROLLER ?
3506 017122 001411          BEQ     9$               ;SKIP NEXT IF NOT
3507 017124 013746 002462  MOV      @READIN,-(SP)    ;GET READY TO MOVE COMMAND
3508 017130 052716 000001  BIS      #GO,(SP)         ;GET READY TO SET GO
3509 017134 012677 163140  MOV      (SP)+,@RHCS1     ;GO WITH
3510 017140 011100          MOV     @R1,R0           ;SAVE RHCS1 DURING ABOVE OPERATION
3511 017142 011305          MOV     @R3,R5           ;SAVE RHDS1 DURING ABOVE OPERATION
3512 017144 000406          BR      10$             ;SKIP NEXT RH11 CODE
3513
3514 017146          9$:
3515 017146 013746 002462  MOV      @READIN,-(SP)    ;GET READY TO MOVE COMMAND
3516 017152 052716 000101  BIS      #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
3517 017156 012677 163116  MOV      (SP)+,@RHCS1     ;GO WITH
3518
3519 017162          10$:
3520 017162 104413          WAT
3521 017164 002300          RHCS1                ;WAIT FOR RDY BIT TO SET
                        ;WAIT FOR RHCS1 REGISTER

```

```

3522 017166 000200          RDY          ;WAIT FOR RDY BIT IN RHCS1 REGISTER
3523 017170 000001          1.          ;ALLOW 10 MICRO SECONDS
3524 017172 000001          1.          ;RDY MUST SET BETWEEN
3525
3526          ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
3527
3528 017174 004037 041360    JSR      RO,@#FILLRE ;MOV 0 INTO SAVED RHCA
3529 017200 002312          RHCA        ;SAVED REGISTER TO CHANGE
3530 017202 000000          0          ;DATA
3531 017204 004037 041360    JSR      RO,@#FILLRE ;MOV 0 INTO SAVED RHDST
3532 017210 002304          RHDST      ;SAVED REGISTER TO CHANGE
3533 017212 000000          0          ;DATA
3534 017214 004037 042336    JSR      RO,@#CHREG  ;CHANGE BITS IN SAVED REGISTER
3535 017220 002310          RHOF       ;CHANGE RHOF REGISTER
3536 017222 000003          3          ;3 BIT/BITS TO BE CHANGED
3537 017224 000000          0          ;NEW VALUE OF FMT22 IS 0
3538 017226 010000          FMT22     ;CHANGE FMT22 BIT
3539 017230 000000          0          ;NEW VALUE OF ECI IS 0
3540 017232 004000          ECI       ;CHANGE ECI BIT
3541 017234 000000          0          ;NEW VALUE OF HCI IS 0
3542 017236 002000          HCI       ;CHANGE HCI BIT
3543 017240 004037 042336    JSR      RO,@#CHREG  ;CHANGE BITS IN SAVED REGISTER
3544 017244 002322          RHDS1     ;CHANGE RHDS1 REGISTER
3545 017246 000001          1          ;1 BIT/BITS TO BE CHANGED
3546 017250 000001          1          ;NEW VALUE OF VV IS 1
3547 017252 000100          VV        ;CHANGE VV BIT
3548
3549          ;*NOW THAT SAVERE TABLE WITH SAVED REGISTERS HAVE
3550          ;*THE EXPECTED VALUE AFTER A READ-IN COMMAND
3551          ;*COMPARISONS ARE MADE
3552
3553 017254 004037 042444    JSR      RO,@#COMREG ;COMPARE SAVED REGISTERS WITH
3554 017260 004612          SAVERE    ;GOOD DATA SAVED IN 'SAVERE'
3555 017262 002354          WC      ;TEST DATA STARTING FROM 'RHWC'
3556 017264 000022          18.     ;18. REGISTERS TO BE COMPARED
3557 017266 017272          5$     ;RETURN TO 5$ ON ERROR
3558 017270 017276          6$     ;RETURN TO 6$ ON NO ERROR
3559
3560          5$:          ERROR  20      ;READ-IN COMMAND GAVE AN
3561 017274 000207          RTS    PC      ;ERROR
3562          ;GOOD DATA HAS WHAT SHOULD
3563          ;BE IN REGISTER AFTER A
3564          ;READ-IN COMMAND
3565          ;RECEIVED DATA HAS WHAT
3566          ;THE REGISTER ACTUALLY CONTAIN
3567          ;THE FOLLOWING SHOULD
3568          ;BE THE REGISTER CONTENTS
3569          ;RHCA = 0, RHDST = 0
3570          ;RHOF SHOULD HAVE FMT22  0,
3571          ;HCI = 0, ECI = 0,
3572          ;RHDS1 SHOULD HAVE VV = 1
3573          ;ALL OTHER BITS SHOULD
3574          ;BE UNCHANGED
3575 017276          6$:

```

```

3576 017276 005737 004744          TST    @#RPO6          ;TEST FOR RPO6 DRIVE
3577 017302 001401                   BEQ    7$              ;IF = 0, TREAT DRIVE AS AN RPO4
3578 017304 000402                   BR     8$              ;TREAT AS RPO6 - DO NEXT 'MAKECL'
3579 017306 000137 017352          7$:   JMP    @#DOG          ;DO SECOND FOLLOWING 'MAKECL'
3580 017312 8$:
3581
3582
3583
3584 017312 000004          TST22: SCOPE
3585 017314 012706 001000          MOV    #STACK,SP      ;RESET STACK
3586 017320 012737 000022 004604    MOV    #23-1,@#TSTNM  ;THIS SAVES TEST NUMBER
3587 017326 004737 041456          JSR    PC,@#CLDISK    ;INIT DRIVE
3588 017332 012777 000001 162760    MOV    #DMD,@RHMR     ;SET DIAGNOSTIC MODE
3589 017340 004037 041126          JSR    RO,@#MAKECYL   ;SUBROUTINE TO GIVE A SEEK
3590 017344 000777          777
3591 017346 000137 017406          JMP    @#FISH         ;DON'T DO NEXT 'MAKECL'
3592
3593
3594 017352          DOG:
3595 017352 000004          TST23: SCOPE
3596 017354 012706 001000          MOV    #STACK,SP      ;RESET STACK
3597 017360 012737 000023 004604    MOV    #24-1,@#TSTNM  ;THIS SAVES TEST NUMBER
3598 017366 004737 041456          JSR    PC,@#CLDISK    ;INIT DRIVE
3599 017372 012777 000001 162720    MOV    #DMD,@RHMR     ;SET DIAGNOSTIC MODE
3600 017400 004037 041126          JSR    RO,@#MAKECYL   ;SUBROUTINE TO GIVE A SEEK
3601 017404 000377          377
3602

```

CZ
CZ

3603							
3604							
3605							
3606	017406						
3607	017406	000004			FISH:	SCOPE	
3608	017410	012706	001000		TST24:	MOV	#STACK,SP ;RESET STACK
3609	017414	012737	000024	004604		MOV	#24,@TSTNM ;SAVE TEST NUMBER
3610	017422	004737	041456			JSR	PC,@CLDISK ;SET R1-RHCS1, R2-RHCS2
3611	017426	004737	041536			JSR	PC,@CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
3612	017432	104401	066720			TYPE	,CPHALT ;CANNOT CONTINUE TESTING IF ANY OF
3613	017436	000000				HALT	;STOP
3614	017440	012700	002272			MOV	#RHWC,RO ;ADDR. OF ADDR. OF RHWC IN RO
3615	017444	012730	177777			MOV	#177777,@(RO)+ ;LOAD 177777 INTO RHWC
3616	017450	012730	177776			MOV	#177776,@(RO)+ ;LOAD 177776 INTO RHBA
3617	017454	052730	000010			BIS	#010,@(RO)+ ;LOAD 010 INTO RHCS2
3618	017460	012730	001400			MOV	#1400,@(RO)+ ;LOAD 1400 INTO RHCS1
3619	017464	012730	000000			MOV	#0,@(RO)+ ;LOAD 0 INTO RHER1
3620	017470	012730	177777			MOV	#177777,@(RO)+ ;LOAD 177777 INTO RHDST
3621	017474	012730	000000			MOV	#0,@(RO)+ ;LOAD 0 INTO RHER2
3622	017500	012730	177777			MOV	#177777,@(RO)+ ;LOAD 177777 INTO RHOF
3623	017504	012730	177777			MOV	#177777,@(RO)+ ;LOAD 177777 INTO RHCA
3624	017510	012730	000000			MOV	#0,@(RO)+ ;LOAD 0 INTO RHER3
3625	017514	010046				MOV	RO,-(SP) ;PUSH RO ON STACK
3626	017516	011446				MOV	@R4,-(SP) ;SAVE RHER1 TO REINSTATE LATER
3627	017520	011246				MOV	@R2,-(SP) ;SAVE RHCS2 TO BE REINSTATED
3628	017522	013700	004742			MOV	@TOTALAT,RO ;GET DRIVES PRESENT
3629	017526	005012				CLR	@R2 ;CLEAR RHCS2 AND CARRY
3630	017530	012705	000010			MOV	#8.,R5 ;COUNTER
3631	017534	006000			84\$:	ROR	RO ;GET BIT INTO CARRY
3632	017536	103002				BCC	85\$;BRANCH IF NO UNIT ON THIS BIT
3633	017540	012714	177777			MOV	#-1,@R4 ;MOVE INTO ERROR REGISTER
3634	017544	005212			85\$:	INC	@R2 ;INCREMENT RHCS2 TO NEXT UNIT
3635	017546	005305				DEC	R5 ;COUNT
3636	017550	001371				BNE	84\$;BRANCH IF 8 NOT DONE
3637	017552	012612				MOV	(SP)+,@R2 ;REINSTATE RHCS2
3638	017554	012614				MOV	(SP)+,@R4 ;REINSTATE RHER1
3639	017556	012600				MOV	(SP)+,RO ;POP STACK INTO RO
3640	017560	005720				TST	(RO)+ ;GET OVER PHAS IN RO
3641	017562	012730	177776			MOV	#177776,@(RO)+ ;LOAD 177776 INTO RHMR
3642							
3643	017566	013777	002426	162504		MOV	@RECALI,@RHCS1 ;GET READY FOR RECAL!
3644							
3645							;*NOW SAVE REGISTERS FOR COMPARISON AFTER RECALIBRATE
3646							
3647	017574	004037	041624			JSR	RO,@SAVER ;SAVE REGISTERS
3648	017600	002272				RHWC	;RHWC IS THE FIRST REGISTER SAVED
3649	017602	004612				SAVERE	;STARTING ADDRESS OF WHERE
3650	017604	000022				18.	;NUMBER OF REGISTERS
3651	017606	013777	004606	162452		MOV	@RP4VEC,@RPVEC ;SET RPO4 VECTOR ADDRESS
3652	017614	013746	002426			MOV	@RECALI,-(SP) ;GET READY TO MOVE COMMAND
3653	017620	052716	000101			BIS	#GO!IE,(SP) ;GET READY TO SET 'GO' AND
3654	017624	012677	162450			MOV	(SP)+,@RHCS1 ;GO WITH
3655	017630	011100				MOV	@R1,RO ;SAVE RHCS1 DURING ABOVE OPERATION
3656	017632	011305				MOV	@R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
3657	017634	104413				WAT	;WAIT FOR DRY BIT TO SET
3658	017636	002322				RHDS1	;WAIT FOR RHDS1 REGISTER

```

3659 017640 000200 DRY ;WAIT FOR DRY BIT IN RHDS1 REGISTER
3660 017642 076377 31999. ;ALLOW 319990 MICRO SECONDS
3661 017644 056701 24001. ;DRY MUST SET BETWEEN
3662 017646 013746 002426 MOV @#RECALI,-(SP) ;SAVE COMMAND
3663 017652 052716 004301 BIS #DVA!GO!IE!RDY,(SP) ;INCLUDE DVA!GO!IE!RDY
3664 017656 005737 004722 TST @#NUNIT ;ARE THERE MORE THAN ONE UNIT
3665 017662 001413 BEQ 89$ ;BRANCH IF ONLY ONE UNIT
3666 017664 010037 004760 MOV R0,@#TMP4 ;GET RHCS1
3667 017670 042737 177677 004760 BIC #*CIE,@#TMP4 ;KEEP IE BIT
3668 017676 042716 000100 BIC #IE,(SP) ;CLEAR IE IN GOOD DATA
3669 017702 053716 004760 BIS @#TMP4,(SP) ;GET IE AS IS
3670 017706 052716 100000 BIS #SC,(SP) ;SET SC IN RHCS1
3671 017712 011637 001124 MOV (SP),@#SGDDAT ;SAVE FOR PRINTOUT
3672 017716 022600 CMP (SP)+,R0 ;DURING ABOVE OPERATION ONLY DVA.GO!IE.RDY
3673 017720 001405 BEQ 88$ ;BRANCH IF GOOD
3674 017722 010037 001126 MOV R0,@#SBDDAT ;BAD DATA
3675 017726 010137 004600 MOV R1,@#REGADR ;FAILING REGISTER RHCS1
3676 017732 104021 ERROR 21 ;DURING ABOVE OPERATION ONLY
3677 017734 012746 030500 88$: MOV #MOL!DPR!VV.PIP,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
3678 017740 011637 001124 MOV (SP),@#SGDDAT ;SAVE FOR PRINTOUT
3679 017744 022605 CMP (SP)+,R5 ;DURING ABOVE OPERATION ONLY MOL!DPR!VV.PIP
3680 017746 001405 BEQ 90$ ;BRANCH IF GOOD
3681 017750 010537 001126 MOV R5,@#SBDDAT ;BAD DATA
3682 017754 010337 004600 MOV R3,@#REGADR ;FAILING REGISTER RHDS1
3683 017760 104063 ERROR 63 ;DURING ABOVE OPERATION ONLY
3684
3685 ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
3686
3687 017762 004037 041360 JSR R0,@#FILLRE ;MOV 0 INTO SAVED RHCC
3688 017766 002334 RHCC ;SAVED REGISTER TO CHANGE
3689 017770 000000 0 ;DATA
3690 017772 004037 041360 JSR R0,@#FILLRE ;MOV 116377 INTO SAVED RHOF
3691 017776 002310 RHOF ;SAVED REGISTER TO CHANGE
3692 020000 116377 116377 ;DATA
3693 020002 053737 004740 004636 BIS @#ATTENT,@#SAVERE+24 ;SET APPROPRIATE 'ATA' BITS
3694 020010 004037 041360 JSR R0,@#FILLRE ;MOV 104206 INTO SAVED RHCS1
3695 020014 002300 RHCS1 ;SAVED REGISTER TO CHANGE
3696 020016 104206 104206 ;DATA
3697 020020 004037 041360 JSR R0,@#FILLRE ;MOV 110700 INTO SAVED RHDS1
3698 020024 002322 RHDS1 ;SAVED REGISTER TO CHANGE
3699 020026 110700 110700 ;DATA
3700
3701
3702 ;*NOW COMPARE REGISTERS AFTER A RECALIBRATE COMMAND
3703
3704 020030 004037 042444 JSR R0,@#COMREG ;COMPARE SAVED REGISTERS WITH
3705 020034 004612 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
3706 020036 002354 WC ;TEST DATA STARTING FROM 'RHWC'
3707 020040 000022 18. ;18. REGISTERS TO BE COMPARED
3708 020042 020046 1$ ;RETURN TO 1$ ON ERROR
3709 020044 020052 2$ ;RETURN TO 2$ ON NO ERROR
3710
3711 020046 104064 1$: ERROR 64 ;RECALIBRATE COMMAND CAUSED
3712 020050 000207 RTS PC ;AN ERROR
3713 ;GOOD DATA GIVES WHAT SHOULD
3714 ;BE THERE

```


CZRJICO, RPO4/5/6 FCTNL CTLR1
CZRJIC.P11 27-JUL-78 13:05

MACY11 30A(1052) 27-JUL-78 13:06 PAGE 87
T24 RECALIBRATE COMMAND

SEQ 0086

3715
3716
3717
3718 020052

28:

;RECEIVED DATA GIVES WHAT WAS
;THERE AFTER COMMAND

```

3719
3720 020052 005737 004744          TST    @#RPO6          ;TEST FOR RPO6 DRIVE
3721 020056 001401                   BEQ    3$              ;IF = 0, TREAT DRIVE AS AN RPO4
3722 020060 000402                   BR     4$              ;TREAT AS RPO6 - DO NEXT 'MAKECL'
3723 020062 000137 020126          3$:   JMP    @#CAT          ;DO SECOND FOLLOWING 'MAKECL'
3724 020066                   4$:
3725
3726
3727 020066 000004          TST25: SCOPE
3728 020070 012706 001000          MOV    #STACK,SP      ;RESET STACK
3729 020074 012737 000025 004604          MOV    #26-1,@#TSTNM ;THIS SAVES TEST NUMBER
3730 020102 004737 041456          JSR    PC,@#CLDISK    ;INIT DRIVE
3731 020106 012777 000001 162204          MOV    #DMD,@#RHMR    ;SET DIAGNOSTIC MODE
3732 020114 004037 041126          JSR    R0,@#MAKECYL   ;SUBROUTINE TO GIVE A SEEK
3733 020120 000777 777
3734 020122 000137 020162          JMP    @#BIRD          ;DON'T DO NEXT 'MAKECL'
3735
3736
3737 020126          CAT:
3738 020126 000004          TST26: SCOPE
3739 020130 012706 001000          MOV    #STACK,SP      ;RESET STACK
3740 020134 012737 000026 004604          MOV    #27-1,@#TSTNM ;THIS SAVES TEST NUMBER
3741 020142 004737 041456          JSR    PC,@#CLDISK    ;INIT DRIVE
3742 020146 012777 000001 162144          MOV    #DMD,@#RHMR    ;SET DIAGNOSTIC MODE
3743 020154 004037 041126          JSR    R0,@#MAKECYL   ;SUBROUTINE TO GIVE A SEEK
3744 020160 000377 377
  
```

```
3745
3746
3747
3748 020162          BIRD:
3749 020162 000004    TST27: SCOPE
3750 020164 012706 001000      MOV #STACK,SP ;RESET STACK
3751 020170 012737 000027 004604  MOV #27,@TSTNM ;SAVE TEST NUMBER
3752 020176 004737 041456      JSR PC,@CLDISK ;SET R1-RHCS1, R2-RHCS2
3753 020202 004737 041536      JSR PC,@CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV - 1
3754 020206 104401 066720      TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF ANY OF
3755 020212 000000      HALT ;STOP
3756 020214 012700 002272      MOV #RHWC,R0 ;ADDR. OF ADDR OF RHWC IN R0
3757
3758 020220 012730 000000      MOV #0,@(R0)+ ;LOAD 0 INTO RHWC
3759 020224 012730 000000      MOV #0,@(R0)+ ;LOAD 0 INTO RHBA
3760 020230 052730 000000      BIS #0,@(R0)+ ;LOAD 0 INTO RHCS2
3761 020234 012730 000000      MOV #0,@(R0)+ ;LOAD 0 INTO RHCS1
3762 020240 012730 000000      MOV #0,@(R0)+ ;LOAD 0 INTO RHER1
3763 020244 012730 000000      MOV #0,@(R0)+ ;LOAD 0 INTO RHD5T
3764 020250 012730 000000      MOV #0,@(R0)+ ;LOAD 0 INTO RHER2
3765 020254 012730 000000      MOV #0,@(R0)+ ;LOAD 0 INTO RHOF
3766 020260 012730 000000      MOV #0,@(R0)+ ;LOAD 0 INTO RHCA
3767 020264 012730 000000      MOV #0,@(R0)+ ;LOAD 0 INTO RHER3
3768 020270 012730 177777      MOV #-1,@(R0)+ ;CLEAR ALL BITS OF RHAS
3769 020274 012730 000000      MOV #0,@(R0)+ ;LOAD 0 INTO RHMR
3770 020300 013777 002426 161772  MOV @#RECALI,@RHCS1 ;GET READY FOR RECALI
3771
3772 ;*NOW SAVE REGISTERS FOR COMPARISON AFTER RECALIBRATE
3773
3774 020306 004037 041624      JSR R0,@#SAVER ;SAVE REGISTERS
3775 020312 002272          RHWC ;RHWC IS THE FIRST REGISTER SAVED
3776 020314 004612          SAVERE ;STARTING ADDRESS OF WHERE
3777 020316 000022          18. ;NUMBER OF REGISTERS
3778 020320 013777 004606 161740  MOV @#RP4VEC,@RPVEC ;SET RPO4 VECTOR ADDRESS
3779 020326 013746 002426      MOV @#RECALI,-(SP) ;GET READY TO MOVE COMMAND
3780 020332 052716 000101      BIS #GO!IE,(SP) ;GET READY TO SET 'GO' AND
3781 020336 012677 161736      MOV (SP)+,@RHCS1 ;GO WITH
3782 020342 011100          MOV @R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
3783 020344 011305          MOV @R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
3784 020346 104413          WAT ;WAIT FOR DRY BIT TO SET
3785 020350 002322          RHDS1 ;WAIT FOR RHDS1 REGISTER
3786 020352 000200          DRY ;WAIT FOR DRY BIT IN RHDS1 REGISTER
3787 020354 076377          31999. ;ALLOW 319990 MICRO SECONDS
3788 020356 056701          24001. ;DRY MUST SET BETWEEN
3789 020360 013746 002426      MOV @#RECALI,-(SP) ;SAVE COMMAND
3790 020364 052716 004301      BIS #DVA!GO!IE.RDY,(SP) ;INCLUDE DVA!GO!IE!RDY
3791 020370 011637 001124      MOV (SP),@#SGDDAT ;SAVE FOR PRINTOUT
3792 020374 022600          CMP (SP)+,R0 ;DURING ABOVE OPERATION ONLY DVA.GO!IE!RDY
3793 020376 001405          BEQ 88$ ;BRANCH IF GOOD
3794 020400 010037 001126      MOV R0,@#SBDDAT ;BAD DATA
3795 020404 010137 004600      MOV R1,@#REGADR ;FAILING REGISTER RHCS1
3796 020410 104021          ERROR 21 ;DURING ABOVE OPERATION ONLY
3797 020412 012746 030500      MOV #MOL!DPR!VV!PIP,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
3798 020416 011637 001124      MOV (SP),@#SGDDAT ;SAVE FOR PRINTOUT
3799 020422 022605          CMP (SP)+,R5 ;DURING ABOVE OPERATION ONLY MOL!DPR!VV!PIP
3800 020424 001405          BEQ 90$ ;BRANCH IF GOOD
```

L 7

3801	020426	010537	001126	MOV	R5,@#BDDAT		:BAD DATA
3802	020432	010337	004600	MOV	R3,@#REGADR		:FAILING REGISTER RHDS1
3803	020436	104063		ERROR	63		:DURING ABOVE OPERATION ONLY
3804				:*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES			
3805				JSR	RO,@#FILLRE		:MOV 0 INTO SAVED RHCC
3806	020440	004037	041360	RHCC			:SAVED REGISTER TO CHANGE
3807	020444	002334		0			:DATA
3808	020446	000000		BIS	@#ATTENT,@#SAVERE	+24	:SET APPROPRIATE 'ATA' BITS
3809	020450	053737	004740	004636	JSR	RO,@#CHREG	:CHANGE BITS IN SAVED REGISTER
3810	020456	004037	042336	RHDS1			:CHANGE RHDS1 REGISTER
3811	020462	002322		1			:1 BIT/BITS TO BE CHANGED
3812	020464	000001		1			:NEW VALUE OF ATA IS 1
3813	020466	000001		ATA			:CHANGE ATA BIT
3814	020470	100000		JSR	RO,@#CHREG		:CHANGE BITS IN SAVED REGISTER
3815	020472	004037	042336	RHCS1			:CHANGE RHCS1 REGISTER
3816	020476	002300		1			:1 BIT/BITS TO BE CHANGED
3817	020500	000001		1			:NEW VALUE OF SC IS 1
3818	020502	000001		SC			:CHANGE SC BIT
3819	020504	100000		:*NOW COMPARE REGISTERS AFTER A RECALIBRATE COMMAND			
3820							
3821				JSR	RO,@#COMREG		:COMPARE SAVED REGISTERS WITH
3822				SAVERE			:GOOD DATA SAVED IN 'SAVERE'
3823	020506	004037	042444	WC			:TEST DATA STARTING FROM 'RHWC'
3824	020512	004612		18.			:18. REGISTERS TO BE COMPARED
3825	020514	002354		1\$:RETURN TO 1\$ ON ERROR
3826	020516	000022		2\$:RETURN TO 2\$ ON NO ERROR
3827	020520	020524					
3828	020522	020530					
3829							
3830	020524	104064		1\$:	ERROR	64	:RECALIBRATE COMMAND CAUSED
3831	020526	000207		RTS	PC		:AN ERROR
3832							:GOOD DATA GIVES WHAT SHOULD BE
3833							:THERE
3834							:RECEIVED DATA GIVES WHAT WAS
3835	020530			2\$:			:THERE AFTER A RECALIBRATE
3836							


```
3944
3945
3946
3947
3948 021260 013746 002424
3949 021264 052716 004201
3950 021270 005737 004722
3951 021274 001413
3952 021276 010037 004760
3953 021302 042737 177677 004760
3954 021310 042716 000100
3955 021314 053716 004760
3956 021320 052716 100000
3957
3958 021324 011637 001124 9$:
3959 021330 022600
3960
3961 021332 001405
3962 021334 010037 001126
3963 021340 010137 004600
3964 021344 104021
3965
3966
3967
3968 021346 012746 020400 10$:
3969 021352 010537 004760
3970 021356 042737 167677 004760
3971 021364 042716 010100
3972 021370 053716 004760
3973
3974 021374 011637 001124
3975 021400 022605
3976
3977 021402 001405
3978 021404 010537 001126
3979 021410 010337 004600
3980 021414 104063
3981
3982
3983 021416 11$:
3984
3985 021416 104401 021424
3986 021422 000425
3987
3988
```

;*COMPARE CONTENTS OF RHCS1 AND RHDS1, WHICH WERE SAVED
;*DURING THE UNLOAD COMMAND, WITH THE EXPECTED RESULTS

MOV @UNLOAD,-(SP) ;PUSH COMMAND ON STACK
BIS #DVA!GO!RDY,(SP) ;INCLUDE THESE BITS SET
TST @NUNIT ;IS THERE MORE THAN ONE UNIT ?
BEQ 9\$;SKIP NEXT IF ONLY ONE UNIT
MOV R0,@TMP4 ;PUT SAVED RHCS1 INTO TMP4
BIC #^CIE,@TMP4 ;MASK ALL BUT THE 'IE' BIT IN RHCS1
BIC #IE,(SP) ;CLEAR 'IE' IN EXPECTED DATA
BIS @TMP4,(SP) ;SET 'IE' STATE FROM ACTUAL RHCS1 DATA
BIS #SC,(SP) ;SET 'SC' IN RHCS1 SAVED DATA

9\$: MOV (SP),@SGDDAT ;SAVE EXPECTED DATA FOR PRINTOUT
CMP (SP)+,R0 ;COMPARE EXPECTED DATA WITH SAVED
;RHCS1 DATA AND RESET THE STACK
BEQ 10\$;CHECK NEXT BITS IF THESE OK
MOV R0,@SBDDAT ;RHCS1 IS BAD - PRINT IT OUT
MOV R1,@REGADR ;REGISTER ADDRESS
ERROR 21 ;DURING ABOVE OPERATION ONLY THE
;'DVA', 'GO', 'RDY' AND COMMAND BITS
;SHOULD BE SET

10\$: MOV #PIP!DPR,-(SP) ;PUT SOME EXPECTED RHDS1 BITS ON STACK
MOV R5,@TMP4 ;PUT SAVED RHDS1 INTO TMP4
BIC #^C<MOL!VV>,@TMP4 ;MASK ALL BUT 'MOL' & 'VV' IN RHDS1
BIC #MOL!VV,(SP) ;CLEAR 'MOL' & 'VV' IN EXPECTED RHDS1
BIS @TMP4,(SP) ;SET EXPECTED 'MOL' & 'VV' BIT STATES
;FROM THE ACTUAL DATA (DON'T CARE)
MOV (SP),@SGDDAT ;SAVE EXPECTED DATA FOR PRINTOUT
CMP (SP)+,R5 ;COMPARE EXPECTED DATA WITH SAVED
;RHDS1 DATA AND RESET THE STACK
BEQ 11\$;CONTINUE IF EXPECTED=SAVED
MOV R5,@SBDDAT ;RHDS1 IS BAD - PRINT IT OUT
MOV R3,@REGADR ;REGISTER ADDRESS
ERROR 63 ;DURING THE ABOVE OPERATION, ONLY 'PIP
;AND 'DPR' SHOULD BE SET
;'MOL' & 'VV' ARE DON'T CARES

11\$: TYPE ,96\$;;TYPE ASCIZ STRING
BR ,95\$;;GET OVER THE ASCIZ
;THIS PROVIDES A 1 SECOND 'STALL'

```

3989
3990
3991      ;*NOW CHANGE REGISTERS SAVED BEFORE UNLOAD COMMAND
3992      ;*TO EXPECTED VALUES AFTER UNLOAD COMMAND
3993
3994      ;* - AGAIN 'MOL' $ 'VV' ARE DON'T CARES
3995
3996
3997 021476 012746 020400      MOV      #PIP!DPR,-(SP)      ;SET EXPECTED FINAL RHDS1 BITS
3998 021502 017737 160614      MOV      @RHDS1,@TMP4      ;GET PRESENT ACTUAL RHDS1 CONTENTS
3999 021510 042737 167677      BIC      #^C<MOL!VV>,@TMP4 ;MASK OUT ALL BUT 'MOL' & 'VV'
4000 021516 042716 010100      BIC      #MOL!VV,(SP)      ;CLEAR 'MOL' & 'VV' IN EXPECTED RHDS1
4001 021522 053716 004760      BIS      @TMP4,(SP)      ;SET EXPECTED 'MOL' & 'VV' STATES
4002      ;FROM THE ACTUAL (DON'T CARE COND.)
4003 021526 042716 100200      BIC      #ATA!DRY,(SP)      ;CLEAR THESE ADDITIONAL RHDS1 BITS
4004 021532 012637 004642      MOV      (SP)+,@SAVERE+30 ;CHANGE THE SAVED RHDS1 REGISTER
4005      ;AND ADJUST THE STACK
4006
4007 021536 004037 042336      JSR      RO,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
4008 021542 002300      RHCS1      ;CHANGE RHCS1 REGISTER
4009 021544 000001      1          ;1 BIT/BITS TO BE CHANGED
4010 021546 000001      1          ;NEW VALUE OF GO IS 1
4011 021550 000001      GO        ;CHANGE GO BIT
4012
4013 021552 005737 004722      TST      @#NUNIT      ;IS THERE MORE THAN ONE UNIT ?
4014 021556 001006      BNE      7$          ;SKIP NEXT IF MORE THAN ONE UNIT
4015 021560 004037 042336      JSR      RO,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
4016 021564 002300      RHCS1      ;CHANGE RHCS1 REGISTER
4017 021566 000001      1          ;1 BIT/BITS TO BE CHANGED
4018 021570 000000      0          ;NEW VALUE OF SC IS 0
4019 021572 100000      SC        ;CHANGE SC BIT
4020 021574
4021 021574 043737 004740 004636 7$: BIC      @#ATTENT,@#SAVERE+24 ;CLEAR APPROPRIATE ATA BITS
4022
4023      ;*NOW COMPARE REGISTERS AFTER THE UNLOAD COMMAND
4024      ;*WITH EXPECTED VALUES
4025
4026 021602 004037 042444      JSR      RO,@#COMREG      ;COMPARE SAVED REGISTERS WITH
4027 021606 004612      SAVERE      ;GOOD DATA SAVED IN 'SAVERE'
4028 021610 002354      WC        ;TEST DATA STARTING FROM 'RHWC'
4029 021612 000021      17.       ;17. REGISTERS TO BE COMPARED
4030 021614 021620      3$        ;RETURN TO 3$ ON ERROR
4031 021616 021624      4$        ;RETURN TO 4$ ON NO ERROR
4032
4033 021620 104023      3$:      ERROR 23      ;UNLOAD COMMAND GAVE
4034 021622 000207      RTS      PC        ;AN ERROR
4035      ;GOOD DATA GIVES WHAT SHOULD
4036      ;BE THERE
4037      ;RECEIVED DATA GIVES WHAT WAS
4038      ;THERE AFTER UNLOAD COMMAND
4039
4040 021624      4$:
4041 021624 104401 021632      TYPE      ,98$      ;;TYPE ASCIZ STRING
4042 021630 000406      BR        ,97$      ;;GET OVER THE ASCIZ
4043 021646 013746 004716      MOV      @#UNIT,-(SP) ;GET UNIT UNDER TEST
4044 021652 104405      TYPDS
    
```



```

4045 021654 104401 021662      TYPE      ,100$      ;;TYPE ASCIZ STRING
4046 021660 000444      BR        99$      ;;GET OVER THE ASCIZ
4047
4048 021772 005037 047274      CLR      @#PRITEM  ;CLEAR PREVIOUS ERROR NUMBER
4049 021776 000000      HALT      ;WAIT FOR CONTINUE
4050 022000 004737 041514      JSR      PC,@#CHECK ;CHECKS THAT DVA,RDY,MOL & DPR = 1
4051 022004 000240      NOP      ;CHECKS THAT ALL OTHER BITS = 0
4052 022006 000240      NOP      ;THERE IS NO HALT IF IT FAILS - IT
4053 022010 000240      NOP      ;IS USED IN THE MIDDLE OF A TEST
4054
4055
4056
4057      ;*SET VV IN RHDS1 AFTER RESET FROM ACTUATING
4058      ;*THE STANDBY SWITCH AND CYCLING UP (MOL = 1)
4059 022012 013746 002460      MOV      @#PKACK,-(SP) ;GET READY TO MOVE COMMAND
4060 022016 052716 000001      BIS      #GO,(SP)     ;GET READY TO SET GO
4061 022022 012677 160252      MOV      (SP)+,@RHCS1 ;GO WITH
4062 022026 011100      MOV      @R1,R0      ;SAVE R-CS1 DURING ABOVE OPERATION
4063 022030 011305      MOV      @R3,R5      ;SAVE R.DS1 DURING ABOVE OPERATION
4064 022032 104413      WAT      ;WAIT FOR VV BIT TO SET
4065 022034 002322      RHDS1     ;WAIT FOR RHDS1 REGISTER
4066 022036 000100      VV        ;WAIT FOR VV BIT IN RHDS1 REGISTER
4067 022040 000001      1.        ;ALLOW 10 MICRO SECONDS
4068 022042 000001      1.        ;VV MUST SET BETWEEN
4069
  
```

```

4070 022044 000004          TST31: SCOPE
4071 022046 012737 022120 001106  MOV    #1$, $LPADR      ;;SET SCOPE LOOP ADDRESS
4072 022054 012706 001000          MOV    #STACK, SP      ;;RESET STACK
4073 022060 012737 000031 004604  MOV    #31, @#TSTNM    ;;SAVE TEST NUMBER
4074 022066 004737 041456          JSR    PC, @#CLDISK    ;;SET R1-RHCS1, R2-RHCS2
4075 022072 004737 041514          JSR    PC, @#CHECK     ;;CHECK THAT DVA, RDY, MOL, DPR, DRY = 1
4076 022076 104401 066720          TYPE   ,CPHALT        ;;CANNOT CONTINUE TESTS IF THEY AREN'T
4077 022102 000000          HALT                               ;STOP
4078 022104 112737 000001 004610  MOV    #1, @#OFSTVL    ;;SET OFFSET VALUE TO 1
4079 022112 112737 000034 004611  MOV    #34, @#GFSTVL+1 ;;SET HCI, ECI, FMT22
4080
4081
4082 022120          1$:
4083 022120 004737 041456          JSR    PC, @#CLDISK    ;;SET R1-RHCS1, R2-RHCS2
4084 022124 004737 041536          JSR    PC, @#CHECKT    ;;CHECK DVA, RDY, MOL, DPR & VV = 1
4085 022130 000240          NOP                               ;;CHECK THAT ALL OTHER BITS = 0
4086 022132 000240          NOP                               ;;UNLIKE THE OTHER STATUS BIT TESTS,
4087 022134 000240          NOP                               ;;THERE IS NO HALT IF IT FAILS - IT IS
4088 022136 012700 002272          MOV    #RHC, R0        ;;ADDR. OF ADDR OF RHC IN R0
4089 022142 012730 177777          MOV    #177777, @ (R0)+ ;;LOAD 177777 INTO RHC
4090 022146 012730 177777          MOV    #177777, @ (R0)+ ;;LOAD 177777 INTO RHBA
4091 022152 052730 000010          BIS    #10, @ (R0)+    ;;LOAD 10 INTO RHCS2
4092 022156 012730 001400          MOV    #1400, @ (R0)+  ;;LOAD 1400 INTO RHCS1
4093 022162 012730 000000          MOV    #0, @ (R0)+    ;;LOAD 0 INTO RHER1
4094 022166 012730 177777          MOV    #177777, @ (R0)+ ;;LOAD 177777 INTO RHDST
4095 022172 012730 000000          MOV    #0, @ (R0)+    ;;LOAD 0 INTO RHER2
4096
4097
4098 022176 013730 004610          MOV    @#OFSTVL, @ (R0)+ ;;SET OFFSET REGISTER
4099
4100 022202 012730 177777          MOV    #177777, @ (R0)+ ;;LOAD 177777 INTO RHCA
4101 022206 012730 000000          MOV    #0, @ (R0)+    ;;LOAD 0 INTO RHER3
4102 022212 010046          MOV    R0, -(SP)      ;;PUSH R0 ON STACK
4103 022214 011446          MOV    @R4, -(SP)    ;;SAVE RHER1 TO REINSTATE LATER
4104 022216 011246          MOV    @R2, -(SP)    ;;SAVE RHCS2 TO BE EINSTATED
4105 022220 013700 004742          MOV    @#T: TALAT, R0 ;;GET DRIVES PRESENT
4106 022224 005012          CLR    @R2           ;;CLEAR RHCS2 AND CARRY
4107 022226 012705 000010          MOV    #8, R5        ;;COUNTER
4108 022232 006000          82$: ROR    R0        ;;GET BIT INTO CARRY
4109 022234 103002          BCC    83$          ;;BRANCH IF NO UNIT ON THIS BIT
4110 022236 012714 177777          MOV    #-1, @R4     ;;MOVE INTO ERROR REGISTER
4111 022242 005212          83$: INC    @R2     ;;INCREMENT RHCS2 TO NEXT UNIT
4112 022244 005305          DEC    R5           ;;COUNT
4113 022246 001371          BNE    82$         ;;BRANCH IF 8 NOT DONE
4114 022250 012612          MOV    (SP)+, @R2   ;;REINSTATE RHCS2
4115 022252 012614          MOV    (SP)+, @R4   ;;REINSTATE RHER1
4116 022254 012600          MOV    (SP)+, R0    ;;POP STACK INTO R0
4117 022256 005720          TST    (R0)+       ;;GET OVER PHAS IN R0
4118 022260 012730 177776          MOV    #177776, @ (R0)+ ;;LOAD 177776 INTO RHMR
4119 022264 013777 002454 160006  MOV    @#OFSETC, @RHCS1 ;;GET READY FOR OFSETC
4120
4121
4122
4123 022272 004037 041624          JSR    R0, @#SAVER    ;;SAVE REGISTERS
4124 022276 002272          RHC    ;;RHC IS THE FIRST REGISTER SAVED
4125 022300 004612          SAVER  ;;STARTING ADDRESS OF WHERE
  
```

```

4126 022302 000022      18.          ;NUMBER OF REGISTERS
4127
4128 022304 013777 004606 157754      MOV      @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
4129
4130 022312 013746 002454      MOV      @#OFSETC,-(SP) ;GET READY TO MOVE COMMAND
4131 022316 052716 000101      BIS      #GO!IE,(SP) ;GET READY TO SET 'GO' AND
4132 022322 012677 157752      MOV      (SP)+,@RHCS1 ;GO WITH
4133 022326 011100      MOV      @R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
4134 022330 011305      MOV      @R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
4135
4136 022332 104413      WAT          ;WAIT FOR DRY BIT TO SET
4137 022334 002322      RHDS1      ;WAIT FOR RHDS1 REGISTER
4138 022336 000200      DRY          ;WAIT FOR DRY BIT IN RHDS1 REGISTER
4139 022340 001750      1000.      ;ALLOW 1000 MICRO SECONDS
4140 022342 000454      300.      ;DRY MUST SET BETWEEN
4141 022344 013746 002454      MOV      @#OFSETC,-(SP) ;SAVE COMMAND
4142 022350 052716 004301      BIS      #DVA!GO!IE!RDY,(SP) ;INCLUDE DVA!GO!IE!RDY
4143 022354 005737 004722      TST      @#NUNIT ;ARE THERE MORE THAN ONE UNIT
4144 022360 001413      BEQ      87$ ;BRANCH IF ONLY ONE UNIT
4145 022362 010037 004760      MOV      R0,@#TMP4 ;GET RHCS1
4146 022366 042737 177677 004760      BIC      #^CIE,@#TMP4 ;KEEP IE BIT
4147 022374 042716 000100      BIC      #IE,(SP) ;CLEAR IE IN GOOD DATA
4148 022400 053716 004760      BIS      @#TMP4,(SP) ;GET IE AS IS
4149 022404 052716 100000      BIS      #SC,(SP) ;SET SC IN RHCS1
4150 022410 011637 001124      MOV      (SP),@#SGDDAT ;SAVE FOR PRINTOUT
4151 022414 022600      CMP      (SP)+,R0 ;DURING ABOVE OPERATION ONLY DVA.GO!IE.RDY
4152 022416 001405      BEQ      86$ ;BRANCH IF GOOD
4153 022420 010037 001126      MOV      R0,@#SBDDAT ;BAD DATA
4154 022424 010137 004600      MOV      R1,@#REGADR ;FAILING REGISTER RHCS1
4155 022430 104021      ERROR     21 ;DURING ABOVE OPERATION ONLY
4156 022432 012746 030500 86$:      MOV      #PIP!MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
4157 022436 011637 001124      MOV      (SP),@#SCDDAT ;SAVE FOR PRINTOUT
4158 022442 022605      CMP      (SP)+,R5 ;DURING ABOVE OPERATION ONLY PIP!MOL!DPR.VV
4159 022444 001405      BEQ      88$ ;BRANCH IF GOOD
4160 022446 010537 001126      MOV      R5,@#SBDDAT ;BAD DATA
4161 022452 010337 004600      MOV      R3,@#REGADR ;FAILING REGISTER RHDS1
4162 022456 104063      ERROR     63 ;DURING ABOVE OPERATION ONLY
4163
4164          ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
4165
4166 022460 004037 042336      JSR      R0,@#CHREG ;CHANGE BITS IN SAVED REGISTER
4167 022464 002300      RHCS1     ;CHANGE RHCS1 REGISTER
4168 022466 000001      1 ;1 BIT/BITS TO BE CHANGED
4169 022470 000001      1 ;NEW VALUE OF SC IS 1
4170 022472 100000      SC ;CHANGE SC BIT
4171 022474 004037 042336      JSR      R0,@#CHREG ;CHANGE BITS IN SAVED REGISTER
4172 022500 002322      RHDS1     ;CHANGE RHDS1 REGISTER
4173 022502 000001      1 ;1 BIT/BITS TO BE CHANGED
4174 022504 000001      1 ;NEW VALUE OF ATA IS 1
4175 022506 100000      ATA ;CHANGE ATA BIT
4176 022510 053737 004740 004636      BIS      @#ATTENT,@#SAVERE+24 ;SET APPROPRIATE 'ATA' BITS
4177
4178          ;*NOW COMPARE REGISTERS AFTER AN OFSET COMMAND
4179
4180 022516 004037 042444      JSR      R0,@#COMREG ;COMPARE SAVED REGISTERS WITH
4181 022522 004612      SAVERE    ;GOOD DATA SAVED IN 'SAVERE'

```

```

4182 022524 002354          WC          ;TEST DATA STARTING FROM 'RHWC'
4183 022526 000022          18.         ;18. REGISTERS TO BE COMPARED
4184 022530 022534          2$          ;RETURN TO 2$ ON ERROR
4185 022532 022540          3$          ;RETURN TO 3$ ON NO ERROR
4186
4187 022534 104024          2$:        ERROR 24          ;OFFSET COMMAND CAUSED AN ERROR
4188 022536 000207          RTS        PC          ;GOOD DATA IS WHAT SHOULD BE THERE
4189                                ;RECEIVED DATA GIVES WHAT WAS THERE
4190                                ;AFTER AN OFFSET COMMAND
4191
4192 022540 013777 004740 157550 3$:        MOV        @#ATTENT,@RHAS ;C EAR WORKING DRIVE ATTENTION
4193
4194
4195
4196
4197                                ;*NOW A RETURN TO CENTER LINE COMMAND WILL BE GIVEN
4198
4199 022546 004737 041536          JSR        PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR & VV = 1
4200 022552 000240          NOP          ;CHECK THAT ALL OTHER BITS = 0
4201 022554 000240          NOP          ;UNLIKE THE OTHER STATUS BIT TESTS,
4202 022556 000240          NOP          ;THERE IS NO HALT IF IT FAILS - IT IS
4203 022560 013777 002456 157512        MOV        @#RETCL,@RHCS1 ;GET READY FOR RETCL
4204
4205                                ;*NOW REGISTERS ARE SAVED FOR COMPARISON AFTER COMMAND
4206
4207 022566 004037 041624          JSR        R0,@#SAVER  ;SAVE REGISTERS
4208 022572 002272          RHWC         ;RHWC IS THE FIRST REGISTER SAVED
4209 022574 004612          SAVERE        ;STARTING ADDRESS OF WHERE
4210 022576 000022          18.         ;NUMBER OF REGISTERS
4211
4212 022600 013777 004606 157460        MOV        @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
4213
4214 022606 013746 002456          MOV        @#RETCL,-(SP) ;GET READY TO MOVE COMMAND
4215 022612 052716 000101          BIS        #GO!IE,(SP)  ;GET READY TO SET 'GO' AND
4216 022616 012677 157456          MOV        (SP)+,@RHCS1 ;GO WITH
4217 022622 011100          MOV        @R1,R0      ;SAVE RHCS1 DURING ABOVE OPERATION
4218 022624 011305          MOV        @R3,R5      ;SAVE RHDS1 DURING ABOVE OPERATION
4219
4220 022626 104413          WAIT         ;WAIT FOR DRY BIT TO SET
4221 022630 002322          RHDS1        ;WAIT FOR RHDS1 REGISTER
4222 022632 000200          DRY          ;WAIT FOR DRY BIT IN RHDS1 REGISTER
4223 022634 001750          1000.        ;ALLOW 10000 MICRO SECONDS
4224 022636 001750          1000.        ;DRY MUST SET BETWEEN
4225 022640 013746 002456          MOV        @#RETCL,-(SP) ;SAVE COMMAND
4226 022644 052716 004301          BIS        #DVA!GO!IE.RDY,(SP) ;INCLUDE DVA.GO!IE!RDY
4227 022650 005737 004722          TST        @#NUNIT     ;ARE THERE MORE THAN ONE UNIT
4228 022654 001413          BEQ        90$         ;BRANCH IF ONLY ONE UNIT
4229 022656 010037 004760          MOV        R0,@#TMP4   ;GET RHCS1
4230 022662 042737 177677 004760        BIC        #^CIE,@#TMP4 ;KEEP IE BIT
4231 022670 042716 000100          BIC        #IE,(SP)    ;CLEAR IE IN GOOD DATA
4232 022674 053716 004760          BIS        @#TMP4,(SP) ;GET IE AS IS
4233 022700 052716 100000          BIS        #SC,(SP)   ;SET SC IN RHCS1
4234 022704 011637 001124          MOV        (SP),@#SGDDAT ;SAVE FOR PRINTOUT
4235 022710 022600          CMP        (SP)+,R0    ;DURING ABOVE OPERATION ONLY DVA.GO!IE.RDY
4236 022712 001405          BEQ        89$         ;BRANCH IF GOOD
4237 022714 010037 001126          MOV        R0,@#SBDDAT ;BAD DATA
  
```



```

4495 023724 104401 066720 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF ANY OF
4496 023730 000000 HALT ;STOP
4497
4498 023732 013777 004606 156326 MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
4499
4500 023740 013746 002450 MOV @#REFOR,-(SP) ;GET READY TO MOVE COMMAND
4501 023744 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET 'GO' AND
4502 023750 012677 156324 MOV (SP)+,@RHCS1 ;GO WITH
4503 023754 011100 MOV @R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
4504 023756 011305 MOV @R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
4505
4506 023760 104413 WAT ;WAIT FOR RDY BIT TO SET
4507 023762 002300 RHCS1 ;WAIT FOR RHCS1 REGISTER
4508 023764 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
4509 023766 001614 908. ;ALLOW 9080 MICRO SECONDS
4510 023770 001507 839. ;RDY MUST SET BETWEEN
4511 023772 013746 002450 MOV @#REFOR,-(SP) ;SAVE COMMAND
4512 023776 052716 004101 BIS #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
4513 024002 011637 001124 MOV (SP),@#SGDDAT ;SAVE FOR PRINTOUT
4514 024006 022600 CMP (SP)+,R0 ;DURING ABOVE OPERATION ONLY IE.GO!DVA
4515 024010 001405 BEQ 67$ ;BRANCH IF GOOD
4516 024012 010037 001126 MOV R0,@#SBDDAT ;BAD DATA
4517 024016 010137 004600 MOV R1,@#REGADR ;FAILING REGISTER RHCS1
4518 024022 104021 ERROR 21 ;DURING ABOVE OPERATION ONLY
4519 024024 012746 010500 67$: MOV #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
4520 024030 011637 001124 MOV (SP),@#SGDDAT ;SAVE FOR PRINTOUT
4521 024034 022605 CMP (SP)+,R5 ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
4522 024036 001405 BEQ 69$ ;BRANCH IF GOOD
4523 024040 010537 001126 MOV R5,@#SBDDAT ;BAD DATA
4524 024044 010337 004600 MOV R3,@#REGADR ;FAILING REGISTER RHDS1
4525 024050 104063 ERROR 63 ;DURING ABOVE OPERATION ONLY
4526
4527 ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
4528
4529 024052 004037 041360 JSR R0,@#FILLRE ;MOV 0 INTO SAVED RHWC
4530 024056 002272 RHWC ;SAVED REGISTER TO CHANGE
4531 024060 000000 0 ;DATA
4532 024062 004037 041360 JSR R0,@#FILLRE ;MOV REINTO+<260.*2> INTO SAVED RHBA
4533 024066 002274 RHBA ;SAVED REGISTER TO CHANGE
4534 024070 004544 REINTO+<260.*2> ;DATA
4535 024072 004037 041360 JSR R0,@#FILLRE ;MOV 1 INTO SAVED RHDST
4536 024076 002304 RHDST ;SAVED REGISTER TO CHANGE
4537 024100 000001 1 ;DATA
4538
4539 ;*COMPARE REGISTER BEFORE READ HEADER AND DATA
4540 ;*WITH REGISTERS AFTER COMMAND
4541
4542 024102 004037 042444 JSR R0,@#COMREG ;COMPARE SAVED REGISTERS WITH
4543 024106 004612 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
4544 024110 002354 WC ;TEST DATA STARTING FROM 'RHWC'
4545 024112 000022 18. ;18. REGISTERS TO BE COMPARED
4546 024114 024120 5$ ;RETURN TO 5$ ON ERROR
4547 024116 024124 6$ ;RETURN TO 6$ ON NO ERROR
4548
4549 024120 104031 5$: ERROR 31 ;READ HEADER AND DATA CAUSED
4550 024122 000207 RTS PC ;IMPROPER REGISTER CHANGE

```

CZR
CZR
5
5
5
5


```

4573
4574 024146 000004
4575 024150 012706 001000
4576 024154 012737 000034 004604
4577 024162 004737 041456
4578
4579
4580
4581 024166 004037 041326
4582 024172 002470
4583 024174 000400
4584 024176 000000
4585
4586
4587
4588 024200 004037 043430
4589 024204 000000
4590 024206 000
4591 024207 000
4592 024210 177400
4593 024212 003534
4594 024214 000000
4595 024216 014000
4596 024220 002446
4597
4598
4599
4600 024222 004037 041624
4601 024226 002272
4602 024230 004612
4603 024232 000022
4604
4605 024234 004737 041536
4606 024240 104401 066720
4607 024244 000000
4608
4609 024246 013777 004606 156012
4610
4611 024254 013746 002446
4612 024260 052716 000101
4613 024264 012677 156010
4614 024270 011100
4615 024272 011305
4616
4617 024274 104413
4618 024276 002300
4619 024300 000200
4620 024302 001614
4621 024304 001507
4622 024306 013746 002446
4623 024312 052716 004101
4624 024316 011637 001124
4625 024322 022600
4626 024324 001405
4627 024326 010037 001126
4628 024332 010137 004600

TST34: SCOPE
MOV #STACK,SP ;RESET STACK
MOV #34,@TSTNM ;SAVE TEST NUMBER
JSR PC,@CLDISK ;SET R1-RHCS1, R2-RHCS2

;*FILL WRITE FROM BUFFER WITH EXPECTED DATA

JSR R0,@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 R0,@RUN ;SETUP TO RUN FOR DATA COMMAND
0 ;CYLINDER 0
0 ;SECTOR 0
0 ;TRACK 0
-256. ;WORD COUNT = 256.
REINTO ;BUS ADDRESS
0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
ECI!FMT22 ;16 BITS PER WORD FORMAT
READAT ;GET READY TO DO A READAT

;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ DATA COMMAND

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

JSR PC,@CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF ANY OF
HALT ;STOP

MOV @RP4VEC,@RPVEC ;SET RPO4 VECTOR ADDRESS

MOV @READAT,-(SP) ;GET READY TO MOVE COMMAND
BIS #GO!IE,(SP) ;GET READY TO SET 'GO' AND
MOV (SP)+,@RHCS1 ;GO WITH
MOV @R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
MOV @R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION

WAT ;WAIT FOR RDY BIT TO SET
RHCS1 ;WAIT FOR RHCS1 REGISTER
RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
908. ;ALLOW 9080 MICRO SECONDS
839. ;RDY MUST SET BETWEEN
MOV @READAT,-(SP) ;SAVE COMMAND
BIS #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
MOV (SP)+,@SGDDAT ;SAVE FOR PRINTOUT
CMP (SP)+,R0 ;DURING ABOVE OPERATION ONLY IE.GO!DVA
BEQ 64$ ;BRANCH IF GOOD
MOV R0,@SBDDAT ;BAD DATA
MOV R1,@REGADR ;FAILING REGISTER RHCS1

.BYTE
.BYTE

```

```

4629 024336 104021
4630 024340 012746 010500
4631 024344 011637 001124
4632 024350 022605
4633 024352 001405
4634 024354 010537 001126
4635 024360 010337 004600
4636 024364 104063
4637
4638
4639
4640 024366 004037 041360
4641 024372 002272
4642 024374 000000
4643 024376 004037 041360
4644 024402 002274
4645 024404 004534
4646 024406 004037 041360
4647 024412 002304
4648 024414 000001
4649
4650
4651
4652
4653 024416 004037 042444
4654 024422 004612
4655 024424 002354
4656 024426 000022
4657 024430 024434
4658 024432 024440
4659
4660 024434 104033 1$:
4661 024436 000207
4662
4663
4664
4665
4666
4667 024440 2$:
4668 024440 004037 043474
4669 024444 002470
4670 024446 003534
4671 024450 000400
4672 024452 024456
4673 024454 024462
4674
4675 024456 104034 3$:
4676 024460 000207
4677
4678 024462 4$:

```

ERROR 21 ; DURING ABOVE OPERATION ONLY
 MOV #MOL!DPR!VV,-(SP) ; SAVE BITS SET DURING OPERATION IN RHDS1
 MOV (SP),@#SGDDAT ; SAVE FOR PRINTOUT
 CMP (SP)+,R5 ; DURING ABOVE OPERATION ONLY MOL.DPR!VV
 BEQ 66\$; BRANCH IF GOOD
 MOV R5,@#SBDDAT ; BAD DATA
 MOV R3,@#REGADR ; FAILING REGISTER RHDS1
 ERROR 63 ; DURING ABOVE OPERATION ONLY
 ; *NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
 JSR RO,@#FILLRE ; MOV 0 INTO SAVED RHWC
 RHWC ; SAVED REGISTER TO CHANGE
 0 ; DATA
 JSR RO,@#FILLRE ; MOV REINTO+<256.*2> INTO SAVED RHBA
 RHBA ; SAVED REGISTER TO CHANGE
 REINTO+<256.*2> ; DATA
 JSR RO,@#FILLRE ; MOV 1 INTO SAVED RHDST
 RHDST ; SAVED REGISTER TO CHANGE
 1 ; DATA
 ; *NOW COMPARE REGISTERS BEFORE READ DATA WITH
 ; *AFTER COMMAND
 JSR RO,@#COMREG ; COMPARE SAVED REGISTERS WITH
 SAVERE ; GOOD DATA SAVED IN 'SAVERE'
 WC ; TEST DATA STARTING FROM 'RHWC'
 18. ; 18. REGISTERS TO BE COMPARED
 1\$; RETURN TO 1\$ ON ERROR
 2\$; RETURN TO 2\$ ON NO ERROR
 ERROR 33 ; READ DATA CAUSED IMPROPER REGISTER
 RTS PC ; CHANGE
 ; GOOD DATA GIVES WHAT SHOULD BE THERE
 ; RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND
 ; *NOW READ INTO BUFFER WILL BE CHECKED TO SEE THAT READ
 ; *WAS GOOD
 JSR RO,@#COMPAR ; COMPARE TWO BLOCKS OF MEMORY
 WRFROM ; GOOD DATA STARTS FROM WRFROM
 REINTO ; TEST DATA STARTS FROM REINTO
 256. ; 256. WORDS TO BE COMPARED
 3\$; RETURN TO 3\$ ON ERROR
 4\$; RETURN TO 4\$ ON NO ERROR
 ERROR 54 ; READ DATA COMMAND
 RTS PC ; READ INCORRECTLY

```

4679
4680 024462 000004          TST35: SCOPE
4681 024464 012706 001000  MOV      #STACK,SP      ;RESET STACK
4682 024470 012737 000035 004604  MOV      #35,@TSTNM     ;SAVE TEST NUMBER
4683 024476 004737 041456  JSR      PC,@CLDISK     ;SET R1-RHCS1, R2-RHCS2
4684
4685                      ;*NOW FILL WRITE FROM BUFFER -200 OF 1'S AND 56 OF 125252
4686
4687 024502 004037 041326  JSR      RO,@CLAREA     ;CLEAR 200. WORDS, FROM WRFROM
4688 024506 002470          WRFROM     ;STARTING FROM WRFROM
4689 024510 000310          200.      ;200. WORDS
4690 024512 177777          -1        ;FILL WITH -1
4691 024514 004037 041326  JSR      RO,@CLAREA     ;CLEAR 56. WORDS, FROM WRFROM+<200.*2>
4692 024520 003310          WRFROM+<200.*2> ;STARTING FROM WRFROM+<200.*2>
4693 024522 000070          56.      ;56. WORDS
4694 024524 125252          125252   ;FILL WITH 125252
4695
4696                      ;*NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA AS
4697                      ;*WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
4698                      ;*CAN BE MADE TO DETERMINE THAT WRITE DID NOT CHANGE BUFFER
4699
4700 024526 004037 041326  JSR      RO,@CLAREA     ;CLEAR 200. WORDS, FROM REINTO
4701 024532 003534          REINTO    ;STARTING FROM REINTO
4702 024534 000310          200.      ;200. WORDS
4703 024536 177777          -1        ;FILL WITH -1
4704 024540 004037 041326  JSR      RO,@CLAREA     ;CLEAR 56. WORDS, FROM REINTO+<200.*2>
4705 024544 004354          REINTO+<200.*2> ;STARTING FROM REINTO+<200.*2>
4706 024546 000070          56.      ;56. WORDS
4707 024550 125252          125252   ;FILL WITH 125252
4708
4709                      ;*NOW WRITE DATA COMMAND WILL BE LOADED
4710
4711 024552 004037 043430  JSR      RO,@#RUN       ;SETUP TO RUN FOR DATA COMMAND
4712 024556 000000          0         ;CYLINDER 0
4713 024560 000          .BYTE    0         ;SECTOR 0
4714 024561 000          .BYTE    0         ;TRACK 0
4715 024562 177470          -200.     ;WORD COUNT = 200.
4716 024564 002470          WRFROM    ;BUS ADDRESS
4717 024566 000000          0         ;DO NOT INHIBIT BUS ADDRESS INCREMENT
4718 024570 010000          FMT22     ;16 BITS PER WORD FORMAT
4719 024572 002442          WRIDAT    ;GET READY TO DO A WRIDAT
4720
4721                      ;*NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE DATA
4722
4723 024574 004037 041624  JSR      RO,@#SAVER     ;SAVE REGISTERS
4724 024600 002272          RHWC      ;RHWC IS THE FIRST REGISTER SAVED
4725 024602 004612          SAVER     ;STARTING ADDRESS OF WHERE
4726 024604 000022          18.      ;NUMBER OF REGISTERS
4727
4728 024606 004737 041536  JSR      PC,@#CHECKT    ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
4729 024612 104401 066720  TYPE     .CPHALT     ;CANNOT CONTINUE TESTING IF ANY OF
4730 024616 000000          HALT     ;STOP
4731
4732 024620 013777 004606 155440  MOV      @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
4733
4734 024626 013746 002442  MOV      @#WRIDAT,-(SP) ;GET READY TO MOVE COMMAND

```

```

4735 024632 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET 'GO' AND
4736 024636 012677 155436 MOV (SP)+,@RHCS1 ;GO WITH
4737 024642 011100 MOV @R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
4738 024644 011305 MOV @R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
4739
4740 ;*ONE REVOLUTION = 16670 MICRO SEC, ONE SECTOR=760 MICRO SEC
4741
4742 024646 104413 WAT ;WAIT FOR RDY BIT TO SET
4743 024650 002300 RHCS1 ;WAIT FOR RHCS1 REGISTER
4744 024652 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
4745 024654 001614 908. ;ALLOW 9080 MICRO SECONDS
4746 024656 001507 839. ;RDY MUST SET BETWEEN
4747 024660 013746 002442 MOV @WRIDAT,-(SP) ;SAVE COMMAND
4748 024664 052716 004101 BIS #IE!GO.DVA,(SP) ;INCLUDE IE!GO!DVA
4749 024670 011637 001124 MOV (SP),@SGDDAT ;SAVE FOR PRINTOUT
4750 024674 022600 CMP (SP)+,R0 ;DURING ABOVE OPERATION ONLY IE!GO.DVA
4751 024676 001405 BEQ 64$ ;BRANCH IF GOOD
4752 024700 010037 001126 MOV R0,@SBDDAT ;BAD DATA
4753 024704 010137 004600 MOV R1,@REGADR ;FAILING REGISTER RHCS1
4754 024710 104021 ERROR 21 ;DURING ABOVE OPERATION ONLY
4755 024712 012746 010500 64$: MOV #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
4756 024716 011637 001124 MOV (SP),@SGDDAT ;SAVE FOR PRINTOUT
4757 024722 022605 CMP (SP)+,R5 ;DURING ABOVE OPERATION ONLY MOL.DPR!VV
4758 024724 001405 BEQ 66$ ;BRANCH IF GOOD
4759 024726 010537 001126 MOV R5,@SBDDAT ;BAD DATA
4760 024732 010337 004600 MOV R3,@REGADR ;FAILING REGISTER RHDS1
4761 024736 104063 ERROR 63 ;DURING ABOVE OPERATION ONLY
4762
4763 ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
4764
4765 024740 004037 041360 JSR R0,@FILLRE ;MOV 0 INTO SAVED RHWC
4766 024744 002272 RHWC ;SAVED REGISTER TO CHANGE
4767 024746 000000 0 ;DATA
4768 024750 004037 041360 JSR R0,@FILLRE ;MOV WRFROM+<200.*2> INTO SAVED RHBA
4769 024754 002274 RHBA ;SAVED REGISTER TO CHANGE
4770 024756 003310 WRFROM+<200.*2> ;DATA
4771 024760 004037 041360 JSR R0,@FILLRE ;MOV 1 INTO SAVED RHDST
4772 024764 002304 RHDST ;SAVED REGISTER TO CHANGE
4773 024766 000001 1 ;DATA
4774
4775 ;*NOW COMPARE REGISTERS BEFORE WRITE DATA WITH REGISTERS
4776 ;*AFTER COMMAND
4777
4778 024770 004037 042444 JSR R0,@COMREG ;COMPARE SAVED REGISTERS WITH
4779 024774 004612 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
4780 024776 002354 WC ;TEST DATA STARTING FROM 'RHWC'
4781 025000 000022 18. ;18. REGISTERS TO BE COMPARED
4782 025002 025006 1$ ;RETURN TO 1$ ON ERROR
4783 025004 025012 2$ ;RETURN TO 2$ ON NO ERROR
4784
4785 025006 104035 1$: ERROR 35 ;WRITE DATA COMMAND CAUSED
4786 025010 000207 RTS PC ;IMPROPER REGISTER CHANGE
4787 ;GOOD DATA GIVES WHAT SHOULD
4788 ;BE
4789 ;RECEIVED DATA GIVES WHAT WAS
4790 ;THERE AFTER COMMAND
    
```

```
4791  
4792  
4793  
4794 025012  
4795 025012 004037 043474  
4796 025016 003534  
4797 025020 002470  
4798 025022 000400  
4799 025024 025030  
4800 025026 025034  
4801  
4802 025030 104036  
4803 025032 000207  
4804  
4805  
4806  
4807  
4808  
4809 025034  
4810 025034 004737 041456  
4811 025040 004037 041326  
4812 025044 003534  
4813 025046 000310  
4814 025050 000000  
4815 025052 004037 041326  
4816 025056 004354  
4817 025060 000070  
4818 025062 000377  
4819  
4820  
4821  
4822 025064 004037 041326  
4823 025070 002470  
4824 025072 000310  
4825 025074 177777  
4826 025076 004037 041326  
4827 025102 003310  
4828 025104 000070  
4829 025106 000377  
4830  
4831  
4832  
4833 025110 004037 043430  
4834 025114 000000  
4835 025116 000  
4836 025117 000  
4837 025120 177470  
4838 025122 003534  
4839 025124 000000  
4840 025126 014000  
4841 025130 002446  
4842  
4843  
4844  
4845 025132 004037 041624  
4846 025136 002272
```

:*NOW WRITE FROM BUFFER WILL BE CHECKED FOR NO CHANGE

2\$: JSR RO,@#COMPAR ;COMPARE TWO BLOCKS OF MEMORY
REINTO ;GOOD DATA STARTS FROM REINTO
WRFROM ;TEST DATA STARTS FROM WRFROM
256. ;256. WORDS TO BE COMPARED
3\$;RETURN TO 3\$ ON ERROR
4\$;RETURN TO 4\$ ON NO ERROR

3\$: ERROR 36 ;WRITE DATA COMMAND CHANGED
RTS PC ;WRITE FROM BUFFER

:*NOW A READ DATA COMMAND WILL BE GIVEN

:*FILL READ INTO BUFFER WITH 200 ZEROS AND 56 OF 377

4\$: JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
JSR RO,@#CLAREA ;CLEAR 200. WORDS, FROM REINTO
REINTO ;STARTING FROM REINTO
200. ;200. WORDS
0 ;FILL WITH 0
JSR RO,@#CLAREA ;CLEAR 56. WORDS, FROM REINTO+<200.*2>
REINTO+<200.*2> ;STARTING FROM REINTO+<200.*2>
56. ;56. WORDS
377 ;FILL WITH 377

:*FILL WRITE FROM BUFFER WITH 200 ONES AND 56 OF 377

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
377 ;FILL WITH 377

:*NOW FILL COMMAND

JSR RO,@#RUN ;SETUP TO RUN FOR DATA COMMAND
0 ;CYLINDER 0
0 ;SECTOR 0
0 ;TRACK 0
-200. ;WORD COUNT = 200.
REINTO ;BUS ADDRESS
0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
ECI.FMT22 ;16 BITS PER WORD FORMAT
READAT ;GET READY TO DO A READAT

:*NOW SAVE REGISTERS FOR COMPARISON AFTER READ DATA COMMAND

JSR RO,@#SAVER ;SAVE REGISTERS
RHWC ;RHWC IS THE FIRST REGISTER SAVED


```

4847 025140 004612 SAVERE ;STARTING ADDRESS OF WHERE
4848 025142 000022 18. ;NUMBER OF REGISTERS
4849
4850 025144 004737 041536 JSR PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
4851 025150 104401 066720 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF ANY OF
4852 025154 000000 HALT ;STOP
4853
4854 025156 013777 004606 155102 MOV @#R4VEC,@R4VEC ;SET RPO4 VECTOR ADDRESS
4855
4856 025164 013746 002446 MOV @#READAT,-(SP) ;GET READY TO MOVE COMMAND
4857 025170 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET 'GO' AND
4858 025174 012677 155100 MOV (SP)+,@RHCS1 ;GO WITH
4859 025200 011100 MOV @R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
4860 025202 011305 MOV @R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
4861
4862 025204 104413 WAT ;WAIT FOR RDY BIT TO SET
4863 025206 002300 RHCS1 ;WAIT FOR RHCS1 REGISTER
4864 025210 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
4865 025212 001614 908. ;ALLOW 9080 MICRO SECONDS
4866 025214 001507 839. ;RDY MUST SET BETWEEN
4867 025216 013746 002446 MOV @#READAT,-(SP) ;SAVE COMMAND
4868 025222 052716 004101 BIS #IE!GO.DVA,(SP) ;INCLUDE IE!GO!DVA
4869 025226 011637 001124 MOV (SP)+,@#SGDDAT ;SAVE FOR PRINTOUT
4870 025232 022600 CMP (SP)+,R0 ;DURING ABOVE OPERATION ONLY IE.GO.DVA
4871 025234 001405 BEQ 67$ ;BRANCH IF GOOD
4872 025236 010037 001126 MOV R0,@#SBDDAT ;BAD DATA
4873 025242 010137 004600 MOV R1,@#REGADR ;FAILING REGISTER RHCS1
4874 025246 104021 ERROR 21 ;DURING ABOVE OPERATION ONLY
4875 025250 012746 010500 67$: MOV #MOL!DPR.VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
4876 025254 011637 001124 MOV (SP)+,@#SGDDAT ;SAVE FOR PRINTOUT
4877 025260 022605 CMP (SP)+,R5 ;DURING ABOVE OPERATION ONLY MOL.DPR.VV
4878 025262 001405 BEQ 69$ ;BRANCH IF GOOD
4879 025264 010537 001126 MOV R5,@#SBDDAT ;BAD DATA
4880 025270 010337 004600 MOV R3,@#REGADR ;FAILING REGISTER RHDS1
4881 025274 104063 ERROR 63 ;DURING ABOVE OPERATION ONLY
4882
4883 ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
4884
4885 025276 004037 041360 JSR R0,@#FILLRE ;MOV 0 INTO SAVED RHWC
4886 025302 002272 RHWC ;SAVED REGISTER TO CHANGE
4887 025304 000000 0 ;DATA
4888 025306 004037 041360 JSR R0,@#FILLRE ;MOV REINTO+<200.*2> INTO SAVED RHBA
4889 025312 002274 RHBA ;SAVED REGISTER TO CHANGE
4890 025314 004354 REINTO+<200.*2> ;DATA
4891 025316 004037 041360 JSR R0,@#FILLRE ;MOV 1 INTO SAVED RHDST
4892 025322 002304 RHDST ;SAVED REGISTER TO CHANGE
4893 025324 000001 1 ;DATA
4894
4895 ;*COMPARE REGISTERS BEFORE READ DATA COMMAND
4896 ;*WITH REGISTERS AFTER COMMAND
4897
4898 025326 004037 042444 JSR R0,@#COMREG ;COMPARE SAVED REGISTERS WITH
4899 025332 004612 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
4900 025334 002354 WC ;TEST DATA STARTING FROM 'RHWC'
4901 025336 000022 18. ;18. REGISTERS TO BE COMPARED
4902 025340 025344 5$ ;RETURN TO 5$ ON ERROR
    
```

```
4903 025342 025350      6$      ;RETURN TO 6$ ON NO ERROR
4904
4905 025344 104033      5$:    ERROR 33      ;READ DATA CAUSED IMPROPER
4906 025346 000207      RTS    PC           ;REGISTER CHANGE
4907                                     ;GOOD DATA GIVFS WHAT SHOULD BE THERE
4908                                     ;RECEIVED DATA GIVES WHAT WAS THERE
4909                                     ;AFTER COMMAND
4910
4911                                     ;*NOW READ INTO BUFFER IS CHECKED FOR GOOD READ
4912
4913 025350      6$:
4914 025350 004037 043474 JSR    R0,@#COMPAR ;COMPARE TWO BLOCKS OF MEMORY
4915 025354 002470 WRFROM ;GOOD DATA STARTS FROM WRFROM
4916 025356 003534 REINTO ;TEST DATA STARTS FROM REINTO
4917 025360 000400 256. ;256. WORDS TO BE COMPARED
4918 025362 025366 7$    ;RETURN TO 7$ ON ERROR
4919 025364 025372 10$   ;RETURN TO 10$ ON NO ERROR
4920
4921 025366 104034      7$:    ERROR 34      ;INCORRECT DATA AFTER
4922 025370 000207      RTS    PC           ;WRITE DATA FOLLOWED BY A
4923                                     ;READ DATA
4924 025372      10$:
4925
4926
4927
```

```

4928 025372 000004          TST36: SCOPE
4929 025374 012706 001000  MOV      #STACK,SP      ;RESET STACK
4930 025400 012737 000036 004604  MOV      #36,@TSTNM     ;SAVE TEST NUMBER
4931 025406 004737 041456  JSR      PC,@CLDISK     ;SET R1-RHCS1, R2-RHCS2
4932
4933          ;*NOW FILL WRITE FROM BUFFER - 256 OF 125252
4934
4935 025412 004037 041326  JSR      R0,@CLAREA     ;CLEAR 256. WORDS, FROM WRFROM
4936 025416 002470          WRFROM          ;STARTING FROM WRFROM
4937 025420 000400          256.           ;256. WORDS
4938 025422 125252          125252         ;FILL WITH 125252
4939
4940          ;*NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA AS
4941          ;*WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
4942          ;*CAN BE MADE TO DETERMINE THAT WRITE DID NOT CHANGE BUFFER
4943
4944 025424 004037 041326  JSR      R0,@CLAREA     ;CLEAR 256. WORDS, FROM REINTO
4945 025430 003534          REINTO          ;STARTING FROM REINTO
4946 025432 000400          256.           ;256. WORDS
4947 025434 125252          125252         ;FILL WITH 125252
4948
4949          ;*NOW WRITE DATA COMMAND WILL BE LOADED
4950
4951 025436 004037 043430  JSR      R0,@RJUN       ;SETUP TO RUN FOR DATA COMMAND
4952 025442 000000          0              ;CYLINDER 0
4953 025444 000          .BYTE          0              ;SECTOR 0
4954 025445 000          .BYTE          0              ;TRACK 0
4955 025446 177400          -256.          ;WORD COUNT = 256.
4956 025450 002470          WRFROM          ;BUS ADDRESS
4957 025452 000000          0              ;DO NOT INHIBIT BUS ADDRESS INCREMENT
4958 025454 010000          FMT22          ;16 BITS PER WORD FORMAT
4959 025456 002442          WRIDAT         ;GET READY TO DO A WRIDAT
4960
4961          ;*NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE DATA
4962
4963 025460 004037 041624  JSR      R0,@SAVER      ;SAVE REGISTERS
4964 025464 002272          RHWC           ;RHWC IS THE FIRST REGISTER SAVED
4965 025466 004612          SAVERE         ;STARTING ADDRESS OF WHERE
4966 025470 000022          18.           ;NUMBER OF REGISTERS
4967
4968 025472 004737 041536  JSR      PC,@CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV 1
4969 025476 104401 066720  TYPE      ,CPHALT      ;CANNOT CONTINUE TESTING IF ANY OF
4970 025502 000000          HALT          ;STOP
4971
4972 025504 013777 004606 154554  MOV      @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
4973
4974 025512 013746 002442  MOV      @#WRIDAT,-(SP) ;GET READY TO MOVE COMMAND
4975 025516 052716 000101  BIS      #GO!IE,(SP)    ;GET READY TO SET 'GO' AND
4976 025522 012677 154552  MOV      (SP)+,@RHCS1   ;GO WITH
4977 025526 011100          MOV      @R1,R0        ;SAVE RHCS1 DURING ABOVE OPERATION
4978 025530 011305          MOV      @R3,R5        ;SAVE RHDS1 DURING ABOVE OPERATION
4979
4980          ;*ONE REVOLUTION=16670 MICROSEC, ONE SECTOR=760 MICROSEC
4981
4982 025532 104413          WAT            ;WAIT FOR RDY BIT TO SET
4983 025534 002300          RHCS1         ;WAIT FOR RHCS1 REGISTER
  
```

```
4984 025536 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
4985 025540 001614 908. ;ALLOW 9080 MICRO SECONDS
4986 025542 001507 839. ;RDY MUST SET BETWEEN
4987 025544 013746 002442 MOV @#WRIDAT,-(SP) ;SAVE COMMAND
4988 025550 052716 004101 BIS #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
4989 025554 011637 001124 MOV (SP),@#SGDDAT ;SAVE FOR PRINTOUT
4990 025560 022600 CMP (SP)+,R0 ;DURING ABOVE OPERATION ONLY IE.GO!DVA
4991 025562 001405 BEQ 64$ ;BRANCH IF GOOD
4992 025564 010037 001126 MOV R0,@#SBDDAT ;BAD DATA
4993 025570 010137 004600 MOV R1,@#REGADR ;FAILING REGISTER RHCS1
4994 025574 104021 ERROR 21 ;DURING ABOVE OPERATION ONLY
4995 025576 012746 010500 64$: MOV #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
4996 025602 011637 001124 MOV (SP),@#SGDDAT ;SAVE FOR PRINTOUT
4997 025606 022605 CMP (SP)+,R5 ;DURING ABOVE OPERATION ONLY MOL.DPR VV
4998 025610 001405 BEQ 66$ ;BRANCH IF GOOD
4999 025612 010537 001126 MOV R5,@#SBDDAT ;BAD DATA
5000 025616 010337 004600 MOV R3,@#REGADR ;FAILING REGISTER RHDS1
5001 025622 104063 ERROR 63 ;DURING ABOVE OPERATION ONLY
5002
5003 ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
5004
5005 025624 004037 041360 JSR R0,@#FILLRE ;MOV 0 INTO SAVED RHWC
5006 025630 002272 RHWC ;SAVED REGISTER TO CHANGE
5007 025632 000000 0 ;DATA
5008 025634 004037 041360 JSR R0,@#FILLRE ;MOV WRFROM+<256.*2> INTO SAVED RHBA
5009 025640 002274 RHBA ;SAVED REGISTER TO CHANGE
5010 025642 003470 WRFROM+<256.*2> ;DATA
5011 025644 004037 041360 JSR R0,@#FILLRE ;MOV 1 INTO SAVED RHDST
5012 025650 002304 RHDST ;SAVED REGISTER TO CHANGE
5013 025652 000001 1 ;DATA
5014
5015 ;*NOW COMPARE REGISTERS BEFORE WRITE DATA WITH REGISTERS
5016 ;*AFTER COMMAND
5017
5018 025654 004037 042444 JSR R0,@#COMREG ;COMPARE SAVED REGISTERS WITH
5019 025660 004612 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
5020 025662 002354 WC ;TEST DATA STARTING FROM 'RHWC'
5021 025664 000022 18. ;18. REGISTERS TO BE COMPARED
5022 025666 025672 1$ ;RETURN TO 1$ ON ERROR
5023 025670 025676 2$ ;RETURN TO 2$ ON NO ERROR
5024
5025 025672 104035 1$: ERROR 35 ;WRITE DATA COMMAND CAUSED
5026 025674 000207 RTS PC ;IMPROPER REGISTER CHANGE
5027 ;GOOD DATA GIVES WHAT SHOULD
5028 ;BE
5029 ;RECEIVED DATA GIVES WHAT WAS
5030 ;THERE AFTER COMMAND
5031
5032 ;*NOW WRITE FROM BUFFER WILL CHECKED FOR NO CHANGE
5033
5034 025676 2$: JSR R0,@#COMPAR ;COMPARE TWO BLOCKS OF MEMORY
5035 025676 004037 043474 REINTO ;GOOD DATA STARTS FROM REINTO
5036 025702 003534 WRFROM ;TEST DATA STARTS FROM WRFROM
5037 025704 002470 256. ;256. WORDS TO BE COMPARED
5038 025706 000400 3$ ;RETURN TO 3$ ON ERROR
5039 025710 025714
```

```
5040 025712 025720          4$          ;RETURN TO 4$ ON NO ERROR
5041
5042 025714 104036          3$:      ERROR 36          ;WRITE DATA COMMAND CHANGED
5043 025716 000207          RTS      PC          ;WRITE FROM BUFFER
5044
5045                          ;*NOW A READ DATA COMMAND WILL BE GIVEN
5046                          ;*FILL READ INTO BUFFER WITH 256 ZEROS
5047
5048 025720          4$:
5049 025720 004737 041456      JSR      PC,@#CLDISK      ;SET R1-RHCS1, R2-RHCS2
5050 025724 004037 041326      JSR      RO,@#CLAREA      ;CLEAR 256. WORDS, FROM REINTO
5051 025730 003534          REINTO      ;STARTING FROM REINTO
5052 025732 000400          256.        ;256. WORDS
5053 025734 000000          0          ;FILL WITH 0
5054
5055                          ;*FILL WRITE FROM BUFFER WITH 256 OF 125252
5056
5057 025736 004037 041326      JSR      RO,@#CLAREA      ;CLEAR 256. WORDS, FROM WRFROM
5058 025742 002470          WRFROM      ;STARTING FROM WRFROM
5059 025744 000400          256.        ;256. WORDS
5060 025746 125252          125252      ;FILL WITH 125252
5061
5062                          ;*NOW FILL COMMAND
5063
5064 025750 004037 043430      JSR      RO,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
5065 025754 000000          0          ;CYLINDER 0
5066 025756 000        .BYTE 0          ;SECTOR 0
5067 025757 000        .BYTE 0          ;TRACK 0
5068 025760 177400          -256.      ;WORD COUNT = 256.
5069 025762 003534          REINTO      ;BUS ADDRESS
5070 025764 000000          0          ;DO NOT INHIBIT BUS ADDRESS INCREMENT
5071 025766 014000          ECI.FMT22 ;16 BITS PER WORD FORMAT
5072 025770 002446          READAT      ;GET READY TO DO A READAT
5073
5074                          ;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ DATA COMMAND
5075
5076 025772 004037 041624      JSR      RO,@#SAVER      ;SAVE REGISTERS
5077 025776 002272          RHWC          ;RHWC IS THE FIRST REGISTER SAVED
5078 026000 004612          SAVERE      ;STARTING ADDRESS OF WHERE
5079 026002 000022          18.          ;NUMBER OF REGISTERS
5080
5081 026004 004737 041536      JSR      PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV - 1
5082 026010 104401 066720      TYPE      ,CPHALT      ;CANNOT CONTINUE TESTING IF ANY OF
5083 026014 000000          HALT          ;STOP
5084
5085 026016 013777 004606 154242 MOV      @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
5086
5087 026024 013746 002446      MOV      @#READAT,-(SP) ;GET READY TO MOVE COMMAND
5088 026030 052716 000101      BIS      #GO!IE,(SP)    ;GET READY TO SET 'GO' AND
5089 026034 012677 154240      MOV      (SP)+,@RHCS1   ;GO WITH
5090 026040 011100          MOV      @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
5091 026042 011305          MOV      @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
5092
5093 026044 104413          WAT          ;WAIT FOR RDY BIT TO SET
5094 026046 002300          RHCS1      ;WAIT FOR RHCS1 REGISTER
5095 026050 000200          RDY          ;WAIT FOR RDY BIT IN RHCS1 REGISTER
```

```
5096 026052 001614          908.          ;ALLOW 9080 MICRO SECONDS
5097 026054 001507          839.          ;RDY MUST SET BETWEEN
5098 026056 013746 002446   MOV @#READAT,-(SP) ;SAVE COMMAND
5099 026062 052716 004101   BIS #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
5100 026066 011637 001124   MOV (SP),@#SGDDAT ;SAVE FOR PRINTOUT
5101 026072 022600          CMP (SP)+,R0 ;DURING ABOVE OPERATION ONLY IE.GO.DVA
5102 026074 001405          BEQ 67$ ;BRANCH IF GOOD
5103 026076 010037 001126   MOV R0,@#SBDDAT ;BAD DATA
5104 026102 010137 004600   MOV R1,@#REGADR ;FAILING REGISTER RHCS1
5105 026106 104021          ERROR 21 ;DURING ABOVE OPERATION ONLY
5106 026110 012746 010500   67$: MOV #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
5107 026114 011637 001124   MOV (SP),@#SGDDAT ;SAVE FOR PRINTOUT
5108 026120 022605          CMP (SP)+,R5 ;DURING ABOVE OPERATION ONLY MOL.DPR.VV
5109 026122 001405          BEQ 69$ ;BRANCH IF GOOD
5110 026124 010537 001126   MOV R5,@#SBDDAT ;BAD DATA
5111 026130 010337 004600   MOV R3,@#REGADR ;FAILING REGISTER RHDS1
5112 026134 104063          ERROR 63 ;DURING ABOVE OPERATION ONLY
5113
5114          ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
5115
5116 026136 004037 041360   JSR R0,@#FILLRE ;MOV 0 INTO SAVED RHWC
5117 026142 002272          RHWC          ;SAVED REGISTER TO CHANGE
5118 026144 000000          0            ;DATA
5119 026146 004037 041360   JSR R0,@#FILLRE ;MOV REINTO+<256.*2> INTO SAVED RHBA
5120 026152 002274          RHBA          ;SAVED REGISTER TO CHANGE
5121 026154 004534          REINTO+<256 *2> ;DATA
5122 026156 004037 041360   JSR R0,@#FILLRE ;MOV 1 INTO SAVED RHDST
5123 026162 002304          RHDST        ;SAVED REGISTER TO CHANGE
5124 026164 000001          1            ;DATA
5125
5126          ;*COMPARE REGISTERS BEFORE READ DATA COMMAND
5127          ;*WITH REGISTERS AFTER COMMAND
5128
5129 026166 004037 042444   JSR R0,@#COMREG ;COMPARE SAVED REGISTERS WITH
5130 026172 004612          SAVERE        ;GOOD DATA SAVED IN 'SAVERE'
5131 026174 002354          WC          ;TEST DATA STARTING FROM 'RHWC'
5132 026176 000022          18.         ;18. REGISTERS TO BE COMPARED
5133 026200 026204          5$          ;RETURN TO 5$ ON ERROR
5134 026202 026210          6$          ;RETURN TO 6$ ON NO ERROR
5135
5136 026204 104033          5$: ERROR 33 ;READ DATA CAUSED IMPROPER
5137 026206 000207          RTS PC      ;REGISTER CHANGE
5138          ;GOOD DATA GIVES WHAT SHOULD BE THE
5139          ;RECEIVED DATA GIVES WHAT WAS THERE
5140          ;AFTER COMMAND
5141
5142          ;*NOW READ INTO BUFFER IS CHECKED FOR GOOD READ
5143
5144 026210          6$:
5145 026210 004037 043474   JSR R0,@#COMPAR ;COMPARE TWO BLOCKS OF MEMORY
5146 026214 002470          WRFROM       ;GOOD DATA STARTS FROM WRFROM
5147 026216 003534          REINTO       ;TEST DATA STARTS FROM REINTO
5148 026220 000400          256.        ;256. WORDS TO BE COMPARED
5149 026222 026226          7$          ;RETURN TO 7$ ON ERROR
5150 026224 026232          10$         ;RETURN TO 10$ ON NO ERROR
5151
```


5157	026232	000004		TST37: SCOPE			
5158	026234	012706	001000	MOV	#STACK,SP	:	RESET STACK
5159	026240	012737	000037	MOV	#37,@TSTNM	:	SAVE TEST NUMBER
5160	026246	004737	041456	JSR	PC,@CLDISK	:	SET R1-RHCS1, R2-RHCS2
5161							
5162						:	*NOW FILL WRITE FROM BUFFER-200 OF 52525 AND 56 OF 377
5163							
5164	026252	004037	041326	JSR	RO,@CLAREA	:	CLEAR 200. WORDS, FROM WRFROM
5165	026256	002470		WRFROM		:	STARTING FROM WRFROM
5166	026260	000310		200.		:	200. WORDS
5167	026262	052525		52525		:	FILL WITH 52525
5168	026264	004037	041326	JSR	RO,@CLAREA	:	CLEAR 56. WORDS, FROM WRFROM+<200.*2>
5169	026270	003310		WRFROM+<200.*2>		:	STARTING FROM WRFROM+<200.*2>
5170	026272	000070		56.		:	56. WORDS
5171	026274	000377		377		:	FILL WITH 377
5172							
5173						:	*NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA AS
5174						:	*WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
5175						:	*CAN BE MADE TO DETERMINE THAT WRITE DID NOT CHANGE BUFFER
5176							
5177	026276	004037	041326	JSR	RO,@CLAREA	:	CLEAR 200. WORDS, FROM REINTO
5178	026302	003534		REINTO		:	STARTING FROM REINTO
5179	026304	000310		200.		:	200. WORDS
5180	026306	052525		52525		:	FILL WITH 52525
5181	026310	004037	041326	JSR	RO,@CLAREA	:	CLEAR 56. WORDS, FROM REINTO+<200.*2>
5182	026314	004354		REINTO+<200.*2>		:	STARTING FROM REINTO+<200.*2>
5183	026316	000070		56.		:	56. WORDS
5184	026320	000377		377		:	FILL WITH 377
5185							
5186						:	*NOW WRITE DATA COMMAND WILL BE LOADED
5187							
5188	026322	004037	043430	JSR	RO,@RUN	:	SETUP TO RUN FOR DATA COMMAND
5189	026326	000000		0		:	CYLINDER 0
5190	026330	000		.BYTE 0		:	SECTOR 0
5191	026331	000		.BYTE 0		:	TRACK 0
5192	026332	177470		-200.		:	WORD COUNT = 200.
5193	026334	002470		WRFROM		:	BUS ADDRESS
5194	026336	000000		0		:	DO NOT INHIBIT BUS ADDRESS INCREMENT
5195	026340	010000		FMT22		:	16 BITS PER WORD FORMAT
5196	026342	002442		WRIDAT		:	GET READY TO DO A WRIDAT
5197							
5198						:	*NOW SAVE REGISTER FOR COMPARISON AFTER WRITE DATA
5199							
5200	026344	004037	041624	JSR	RO,@SAVER	:	SAVE REGISTERS
5201	026350	002272		RHWC		:	RHWC IS THE FIRST REGISTER SAVED
5202	026352	004612		SAVERE		:	STARTING ADDRESS OF WHERE
5203	026354	000022		18.		:	NUMBER OF REGISTERS
5204							
5205	026356	004737	041536	JSR	PC,@CHECKT	:	CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
5206	026362	104401	066720	TYPE	,CPHALT	:	CANNOT CONTINUE TESTING IF ANY OF
5207	026366	000000		HALT		:	STOP
5208							
5209	026370	013777	004606	MOV	@RP4VEC,@RPEC	:	SET RP04 VECTOR ADDRESS
5210			153670				
5211	026376	013746	002442	MOV	@WRIDAT,-(SP)	:	GET READY TO MOVE COMMAND
5212	026402	052716	000101	BIS	#GO'IE,(SP)	:	GET READY TO SET 'GO' AND


```

5213 026406 012677 153666      MOV      (SP)+,@RHCS1      ;GO WITH
5214 026412 011100      MOV      @R1,R0           ;SAVE RHCS1 DURING ABOVE OPERATION
5215 026414 011305      MOV      @R3,R5           ;SAVE RHDS1 DURING ABOVE OPERATION
5216
5217                          ;*ONE REVOLUTION=16670 MICROSEC, ONE SECTOR=760 MICROSEC
5218
5219 026416 104413      WAT                          ;WAIT FOR RDY BIT TO SET
5220 026420 002300      RHCS1                       ;WAIT FOR RHCS1 REGISTER
5221 026422 000200      RDY                         ;WAIT FOR RDY BIT IN RHCS1 REGISTER
5222 026424 001614      908.                       ;ALLOW 9080 MICRO SECONDS
5223 026426 001507      839.                       ;RDY MUST SET BETWEEN
5224 026430 013746 002442      MOV      @WRIDAT,-(SP)     ;SAVE COMMAND
5225 026434 052716 004101      BIS      #IE!GO!DVA,(SP)  ;INCLUDE IE!GO!DVA
5226 026440 011637 001124      MOV      (SP),@SGDDAT     ;SAVE FOR PRINTOUT
5227 026444 022600      CMP      (SP)+,R0         ;DURING ABOVE OPERATION ONLY IE!GO!DVA
5228 026446 001405      BEQ      64$              ;BRANCH IF GOOD
5229 026450 010037 001126      MOV      R0,@SBDDAT       ;BAD DATA
5230 026454 010137 004600      MOV      R1,@REGADR       ;FAILING REGISTER RHCS1
5231 026460 104021      ERROR   21                ;DURING ABOVE OPERATION ONLY
5232 026462 012746 010500      64$: MOV      #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
5233 026466 011637 001124      MOV      (SP),@SGDDAT     ;SAVE FOR PRINTOUT
5234 026472 022605      CMP      (SP)+,R5         ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
5235 026474 001405      BEQ      66$              ;BRANCH IF GOOD
5236 026476 010537 001126      MOV      R5,@SBDDAT       ;BAD DATA
5237 026502 010337 004600      MOV      R3,@REGADR       ;FAILING REGISTER RHDS1
5238 026506 104063      ERROR   63                ;DURING ABOVE OPERATION ONLY
5239
5240                          ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
5241
5242 026510 004037 041360      JSR      R0,@FILLRE        ;MOV 0 INTO SAVED RHWC
5243 026514 002272      RHWC                       ;SAVED REGISTER TO CHANGE
5244 026516 000000      0                           ;DATA
5245 026520 004037 041360      JSR      R0,@FILLRE        ;MOV WRFROM+<200.*2> INTO SAVED RHBA
5246 026524 002274      RHBA                       ;SAVED REGISTER TO CHANGE
5247 026526 003310      WRFROM+<200.*2>           ;DATA
5248 026530 004037 041360      JSR      R0,@FILLRE        ;MOV 1 INTO SAVED RHDST
5249 026534 002304      RHDST                      ;SAVED REGISTER TO CHANGE
5250 026536 000001      1                           ;DATA
5251
5252                          ;*NOW COMPARE REGISTERS BEFORE WRITE DATA WITH REGISTERS
5253                          ;*AFTER COMMAND
5254
5255 026540 004037 042444      JSR      R0,@COMREG        ;COMPARE SAVED REGISTERS WITH
5256 026544 004612      SAVERE                     ;GOOD DATA SAVED IN 'SAVERE'
5257 026546 002354      WC                         ;TEST DATA STARTING FROM 'RHWC'
5258 026550 000022      18.                       ;18. REGISTERS TO BE COMPARED
5259 026552 026556      1$                         ;RETURN TO 1$ ON ERROR
5260 026554 026562      2$                         ;RETURN TO 2$ ON NO ERROR
5261
5262 026556 104035      1$: ERROR 35                ;WRITE DATA COMMAND CAUSED
5263 026560 000207      RTS      PC                 ;IMPROPER REGISTER CHANGE
5264
5265                          ;GOOD DATA GIVES WHAT SHOULD
5266                          ;BE
5267                          ;RECEIVED DATA GIVES WHAT WAS
5268                          ;THERE AFTER COMMAND

```

```

5269                                     ;*NOW WRITE FROM BUFFER WILL BE CHECKED FOR NO CHANGE
5270
5271 026562                               2$:
5272 026562 004037 043474                JSR    RO,@#COMPAR    ;COMPARE TWO BLOCKS OF MEMORY
5273 026566 003534                        REINTO   ;GOOD DATA STARTS FROM REINTO
5274 026570 002470                        WRFROM   ;TEST DATA STARTS FROM WRFROM
5275 026572 000400                        256.    ;256. WORDS TO BE COMPARED
5276 026574 026600                        3$      ;RETURN TO 3$ ON ERROR
5277 026576 026604                        4$      ;RETURN TO 4$ ON NO ERROR
5278
5279 026600 104036                        3$:     ERROR 36      ;WRITE DATA COMMAND CHANGED
5280 026602 000207                        RTS     PC          ;WRITE FROM BUFFER
5281
5282                                     ;*NOW A READ DATA COMMAND WILL BE GIVEN
5283
5284                                     ;*FILL READ INTO BUFFER WITH 200 ZEROS AND 56 OF ALL ONES
5285
5286 026604                               4$:
5287 026604 004737 041456                JSR    PC,@#CLDISK   ;SET R1-RHCS1, R2-RHCS2
5288 026610 004037 041326                JSR    RO,@#CLAREA   ;CLEAR 200. WORDS, FROM REINTO
5289 026614 003534                        REINTO   ;STARTING FROM REINTO
5290 026616 000310                        200.    ;200. WORDS
5291 026620 000000                        0       ;FILL WITH 0
5292 026622 004037 041326                JSR    RO,@#CLAREA   ;CLEAR 56. WORDS, FROM REINTO+<200.*2>
5293 026626 004354                        REINTO+<200.*2>     ;STARTING FROM REINTO+<200.*2>
5294 026630 000070                        56.    ;56. WORDS
5295 026632 000377                        377    ;FILL WITH 377
5296
5297                                     ;*FILL WRITE FROM BUFFER WITH 200 OF 52525 AND 56 OF 0
5298
5299 026634 004037 041326                JSR    RO,@#CLAREA   ;CLEAR 200. WORDS, FROM WRFROM
5300 026640 002470                        WRFROM   ;STARTING FROM WRFROM
5301 026642 000310                        200.    ;200. WORDS
5302 026644 052525                        52525   ;FILL WITH 52525
5303 026646 004037 041326                JSR    RO,@#CLAREA   ;CLEAR 56. WORDS, FROM WRFROM+<200.*2>
5304 026652 003310                        WRFROM+<200.*2>     ;STARTING FROM WRFROM+<200.*2>
5305 026654 000070                        56.    ;56. WORDS
5306 026656 000377                        377    ;FILL WITH 377
5307
5308                                     ;*NOW FILL COMMAND
5309
5310 026660 004037 043430                JSR    RO,@#RUN      ;SETUP TO RUN FOR DATA COMMAND
5311 026664 000000                        0       ;CYLINDER 0
5312 026666 000      .BYTE                0       ;SECTOR 0
5313 026667 000      .BYTE                0       ;TRACK 0
5314 026670 177470                        -200.   ;WORD COUNT = 200.
5315 026672 003534                        REINTO   ;BUS ADDRESS
5316 026674 000000                        0       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
5317 026676 014000                        ECI!FMT22 ;16 BITS PER WORD FORMAT
5318 026700 002446                        READAT   ;GET READY TO DO A READAT
5319
5320                                     ;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ DATA COMMAND
5321
5322 026702 004037 041624                JSR    RO,@#SAVER    ;SAVE REGISTERS
5323 026706 002272                        RHWC     ;RHWC IS THE FIRST REGISTER SAVED
5324 026710 004612                        SAVERE   ;STARTING ADDRESS OF WHERE
    
```



```
5381
5382 027114 104033          5$:  ERROR 33          ;READ DATA CAUSED IMPROPER
5383 027116 000207          RTS   PC          ;REGISTER CHANGE
5384                                     ;GOOD DATA GIVES WHAT SHOULD BE THE
5385                                     ;RECEIVED DATA GIVES WHAT WAS THERE
5386                                     ;AFTER COMMAND
5387
5388                                     ;*NOW READ INTO BUFFER IS CHECKED FOR GOOD READ
5389
5390 027120          6$:
5391 027120 004037 043474    JSR   RO,@#COMPAR    ;COMPARE TWO BLOCKS OF MEMORY
5392 027124 002470          WRFROM          ;GOOD DATA STARTS FROM WRFROM
5393 027126 003534          REINTO          ;TEST DATA STARTS FROM REINTO
5394 027130 000400          256.          ;256. WORDS TO BE COMPARED
5395 027132 027136          7$          ;RETURN TO 7$ ON ERROR
5396 027134 027142          10$         ;RETURN TO 10$ ON NO ERROR
5397
5398 027136 104034          7$:  ERROR 34          ;INCORRECT DATA AFTER
5399 027140 000207          RTS   PC          ;WRITE DATA FOLLOWED BY A
5400                                     ;READ DATA
5401 027142          10$:
5402
5403
5404
5405
```

```

5406 027142 000004          TST40: SCOPE
5407 027144 012737 000001 001212 MOV #1,STIIES          ;;DO 1 ITERATION
5408 027152 012706 001000          MOV #STACK,SP          ;RESET STACK
5409 027156 012737 000040 004604 MOV #40,@TSTNM          ;SAVE TEST NUMBER
5410 027164 004737 041456          JSR PC,@#CLDISK          ;SET R1-RHCS1, R2-RHCS2
5411 027170 005737 004750          TST @#RH70              ;TEST FOR RH70 CONTROLLER
5412 027174 001402          BEQ 30$                  ;IF FLAG = 1, THIS TEST IS SKIPPED
5413 027176 000137 030312          JMP TST41 ;              JUMP TO NEXT TEST -----)
5414
5415 027202 005037 004732          CLR @#UBUSB             ;CLEAR UNIBUS INDICATOR
5416
5417          ;*NOW FILL WRITE FROM BUFFER-200 OF 52525 AND 56 OF 377
5418
5419 027206 004037 041326          JSR RO,@#CLAREA          ;CLEAR 200. WORDS, FROM WRFROM
5420 027212 002470          WRFROM                   ;STARTING FROM WRFROM
5421 027214 000310          200.                     ;200. WORDS
5422 027216 052525          52525                    ;FILL WITH 52525
5423 027220 004037 041326          JSR RO,@#CLAREA          ;CLEAR 56. WORDS, FROM WRFROM+<200.*2>
5424 027224 003310          WRFROM+<200.*2>          ;STARTING FROM WRFROM+<200.*2>
5425 027226 000070          56.                      ;56. WORDS
5426 027230 000377          377                      ;FILL WITH 377
5427
5428          ;*NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA AS
5429          ;*WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
5430          ;*CAN BE MADE TO DETERMINE THAT WRITE DID NOT CHANGE BUFFER
5431
5432 027232 004037 041326          JSR RO,@#CLAREA          ;CLEAR 200. WORDS, FROM REINTO
5433 027236 003534          REINTO                   ;STARTING FROM REINTO
5434 027240 000310          200.                     ;200. WORDS
5435 027242 052525          52525                    ;FILL WITH 52525
5436 027244 004037 041326          JSR RO,@#CLAREA          ;CLEAR 56. WORDS, FROM REINTO+<200.*2>
5437 027250 004354          REINTO+<200.*2>          ;STARTING FROM REINTO+<200.*2>
5438 027252 000070          56.                      ;56. WORDS
5439 027254 000377          377                      ;FILL WITH 377
5440
5441          ;*NOW WRITE DATA COMMAND WILL BE LOADED
5442
5443 027256 004037 043430          JSR RO,@#RUN             ;SETUP TO RUN FOR DATA COMMAND
5444 027262 000000          0                         ;CYLINDER 0
5445 027264 000          .BYTE 0                  ;SECTOR 0
5446 027265 000          .BYTE 0                  ;TRACK 0
5447 027266 177470          -200.                    ;WORD COUNT = 200.
5448 027270 002470          WRFROM                   ;BUS ADDRESS
5449 027272 000000          0                         ;DO NOT INHIBIT BUS ADDRESS INCREMENT
5450 027274 010000          FMT22                    ;16 BITS PER WORD FORMAT
5451 027276 002442          WRIDAT                   ;GET READY TO DO A WRIDAT
5452 027300 052777 002000 152772          BIS #PSEL,@RHCS1        ;SET PORT B
5453          ;THAT IS UNIBUS B
5454
5455          ;*NOW SAVE REGISTER FOR COMPARISON AFTER WRITE DATA
5456
5457 027306 004037 041624          JSR RO,@#SAVER          ;SAVE REGISTERS
5458 027312 002272          RHWC                     ;RHWC IS THE FIRST REGISTER SAVED
5459 027314 004612          SAVERE                   ;STARTING ADDRESS OF WHERE
5460 027316 000022          18.                      ;NUMBER OF REGISTERS
5461

```

```

5462 027320 004737 041536 JSR PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV 1
5463 027324 104401 066720 TYPE ,CPHAL ;CANNOT CONTINUE TESTING IF ANY OF
5464 027330 000000 HALT ;STOP
5465
5466 027332 013777 004606 152726 MOV @#R4VEC,@#RVEC ;SET RPO4 VECTOR ADDRESS
5467 027340 013746 002442 MOV @#WRIDAT,-(SP) ;GET READY TO MOVE COMMAND
5468 027344 052716 002101 BIS #GO!IE!PSEL,(SP) ;GET READY TO SET 'GO' AND
5469 027350 012677 152724 MOV (SP)+,@#RHCS1 ;GO WITH
5470 027354 011100 MOV @#R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
5471 027356 011305 MOV @#R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
5472
5473 ;*ONE REVOLUTION=16670 MICROSEC, ONE SECTOR=760 MICROSEC
5474
5475 027360 104413 WAT ;WAIT FOR RDY BIT TO SET
5476 027362 002300 RHCS1 ;WAIT FOR RHCS1 REGISTER
5477 027364 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
5478 027366 001614 908. ;ALLOW 9080 MICRO SECONDS
5479 027370 001507 839. ;RDY MUST SET BETWEEN
5480 027372 013746 002442 MOV @#WRIDAT,-(SP) ;SAVE COMMAND
5481 027376 052716 006101 BIS #IE!GO!DVA!PSEL,(SP) ;INCLUDE IE!GO!DVA!PSEL
5482 027402 011637 001124 MOV (SP),@#SGDDAT ;SAVE FOR PRINTOUT
5483 027406 022600 CMP (SP)+,R0 ;DURING ABOVE OPERATION ONLY IE!GO!DVA!PSEL
5484 027410 001405 BEQ 64$ ;BRANCH IF GOOD
5485 027412 010037 001126 MOV R0,@#SBDDAT ;BAD DATA
5486 027416 010137 004600 MOV R1,@#REGADR ;FAILING REGISTER RHCS1
5487 027422 104021 ERROR 21 ;DURING ABOVE OPERATION ONLY
5488 027424 012746 010500 64$: MOV #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
5489 027430 011637 001124 MOV (SP),@#SGDDAT ;SAVE FOR PRINTOUT
5490 027434 022605 CMP (SP)+,R5 ;DURING ABOVE OPERATION ONLY MOL!DPR.VV
5491 027436 001405 BEQ 66$ ;BRANCH IF GOOD
5492 027440 010537 001126 MOV R5,@#SBDDAT ;BAD DATA
5493 027444 010337 004600 MOV R3,@#REGADR ;FAILING REGISTER RHDS1
5494 027450 104063 ERROR 63 ;DURING ABOVE OPERATION ONLY
5495
5496
5497 ;*CHECK IF NEM NON EXISTANT MEMORY IS SET
5498 ;*IF SET IT MEANS UNIBUS B IS NOT CONNECTED
5499 ;*SO THIS TEST IS NOT PERFORMED
5500
5501 027452 032777 004000 152616 BIT #NEM,@#RHCS2 ;TEST NEM
5502 027460 001441 BEQ 11$ ;BRANCH IF UNIBUS B THERE
5503 027462 012737 177777 004732 MOV #-1,@#UBUSB ;UNIBUS B NOT THERE
5504 027470 104401 027476 TYPE ,68$ ;:TYPE ASCIZ STRING
5505 027474 000425 BR 67$ ;:GET OVER THE ASCIZ
5506 027550 104401 001223 TYPE ,$CRLF
5507 027554 104401 001223 TYPE ,$CRLF
5508 027560 000137 030312 JMP @#10$ ;JUMP TO NEXT TEST - NO UNIBUS B
5509 027564 11$:
5510 027564 104401 027572 TYPE ,70$ ;:TYPE ASCIZ STRING
5511 027570 000424 BR 69$ ;:GET OVER THE ASCIZ
5512 027642 104401 001223 TYPE ,$CRLF
5513 027646 104401 001223 TYPE ,$CRLF
5514
5515 ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
5516
5517 027652 004037 041360 JSR R0,@#FILLRE ;MOV 0 INTO SAVED RHWC

```

```

5518 027656 002272          RHWC          ;SAVED REGISTER TO CHANGE
5519 027660 000000          0              ;DATA
5520 027662 004037 041360  JSR      RO,@#FILLRE ;MOV WRFROM+<200.*2> INTO SAVED RHBA
5521 027666 002274          RHBA          ;SAVED REGISTER TO CHANGE
5522 027670 003310          WRFROM+<200.*2> ;DATA
5523 027672 004037 041360  JSR      RO,@#FILLRE ;MOV 1 INTO SAVED RHDST
5524 027676 002304          RHDST         ;SAVED REGISTER TO CHANGE
5525 027700 000001          1              ;DATA
5526
5527                          ;*NOW COMPARE REGISTERS BEFORE WRITE DATA WITH REGISTERS
5528                          ;*AFTER COMMAND
5529
5530 027702 004037 042444  JSR      RO,@#COMREG ;COMPARE SAVED REGISTERS WITH
5531 027706 004612          SAVERE         ;GOOD DATA SAVED IN 'SAVERE'
5532 027710 002354          WC              ;TEST DATA STARTING FROM 'RHWC'
5533 027712 000022          18.           ;18. REGISTERS TO BE COMPARED
5534 027714 027720          1$             ;RETURN TO 1$ ON ERROR
5535 027716 027724          2$             ;RETURN TO 2$ ON NO ERROR
5536
5537 027720 104074          1$:      ERROR 74      ;WHILE USING UNIBUS B
5538                          ;WRITE DATA COMMAND CAUSED
5539 027722 000207          RTS      PC      ;IMPROPER REGISTER CHANGE
5540                          ;GOOD DATA GIVES WHAT SHOULD
5541                          ;BE
5542                          ;RECEIVED DATA GIVES WHAT WAS
5543                          ;THERE AFTER COMMAND
5544
5545                          ;*NOW WRITE FROM BUFFER WILL BE CHECKED FOR NO CHANGE
5546
5547 027724          2$:
5548 027724 004037 043474  JSR      RO,@#COMPAR ;COMPARE TWO BLOCKS OF MEMORY
5549 027730 003534          REINTO        ;GOOD DATA STARTS FROM REINTO
5550 027732 00247C          WRFROM        ;TEST DATA STARTS FROM WRFROM
5551 027734 000400          256.         ;256. WORDS TO BE COMPARED
5552 027736 027742          3$             ;RETURN TO 3$ ON ERROR
5553 027740 027746          4$             ;RETURN TO 4$ ON NO ERROR
5554
5555 027742 104075          3$:      ERROR 75      ;WHILE USING UNIBUS B
5556                          ;WRITE DATA COMMAND CHANGED
5557 027744 000207          RTS      PC      ;WRITE FROM BUFFER
5558
5559                          ;*NOW A READ DATA COMMAND WILL BE GIVEN
5560                          ;*FILL READ INTO BUFFER WITH 200 ZEROS AND 56 OF ALL ONES
5561
5562 027746          4$:
5563 027746 004737 041456  JSR      PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
5564 027752 004037 041326  JSR      RO,@#CLAREA ;CLEAR 200. WORDS, FROM REINTO
5565 027756 003534          REINTO        ;STARTING FROM REINTO
5566 027760 000310          200.         ;200. WORDS
5567 027762 000000          0              ;FILL WITH 0
5568 027764 004037 041326  JSR      RO,@#CLAREA ;CLEAR 56. WORDS, FROM REINTO+<200.*2>
5569 027770 004354          REINTO+<200.*2> ;STARTING FROM REINTO+<200.*2>
5570 027772 000070          56.           ;56. WORDS
5571 027774 000377          377          ;FILL WITH 377
5572
5573                          ;*FILL WRITE FROM BUFFER WITH 200 OF 52525 AND 56 OF 0

```

```
5574
5575 027776 004037 041326 JSR RO,@#CLAREA ;CLEAR 200. WORDS, FROM WRFROM
5576 030002 002470 WRFROM ;STARTING FROM WRFROM
5577 030004 000310 200. ;200. WORDS
5578 030006 052525 52525 ;FILL WITH 52525
5579 030010 004037 041326 JSR RO,@#CLAREA ;CLEAR 56. WORDS, FROM WRFROM+<200.*2>
5580 030014 003310 WRFROM+<200.*2> ;STARTING FROM WRFROM+<200.*2>
5581 030016 000070 56. ;56. WORDS
5582 030020 000377 377 ;FILL WITH 377
5583
5584 ;*NOW FILL COMMAND
5585
5586 030022 004037 043430 JSR RO,@#RUN ;SETUP TO RUN FOR DATA COMMAND
5587 030026 000000 0 ;CYLINDER 0
5588 030030 000 .BYTE 0 ;SECTOR 0
5589 030031 000 .BYTE 0 ;TRACK 0
5590 030032 177470 -200. ;WORD COUNT = 200.
5591 030034 003534 REINTO ;BUS ADDRESS
5592 030036 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
5593 030040 014000 ECI!FMT22 ;16 BITS PER WORD FORMAT
5594 030042 002446 READAT ;GET READY TO DO A READAT
5595 030044 052777 002000 152226 BIS #PSEL,@RHCS1 ;SET PORT B
5596 ;THAT IS UNIBUS B
5597
5598 ;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ DATA COMMAND
5599
5600 030052 004037 041624 JSR RO,@#SAVER ;SAVE REGISTERS
5601 030056 002272 RHWC ;RHWC IS THE FIRST REGISTER SAVED
5602 030060 004612 SAVERE ;STARTING ADDRESS OF WHERE
5603 030062 000022 18. ;NUMBER OF REGISTERS
5604
5605 030064 004737 041536 JSR PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV - 1
5606 030070 104401 066720 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF ANY OF
5607 030074 000000 HALT ;STOP
5608
5609 030076 013777 004606 152162 MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
5610 030104 013746 002446 MOV @#READAT,-(SP) ;GET READY TO MOVE COMMAND
5611 030110 052716 002101 BIS #GO!IE!PSEL,(SP) ;GET READY TO SET 'GO' AND
5612 030114 012677 152160 MOV (SP)+,@RHCS1 ;GO WITH
5613 030120 011100 MOV @R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
5614 030122 011305 MOV @R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
5615
5616 030124 104413 WAT ;WAIT FOR RDY BIT TO SET
5617 030126 002300 RHCS1 ;WAIT FOR RHCS1 REGISTER
5618 030130 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
5619 030132 001614 908. ;ALLOW 9080 MICRO SECONDS
5620 030134 001507 839. ;RDY MUST SET BETWEEN
5621 030136 013746 002446 MOV @#READAT,-(SP) ;SAVE COMMAND
5622 030142 052716 006101 BIS #IE!GO!DVA!PSEL,(SP) ;INCLUDE IE!GO!DVA!PSEL
5623 030146 011637 001124 MOV (SP),@#SGDDAT ;SAVE FOR PRINTOUT
5624 030152 022600 CMP (SP)+,R0 ;DURING ABOVE OPERATION ONLY IE!GO.DVA!PSEL
5625 030154 001405 BEQ 71$ ;BRANCH IF GOOD
5626 030156 010037 001126 MOV RO,@#SBDDAT ;BAD DATA
5627 030162 010137 004600 MOV R1,@#REGADR ;FAILING REGISTER RHCS1
5628 030166 104021 ERROR 21 ;DURING ABOVE OPERATION ONLY
5629 030170 012746 010500 71$: MOV #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
```



```

5630 030174 011637 001124      MOV      (SP),@#SGDDAT      ;SAVE FOR PRINTOUT
5631 030200 022605              CMP      (SP)+,R5          ;DURING ABOVE OPERATION ONLY MOL.DPR.VV
5632 030202 001405              BEQ      73$              ;BRANCH IF GOOD
5633 030204 010537 001126      MOV      R5,@#SDDAT        ;BAD DATA
5634 030210 010337 004600      MOV      R3,@#REGADR       ;FAILING REGISTER RHDS1
5635 030214 104063              ERROR    63              ;DURING ABOVE OPERATION ONLY
5636
5637
5638
5639 030216 004037 041360      JSR      RO,@#FILLRE       ;MOV 0 INTO SAVED RHWC
5640 030222 002272              RHWC
5641 030224 000000              0                          ;SAVED REGISTER TO CHANGE
5642 030226 004037 041360      JSR      RO,@#FILLRE       ;DATA
5643 030232 002274              RHBA                        ;MOV REINTO+<200.*2> INTO SAVED RHBA
5644 030234 004354              REINTO+<200.*2>           ;SAVED REGISTER TO CHANGE
5645 030236 004037 041360      JSR      RO,@#FILLRE       ;DATA
5646 030242 002304              RHDST                      ;MOV 1 INTO SAVED RHDST
5647 030244 000001              1                          ;SAVED REGISTER TO CHANGE
5648
5649
5650
5651
5652 030246 004037 042444      JSR      RO,@#COMREG       ;COMPARE SAVED REGISTERS WITH
5653 030252 004612              SAVERE                     ;GOOD DATA SAVED IN 'SAVERE'
5654 030254 002354              WC                         ;TEST DATA STARTING FROM 'RHWC'
5655 030256 000022              18.                       ;18. REGISTERS TO BE COMPARED
5656 030260 030264              5$                        ;RETURN TO 5$ ON ERROR
5657 030262 030270              6$                        ;RETURN TO 6$ ON NO ERROR
5658
5659 030264 104072              5$: ERROR 72              ;WHILE USING UNIBUS B
5660
5661 030266 000207              RTS      PC                ;READ DATA CAUSED IMPROPER
5662
5663
5664
5665
5666
5667
5668
5669 030270 004037 043474      JSR      RO,@#COMPAR       ;COMPARE TWO BLOCKS OF MEMORY
5670 030274 002470              WRFROM                     ;GOOD DATA STARTS FROM WRFROM
5671 030276 003534              REINTO                      ;TEST DATA STARTS FROM REINTO
5672 030300 000400              256.                       ;256. WORDS TO BE COMPARED
5673 030302 030306              7$                        ;RETURN TO 7$ ON ERROR
5674 030304 030312              10$                       ;RETURN TO 10$ ON NO ERROR
5675
5676 030306 104073              7$: ERROR 73              ;WHILE USING UNIBUS B
5677
5678 030310 000207              RTS      PC                ;INCORRECT DATA AFTER
5679
5680 030312              10$:                      ;WRITE DATA FOLLOWED BY A
5681
5682
5683
5684
5685

```

CZRJICO, RPO4/5/6 FCTNL CTLR1
CZRJIC.P11 27-JUL-78 13:05

MACY11 30A(1052) 27-JUL-78 13:06 PAGE 128
T40 WRITE/READ DATA USING UNIBUS B

K 10

SEQ 0127

5686

CZF
CZF

```

5687
5688 030312 000004          TST41: SCOPE
5689 030314 012706 001000  MOV    #STACK,SP      ;RESET STACK
5690 030320 012737 000041 004604  MOV    #41,@#TSTNM    ;SAVE TEST NUMBER
5691 030326 004737 041456  JSR    PC,@#CLDISK    ;SET R1-RHCS1, R2-RHCS2
5692
5693          ;*FILL WRITE FROM BUFFER WITH HEADER
5694 030332 004037 041302  JSR    R0,@#FLHEAD    ;SAVE HEADER DATA IN WRFROM
5695 030336 002470          WRFROM    ;LOCATION WHERE SAVED
5696 030340 000004          4          ;NUMBER OF WORDS SAVED
5697 030342 010000          10000     ;FIRST DATA WORD
5698 030344 000000          0          ;SECOND DATA WORD
5699 030346 000000          0          ;THIRD DATA WORD
5700 030350 000000          0          ;FOURTH DATA WORD
5701
5702          ;*FILL WRITE FROM BUFFER WITH DATA
5703 030352 004037 041326  JSR    R0,@#CLAREA    ;CLEAR 256. WORDS, FROM WRFROM+10
5704 030356 002500          WRFROM+10 ;STARTING FROM WRFROM+10
5705 030360 000400          256.      ;256. WORDS
5706 030362 000000          0          ;FILL WITH 0
5707
5708          ;*FILL WRITE FROM BUFFER WITH NEXT SECTOR HEADER
5709 030364 004037 041302  JSR    R0,@#FLHEAD    ;SAVE HEADER DATA IN WRFROM+<260.*2>
5710 030370 003500          WRFROM+<260.*2> ;LOCATION WHERE SAVED
5711 030372 000004          4          ;NUMBER OF WORDS SAVED
5712 030374 010000          10000     ;FIRST DATA WORD
5713 030376 000001          1          ;SECOND DATA WORD
5714 030400 000000          0          ;THIRD DATA WORD
5715 030402 000000          0          ;FOURTH DATA WORD
5716
5717          ;*FILL WRITE FROM BUFFER WITH NEXT SECTOR DATA
5718 030404 004037 041326  JSR    R0,@#CLAREA    ;CLEAR 2 WORDS, FROM WRFROM+<264.*2>
5719 030410 003510          WRFROM+<264.*2> ;STARTING FROM WRFROM+<264.*2>
5720 030412 000002          2          ;2 WORDS
5721 030414 000001          1          ;FILL WITH 1
5722
5723          ;*NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA
5724          ;*AS WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
5725          ;*CAN BE MADE TO MAKE SURE THAT WRITE DID NOT
5726          ;*CHANGE WRITE FROM BUFFER.
5727
5728 030416 004037 041302  JSR    R0,@#FLHEAD    ;SAVE HEADER DATA IN REINTO
5729 030422 003534          REINTO    ;LOCATION WHERE SAVED
5730 030424 000004          4          ;NUMBER OF WORDS SAVED
5731 030426 010000          10000     ;FIRST DATA WORD
5732 030430 000000          0          ;SECOND DATA WORD
5733 030432 000000          0          ;THIRD DATA WORD
5734 030434 000000          0          ;FOURTH DATA WORD
5735 030436 004037 041326  JSR    R0,@#CLAREA    ;CLEAR 256. WORDS, FROM REINTO+10
5736 030442 003544          REINTO+10 ;STARTING FROM REINTO+10
5737 030444 000400          256.      ;256. WORDS
5738 030446 000000          0          ;FILL WITH 0
5739 030450 004037 041302  JSR    R0,@#FLHEAD    ;SAVE HEADER DATA IN REINTO+<260.*2>
5740 030454 004544          REINTO+<260.*2> ;LOCATION WHERE SAVED
5741 030456 000004          4          ;NUMBER OF WORDS SAVED
5742 030460 010000          10000     ;FIRST DATA WORD
  
```

```

5743 030462 000001      1      ;SECOND DATA WORD
5744 030464 000000      0      ;THIRD DATA WORD
5745 030466 000000      0      ;FOURTH DATA WORD
5746 030470 004037 041326 JSR    R0,@#CLAREA ;CLEAR 2 WORDS, FROM REINTO+<264.*2>
5747 030474 004554      REINTO+<264.*2> ;STARTING FROM REINTO+<264.*2>
5748 030476 000002      2      ;2 WORDS
5749 030500 000001      1      ;FILL WITH 1
5750
5751      ;*NOW THE WRITE HEADER AND DATA COMMAND WILL BE FILLED
5752
5753 030502 004037 043430 JSR    R0,@#RUN      ;SETUP TO RUN FOR DATA COMMAND
5754 030506 000000      0      ;CYLINDER 0
5755 030510      000      .BYTE 0 ;SECTOR 0
5756 030511      000      .BYTE 0 ;TRACK 0
5757 030512 177366      -262.-4 ;WORD COUNT (DATA) = 262. +
5758 030514 002470      WRFROM ;BUS ADDRESS
5759 030516 000000      0      ;DO NOT INHIBIT BUS ADDRESS INCREMENT
5760 030520 010000      FMT22 ;16 BITS PER WORD FORMAT
5761 030522 002444      WRIFOR ;GET READY TO DO A WRIFOR
5762
5763      ;*NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE HEADER AND DAT1
5764 030524 004037 041624 JSR    R0,@#SAVER   ;SAVE REGISTERS
5765 030530 002272      RHWC ;RHWC IS THE FIRST REGISTER SAVED
5766 030532 004612      SAVERE ;STARTING ADDRESS OF WHERE
5767 030534 000022      18. ;NUMBER OF REGISTERS
5768
5769 030536 004737 041536 JSR    PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV - 1
5770 030542 104401 066720 TYPE  ,CPHALT ;CANNOT CONTINUE TESTING IF ANY OF
5771 030546 000000      HALT ;STOP
5772
5773 030550 013777 004606 151510 MOV    @#RP4VEC,@#RPVEC ;SET RP04 VECTOR ADDRESS
5774
5775 030556 013746 002444 MOV    @#WRIFOR,-(SP) ;GET READY TO MOVE COMMAND
5776 030562 052716 000101 BIS    #GO!IE,(SP) ;GET READY TO SET 'GO' AND
5777 030566 012677 151506 MOV    (SP)+,@#RHCS1 ;GO WITH
5778 030572 011100      MOV    @R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
5779 030574 011305      MOV    @R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
5780
5781      ;*ONE REVOLUTION = 16670 MICRO SEC, ONE SECTOR = 760 MICRO SEC
5782
5783 030576 104413      WAT ;WAIT FOR RDY BIT TO SET
5784 030600 002300      RHCS1 ;WAIT FOR RHCS1 REGISTER
5785 030602 000200      RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
5786 030604 002001      1025. ;ALLOW 10250 MICRO SECONDS
5787 030606 001553      875. ;RDY MUST SET BETWEEN
5788 030610 013746 002444 MOV    @#WRIFOR,-(SP) ;SAVE COMMAND
5789 030614 052716 004101 BIS    #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
5790 030620 011637 001124 MOV    (SP),@#SGDDAT ;SAVE FOR PRINTOUT
5791 030624 022600      CMP    (SP)+,R0 ;DURING ABOVE OPERATION ONLY IE!GO.DVA
5792 030626 001405      BEQ    64$ ;BRANCH IF GOOD
5793 030630 010037 001126 MOV    R0,@#SBDDAT ;BAD DATA
5794 030634 010137 004600 MOV    R1,@#REGADR ;FAILING REGISTER RHCS1
5795 030640 104021      ERROR 21 ;DURING ABOVE OPERATION ONLY
5796 030642 012746 010500 MOV    #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
5797 030646 011637 001124 MOV    (SP),@#SGDDAT ;SAVE FOR PRINTOUT
5798 030652 022605      CMP    (SP)+,R5 ;DURING ABOVE OPERATION ONLY MOL DPR.VV
  
```

```

5799 030654 001405          BEQ      66$          ;BRANCH IF GOOD
5800 030656 010537 001126    MOV      R5,@#SDDAT  ;BAD DATA
5801 030662 010337 004600    MOV      R3,@#REGADR ;FAILING REGISTER RHDS1
5802 030666 104063          ERROR    63          ;DURING ABOVE OPERATION ONLY
5803
5804          ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
5805
5806 030670 004037 041360    JSR      R0,@#FILLRE ;MOV 0 INTO SAVED RHWC
5807 030674 002272          RHWC          ;SAVED REGISTER TO CHANGE
5808 030676 000000          0            ;DATA
5809 030700 004037 041360    JSR      R0,@#FILLRE ;MOV WRFROM+<266.*2> INTO SAVED RHBA
5810 030704 002274          RHBA          ;SAVED REGISTER TO CHANGE
5811 030706 003514          WRFROM+<266.*2> ;DATA
5812 030710 004037 041360    JSR      R0,@#FILLRE ;MOV 2 INTO SAVED RHDST
5813 030714 002304          RHDST        ;SAVED REGISTER TO CHANGE
5814 030716 000002          2            ;DATA
5815
5816          ;*NOW COMPARE REGISTERS BEFORE WRITE HEADER AND DATA
5817          ;*WITH REGISTERS AFTER COMMAND
5818
5819 030720 004037 042444    JSR      R0,@#COMREG ;COMPARE SAVED REGISTERS WITH
5820 030724 004612          SAVERE        ;GOOD DATA SAVED IN 'SAVERE'
5821 030726 002354          WC           ;TEST DATA STARTING FROM 'RHWC'
5822 030730 000022          18.          ;18. REGISTERS TO BE COMPARED
5823 030732 030736          1$           ;RETURN TO 1$ ON ERROR
5824 030734 030742          2$           ;RETURN TO 2$ ON NO ERROR
5825
5826 030736 104027          1$:          ERROR    27          ;WRITE HEADER AND DATA
5827 030740 000207          RTS         PC          ;CAUSED IMPROPER REGISTER
5828          ;CHANGE
5829          ;GOOD DATA GIVES WHAT SHOULD
5830          ;BE THERE
5831          ;RECEIVED DATA GIVES WHAT
5832          ;WAS THERE AFTER COMMANT
5833
5834          ;*NOW WRITE FROM BUFFER WILL BE CHECKED TO SEE THAT
5835          ;*NOTHING GOT CHANGED
5836
5837 030742          2$:
5838 030742 004037 043474    JSR      R0,@#COMPAR ;COMPARE TWO BLOCKS OF MEMORY
5839 030746 003534          REINTO       ;GOOD DATA STARTS FROM REINTO
5840 030750 002470          WRFROM       ;TEST DATA STARTS FROM WRFROM
5841 030752 000412          266.         ;266. WORDS TO BE COMPARED
5842 030754 030760          3$           ;RETURN TO 3$ ON ERROR
5843 030756 030764          4$           ;RETURN TO 4$ ON NO ERROR
5844
5845 030760 104030          3$:          ERROR    30          ;WRITE HEADER AND DATA
5846 030762 000207          RTS         PC          ;CHANGED WRITE FROM BUFFER
5847
5848          ;*NOW A READ HEADER AND DATA COMMAND WILL BE GIVEN
5849          ;*FOR SECTOR 1, 256 WORDS
5850          ;*READ INTO BUFFER IS FILLED WITH ONES
5851 030764          4$:
5852 030764 004737 041456    JSR      PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
5853 030770 004037 041326    JSR      R0,@#CLAREA ;CLEAR 260. WORDS, FROM REINTO
5854 030774 003534          REINTO       ;STARTING FROM REINTO
    
```

```

5855 030776 000404          260.          :260. WORDS
5856 031000 000377          377          :FILL WITH 377
5857
5858          :*WRITE FROM BUFFER IS FILLED WITH EXPECTED DATA
5859
5860 031002 004037 041302    JSR      RO,@#FLHEAD    :SAVE HEADER DATA IN WRFROM
5861 031006 002470          WRFROM          :LOCATION WHERE SAVED
5862 031010 000006          6              :NUMBER OF WORDS SAVED
5863 031012 010000          10000         :FIRST DATA WORD
5864 031014 000001          1              :SECOND DATA WORD
5865 031016 000000          0              :THIRD DATA WORD
5866 031020 000000          0              :FOURTH DATA WORD
5867 031022 000001          1              :FIFTH DATA WORD
5868 031024 000001          1              :SIXTH DATA WORD
5869 031026 004037 041326    JSR      RO,@#CLAREA    :CLEAR 254 WORDS, FROM WRFROM+<6.*2>
5870 031032 002504          WRFROM+<6.*2>      :STARTING FROM WRFROM+<6.*2>
5871 031034 000254          254           :254 WORDS
5872 031036 000000          0              :FILL WITH 0
5873
5874          :*NOW FILL COMMAND
5875
5876 031040 004037 043430    JSR      RO,@#RUN       :SETUP TO RUN FOR DATA COMMAND
5877 031044 000000          0              :CYLINDER 0
5878 031046 001          .BYTE          :SECTOR 1
5879 031047 000          .BYTE          :TRACK 0
5880 031050 177374          -256.-4       :WORD COUNT (DATA) = 256. +
5881 031052 003534          REINTO        :BUS ADDRESS
5882 031054 000000          0              :DO NOT INHIBIT BUS ADDRESS INCREMENT
5883 031056 014000          ECI!FMT22    :16 BITS PER WORD FORMAT
5884 031060 002450          REFOR        :GET READY TO DO A REFOR
5885
5886          :*NOW SAVE REGISTERS FOR COMPARISON AFTER READ HEADER AND DATA
5887
5888 031062 004037 041624    JSR      RO,@#SAVER    :SAVE REGISTERS
5889 031066 002272          RHWC          :RHWC IS THE FIRST REGISTER SAVED
5890 031070 004612          SAVERE       :STARTING ADDRESS OF WHERE
5891 031072 000022          18.         :NUMBER OF REGISTERS
5892
5893 031074 004737 041536    JSR      PC,@#CHECKT   :CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
5894 031100 104401 066720    TYPE        .CPHALT   :CANNOT CONTINUE TESTING IF ANY OF
5895 031104 000000          HALT        :STOP
5896
5897 031106 013777 004606 151152 MOV      @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
5898
5899 031114 013746 002450    MOV      @#REFOR,-(SP) :GET READY TO MOVE COMMAND
5900 031120 052716 000101    BIS      #GO!IE,(SP)  :GET READY TO SET 'GO' AND
5901 031124 012677 151150    MOV      (SP)+,@RHCS1 :GO WITH
5902 031130 011100          MOV      @R1,R0       :SAVE RHCS1 DURING ABOVE OPERATION
5903 031132 011305          MOV      @R3,R5       :SAVE RHDS1 DURING ABOVE OPERATION
5904
5905          :*ONE REVOLUTION = 16670 MICRO SEC, ONE SECTOR 760 MICRO SECONDS
5906
5907 031134 104413          WAT          :WAIT FOR RDY BIT TO SET
5908 031136 002300          RHCS1       :WAIT FOR RHCS1 REGISTER
5909 031140 000200          RDY         :WAIT FOR RDY BIT IN RHCS1 REGISTER
5910 031142 001614          908.       :ALLOW 9080 MICRO SECONDS
  
```

```

5911 031144 001507      839.      :RDY MUST SET BETWEEN
5912 031146 013746 002450  MOV      @#REFOR,-(SP) :SAVE COMMAND
5913 031152 052716 004101  BIS      #IE!GO!DVA,(SP) :INCLUDE IE!GO!DVA
5914 031156 011637 001124  MOV      (SP),@#SGDDAT :SAVE FOR PRINTOUT
5915 031162 022600      CMP      (SP)+,R0      :DURING ABOVE OPERATION ONLY IE!GO!DVA
5916 031164 001405      BEQ      67$          :BRANCH IF GOOD
5917 031166 010037 001126  MOV      R0,@#SBDDAT  :BAD DATA
5918 031172 010137 004600  MOV      R1,@#REGADR  :FAILING REGISTER RHCS1
5919 031176 104021      ERROR   21          :DURING ABOVE OPERATION ONLY
5920 031200 012746 010500  67$: MOV      #MOL!DPR!VV,-(SP) :SAVE BITS SET DURING OPERATION IN RHDS1
5921 031204 011637 001124  MOV      (SP),@#SGDDAT :SAVE FOR PRINTOUT
5922 031210 022605      CMP      (SP)+,R5      :DURING ABOVE OPERATION ONLY MOL!DPR!VV
5923 031212 001405      BEQ      69$          :BRANCH IF GOOD
5924 031214 010537 001126  MOV      R5,@#SBDDAT  :BAD DATA
5925 031220 010337 004600  MOV      R3,@#REGADR  :FAILING REGISTER RHDS1
5926 031224 104063      ERROR   63          :DURING ABOVE OPERATION ONLY
5927
5928      :*CHANGE SAVED REGISTERS TO EXPECTED VALUES
5929
5930 031226 004037 041360  JSR      R0,@#FILLRE  :MOV 0 INTO SAVED RHWC
5931 031232 002272      RHWC      :SAVED REGISTER TO CHANGE
5932 031234 000000      0          :DATA
5933 031236 004037 041360  JSR      R0,@#FILLRE  :MOV REINTO+<260.*2> INTO SAVED RHBA
5934 031242 002274      RHBA      :SAVED REGISTER TO CHANGE
5935 031244 004544      REINTO+<260.*2> :DATA
5936 031246 004037 041360  JSR      R0,@#FILLRE  :MOV 2 INTO SAVED RHDST
5937 031252 002304      RHDST     :SAVED REGISTER TO CHANGE
5938 031254 000002      2          :DATA
5939
5940      :*COMPARE REGISTERS BEFORE READ HEADER AND DATA
5941      :*WITH REGISTERS AFTER COMMAND
5942
5943 031256 004037 042444  JSR      R0,@#COMREG  :COMPARE SAVED REGISTERS WITH
5944 031262 004612      SAVERE    :GOOD DATA SAVED IN 'SAVERE'
5945 031264 002354      WC        :TEST DATA STARTING FROM 'RHWC'
5946 031266 000022      18.      :18. REGISTERS TO BE COMPARED
5947 031270 031274      5$        :RETURN TO 5$ ON ERROR
5948 031272 031300      6$        :RETURN TO 6$ ON NO ERROR
5949
5950
5951 031274 104031 5$: ERROR 31      :READ HEADER AND DATA CAUSED
5952 031276 000207      RTS      PC        :IMPROPER REGISTER CHANGE
5953      :GOOD DATA GIVES WHAT SHOULD
5954      :BE THERE
5955      :RECEIVED DATA GIVES WHAT WAS
5956      :THERE AFTER COMMAND
5957
5958      :*NOW READ INTO BUFFER WILL BE CHECKED TO SEE
5959      :*THAT READ WAS GOOD
5960
5961 031300      6$:
5962 031300 004037 043474  JSR      R0,@#COMPAR  :COMPARE TWO BLOCKS OF MEMORY
5963 031304 002470      WRFROM    :GOOD DATA STARTS FROM WRFROM
5964 031306 003534      REINTO    :TEST DATA STARTS FROM REINTO
5965 031310 000404      260.     :260. WORDS TO BE COMPARED
5966 031312 031316      7$        :RETURN TO 7$ ON ERROR
    
```

5967	031314	031322		10\$;RETURN TO 10\$ ON NO ERROR
5968											
5969											
5970	031316	104032		7\$:	ERROR	32					:WRITE HEADER AND DATA
5971	031320	000207			RTS	PC					:FOLLOWED BY A READ HEADER
5972											:AND DATA GAVE A READ ERROR
5973											:ERROR MAY BE IN READ OR WRITE
5974	031322			10\$:							


```

5975
5976 031322 000004*          TST42: SCOPE
5977 031324 012706 001000    MOV      #STACK,SP          ;RESET STACK
5978 031330 012737 000042    MOV      #42,@TSTNM        ;SAVE TEST NUMBER
5979 031336 004737 041456    JSR      PC,@CLDISK        ;SET R1-RHCS1, R2-RHCS2
5980
5981                                ;*FILL WRITE FROM BUFFER WITH HEADER
5982 031342 004037 041302    JSR      RO,@#FLHEAD        ;SAVE HEADER DATA IN WRFROM
5983 031346 002470            WRFROM          ;LOCATION WHERE SAVED
5984 031350 000004            4                ;NUMBER OF WORDS SAVED
5985 031352 010000            10000           ;FIRST DATA WORD
5986 031354 011025            <18.*40>.<21.> ;SECOND DATA WORD
5987 031356 000000            0                ;THIRD DATA WORD
5988 031360 000000            0                ;FOURTH DATA WORD
5989
5990                                ;*FILL WRITE FROM BUFFER WITH DATA
5991 031362 004037 041326    JSR      RO,@#CLAREA        ;CLEAR 256. WORDS, FROM WRFROM+10
5992 031366 002500            WRFROM+10        ;STARTING FROM WRFROM+10
5993 031370 000400            256.            ;256. WORDS
5994 031372 001125            <18.*40>!21.    ;FILL WITH <18.*40>!21.
5995
5996                                ;*FILL WRITE FROM BUFFER WITH NEXT TRACK HEADER
5997 031374 004037 041302    JSR      RO,@#FLHEAD        ;SAVE HEADER DATA IN WRFROM+<260.*2>
5998 031400 003500            WRFROM+<260.*2> ;LOCATION WHERE SAVED
5999 031402 000004            4                ;NUMBER OF WORDS SAVED
6000 031404 010001            10001           ;FIRST DATA WORD
6001 031406 000000            0                ;SECOND DATA WORD
6002 031410 000000            0                ;THIRD DATA WORD
6003 031412 000000            0                ;FOURTH DATA WORD
6004
6005                                ;*FILL WRITE FROM BUFFER WITH NEXT TRACK DATA
6006 031414 004037 041326    JSR      RO,@#CLAREA        ;CLEAR 2 WORDS, FROM WRFROM+<264.*2>
6007 031420 003510            WRFROM+<264.*2> ;STARTING FROM WRFROM+<264.*2>
6008 031422 000002            2                ;2 WORDS
6009 031424 002000            2000            ;FILL WITH 2000
6010
6011                                ;*NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA
6012                                ;*AS WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
6013                                ;*CAN BE MADE TO MAKE SURE THAT WRITE DID NOT
6014                                ;*CHANGE WRITE FROM BUFFER.
6015
6016 031426 004037 041302    JSR      RO,@#FLHEAD        ;SAVE HEADER DATA IN REINTO
6017 031432 003534            REINTO          ;LOCATION WHERE SAVED
6018 031434 000004            4                ;NUMBER OF WORDS SAVED
6019 031436 010000            10000           ;FIRST DATA WORD
6020 031440 011025            <18.*400>!<21.> ;SECOND DATA WORD
6021 031442 000000            0                ;THIRD DATA WORD
6022 031444 000000            0                ;FOURTH DATA WORD
6023 031446 004037 041326    JSR      RO,@#CLAREA        ;CLEAR 256. WORDS, FROM REINTO+10
6024 031452 003544            REINTO+10        ;STARTING FROM REINTO+10
6025 031454 000400            256.            ;256. WORDS
6026 031456 001125            <18.*40>!21.    ;FILL WITH <18.*40>!21.
6027 031460 004037 041302    JSR      RO,@#FLHEAD        ;SAVE HEADER DATA IN REINTO+<260.*2>
6028 031464 004544            REINTO+<260.*2> ;LOCATION WHERE SAVED
6029 031466 000004            4                ;NUMBER OF WORDS SAVED
6030 031470 010001            10001           ;FIRST DATA WORD
    
```

```
6031 031472 000000 0 ;SECOND DATA WORD
6032 031474 000000 0 ;THIRD DATA WORD
6033 031476 000000 0 ;FOURTH DATA WORD
6034 031500 004037 041326 JSR RO,@#CLAREA ;CLEAR 2 WORDS, FROM REINTO+<264.*2>
6035 031504 004554 REINTO+<264.*2> ;STARTING FROM REINTO+<264.*2>
6036 031506 000002 2 ;2 WORDS
6037 031510 002000 2000 ;FILL WITH 2000
6038
6039 ;*NOW THE WRITE HEADER AND DATA COMMAND WILL BE FILLED
6040
6041 031512 004037 043430 JSR RO,@#RUN ;SETUP TO RUN FOR DATA COMMAND
6042 031516 000000 0 ;CYLINDER 0
6043 031520 025 .BYTE 21. ;SECTOR 21.
6044 031521 022 .BYTE 18. ;TRACK 18.
6045 031522 177366 -262.-4 ;WORD COUNT (DATA) = 262. +
6046 031524 002470 WRFROM ;BUS ADDRESS
6047 031526 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
6048 031530 010000 FMT22 ;16 BITS PER WORD FORMAT
6049 031532 002444 WRIFOR ;GET READY TO DO A WRIFOR
6050
6051 ;*NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE HEADER AND DATA
6052
6053 031534 004037 041624 JSR RO,@#SAVER ;SAVE REGISTERS
6054 031540 002272 RHWC ;RHWC IS THE FIRST REGISTER SAVED
6055 031542 004612 SAVERE ;STARTING ADDRESS OF WHERE
6056 031544 000022 18. ;NUMBER OF REGISTERS
6057
6058 031546 004737 041536 JSR PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV - 1
6059 031552 104401 066720 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF ANY OF
6060 031556 000000 HALT ;STOP
6061
6062 031560 013777 004606 150500 MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
6063
6064 031566 013746 002444 MOV @#WRIFOR,-(SP) ;GET READY TO MOVE COMMAND
6065 031572 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET 'GO' AND
6066 031576 012677 150476 MOV (SP)+,@RHCS1 ;GO WITH
6067 031602 011100 MOV @R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
6068 031604 011305 MOV @R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
6069
6070 ;*ONE REVOLUTION 16670 MICRO1 SEC, ONE SECTOR = 760 MICRO1 SEC
6071
6072 031606 104413 WAT ;WAIT FOR RDY BIT TO SET
6073 031610 002300 RHCS1 ;WAIT FOR RHCS1 REGISTER
6074 031612 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
6075 031614 003237 1695. ;ALLOW 16950 MICRO SECONDS
6076 031616 001515 845. ;RDY MUST SET BETWEEN
6077 031620 013746 002444 MOV @#WRIFOR,-(SP) ;SAVE COMMAND
6078 031624 052716 004101 BIS #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
6079 031630 011637 001124 MOV (SP)+,@#SGDDAT ;SAVE FOR PRINTOUT
6080 031634 022600 CMP (SP)+,R0 ;DURING ABOVE OPERATION ONLY IE.GO.DVA
6081 031636 001405 BEQ 64$ ;BRANCH IF GOOD
6082 031640 010037 001126 MOV RO,@#SBDDAT ;BAD DATA
6083 031644 010137 004600 MOV R1,@#REGADR ;FAILING REGISTER RHCS1
6084 031650 104021 ERROR 21 ;DURING ABOVE OPERATION ONLY
6085 031652 012746 010500 64$: MOV #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
6086 031656 011637 001124 MOV (SP)+,@#SGDDAT ;SAVE FOR PRINTOUT
```

```
6087 031662 022605          CMP      (SP)+,R5          ;DURING ABOVE OPERATION ONLY MOL.DPR.VV
6088 031664 001405          BEQ      66$              ;BRANCH IF GOOD
6089 031666 010537 001126    MOV      R5,@#SDDAT      ;BAD DATA
6090 031672 010337 004600    MOV      R3,@#REGADR     ;FAILING REGISTER RHDS1
6091 031676 104063          ERROR    63              ;DURING ABOVE OPERATION ONLY
6092
6093                          ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
6094
6095 031700 004037 041360    JSR      R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
6096 031704 002272          RHWC                    ;SAVED REGISTER TO CHANGE
6097 031706 000000          0                       ;DATA
6098 031710 004037 041360    JSR      R0,@#FILLRE     ;MOV WRFROM+<266.*2> INTO SAVED RHBA
6099 031714 002274          RHBA                    ;SAVED REGISTER TO CHANGE
6100 031716 003514          WRFROM+<266.*2>        ;DATA
6101 031720 004037 041360    JSR      R0,@#FILLRE     ;MOV 1 INTO SAVED RHCA
6102 031724 002312          RHCA                    ;SAVED REGISTER TO CHANGE
6103 031726 000001          1                       ;DATA
6104 031730 004037 041360    JSR      R0,@#FILLRE     ;MOV 1 INTO SAVED RHCC
6105 031734 002334          RHCC                    ;SAVED REGISTER TO CHANGE
6106 031736 000001          1                       ;DATA
6107 031740 004037 041360    JSR      R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
6108 031744 002304          RHDST                   ;SAVED REGISTER TO CHANGE
6109 031746 000001          1                       ;DATA
6110
6111                          ;*NOW COMARE REGISTERS BEFORE WRITE HEADER AND DATA
6112                          ;*WITH REGISTERS AFTER COMMAND
6113
6114 031750 004037 042444    JSR      R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
6115 031754 004612          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
6116 031756 002354          WC                     ;TEST DATA STARTING FROM 'RHWC'
6117 031760 000022          18.                    ;18. REGISTERS TO BE COMPARED
6118 031762 031766          1$                      ;RETURN TO 1$ ON ERROR
6119 031764 031772          2$                      ;RETURN TO 2$ ON NO ERROR
6120
6121 031766 104027 1$:      ERROR    27          ;WRITE HEADER AND DATA
6122 031770 000207          RTS      PC             ;CAUSED IMPROPER REGISTER
6123                          ;CHANGE
6124                          ;GOOD DATA GIVES WHAT SHOULD
6125                          ;BE THERE
6126                          ;RECEIVED DATA GIVES WHAT
6127                          ;WAS THERE AFTER COMMANT
6128
6129                          ;*NOW WRITE FROM BUFFER WILL BE CHECKED TO SEE THAT
6130                          ;*NOTHING GOT CHANGED
6131
6132 031772 004037 043474    2$:      JSR      R0,@#COMPAR    ;COMPARE TWO BLOCKS OF MEMORY
6133 031772 003534          REINTO                  ;GOOD DATA STARTS FROM REINTO
6134 031776 002470          WRFROM                  ;TEST DATA STARTS FROM WRFROM
6135 032000 000412          266.                    ;266. WORDS TO BE COMPARED
6136 032002 000412          3$                      ;RETURN TO 3$ ON ERROR
6137 032004 032010          4$                      ;RETURN TO 4$ ON NO ERROR
6138 032006 032014
6139
6140 032010 104030 3$:      ERROR    30          ;WRITE HEADER AND DATA
6141 032012 000207          RTS      PC             ;CHANGED WRITE FROM BUFFER
6142
```



```
6199 032166 002300          RHCS1          ;WAIT FOR RHCS1 REGISTER
6200 032170 000200          RDY           ;WAIT FOR RDY BIT IN RHCS1 REGISTER
6201 032172 001614          908.         ;ALLOW 9080 MICRO SECONDS
6202 032174 001507          839.         ;RDY MUST SET BETWEEN
6203 032176 013746 002450  MOV @#REFOR,-(SP) ;SAVE COMMAND
6204 032202 052716 004101  BIS #IE.GO!DVA,(SP) ;INCLUDE IE!GO!DVA
6205 032206 011637 001124  MOV (SP),@#SGDDAT ;SAVE FOR PRINTOUT
6206 032212 022600          CMP (SP)+,R0  ;DURING ABOVE OPERATION ONLY IE.GO!DVA
6207 032214 001405          BEQ 67$      ;BRANCH IF GOOD
6208 032216 010037 001126  MOV R0,@#SBDDAT ;BAD DATA
6209 032222 010137 004600  MOV R1,@#REGADR ;FAILING REGISTER RHCS1
6210 032226 104021          ERROR 21      ;DURING ABOVE OPERATION ONLY
6211 032230 012746 010500  67$: MOV #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
6212 032234 011637 001124  MOV (SP),@#SGDDAT ;SAVE FOR PRINTOUT
6213 032240 022605          CMP (SP)+,R5  ;DURING ABOVE OPERATION ONLY MOL.DPR VV
6214 032242 001405          BEQ 69$      ;BRANCH IF GOOD
6215 032244 010537 001126  MOV R5,@#SBDDAT ;BAD DATA
6216 032250 010337 004600  MOV R3,@#REGADR ;FAILING REGISTER RHDS1
6217 032254 104063          ERROR 63      ;DURING ABOVE OPERATION ONLY
6218
6219          ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
6220 032256 004037 041360  JSR R0,@#FILLRE ;MOV 0 INTO SAVED RHWC
6221 032262 002272          RHWC         ;SAVED REGISTER TO CHANGE
6222 032264 000000          0           ;DATA
6223 032266 004037 041360  JSR R0,@#FILLRE ;MOV REINTO+<260.*2> INTO SAVED RHBA
6224 032272 002274          RHBA         ;SAVED REGISTER TO CHANGE
6225 032274 004544          REINTO+<260.*2> ;DATA
6226 032276 004037 041360  JSR R0,@#FILLRE ;MOV 1 INTO SAVED RHDST
6227 032302 002304          RHDST       ;SAVED REGISTER TO CHANGE
6228 032304 000001          1           ;DATA
6229
6230          ;*COMPARE REGISTERS BEFORE READ HEADER AND DATA
6231          ;*WITH REGISTERS AFTER COMMAND
6232 032306 004037 042444  JSR R0,@#COMREG ;COMPARE SAVED REGISTERS WITH
6233 032312 004612          SAVERE       ;GOOD DATA SAVED IN 'SAVERE'
6234 032314 002354          WC        ;TEST DATA STARTING FROM 'RHWC'
6235 032316 000022          18.        ;18. REGISTERS TO BE COMPARED
6236 032320 032324          5$         ;RETURN TO 5$ ON ERROR
6237 032322 032330          6$         ;RETURN TO 6$ ON NO ERROR
6238
6239
6240 032324 104031 5$: ERROR 31      ;READ HEADER AND DATA CAUSED
6241 032326 000207          RTS PC      ;IMPROPER REGISTER CHANGE
6242          ;GOOD DATA GIVES WHAT SHOULD
6243          ;BE THERE
6244          ;RECEIVED DATA GIVES WHAT WAS
6245          ;THERE AFTER COMMAND
6246
6247          ;*NOW READ INTO BUFFER WILL BE CHECKED TO SEE
6248          ;*THAT READ WAS GOOD
6249
6250 032330          6$:
6251 032330 004037 043474  JSR R0,@#COMPAR ;COMPARE TWO BLOCKS OF MEMORY
6252 032334 002470          WRFROM     ;GOOD DATA STARTS FROM WR.ROM
6253 032336 003534          REINTO     ;TEST DATA STARTS FROM REINTO
6254 032340 000404          260.     ;260. WORDS TO BE COMPARED
```

6255 032342 032346
6256 032344 032352
6257
6258
6259 032346 104032
6260 032350 000207
6261
6262
6263 032352
6264

7\$
10\$

7\$: ERROR 32
RTS PC

10\$:

:RETURN TO 7\$ ON ERROR
:RETURN TO 10\$ ON NO ERROR

:WRITE HEADER AND DATA
:FOLLOWED BY A READ HEADER
:AND DATA GAVE A READ ERROR
:ERROR MAY BE IN READ OR WRITE

```

6265
6266 032352 000004          TST43: SCOPE
6267 032354 012706 001000  MOV      #STACK,SP      ;RESET STACK
6268 032360 012737 000043 004604  MOV      #43,@TSTNM     ;SAVE TEST NUMBER
6269 032366 004737 041456  JSR      PC,@CLDISK     ;SET R1-RHCS1, R2-RHCS2
6270
6271                               ;*THE FOLLOWING MOVES ARE TO INITIALIZE TEST FROM
6272                               ;*CYLINDER 0
6273                               ;*THESE LOCATIONS ARE CHANGED DURING TEST TO ENABLE
6274                               ;*GOING TO NEXT CYLINDER
6275
6276 032372 012737 010000 032550  MOV      #10000,@ST1+10
6277 032400 012737 001125 032570  MOV      #<<18.*40>!21.>,@ST2+10
6278 032406 012737 010001 032602  MOV      #10001,@ST3+10
6279 032414 012737 002000 032622  MOV      #2000,@ST4+10
6280 032422 012737 000000 032630  MOV      #0,@ST5+4
6281 032430 012737 000001 033040  MOV      #1,@ST6+6
6282 032436 012737 000001 033050  MOV      #1,@ST6+16
6283 032444 012737 010001 033132  MOV      #10001,@ST9+10
6284 032452 012737 002000 033152  MOV      #2000,@ST10+10
6285 032460 012737 000001 033172  MOV      #1,@ST11+4
6286
6287                               ;*THIS IS TO GET THE HEADS TO CYLINDER 0
6288
6289 032466 013777 004606 147572  MOV      @RPP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
6290 032474 013746 002426      MOV      @RRECALI,-(SP) ;GET READY TO MOVE COMMAND
6291 032500 052716 000101      BIS      #GO!IE,(SP)    ;GET READY TO SET 'GO' AND
6292 032504 012677 147570      MOV      (SP)+,@RHCS1  ;GO WITH
6293 032510 011100      MOV      @R1,R0        ;SAVE RHCS1 DURING ABOVE OPERATION
6294 032512 011305      MOV      @R3,R5        ;SAVE RHDS1 DURING ABOVE OPERATION
6295 032514 104413      WAT                    ;WAIT FOR DRY BIT TO SET
6296 032516 002322      RHDS1                 ;WAIT FOR RHDS1 REGISTER
6297 032520 000200      DRY                    ;WAIT FOR DRY BIT IN RHDS1 REGISTER
6298 032522 076377      31999.                 ;ALLOW 319990 MICRO SECONDS
6299 032524 056701      24001.                 ;DRY MUST SET BETWEEN
6300 032526 012737 000012 001200  MOV      #10,@STMP1     ;TEN COUNT TO GET TO CYLINDER 10
6301 032534 004737 041456  JSR      PC,@CLDISK     ;SET R1-RHCS1, R2-RHCS2
6302
6303                               ;*FILL WRITE FROM BUFFER WITH HEADER
6304 032540          ST1:
6305 032540 004037 041302  JSR      R0,@FLHEAD     ;SAVE HEADER DATA IN WRFROM
6306 032544 002470      WRFROM                 ;LOCATION WHERE SAVED
6307 032546 000004      4                       ;NUMBER OF WORDS SAVED
6308 032550 010000      10000                   ;FIRST DATA WORD
6309 032552 011025      <18.*400>.21.           ;SECOND DATA WORD
6310 032554 000000      0                       ;THIRD DATA WORD
6311 032556 000000      0                       ;FOURTH DATA WORD
6312
6313                               ;*FILL WRITE FROM BUFFER WITH DATA
6314 032560          ST2:
6315 032560 004037 041326  JSR      R0,@CLAREA     ;CLEAR 256. WORDS, FROM WRFROM+10
6316 032564 002500      WRFROM+10               ;STARTING FROM WRFROM+10
6317 032566 000400      256.                   ;256. WORDS
6318 032570 001125      <0.*2000>!<18.*40>.21. ;FILL WITH <0.*2000>.<18.*40>!21.
6319
6320                               ;*FILL WRITE FROM BUFFER WITH NEXT TRACK HEADER
    
```

```

6321 032572
6322 032572 004037 041302
6323 032576 003500
6324 032600 000004
6325 032602 010001
6326 032604 000000
6327 032606 000000
6328 032610 000000
6329
6330
6331 032612
6332 032612 004037 041326
6333 032616 003510
6334 032620 000002
6335 032622 002000
6336
6337
6338 032624
6339 032624 004037 043430
6340 032630 000000
6341 032632 025
6342 032633 022
6343 032634 177366
6344 032636 002470
6345 032640 000000
6346 032642 010000
6347 032644 002444
6348
6349
6350
6351 032646 004037 041624
6352 032652 002272
6353 032654 004612
6354 032656 000022
6355
6356 032660 004737 041536
6357 032664 104401 066720
6358 032670 000000
6359
6360 032672 013777 004606 147366
6361
6362 032700 013746 002444
6363 032704 052716 000101
6364 032710 012677 147364
6365 032714 011100
6366 032716 011305
6367
6368
6369
6370
6371
6372 032720 104413
6373 032722 002300
6374 032724 000200
6375 032726 003237
6376 032730 001515

ST3:
JSR RO,@#FLHEAD ;SAVE HEADER DATA IN WRFROM+<260.*2>
WRFROM+<260.*2> ;LOCATION WHERE SAVED
4 ;NUMBER OF WORDS SAVED
10001 ;FIRST DATA WORD
0 ;SECOND DATA WORD
0 ;THIRD DATA WORD
0 ;FOURTH DATA WORD

;*FILL WRITE FROM BUFFER WITH NEXT TRACK DATA

ST4:
JSR RO,@#CLAREA ;CLEAR 2 WORDS, FROM WRFROM+<264.*2>
WRFROM+<264.*2> ;STARTING FROM WRFROM+<264.*2>
2 ;2 WORDS
1.*2000 ;FILL WITH 1.*2000

;*THE WRITE HEADER AND DATA COMMAND WILL BE FILLED

ST5:
JSR RO,@#RUN ;SETUP TO RUN FOR DATA COMMAND
0 ;CYLINDER 0
.BYTE 21. ;SECTOR 21.
.BYTE 18. ;TRACK 18.
-262.-4 ;WORD COUNT (DATA) = 262. +
WRFROM ;BUS ADDRESS
0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
FMT22 ;16 BITS PER WORD FORMAT
WRIFOR ;GET READY TO DO A WRIFOR

;*SAVE REGISTERS FOR COMPARISON AFTER WRITE HEADER AND DATA

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

JSR PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV - 1
TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF ANY OF
HALT ;STOP

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

MOV @#WRIFOR,-(SP) ;GET READY TO MOVE COMMAND
BIS #GO!IE,(SP) ;GET READY TO SET 'GO' AND
MOV (SP)+,@RHCS1 ;GO WITH
MOV @R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
MOV @R3,R5 ;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

WAT ;WAIT FOR RDY BIT TO SET
RHCS1 ;WAIT FOR RHCS1 REGISTER
RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
1695. ;ALLOW 16950 MICRO SECONDS
845. ;RDY MUST SET BETWEEN
  
```



```

6377 032732 013746 002444      MOV    @#WRIFOR,-(SP) ;SAVE COMMAND
6378 032736 052716 004101      BIS    #IE!GO.DVA,(SP) ;INCLUDE IE!GO.DVA
6379 032742 011637 001124      MOV    (SP),@#SGDDAT ;SAVE FOR PRINTOUT
6380 032746 022600                CMP    (SP)+,R0 ;DURING ABOVE OPERATION ONLY IE.GO.DVA
6381 032750 001405                BEQ    64$ ;BRANCH IF GOOD
6382 032752 010037 001126      MOV    R0,@#SBDDAT ;BAD DATA
6383 032756 010137 004600      MOV    R1,@#REGADR ;FAILING REGISTER RHCS1
6384 032762 104021                ERROR  21 ;DURING ABOVE OPERATION ONLY
6385 032764 012746 010500      64$: MOV    #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
6386 032770 011637 001124      MOV    (SP),@#SGDDAT ;SAVE FOR PRINTOUT
6387 032774 022605                CMP    (SP)+,R5 ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
6388 032776 001405                BEQ    66$ ;BRANCH IF GOOD
6389 033000 010537 001126      MOV    R5,@#SBDDAT ;BAD DATA
6390 033004 010337 004600      MOV    R3,@#REGADR ;FAILING REGISTER RHDS1
6391 033010 104063                ERROR  63 ;DURING ABOVE OPERATION ONLY
6392
6393                ;*NOW CHANGES SAVED REGISTERS TO EXPECTED VALUES
6394
6395 033012 004037 041360      JSR    R0,@#FILLRE ;MOV 0 INTO SAVED RHWC
6396 033016 002272                RHWC ;SAVED REGISTER TO CHANGE
6397 033020 000000                0 ;DATA
6398 033022 004037 041360      JSR    R0,@#FILLRE ;MOV WRFROM+<266.*2> INTO SAVED RHBA
6399 033026 002274                RHBA ;SAVED REGISTER TO CHANGE
6400 033030 003514                WRFROM+<266.*2> ;DATA
6401 033032
6402 033032 004037 041360      ST6: JSR    R0,@#FILLRE ;MOV 1 INTO SAVED RHCC
6403 033036 002334                RHCC ;SAVED REGISTER TO CHANGE
6404 033040 000001                1 ;DATA
6405 033042 004037 041360      JSR    R0,@#FILLRE ;MOV 1 INTO SAVED RHCA
6406 033046 002312                RHCA ;SAVED REGISTER TO CHANGE
6407 033050 000001                1 ;DATA
6408 033052 004037 041360      JSR    R0,@#FILLRE ;MOV 1 INTO SAVED RHDST
6409 033056 002304                RHDST ;SAVED REGISTER TO CHANGE
6410 033060 000001                1 ;DATA
6411
6412                ;*COMPARE REGISTERS BEFORE WRITE HEADER AND DATA
6413                ;*WITH REGISTERS AFTER COMMAND
6414
6415 033062 004037 042444      JSR    R0,@#COMREG ;COMPARE SAVED REGISTERS WITH
6416 033066 004612                SAVERE ;GOOD DATA SAVED IN 'SAVERE'
6417 033070 002354                WC ;TEST DATA STARTING FROM 'RHWC'
6418 033072 000022                18. ;18. REGISTERS TO BE COMPARED
6419 033074 033100                ST7 ;RETURN TO ST7 ON ERROR
6420 033076 033104                ST8 ;RETURN TO ST8 ON NO ERROR
6421 033100 104027      ST7: ERROR  27 ;WRITE HEADER AND DATA CAUSED
6422 033102 000207                RTS    PC ;IMPROPER REGISTER CHANGE
6423                ;GOOD DATA GIVES WHAT SHOULD BE
6424                ;THERE
6425                ;RECEIVED DATA GIVES WHAT WAS BE
6426                ;THERE AFTER COMMAND
6427
6428                ;*SETUP TO READ HEADER AND DATA FOR NEXT TRACK
6429                ;*FILL READ INTO BUFFER WITH ALL ONES
6430
6431 033104
6432 033104 004737 041456      ST8: JSR    PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2

```

```

6433 033110 004037 041326 JSR RO,@#CLAREA ;CLEAR 260. WORDS, FROM REINTO
6434 033114 003534 REINTO ;STARTING FROM REINTO
6435 033116 000404 260. ;260. WORDS
6436 033120 177777 -1 ;FILL WITH -1
6437
6438 ;*FILL WRITE FROM BUFFER WITH EXPECTED DATA
6439
6440 033122 ST9: JSR RO,@#FLHEAD ;SAVE HEADER DATA IN WRFROM
6441 033122 004037 041302 WRFROM ;LOCATION WHERE SAVED
6442 033126 002470 4 ;NUMBER OF WORDS SAVED
6443 033130 000004 10001 ;FIRST DATA WORD
6444 033132 010001 0 ;SECOND DATA WORD
6445 033134 000000 0 ;THIRD DATA WORD
6446 033136 000000 0 ;FOURTH DATA WORD
6447 033140 000000
6448 033142 ST10: JSR RO,@#CLAREA ;CLEAR 2 WORDS, FROM WRFROM+<4*2>
6449 033142 004037 041326 WRFROM+<4*2> ;STARTING FROM WRFROM+<4*2>
6450 033146 002500 2 ;2 WORDS
6451 033150 000002 1*2000 ;FILL WITH 1*2000
6452 033152 002000 JSR RO,@#CLAREA ;CLEAR 254. WORDS, FROM WRFROM+<6*2>
6453 033154 004037 041326 WRFROM+<6*2> ;STARTING FROM WRFROM+<6*2>
6454 033160 002504 254. ;254. WORDS
6455 033162 000376 0 ;FILL WITH 0
6456 033164 000000
6457
6458 ;*FILL COMMAND INTO REGISTERS
6459 033166 ST11: JSR RO,@#RUN ;SETUP TO RUN FOR DATA COMMAND
6460 033166 004037 043430 1 ;CYLINDER 1
6461 033172 000001 .BYTE 0 ;SECTOR 0
6462 033174 000 .BYTE 0 ;TRACK 0
6463 033175 000 -256.-4 ;WORD COUNT (DATA) = 256. +
6464 033176 177374 REINTO ;BUS ADDRESS
6465 033200 003534 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
6466 033202 000000 ECI!FMT22 ;16 BITS PER WORD FORMAT
6467 033204 014000 REFOR ;GET READY TO DO A REFOR
6468 033206 002450
6469
6470 ;*SAVE REGISTERS FOR COMPARISON AFTER READ HEADER
6471 ;*AND DATA
6472
6473 033210 004037 041624 JSR RO,@#SAVER ;SAVE REGISTERS
6474 033214 002272 RHWC ;RHWC IS THE FIRST REGISTER SAVED
6475 033216 004612 SAVERE ;STARTING ADDRESS OF WHERE
6476 033220 000020 16. ;NUMBER OF REGISTERS
6477
6478 033222 004737 041536 JSR PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
6479 033226 104401 066720 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF ANY OF
6480 033232 000000 HALT ;STOP
6481
6482 033234 013777 004606 147024 MOV @#RP4VEC,@#RPVEC ;SET RP04 VECTOR ADDRESS
6483
6484 033242 013746 002450 MOV @#REFOR,-(SP) ;GET READY TO MOVE COMMAND
6485 033246 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET 'GO' AND
6486 033252 012677 147022 MOV (SP)+,@#RHCS1 ;GO WITH
6487 033256 011100 MOV @R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
6488 033260 011305 MOV @R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
  
```

```

6489
6490 033262 104413          WAT          :WAIT FOR RDY BIT TO SET
6491 033264 002300          RHCS1        :WAIT FOR RHCS1 REGISTER
6492 033266 000200          RDY          :WAIT FOR RDY BIT IN RHCS1 REGISTER
6493 033270 001614          908.        :ALLOW 9080 MICRO SECONDS
6494 033272 001507          839.        :RDY MUST SET BETWEEN
6495 033274 013746 002450    MOV @#REFOR,-(SP) :SAVE COMMAND
6496 033300 052716 004101    BIS #IE!GO!DVA,(SP) :INCLUDE IE!GO!DVA
6497 033304 011637 001124    MOV (SP),@#SGDDAT :SAVE FOR PRINTOUT
6498 033310 022600          CMP (SP)+,R0 :DURING ABOVE OPERATION ONLY IE.GO!DVA
6499 033312 001405          BEQ 64$     :BRANCH IF GOOD
6500 033314 010037 001126    MOV R0,@#SBDDAT :BAD DATA
6501 033320 010137 004600    MOV R1,@#REGADR :FAILING REGISTER RHCS1
6502 033324 104021          ERROR 21    :DURING ABOVE OPERATION ONLY
6503 033326 012746 010500    64$: MOV #MOL!DPR!VV,-(SP) :SAVE BITS SET DURING OPERATION IN RHDS1
6504 033332 011637 001124    MOV (SP),@#SGDDAT :SAVE FOR PRINTOUT
6505 033336 022605          CMP (SP)+,R5 :DURING ABOVE OPERATION ONLY MOL!DPR!VV
6506 033340 001405          BEQ 66$     :BRANCH IF GOOD
6507 033342 010537 001126    MOV R5,@#SBDDAT :BAD DATA
6508 033346 010337 004600    MOV R3,@#REGADR :FAILING REGISTER RHDS1
6509 033352 104063          ERROR 63    :DURING ABOVE OPERATION ONLY
6510
6511
6512
6513 033354 004037 041360    JSR R0,@#FILLRE :MOV 0 INTO SAVED RHWC
6514 033360 002272          RHWC        :SAVED REGISTER TO CHANGE
6515 033362 000000          0           :DATA
6516 033364 004037 041360    JSR R0,@#FILLRE :MOV REINTO+<260.*2> INTO SAVED RHBA
6517 033370 002274          RHBA        :SAVED REGISTER TO CHANGE
6518 033372 004544          REINTO+<260.*2> :DATA
6519 033374 004037 041360    JSR R0,@#FILLRE :MOV 1 INTO SAVED RHDST
6520 033400 002304          RHDST       :SAVED REGISTER TO CHANGE
6521 033402 000001          1           :DATA
6522
6523
6524
6525
6526 033404 004037 042444    JSR R0,@#COMREG :COMPARE SAVED REGISTERS WITH
6527 033410 004612          SAVERE      :GOOD DATA SAVED IN 'SAVERE'
6528 033412 002354          WC         :TEST DATA STARTING FROM 'RHWC'
6529 033414 000022          18.        :18. REGISTERS TO BE COMPARED
6530 033416 033422          ST12       :RETURN TO ST12 ON ERROR
6531 033420 033426          ST13       :RETURN TO ST13 ON NO ERROR
6532
6533 033422 104031          ST12: ERROR 31 :READ HEADER AND DATA CAUSED
6534 033424 000207          RTS PC     :IMPROPER REGISTER CHANGE
6535
6536
6537
6538
6539
6540
6541 033426          ST13:
6542 033426 004037 043474    JSR R0,@#COMPAR :COMPARE TWO BLOCKS OF MEMORY
6543 033432 002470          WRFROM     :GOOD DATA STARTS FROM WRFROM
6544 033434 003534          REINTO     :TEST DATA STARTS FROM REINTO

```

```
6545 033436 000404          260.          :260. WORDS TO BE COMPARED
6546 033440 033444          ST14          :RETURN TO ST14 ON ERROR
6547 033442 033450          ST15          :RETURN TO ST15 ON NO ERROR
6548
6549 033444 104032          ST14: ERROR 32 :WRITE HEADER AND DATA
6550 033446 000207          RTS PC        :WITH AN IMPLIED SEEK
6551                                     :FOLLOWED BY A READ
6552                                     :HEADER AND DATA ON THE
6553                                     :NEXT TRACK GAVE A
6554                                     :READ ERROR
6555                                     :ERROR MAY BE READ OR WRITE
6556
6557                                     :*THE HEADS HAVE ADVANCED ONE CYLINDER BY AN IMPLIED
6558                                     :*SEEK
6559                                     :*CHANGES WILL BE MADE TO ENABLE GOING TO THE NEXT
6560                                     :*CYLINDER AND THEN THE ABOVE WILL BE REPEATED
6561                                     :*TILL CYLINDER 10 IS REACHED
6562
6563 033450 005237 032550          ST15: INC @#ST1+10
6564 033454 062737 002000 032570 ADD #<1.*2000>,@#ST2+10
6565 033462 005237 032602          INC @#ST3+10
6566 033466 062737 002000 032622 ADD #<1.*2000>,@#ST4+10
6567 033474 005237 032630          INC @#ST5+4
6568 033500 005237 033040          INC @#ST6+6
6569 033504 005237 033050          INC @#ST6+16
6570 033510 005237 033132          INC @#ST9+10
6571 033514 062737 002000 033152 ADD #<1.*2000>,@#ST10+10
6572 033522 005237 033172          INC @#ST11+4
6573 033526 005337 001200          DEC @#STMP1    :COUNT FOR TEN TIMES
6574 033532 001001          BNE ST16      :BRANCH IF 10 NOT DONE
6575 033534 000402          BR ST17       :10 COMPLETED SO CONTINUE
6576 033536 000137 032540          ST16: JMP ST1   :JUMP AS 10 NOT DONE
6577
6578                                     :*THE HEADS ARE NOW AT CYLINDER 10
6579                                     :*ALL REGISTERS WILL BE SAVED AND A SEEK WILL BE GIVEN
6580                                     :*TO CYLINDER 0
6581                                     :*FILL REGISTERS FOR A SEEK COMMAND
6582
6583 033542          ST17:
6584 033542 004737 041456          JSR PC,@#CLDISK :SET R1-RHCS1, R2-RHCS2
6585 033546 004037 041426          JSR R0,@#SEEKCY :SEEK FOR
6586 033552 000000          0             :CYLINDER 0
6587
6588                                     :*SAVE REGISTERS FOR COMPARISON AFTER SEEK COMMAND
6589 033554 004037 041624          JSR R0,@#SAVER  :SAVE REGISTERS
6590 033560 002272          RHWC          :RHWC IS THE FIRST REGISTER SAVED
6591 033562 004612          SAVERE        :STARTING ADDRESS OF WHERE
6592 033564 000022          18.          :NUMBER OF REGISTERS
6593
6594 033566 013777 004606 146472 MOV @#RP4VEC,@#RPVEC ;SET RPO4 VECTOR ADDRESS
6595
6596 033574 013746 002452          MOV @#SEECOM,-(SP) :GET READY TO MOVE COMMAND
6597 033600 052716 000101          BIS #GO!IE,(SP)  :GET READY TO SET 'GO' AND
6598 033604 012677 146470          MOV (SP)+,@#RHCS1 :GO WITH
6599 033610 011100          MOV @R1,R0      :SAVE RHCS1 DURING ABOVE OPERATION
6600 033612 011305          MOV @R3,R5      :SAVE RHDS1 DURING ABOVE OPERATION
```

```

6601
6602
6603 033614 104413
6604 033616 002322
6605 033620 000200
6606 033622 015530
6607 033624 000043
6608 033626 013746 002452
6609 033632 052716 004301
6610 033636 011637 001124
6611 033642 022600
6612 033644 001405
6613 033646 010037 001126
6614 033652 010137 004600
6615 033656 104021
6616 033660 012746 030500
6617 033664 011637 001124
6618 033670 022605
6619 033672 001405
6620 033674 010537 001126
6621 033700 010337 004600
6622 033704 104063
6623
6624
6625
6626 033706 004037 041360
6627 033712 002334
6628 033714 000000
6629 033716 053737 004740 004636
6630 033724 004037 042336
6631 033730 002322
6632 033732 000001
6633 033734 000001
6634 033736 100000
6635 033740 004037 042336
6636 033744 002300
6637 033746 000001
6638 033750 000001
6639 033752 100000
6640
6641
6642 033754 004037 042444
6643 033760 004612
6644 033762 002354
6645 033764 000022
6646 033766 033772
6647 033770 033776
6648
6649 033772 104037
6650 033774 000207
6651
6652
6653
6654
6655
6656

;*SEEK FOR ONE CYLINDER=7MILI SEC., FOR TEN=70 MILI SEC
WAT ;WAIT FOR DRY BIT TO SET
RHDS1 ;WAIT FOR RHDS1 REGISTER
DRY ;WAIT FOR DRY BIT IN RHDS1 REGISTER
7000. ;ALLOW 7000 MICRO SECONDS
35. ;DRY MUST SET BETWEEN
MOV @#SEECOM,-(SP) ;SAVE COMMAND
BIS #DVA!GO!IE!RDY,(SP) ;INCLUDE DVA!GO!IE!RDY
MOV (SP),@#SGDDAT ;SAVE FOR PRINTOUT
CMP (SP)+,R0 ;DURING ABOVE OPERATION ONLY DVA!GO.IE.RDY
BEQ 64$ ;BRANCH IF GOOD
MOV R0,@#SBDDAT ;BAD DATA
MOV R1,@#REGADR ;FAILING REGISTER RHCS.
ERROR 21 ;DURING ABOVE OPERATION ONLY
MOV #PIP!MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
MOV (SP),@#SGDDAT ;SAVE FOR PRINTOUT
CMP (SP)+,R5 ;DURING ABOVE OPERATION ONLY PIP.MOL DPR VV
BEQ 66$ ;BRANCH IF GOOD
MOV R5,@#SBDDAT ;BAD DATA
MOV R3,@#REGADR ;FAILING REGISTER RHDS1
ERROR 63 ;DURING ABOVE OPERATION ONLY

;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
JSR R0,@#FILLRE ;MOV 0 INTO SAVED RHCC
RHCC ;SAVED REGISTER TO CHANGE
0 ;DATA
BIS @#ATTENT,@#SAVERE+24 ;SET APPROPRIATE 'ATA' BITS
JSR R0,@#CHREG ;CHANGE BITS IN SAVED REGISTER
RHDS1 ;CHANGE RHDS1 REGISTER
1 ;1 BIT/BITS TO BE CHANGED
1 ;NEW VALUE OF ATA IS 1
ATA ;CHANGE ATA BIT
JSR R0,@#CHREG ;CHANGE BITS IN SAVED REGISTER
RHCS1 ;CHANGE RHCS1 REGISTER
1 ;1 BIT/BITS TO BE CHANGED
1 ;NEW VALUE OF SC IS 1
SC ;CHANGE SC BIT

;*COMPARE REGISTERS AFTER A SEEK COMMAND
JSR R0,@#COMREG ;COMPARE SAVED REGISTERS WITH
SAVERE ;GOOD DATA SAVED IN 'SAVERE'
WC ;TEST DATA STARTING FROM 'RHWC'
18. ;18. REGISTERS TO BE COMPARED
ST18 ;RETURN TO ST18 ON ERROR
ST19 ;RETURN TO ST19 ON NO ERROR

ST18: ERROR 37 ;SEEK COMMAND CAUSED AN
RTS PC ;ERROR
;GOOD DATA GIVES WHAT SHOULD
;BE THERE
;RECEIVED DATA GIVES WHAT WAS
;THERE AFTER A SEEK COMMAND

;*AT THIS POINT THE CURRENT CYLINDER IS GOOD AND THERE ARE

```

```
6657 :*NO ERROR BITS
6658 :*A READ HEADER AND DATA WILL BE DONE ON CYLINDER 0
6659 :*SECTOR 21 TRACK 18, EXPECTED DATA IS 1125
6660 :*FOR 10 WORDS
6661 :*CLEAR READ INTO BUFFER WITH ALL ONES
6662
6663 033776 ST19:
6664 033776 004037 041326 JSR R0,@#CLAREA ;CLEAR 260. WORDS, FROM REINTO
6665 034002 003534 REINTO ;STARTING FROM REINTO
6666 034004 000404 260. ;260. WORDS
6667 034006 177777 -1 ;FILL WITH -1
6668
6669 :*FILL WRITE FROM BUFFER WITH EXPECTED DATA
6670 034010 004037 041302 JSR R0,@#FLHEAD ;SAVE HEADER DATA IN WRFROM
6671 034014 002470 WRFROM ;LOCATION WHERE SAVED
6672 034016 000004 4 ;NUMBER OF WORDS SAVED
6673 034020 010000 10000 ;FIRST DATA WORD
6674 034022 011025 <18.*40>!<21.> ;SECOND DATA WORD
6675 034024 000000 0 ;THIRD DATA WORD
6676 034026 000000 0 ;FOURTH DATA WORD
6677 034030 004037 041326 JSR R0,@#CLAREA ;CLEAR 10. WORDS, FROM WRFROM+<4.*2>
6678 034034 002500 WRFROM+<4.*2> ;STARTING FROM WRFROM+<4.*2>
6679 034036 000012 10. ;10. WORDS
6680 034040 001125 <18.*40>!<21.> ;FILL WITH <18.*40>!<21.>
6681 034042 004037 041326 JSR R0,@#CLAREA ;CLEAR 246. WORDS, FROM WRFROM+<14.*2>
6682 034046 002524 WRFROM+<14.*2> ;STARTING FROM WRFROM+<14.*2>
6683 034050 000366 246. ;246. WORDS
6684 034052 177777 -1 ;FILL WITH -1
6685
6686 :*FILL READ HEADER AND DATA COMMAND FOR 10 WORDS
6687 034054 004037 043430 JSR R0,@#RUN ;SETUP TO RUN FOR DATA COMMAND
6688 034060 000000 0 ;CYLINDER 0
6689 034062 025 .BYTE 21. ;SECTOR 21.
6690 034063 022 .BYTE 18. ;TRACK 18.
6691 034064 177762 -10.-4 ;WORD COUNT (DATA) = 10. +
6692 034066 003534 REINTO ;BUS ADDRESS
6693 034070 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
6694 034072 014000 ECI!FMT22 ;16 BITS PER WORD FORMAT
6695 034074 002450 REFOR ;GET READY TO DO A REFOR
6696
6697 034076 004737 041536 JSR PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
6698 034102 104401 066720 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF ANY OF
6699 034106 000000 HALT ;STOP
6700
6701 034110 013777 004606 146150 MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
6702
6703 034116 013746 002450 MOV @#REFOR,-(SP) ;GET READY TO MOVE COMMAND
6704 034122 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET 'GO' AND
6705 034126 012677 146146 MOV (SP)+,@RHCS1 ;GO WITH
6706
6707 034132 104413 WAT ;WAIT FOR RDY BIT TO SET
6708 034134 002300 RHCS1 ;WAIT FOR RHCS1 REGISTER
6709 034136 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
6710 034140 001614 908. ;ALLOW 9080 MICRO SECONDS
6711 034142 001507 839. ;RDY MUST SET BETWEEN
6712
```

```
6713 ;*CHECK READ WORDS
6714 034144 004037 043474 JSR RO, @#COMPAR ;COMPARE TWO BLOCKS OF MEMORY
6715 034150 002470 WRFROM ;GOOD DATA STARTS FROM WRFROM
6716 034152 003534 REINTO ;TEST DATA STARTS FROM REINTO
6717 034154 000404 260. ;260. WORDS TO BE COMPARED
6718 034156 034162 ST26 ;RETURN TO ST26 ON ERROR
6719 034160 034166 ST27 ;RETURN TO ST27 ON NO ERROR
6720
6721 034162 104032 ST26: ERROR 32 ;PEAD HEADER AND DATA
6722 034164 000207 RTS PC ;FOLLOWING A SEEK TO CYLINDER 0
6723 ;FROM CYLINDER 10 GAVE AN
6724 ;ERROR
6725
6726 034166 ST27:
6727
```

```

6728
6729
6730 034166 000004          TST44: SCOPE
6731 034170 012706 001000  MOV #STACK,SP ;RESET STACK
6732 034174 012737 000044 004604  MOV #44,@TSTNM ;SAVE TEST NUMBER
6733 034202 004737 041456  JSR PC,@CLDISK ;SET R1-RHCS1, R2-RHCS2
6734
6735 ;*THIS GETS HEADS TO CYLINDER 0
6736 034206 013777 004606 146052  MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
6737
6738 034214 013746 002426  MOV @#RECALI,-(SP) ;GET READY TO MOVE COMMAND
6739 034220 052716 000101  BIS #GO!IE,(SP) ;GET READY TO SET 'GO' AND
6740 034224 012677 146050  MOV (SP)+,@RHCS1 ;GO WITH
6741 034230 011100  MOV @R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
6742 034232 011305  MOV @R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
6743
6744 034234 104413  WAT ;WAIT FOR DRY BIT TO SET
6745 034236 002322  RHDS1 ;WAIT FOR RHDS1 REGISTER
6746 034240 000200  DRY ;WAIT FOR DRY BIT IN RHDS1 REGISTER
6747 034242 076377  31999. ;ALLOW 319990 MICRO SECONDS
6748 034244 056701  24001. ;DRY MUST SET BETWEEN
6749
6750 ;*FILL REGISTERS FOR A SEEK COMMAND
6751 034246 004037 041426  JSR R0,@#SEEKCY ;SEEK FOR
6752 034252 000012  10. ;CYLINDER 10.
6753
6754 ;*SAVE REGISTERS FOR COMPARISON AFTER SEEK COMMAND
6755 034254 004037 041624  JSR R0,@#SAVER ;SAVE REGISTERS
6756 034260 002272  RHWC ;RHWC IS THE FIRST REGISTER SAVED
6757 034262 004612  SAVERE ;STARTING ADDRESS OF WHERE
6758 034264 000022  18. ;NUMBER OF REGISTERS
6759
6760 034266 013777 004606 145772  MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
6761
6762 034274 013746 002452  MOV @#SEECOM,-(SP) ;GET READY TO MOVE COMMAND
6763 034300 052716 000101  BIS #GO!IE,(SP) ;GET READY TO SET 'GO' AND
6764 034304 012677 145770  MOV (SP)+,@RHCS1 ;GO WITH
6765 034310 011100  MOV @R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
6766 034312 011305  MOV @R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
6767
6768 ;*SEEK FOR ONE CYLINDER=7 MILI SEC., FOR TEN=70 MILI SEC
6769 034314 104413  WAT ;WAIT FOR DRY BIT TO SET
6770 034316 002322  RHDS1 ;WAIT FOR RHDS1 REGISTER
6771 034320 000200  DRY ;WAIT FOR DRY BIT IN RHDS1 REGISTER
6772 034322 015530  7000. ;ALLOW 70000 MICRO SECONDS
6773 034324 000043  35. ;DRY MUST SET BETWEEN
6774 034326 013746 002452  MOV @#SEECOM,-(SP) ;SAVE COMMAND
6775 034332 052716 004301  BIS #DVA.GO!IE!RDY,(SP) ;INCLUDE DVA!GO!IE!RDY
6776 034336 011637 001124  MOV (SP)+,@#SGDDAT ;SAVE FOR PRINTOUT
6777 034342 022600  CMP (SP)+,R0 ;DURING ABOVE OPERATION ONLY DVA.GO!IE.RDY
6778 034344 001405  BEQ 67$ ;BRANCH IF GOOD
6779 034346 010037 001126  MOV R0,@#SBDDAT ;BAD DATA
6780 034352 010137 004600  MOV R1,@#REGADR ;FAILING REGISTER RHCS1
6781 034356 104021  ERROR 21 ;DURING ABOVE OPERATION ONLY
6782 034360 012746 030500 67$: MOV #PIP!MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
6783 034364 011637 001124  MOV (SP)+,@#SGDDAT ;SAVE FOR PRINTOUT

```



```

6784 034370 022605      CMP      (SP)+,R5      ;DURING ABOVE OPERATION ONLY PIP.MOL.DPR VV
6785 034372 001405      BEQ      69$          ;BRANCH IF GOOD
6786 034374 010537 001126  MOV      R5,@#SDDAT   ;BAD DATA
6787 034400 010337 004600  MOV      R3,@#REGADR  ;FAILING REGISTER RHDS1
6788 034404 104063      ERROR    63          ;DURING ABOVE OPERATION ONLY
6789
6790                      ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
6791
6792 034406 004037 041360  JSR      RO,@#FILLRE  ;MOV 10. INTO SAVED RHCC
6793 034412 002334      RHCC          ;SAVED REGISTER TO CHANGE
6794 034414 000012      10.         ;DATA
6795 034416 053737 004740 004636  BIS      @#ATTENT,@#SAVERE+24 ;SET APPROPRIATE 'ATA' BITS
6796 034424 004037 042336  JSR      RO,@#CHREG   ;CHANGE BITS IN SAVED REGISTER
6797 034430 002322      RHDS1       ;CHANGE RHDS1 REGISTER
6798 034432 000001      1           ;1 BIT/BITS TO BE CHANGED
6799 034434 000001      1           ;NEW VALUE OF ATA IS 1
6800 034436 100000      ATA         ;CHANGE ATA BIT
6801 034440 004037 042336  JSR      RO,@#CHREG   ;CHANGE BITS IN SAVED REGISTER
6802 034444 002300      RHCS1       ;CHANGE RHCS1 REGISTER
6803 034446 000001      1           ;1 BIT/BITS TO BE CHANGED
6804 034450 000001      1           ;NEW VALUE OF SC IS 1
6805 034452 100000      SC         ;CHANGE SC BIT
6806
6807                      ;*COMPARE REGISTERS AFTER A SEEK COMMAND
6808 034454 004037 042444  JSR      RO,@#COMREG  ;COMPARE SAVED REGISTERS WITH
6809 034460 004612      SAVERE       ;GOOD DATA SAVED IN 'SAVERE'
6810 034462 002354      WC         ;TEST DATA STARTING FROM 'RHWC'
6811 034464 000022      18.        ;18. REGISTERS TO BE COMPARED
6812 034466 034472      1$         ;RETURN TO 1$ ON ERROR
6813 034470 034476      2$         ;RETURN TO 2$ ON NO ERROR
6814
6815 034472 104037      1$:      ERROR    37      ;SEEK COMMAND CAUSED
6816 034474 000207      RTS      PC        ;ERROR
6817                      ;GOOD DATA GIVES WHAT SHOULD
6818                      ;BE THERE
6819                      ;RECEIVED DATA GIVES WHAT WAS
6820                      ;THERE AFTER A SEEK COMMAND
6821
6822                      ;*AT THIS POINT THE CURRENT CYLINDER IS GOOD AND THERE ARE
6823                      ;*NO ERROR BITS
6824                      ;*A READ HEADER AND DATA WILL BE DONE ON CYLINDER 9
6825                      ;*SECTOR 21 TRACK 18, EXPECTED DATA IS 23125
6826                      ;*FOR 20 WORDS
6827                      ;*CLEAR READ INTO BUFFER WITH ALL ONES
6828
6829 034476      2$:      JSR      PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
6830 034476 004737 041456  JSR      RO,@#CLAREA ;CLEAR 260. WORDS, FROM REINTO
6831 034502 004037 041326  REINTO   ;STARTING FROM REINTO
6832 034506 003534      260.       ;260. WORDS
6833 034510 000404      -1         ;FILL WITH -1
6834 034512 177777
6835
6836                      ;*FILL WRITE FROM BUFFER WITH EXPECTED DATA
6837
6838 034514 004037 041302  JSR      RO,@#FLHEAD ;SAVE HEADER DATA IN WRFROM
6839 034520 002470      WRFROM     ;LOCATION WHERE SAVED

```

```

6840 034522 000004 4 ;NUMBER OF WORDS SAVED
6841 034524 010011 10011 ;FIRST DATA WORD
6842 034526 011025 <18.*400>.<21.> ;SECOND DATA WORD
6843 034530 000000 0 ;THIRD DATA WORD
6844 034532 000000 0 ;FOURTH DATA WORD
6845 034534 004037 041326 JSR RO,@#CLAREA ;CLEAR 20. WORDS, FROM WRFROM+<4.*2>
6846 034540 002500 WRFROM+<4.*2> ;STARTING FROM WRFROM+<4.*2>
6847 034542 000024 20. ;20. WORDS
6848 034544 023125 <9.*2000>.<18.*40>.<21.> ;FILL WITH <9.*2000>.<18.*40>!<2
6849 034546 004037 041326 JSR RO,@#CLAREA ;CLEAR 250. WORDS, FROM WRFROM+<24.*2>
6850 034552 002550 WRFROM+<24.*2> ;STARTING FROM WRFROM+<24.*2>
6851 034554 000372 250. ;250. WORDS
6852 034556 177777 -1 ;FILL WITH -1
6853
6854 ;*FILL READ HEADER AND DATA COMMAND FOR 10 WORDS
6855 034560 004037 043430 JSR RO,@#RUN ;SETUP TO RUN FOR DATA COMMAND
6856 034564 000011 9. ;CYLINDER 9.
6857 034566 025 .BYTE 21. ;SECTOR 21.
6858 034567 022 .BYTE 18. ;TRACK 18.
6859 034570 177750 -20.-4 ;WORD COUNT (DATA) = 20. +
6860 034572 003534 REINTO ;BUS ADDRESS
6861 034574 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
6862 034576 014000 ECI.FMT22 ;16 BITS PER WORD FORMAT
6863 034600 002450 REFOR ;GET READY TO DO A REFOR
6864
6865 034602 004737 041536 JSR PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV 1
6866 034606 104401 066720 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF ANY OF
6867 034612 000000 HALT ;STOP
6868
6869 034614 013777 004606 145444 MOV @#RPO4VEC,@#RPOVEC ;SET RPO4 VECTOR ADDRESS
6870
6871 034622 013746 002450 MOV @#REFOR,-(SP) ;GET READY TO MOVE COMMAND
6872 034626 052716 000101 BIS #GO.IE,(SP) ;GET READY TO SET 'GO' AND
6873 034632 012677 145442 MOV (SP)+,@#RHCS1 ;GO WITH
6874
6875 034636 104413 WAT ;WAIT FOR RDY BIT TO SET
6876 034640 002300 RHCS1 ;WAIT FOR RHCS1 REGISTER
6877 034642 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
6878 034644 001614 908. ;ALLOW 9080 MICRO SECONDS
6879 034646 001507 839. ;RDY MUST SET BETWEEN
6880
6881 ;*CHECK READ WORDS
6882 034650 004037 043474 JSR RO,@#COMPAR ;COMPARE TWO BLOCKS OF MEMORY
6883 034654 002470 WRFROM ;GOOD DATA STARTS FROM WRFROM
6884 034656 003534 REINTO ;TEST DATA STARTS FROM REINTO
6885 034660 000404 260. ;260. WORDS TO BE COMPARED
6886 034662 034666 3$ ;RETURN TO 3$ ON ERROR
6887 034664 034672 4$ ;RETURN TO 4$ ON NO ERROR
6888
6889 034666 104032 3$: ERROR 32 ;READ HEADER AND DATA
6890 034670 000207 RTS PC ;FOLLOWING A SEEK TO CYLINDER 9
6891 ;FROM CYLINDER 0 GAVE AN
6892 ;ERROR
6893 034672 4$:
6894
6895

```

CZRJICO, RPO4/5/6 FCTNL CTLR1
CZRJIC.P11 27-JUL-78 13:05

J 12
MACY11 30A(1052) 27-JUL-78 13:06 PAGE 153
T44 SEEK & READ TEST (CYL = 009)

SEQ 0152

6896
6897

CZ
CZ

CZRJICO, RP04/5/6 FCTNL CTLR1
CZRJIC.P11 27-JUL-78 13:05

MACY11 30A(1052) 27-JUL-78 13:06 PAGE 154
WRITE CHECK DATA & WRITE PROTECT TESTS

K 12

SEQ 0153

6898
6899

.SBTTL WRITE CHECK DATA & WRITE PROTECT TESTS

CZ
CZ

```

6900 034672 000004          TST45: SCOPE
6901 034674 012706 001000      MOV    #STACK,SP      ;RESET STACK
6902 034700 012737 000045 004604  MOV    #45,@TSTNM     ;SAVE TEST NUMBER
6903 034706 004737 041456      JSR    PC,@CLDISK     ;SET R1-RHCS1, R2-RHCS2
6904 034712 004737 041536      JSR    PC,@CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV 1
6905 034716 104401 066720      TYPE   ,CPHALT       ;CANNOT CONTINUE TESTING IF ANY OF
6906 034722 000000          HALT                   ;STOP
6907
6908                          ;*GET HEADS TO CYLINDER 5
6909 034724 004037 041426      JSR    RO,@SEEKCY     ;SEEK FOR
6910 034730 000005          5                       ;CYLINDER 5
6911
6912 034732 013777 004606 145326  MOV    @RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
6913
6914 034740 013746 002452      MOV    @SEECOM,-(SP)  ;GET READY TO MOVE COMMAND
6915 034744 052716 000101      BIS    #GO!IE,(SP)   ;GET READY TO SET 'GO' AND
6916 034750 012677 145324      MOV    (SP)+,@RHCS1  ;GO WITH
6917 034754 011100          MOV    @R1,R0        ;SAVE RHCS1 DURING ABOVE OPERATION
6918 034756 011305          MOV    @R3,R5        ;SAVE RHDS1 DURING ABOVE OPERATION
6919
6920 034760 104413          WAT                   ;WAIT FOR DRY BIT TO SET
6921 034762 002322          RHDS1                ;WAIT FOR RHDS1 REGISTER
6922 034764 000200          DRY                   ;WAIT FOR DRY BIT IN RHDS1 REGISTER
6923 034766 015530          7000.                ;ALLOW 70000 MICRO SECONDS
6924 034770 000043          35.                  ;DRY MUST SET BETWEEN
6925 034772 004737 041456      JSR    PC,@CLDISK     ;SET R1-RHCS1, R2-RHCS2
6926
6927                          ;*FILL WRITE FROM BUFFER WITH HEADER
6928 034776 004037 041302      JSR    RO,@WFLHEAD   ;SAVE HEADER DATA IN WRFROM
6929 035002 002470          WRFROM                ;LOCATION WHERE SAVED
6930 035004 000004          4                       ;NUMBER OF WORDS SAVED
6931 035006 010005          10005                 ;FIRST DATA WORD
6932 035010 003404          <7*40> .4             ;SECOND DATA WORD
6933 035012 000000          0                       ;THIRD DATA WORD
6934 035014 000000          0                       ;FOURTH DATA WORD
6935
6936                          ;*10 WORDS OF OF THE FOLLOWING DATA
6937                          ;* 12344,17777,0,52525,125252
6938
6939 035016 004037 041326      JSR    RO,@CLAREA    ;CLEAR 10. WORDS, FROM WRFROM+<4*2>
6940 035022 002500          WRFROM+<4*2>          ;STARTING FROM WRFROM+<4*2>
6941 035024 000012          10.                   ;10. WORDS
6942 035026 012344          <5*2000>!<7*40>!4    ;FILL WITH <5*2000>!<7*40>!4
6943 035030 004037 041326      JSR    RO,@CLAREA    ;CLEAR 10. WORDS, FROM WRFROM+<14.*2>
6944 035034 002524          WRFROM+<14.*2>       ;STARTING FROM WRFROM+<14.*2>
6945 035036 000012          10.                   ;10. WORDS
6946 035040 177777          -1                    ;FILL WITH -1
6947 035042 004037 041326      JSR    RO,@CLAREA    ;CLEAR 10. WORDS, FROM WRFROM+<24.*2>
6948 035046 002550          WRFROM+<24.*2>       ;STARTING FROM WRFROM+<24.*2>
6949 035050 000012          10.                   ;10. WORDS
6950 035052 000000          0                     ;FILL WITH 0
6951 035054 004037 041326      JSR    RO,@CLAREA    ;CLEAR 10. WORDS, FROM WRFROM+<34.*2>
6952 035060 002574          WRFROM+<34.*2>       ;STARTING FROM WRFROM+<34.*2>
6953 035062 000012          10.                   ;10. WORDS
6954 035064 052525          52525                 ;FILL WITH 52525
6955 035066 004037 041326      JSR    RO,@CLAREA    ;CLEAR 10. WORDS, FROM WRFROM+<44.*2>

```

```

6956 035072 002620 WRFROM+<44.*2> ;STARTING FROM WRFROM+<44.*2>
6957 035074 000012 10. ;10. WORDS
6958 035076 125252 125252 ;FILL WITH 125252
6959
6960 ;*FILL LEFT ROTATING ZEROS FROM WRFROM+<54.*2>
6961
6962 035100 010146 MOV R1,-(SP) ;;PUSH R1 ON STACK
6963 035102 012700 177776 MOV #177776,R0 ;DATA
6964 035106 012705 000020 MOV #16.,R5 ;COUNT
6965 035112 012701 002644 MOV #WRFROM+<54.*2>,R1 ;WHERE DATA GOES
6966 035116 000261
6967 035120 010021 1$: MOV R0,(R1)+ ;STORE DATA
6968 035122 006100 ROL R0 ;GET ZERO ONE BIT LEFT
6969 035124 005305 DEC R5 ;COUNT 16
6970 035126 001374 BNE 1$ ;BRANCH IF 16 NOT DONE
6971
6972 ;*FILL LEFT ROTATING ONE INTO WRFROM+<65.*2>
6973
6974 035130 000241 CLC
6975 035132 012700 000001 2$: MOV #1,R0
6976 035136 010021 MOV R0,(R1)+
6977 035140 006300 ASL R0
6978 035142 103375 BCC 2$
6979 035144 012601 MOV (SP)+,R1 ;;POP STACK INTO R1
6980
6981
6982 ;*FILL REST OF DATA
6983
6984 035146 004037 041326 JSR R0,@#CLAREA ;CLEAR 174. WORDS, FROM WRFROM+<86.*2>
6985 035152 002744 WRFROM+<86.*2> ;STARTING FROM WRFROM+<86.*2>
6986 035154 000256 174. ;174. WORDS
6987 035156 000377 377 ;FILL WITH 377
6988 035160 004037 041302 JSR R0,@#FLHEAD ;SAVE HEADER DATA IN WRFROM+<260.*2>
6989 035164 003500 WRFROM+<260.*2> ;LOCATION WHERE SAVED
6990 035166 000004 4 ;NUMBER OF WORDS SAVED
6991 035170 010005 10005 ;FIRST DATA WORD
6992 035172 003405 <7*400>!5 ;SECOND DATA WORD
6993 035174 000000 0 ;THIRD DATA WORD
6994 035176 000000 0 ;FOURTH DATA WORD
6995 035200 004037 041326 JSR R0,@#CLAREA ;CLEAR 2 WORDS, FROM WRFROM+<264.*2>
6996 035204 003510 WRFROM+<264.*2> ;STARTING FROM WRFROM+<264.*2>
6997 035206 000002 2 ;2 WORDS
6998 035210 012345 <5*2000>!<7*40>!5 ;FILL WITH <5*2000>.<7*40>!5
6999
7000
7001 ;*READ INTO BUFFER WILL BE CLEARED
7002 035212 004037 041326 JSR R0,@#CLAREA ;CLEAR 266. WORDS, FROM REINTO
7003 035216 003534 REINTO ;STARTING FROM REINTO
7004 035220 000412 266. ;266. WORDS
7005 035222 177400 177400 ;FILL WITH 177400
7006
7007 ;*THE WRITE HEADER AND DATA COMMAND WILL BE LOADED
7008 035224 004037 043430 JSR R0,@#RUN ;SETUP TO RUN FOR DATA COMMAND
7009 035230 000005 5 ;CYLINDER 5
7010 035232 004 ;SECTOR 4
7011 035233 007 .BYTE 4 ;TRACK 7

```

```

7012 035234 177366 -262.-4 ;WORD COUNT (DATA) - 262. +
7013 035236 002470 WRFROM ;BUS ADDRESS
7014 035240 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
7015 035242 010000 FMT22 ;16 BITS PER WORD FORMAT
7016 035244 002444 WRIFOR ;GET READY TO DO A WRIFOR
7017
7018 ;*NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE HEADER AND DATA
7019
7020 035246 004037 041624 JSR RO,@#SAVER ;SAVE REGISTERS
7021 035252 002272 RHWC ;RHWC IS THE FIRST REGISTER SAVED
7022 035254 004612 SAVERE ;STARTING ADDRESS OF WHERE
7023 035256 000022 18. ;NUMBER OF REGISTERS
7024
7025 035260 004737 041536 JSR PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV - 1
7026 035264 104401 066720 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF ANY OF
7027 035270 000000 HALT ;STOP
7028
7029 035272 013777 004606 144766 MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
7030
7031 035300 013746 002444 MOV @#WRIFOR,-(SP) ;GET READY TO MOVE COMMAND
7032 035304 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET 'GO' AND
7033 035310 012677 144764 MOV (SP)+,@RHCS1 ;GO WITH
7034
7035 035314 104413 WAT ;WAIT FOR RDY BIT TO SET
7036 035316 002300 RHCS1 ;WAIT FOR RHCS1 REGISTER
7037 035320 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
7038 035322 001732 986. ;ALLOW 9860 MICRO SECONDS
7039 035324 001502 834. ;RDY MUST SET BETWEEN
7040
7041 ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
7042 035326 004037 041360 JSR RO,@#FILLRE ;MOV 0 INTO SAVED RHWC
7043 035332 002272 RHWC ;SAVED REGISTER TO CHANGE
7044 035334 000000 0 ;DATA
7045 035336 004037 041360 JSR RO,@#FILLRE ;MOV WRFROM+<266.*2> INTO SAVED RHBA
7046 035342 002274 RHBA ;SAVED REGISTER TO CHANGE
7047 035344 003514 WRFROM+<266.*2> ;DATA
7048 035346 004037 041360 JSR RO,@#FILLRE ;MOV 3406 INTO SAVED RHDST
7049 035352 002304 RHDST ;SAVED REGISTER TO CHANGE
7050 035354 003406 3406 ;DATA
7051
7052 ;*NOW COMPARE REGISTERS BEFORE WRITE HEADER AND DATA
7053 ;*WITH REGISTERS AFTER COMMAND
7054
7055 035356 004037 042444 JSR RO,@#COMREG ;COMPARE SAVED REGISTERS WITH
7056 035362 004612 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
7057 035364 002354 WC ;TEST DATA STARTING FROM 'RHWC'
7058 035366 000022 18. ;18. REGISTERS TO BE COMPARED
7059 035370 035374 3$ ;RETURN TO 3$ ON ERROR
7060 035372 035400 4$ ;RETURN TO 4$ ON NO ERROR
7061
7062 035374 104027 3$: ERROR 27 ;WRITE HEADER AND DATA
7063 035376 000207 RTS PC ;CAUSED IMPROPER REGISTER
7064 ;CHANGE
7065 ;GOOD DATA GIVES WHAT SHOULD
7066 ;BE THERE
7067 ;RECEIVED DATA GIVES WHAT
  
```

```

7068                                     ;WAS THERE AFTER COMMENT
7069
7070                                     ;*NOW FILL COMMAND FOR READ
7071
7072 035400                               4S:
7073 035400 004737 041456                JSR    PC,@#CLDISK    ;SET R1-RHCS1, R2-RHCS2
7074 035404 004037 043430                JSR    RO,@#RUN      ;SETUP TO RUN FOR DATA COMMAND
7075 035410 000005                        5      ;CYLINDER 5
7076 035412 004      ;.BYTE              4      ;SECTOR 4
7077 035413 007      ;.BYTE              7      ;TRACK 7
7078 035414 177366                        -262.-4 ;WORD COUNT (DATA) = 262. +
7079 035416 003534                        REINTO  ;BUS ADDRESS
7080 035420 000000                        0      ;DO NOT INHIBIT BUS ADDRESS INCREMENT
7081 035422 014000                        ECI'FMT22 ;16 BITS PER WORD FORMAT
7082 035424 002450                        REFOR   ;GET READY TO DO A REFOR
7083
7084                                     ;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ HEADER AND DATA
7085
7086 035426 004037 041624                JSR    RO,@#SAVER    ;SAVE REGISTERS
7087 035432 002272                        RHWC   ;RHWC IS THE FIRST REGISTER SAVED
7088 035434 004612                        SAVERE ;STARTING ADDRESS OF WHERE
7089 035436 000022                        18.   ;NUMBER OF REGISTERS
7090
7091 035440 004737 041536                JSR    PC,@#CHECKT  ;CHECK DVA,RDY,MOL,DPR,DRY,VV - 1
7092 035444 104401 066720                TYPE  ,CPHALT      ;CANNOT CONTINUE TESTING IF ANY OF
7093 035450 000000                        HALT   ;STOP
7094
7095 035452 013777 004606 144606        MOV    @#RP4VEC,@#RPVEC ;SET RP04 VECTOR ADDRESS
7096
7097 035460 013746 002450                MOV    @#REFOR,-(SP) ;GET READY TO MOVE COMMAND
7098 035464 052716 000101                BIS    #GO!IE,(SP)  ;GET READY TO SET 'GO' AND
7099 035470 012677 144604                MOV    (SP)+,@#RHCS1 ;GO WITH
7100
7101 035474 104413                        WAT    ;WAIT FOR RDY BIT TO SET
7102 035476 002300                        RHCS1  ;WAIT FOR RHCS1 REGISTER
7103 035500 000200                        RDY    ;WAIT FOR RDY BIT IN RHCS1 REGISTER
7104 035502 001732                        986.  ;ALLOW 9860 MICRO SECONDS
7105 035504 001502                        834.  ;RDY MUST SET BETWEEN
7106
7107                                     ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
7108
7109 035506 004037 041360                JSR    RO,@#FILLRE  ;MOV 0 INTO SAVED RHWC
7110 035512 002272                        RHWC   ;SAVED REGISTER TO CHANGE
7111 035514 000000                        0      ;DATA
7112 035516 004037 041360                JSR    RO,@#FILLRE  ;MOV REINTO+<266.*2> INTO SAVED RHBA
7113 035522 002274                        RHBA   ;SAVED REGISTER TO CHANGE
7114 035524 004560                        REINTO+<266.*2> ;DATA
7115 035526 004037 041360                JSR    RO,@#FILLRE  ;MOV 3406 INTO SAVED RHDST
7116 035532 002304                        RHDST  ;SAVED REGISTER TO CHANGE
7117 035534 003406                        3406  ;DATA
7118
7119                                     ;*COMPARE REGISTERS BEFORE READ HEADER AND DATA
7120                                     ;*WITH REGISTERS AFTER COMMAND
7121
7122
7123 035536 004037 042444                JSR    RO,@#COMREG  ;COMPARE SAVED REGISTERS WITH

```



```

7124 035542 004612          SAVERE          :GOOD DATA SAVED IN 'SAVERE'
7125 035544 002354          WC             :TEST DATA STARTING FROM 'RHWC'
7126 035546 000022          18.           :18. REGISTERS TO BE COMPARED
7127 035550 035554          5$            :RETURN TO 5$ ON ERROR
7128 035552 035560          6$            :RETURN TO 6$ ON NO ERROR
7129
7130
7131 035554 104031          5$:          ERROR 31      :READ HEADER AND DATA CAUSED
7132 035556 000207          RTS PC        :IMPROPER REGISTER CHANGE
7133                                     :GOOD DATA GIVES WHAT SHOULD
7134                                     :BE THERE RECEIVED DATA GIVES WHAT WAS
7135                                     :RECEIVED DATA GIVES WHAT WAS
7136                                     :THERE AFTER COMMAND
7137
7138                                     ;*NOW READ INTO BUFFER WILL BE CHECKED TO SEE
7139                                     ;*THAT READ WAS GOOD
7140
7141 035560          6$:
7142 035560 004037 043474      JSR    RO,@#COMPAR :COMPARE TWO BLOCKS OF MEMORY
7143 035564 002470          WRFROM        :GOOD DATA STARTS FROM WRFROM
7144 035566 003534          REINTO        :TEST DATA STARTS FROM REINTO
7145 035570 000412          266.         :266. WORDS TO BE COMPARED
7146 035572 035576          7$          :RETURN TO 7$ ON ERROR
7147 035574 035602          10$         :RETURN TO 10$ ON NO ERROR
7148
7149
7150 035576 104032          7$:          ERROR 32      :WRITE HEADER AND DATA
7151 035600 000207          RTS PC        :FOLLOWED BY A READ HEADER
7152                                     :AND DATA GAVE A READ ERROR
7153                                     :ERROR MAY BE IN READ OR WRITE
7154
7155                                     ;*A WRITE,READ HAS BEEN SUCCESSFULLY COMPLETED
7156                                     ;*NOW A WRITE CHECK HEADER AND DATA WILL BE GIVEN
7157                                     ;*FILL THE WRITE CHECK HEADER AND DATA
7158
7159 035602          10$:
7160 035602 004737 041456      JSR    PC,@#CLDISK :SET R1-RHCS1, R2-RHCS2
7161 035606 004037 043430      JSR    RO,@#RUN    :SETUP TO RUN FOR DATA COMMAND
7162 035612 000005          5            :CYLINDER 5
7163 035614 004          .BYTE 4            :SECTOR 4
7164 035615 007          .BYTE 7            :TRACK 7
7165 035616 177366          -262.-4      :WORD COUNT (DATA) = 262. +
7166 035620 002470          WRFROM        :BUS ADDRESS
7167 035622 000000          0            :DO NOT INHIBIT BUS ADDRESS INCREMENT
7168 035624 014000          ECI!FMT22    :16 BITS PER WORD FORMAT
7169 035626 002440          WRCHDT      :GET READY TO DO A WRCHDT
7170
7171                                     ;*SAVE REGISTERS FOR COMPARISON AFTER WRITE CHECK
7172
7173 035630          ST25:
7174 035630 004037 041624      JSR    RO,@#SAVER :SAVE REGISTERS
7175 035634 002272          RHWC         :RHWC IS THE FIRST REGISTER SAVED
7176 035636 004612          SAVERE        :STARTING ADDRESS OF WHERE
7177 035640 000022          18.         :NUMBER OF REGISTERS
7178
7179 035642 004737 041536      JSR    PC,@#CHECKT :CHECK DVA,RDY,MOL,DPR,DRY,VV 1

```

```

7180 035646 104401 066720 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF ANY OF
7181 035652 000000 HALT ;STOP
7182
7183 035654 013777 004606 144404 MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
7184
7185 035662 013746 002440 MOV @#WRCHDT,-(SP) ;GET READY TO MOVE COMMAND
7186 035666 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET 'GO' AND
7187 035672 012677 144402 MOV (SP)+,@RHCS1 ;GO WITH
7188 035676 011100 MOV @R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
7189 035700 011305 MOV @R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
7190
7191 035702 104413 WAT ;WAIT FOR RDY BIT TO SET
7192 035704 002300 RHCS1 ;WAIT FOR RHCS1 REGISTER
7193 035706 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
7194 035710 001732 986. ;ALLOW 9860 MICRO SECONDS
7195 035712 001502 834. ;RDY MUST SET BETWEEN
7196 035714 013746 002440 MOV @#WRCHDT,-(SP) ;SAVE COMMAND
7197 035720 052716 004101 BIS #IE.DVA!GO,(SP) ;INCLUDE IE!DVA!GO
7198 035724 011637 001124 MOV (SP)+,@#SGDDAT ;SAVE FOR PRINTOUT
7199 035730 022600 CMP (SP)+,R0 ;DURING ABOVE OPERATION ONLY IE.DVA!GO
7200 035732 001405 BEQ 64$ ;BRANCH IF GOOD
7201 035734 010037 001126 MOV R0,@#SBDDAT ;BAD DATA
7202 035740 010137 004600 MOV R1,@#REGADR ;FAILING REGISTER RHCS1
7203 035744 104021 ERROR 21 ;DURING ABOVE OPERATION ONLY
7204 035746 012746 010500 64$: MOV #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
7205 035752 011637 001124 MOV (SP)+,@#SGDDAT ;SAVE FOR PRINTOUT
7206 035756 022605 CMP (SP)+,R5 ;DURING ABOVE OPERATION ONLY MOL.DPR!VV
7207 035760 001405 BEQ 66$ ;BRANCH IF GOOD
7208 035762 010537 001126 MOV R5,@#SBDDAT ;BAD DATA
7209 035766 010337 004600 MOV R3,@#REGADR ;FAILING REGISTER RHDS1
7210 035772 104063 ERROR 63 ;DURING ABOVE OPERATION ONLY
7211
7212 ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
7213
7214 035774 004037 041360 JSR R0,@#FILLRE ;MOV 0 INTO SAVED RHWC
7215 036000 002272 RHWC ;SAVED REGISTER TO CHANGE
7216 036002 000000 0 ;DATA
7217 036004 004037 041360 JSR R0,@#FILLRE ;MOV WRFROM+<266.*2> INTO SAVED RHBA
7218 036010 002274 RHBA ;SAVED REGISTER TO CHANGE
7219 036012 003514 WRFROM+<266.*2> ;DATA
7220 036014 004037 041360 JSR R0,@#FILLRE ;MOV 3406 INTO SAVED RHDST
7221 036020 002304 RHDST ;SAVED REGISTER TO CHANGE
7222 036022 003406 3406 ;DATA
7223
7224 ;*COMPARE REGISTERS BEFORE WRITE CHECK HEADER AND DATA
7225 ;*WITH REGISTERS AFTER COMMAND
7226
7227 036024 004037 042444 JSR R0,@#COMREG ;COMPARE SAVED REGISTERS WITH
7228 036030 004612 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
7229 036032 002354 WC ;TEST DATA STARTING FROM 'RHWC'
7230 036034 000022 18. ;18. REGISTERS TO BE COMPARED
7231 036036 036042 8$ ;RETURN TO 8$ ON ERROR
7232 036040 036046 9$ ;RETURN TO 9$ ON NO ERROR
7233
7234 036042 104040 8$: ERROR 40 ;WRITE CHECK CAUSED
7235 036044 000207 RTS PC ;AN IMPROPER REGISTER
  
```

CZRJICO, RPO4/5/6 FCTNL CTLR1
CZRJIC.P11 27-JUL-78 13:05

E 13
MACY11 30A(1052) 27-JUL-78 13:06 PAGE 161
T45 WRITE CHECK HEADER AND DATA

SEQ 0160

7236
7237
7238
7239
7240
7241
7242
7243 036046

:CHANGE
:GOOD DATA GIVES WHAT
:SHOULD BE THERE
:RECEIVED DATA GIVES WHAT
:WAS THERE AFTER COMMAND

98:

```

7244
7245 036046 000004          TST46: SCOPE
7246 036050 012706 001000  MOV      #STACK,SP          ;RESET STACK
7247 036054 012737 000046 004604  MOV      #46,@TSTNM        ;SAVE TEST NUMBER
7248 036062 004737 041456  JSR      PC,@CLDISK        ;SET R1-RHCS1, R2-RHCS2
7249 036066 004737 041536  JSR      PC,@CHECKT        ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
7250 036072 104401 066720  TYPE      ,CPHALT          ;CANNOT CONTINUE TESTING IF ANY OF
7251 036076 000000          HALT                        ;STOP
7252
7253                          ;*GET HEADS TO CYLINDER >
7254
7255 036100 004037 041426  JSR      R0,@SEEKCY        ;SEEK FOR
7256 036104 000005          5                          ;CYLINDER 5
7257
7258 036106 013777 004606 144152  MOV      @RP4VEC,@RPVEC    ;SET RPO4 VECTOR ADDRESS
7259
7260 036114 013746 002452  MOV      @SEECOM,-(SP)     ;GET READY TO MOVE COMMAND
7261 036120 052716 000101  BIS      #GO.IE,(SP)       ;GET READY TO SET 'GO' AND
7262 036124 012677 144150  MOV      (SP)+,@RHCS1      ;GO WITH
7263
7264 036130 104413          WAT                        ;WAIT FOR DRY BIT TO SET
7265 036132 002322          RHDS1                     ;WAIT FOR RHDS1 REGISTER
7266 036134 000200          DRY                       ;WAIT FOR DRY BIT IN RHDS1 REGISTER
7267 036136 004704          2500.                     ;ALLOW 25000 MICRO SECONDS
7268 036140 004704          2500.                     ;DRY MUST SET BETWEEN
7269
7270 036142 004737 041456  JSR      PC,@CLDISK        ;SET R1-RHCS1, R2-RHCS2
7271
7272                          ;*10 WORDS OF EACH 12344,17777,0,52525,125252
7273
7274 036146 004037 041326  JSR      R0,@CLAREA        ;CLEAR 10. WORDS, FROM WRFROM
7275 036152 002470          WRFROM                    ;STARTING FROM WRFROM
7276 036154 000012          10.                       ;10. WORDS
7277 036156 012344          <5*2000>!<7*40>!4        ;FILL WITH <5*2000>!<7*40>!4
7278 036160 004037 041326  JSR      R0,@CLAREA        ;CLEAR 10. WORDS, FROM WRFROM+<10.*2>
7279 036164 002514          WRFROM+<10.*2>            ;STARTING FROM WRFROM+<10.*2>
7280 036166 000012          10.                       ;10. WORDS
7281 036170 177777          -1                         ;FILL WITH -1
7282 036172 004037 041326  JSR      R0,@CLAREA        ;CLEAR 10. WORDS, FROM WRFROM+<20.*2>
7283 036176 002540          WRFROM+<20.*2>            ;STARTING FROM WRFROM+<20.*2>
7284 036200 000012          10.                       ;10. WORDS
7285 036202 000000          0                         ;FILL WITH 0
7286 036204 004037 041326  JSR      R0,@CLAREA        ;CLEAR 10. WORDS, FROM WRFROM+<30.*2>
7287 036210 002564          WRFROM+<30.*2>            ;STARTING FROM WRFROM+<30.*2>
7288 036212 000012          10.                       ;10. WORDS
7289 036214 052525          52525                     ;FILL WITH 52525
7290 036216 004037 041326  JSR      R0,@CLAREA        ;CLEAR 10. WORDS, FROM WRFROM+<40.*2>
7291 036222 002610          WRFROM+<40.*2>            ;STARTING FROM WRFROM+<40.*2>
7292 036224 000012          10.                       ;10. WORDS
7293 036226 125252          125252                    ;FILL WITH 125252
7294
7295                          ;*FILL LEFT ROTATING ZEROS FROM WRFROM+<50.*2>
7296
7297 036230 010146          MOV      R1,-(SP)          ;:PUSH R1 ON STACK
7298 036232 012700 177776          MOV      #177776,R0        ;:DATA
7299 036236 012705 000020          MOV      #16.,R5           ;:COUNT
    
```

```

7300 036242 012701 002634      MOV    #WRFROM+<50.*2>,R1 ;WHERE DATA GOES
7301 036246 000261              SEC
7302 036250 010021      1$:  MOV    RO,(R1)+      ;STORE DATA
7303 036252 006100      ROL    RO              ;GET ZERO ONE BIT LEFT
7304 036254 005305      DEC    R5              ;COUNT 16
7305 036256 001374      BNE    1$              ;BRANCH IF 16 NOT DONE
7306
7307                          ;*FILL LEFT ROTATING ONE INTO WRFROM+<65.*2>
7308
7309 036260 000241      CLC
7310 036262 012700 000001      MOV    #1,R0
7311 036266 010021      2$:  MOV    RO,(R1)+
7312 036270 006300      ASL    RO
7313 036272 103375      BCC    2$
7314 036274 012601      MOV    (SP)+,R1      ;;POP STACK INTO R1
7315
7316                          ;*FILL REST OF DATA
7317
7318 036276 004037 041326      JSR    RO,@#CLAREA    ;CLEAR 174. WORDS, FROM WRFROM+<82.*2>
7319 036302 002734      WRFROM+<82.*2>      ;STARTING FROM WRFROM+<82.*2>
7320 036304 000256      174.              ;174. WORDS
7321 036306 000377      377.              ;FILL WITH 377
7322 036310 004037 041326      JSR    RO,@#CLAREA    ;CLEAR 2 WORDS, FROM WRFROM+<256.*2>
7323 036314 003470      WRFROM+<256.*2>    ;STARTING FROM WRFROM+<256.*2>
7324 036316 000002      2                ;2 WORDS
7325 036320 012345      <5*2000>!<7*40>.5 ;FILL WITH <5*2000>!<7*40>!5
7326
7327                          ;*FILL THE WRITE CHECK HEADER AND DATA
7328 036322 004037 043430      JSR    RO,@#RUN      ;SETUP TO RUN FOR DATA COMMAND
7329 036326 000005      5                ;CYLINDER 5
7330 036330 004      .BYTE 4              ;SECTOR 4
7331 036331 007      .BYTE 7              ;TRACK 7
7332 036332 177400      -256.            ;WORD COUNT = 256.
7333 036334 002470      WRFROM           ;BUS ADDRESS
7334 036336 000000      0                ;DO NOT INHIBIT BUS ADDRESS INCREMENT
7335 036340 014000      ECI.FMT22        ;16 BITS PER WORD FORMAT
7336 036342 002436      WRCHK           ;GET READY TO DO A WRCHK
7337
7338                          ;*SAVE REGISTERS FOR COMPARISON AFTER WRITE CHECK
7339 036344 004037 041624      JSR    RO,@#SAVER    ;SAVE REGISTERS
7340 036350 002272      RHWC            ;RHWC IS THE FIRST REGISTER SAVED
7341 036352 004612      SAVERE          ;STARTING ADDRESS OF WHERE
7342 036354 000022      18.             ;NUMBER OF REGISTERS
7343
7344 036356 004737 041536      JSR    PC,@#CHECKT   ;CHECK DVA,RDY,MOL,DPR,DRY,VV 1
7345 036362 104401 066720      TYPE    ,CPHALT     ;CANNOT CONTINUE TESTING IF ANY OF
7346 036366 000000      HALT            ;STOP
7347
7348 036370 013777 004606 143670      MOV    @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
7349
7350 036376 013746 002436      MOV    @#WRCHK,-(SP) ;GET READY TO MOVE COMMAND
7351 036402 057716 000101      BIS    #GO!IE,(SP)  ;GET READY TO SET 'GO' AND
7352 036406 012677 143666      MOV    (SP)+,@RHCS1 ;GO WITH
7353 036412 011100      MOV    @R1,R0      ;SAVE RHCS1 DURING ABOVE OPERATION
7354 036414 011305      MOV    @R3,R5      ;SAVE RHDS1 DURING ABOVE OPERATION
7355
    
```

```

7356 036416 104413 WAT ;WAIT FOR RDY BIT TO SET
7357 036420 002300 RHCS1 ;WAIT FOR RHCS1 REGISTER
7358 036422 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
7359 036424 001732 986. ;ALLOW 9860 MICRO SECONDS
7360 036426 001502 834. ;RDY MUST SET BETWEEN
7361 036430 013746 002436 MOV @WRCHK,-(SP) ;SAVE COMMAND
7362 036434 052716 004101 BIS #IE.DVA.GO,(SP) ;INCLUDE IE!DVA.GO
7363 036440 011637 001124 MOV (SP),@SGDDAT ;SAVE FOR PRINTOUT
7364 036444 022600 CMP (SP)+,R0 ;DURING ABOVE OPERATION ONLY IE!DVA.GO
7365 036446 001405 BEQ 67$ ;BRANCH IF GOOD
7366 036450 010037 001126 MOV R0,@SBDDAT ;BAD DATA
7367 036454 010137 004600 MOV R1,@REGADR ;FAILING REGISTER RHCS1
7368 036460 104021 ERROR 21 ;DURING ABOVE OPERATION ONLY
7369 036462 012746 010500 67$: MOV #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
7370 036466 011637 001124 MOV (SP),@SGDDAT ;SAVE FOR PRINTOUT
7371 036472 022605 CMP (SP)+,R5 ;DURING ABOVE OPERATION ONLY MOL.DPR.VV
7372 036474 001405 BEQ 69$ ;BRANCH IF GOOD
7373 036476 010537 001126 MOV R5,@SBDDAT ;BAD DATA
7374 036502 010337 004600 MOV R3,@REGADR ;FAILING REGISTER RHDS1
7375 036506 104063 ERROR 63 ;DURING ABOVE OPERATION ONLY
7376
7377 ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
7378
7379 036510 004037 041360 JSR R0,@FILLRE ;MOV 0 INTO SAVED RHWC
7380 036514 002272 RHWC ;SAVED REGISTER TO CHANGE
7381 036516 000000 0 ;DATA
7382 036520 004037 041360 JSR R0,@FILLRE ;MOV WRFROM+<256.*2> INTO SAVED RHBA
7383 036524 002274 RHBA ;SAVED REGISTER TO CHANGE
7384 036526 003470 WRFROM+<256.*2> ;DATA
7385 036530 004037 041360 JSR R0,@FILLRE ;MOV 3405 INTO SAVED RHDST
7386 036534 002304 RHDST ;SAVED REGISTER TO CHANGE
7387 036536 003405 3405 ;DATA
7388
7389 ;*COMPARE REGISTERS BEFORE WRITE CHECK HEADER AND DATA
7390 ;*WITH REGISTER AFTER COMMAND
7391
7392 036540 004037 042444 JSR R0,@COMREG ;COMPARE SAVED REGISTERS WITH
7393 036544 004612 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
7394 036546 002354 WC ;TEST DATA STARTING FROM 'RHWC'
7395 036550 000022 18 ;18. REGISTERS TO BE COMPARED
7396 036552 036556 8$ ;RETURN TO 8$ ON ERROR
7397 036554 036562 9$ ;RETURN TO 9$ ON NO ERROR
7398 036556 104040 8$: ERROR 40 ;WRITE CHECK CAUSED
7399 036560 000207 RTS PC ;AN IMPROPER REGISTER
7400 ;CHANGE
7401 ;GOOD DATA GIVES WHAT
7402 ;SHOULD BE THERE
7403 ;RECEIVED DATA GIVES WHAT
7404 ;WAS THERE AFTER COMMANDS
7405
7406 036562 9$:
7407
7408

```

```

7409
7410 036562 000004          TST47: SCOPE
7411 036564 012706 001000  MOV      #STACK,SP      ;RESET STACK
7412 036570 012737 000047 004604  MOV      #47,@TSTNM     ;SAVE TEST NUMBER
7413 036576 004737 041456  JSR      PC,@CLDISK     ;SET R1-RHCS1, R2-RHCS2
7414 036602 005737 004750  TST      @RH70          ;TEST FOR RH70 CONTROLLER
7415 036606 001402          BEQ      30$            ;IF FLAG = 1, THIS TEST IS SKIPPED
7416 036610 000137 037330  JMP      TST50          ;JUMP TO NEXT TEST -----)
7417 036614 004737 041536  JSR      PC,@CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
7418 036620 104401 066720  TYPE     ,CPHALT       ;CANNOT CONTINUE TESTING IF ANY OF
7419 036624 000000          HALT                    ;STOP
7420
7421 036626 005737 004732  TST      @UBUSB         ;IS UNIBUS B THERE
7422 036632 001402          BEQ      11$           ;UNIBUS B THERE SO CONTINUE
7423 036634 000137 037330  JMP      @9$            ;NO UNIBUS B, GO TO NEXT TEST
7424
7425          ;*GET HEADS TO CYLINDER 5
7426
7427 036640          11$:
7428 036640 004037 041426  JSR      R0,@SEEKCY     ;SEEK FOR
7429 036644 000005          5                        ;CYLINDER 5
7430
7431 036646 013777 004606 143412  MOV      @RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
7432
7433 036654 013746 002452  MOV      @SEECOM,-(SP)  ;GET READY TO MOVE COMMAND
7434 036660 052716 000101  BIS      #GO.IE,(SP)    ;GET READY TO SET 'GO' AND
7435 036664 012677 143410  MOV      (SP)+,@RHCS1   ;GO WITH
7436
7437 036670 104413          WAT                    ;WAIT FOR DRY BIT TO SET
7438 036672 002322          RHDS1                 ;WAIT FOR RHDS1 REGISTER
7439 036674 000200          DRY                    ;WAIT FOR DRY BIT IN RHDS1 REGISTER
7440 036676 004704          2500.                 ;ALLOW 25000 MICRO SECONDS
7441 036700 004704          2500.                 ;DRY MUST SET BETWEEN
7442
7443 036702 004737 041456  JSR      PC,@CLDISK     ;SET R1-RHCS1, R2-RHCS2
7444
7445          ;*10 WORDS OF EACH 12344,17777,0,52525,125252
7446
7447 036706 004037 041326  JSR      R0,@CLAREA     ;CLEAR 10. WORDS, FROM WRFROM
7448 036712 002470          WRFROM                 ;STARTING FROM WRFROM
7449 036714 000012          10.                    ;10. WORDS
7450 036716 012344          <5*2000>!<7*40>.4     ;FILL WITH <5*2000>!<7*40>!4
7451 036720 004037 041326  JSR      R0,@CLAREA     ;CLEAR 10. WORDS, FROM WRFROM+<10.*2>
7452 036724 002514          WRFROM+<10.*2>         ;STARTING FROM WRFROM+<10.*2>
7453 036726 000012          10.                    ;10. WORDS
7454 036730 177777          -1                      ;FILL WITH -1
7455 036732 004037 041326  JSR      R0,@CLAREA     ;CLEAR 10. WORDS, FROM WRFROM+<20.*2>
7456 036736 002540          WRFROM+<20.*2>         ;STARTING FROM WRFROM+<20.*2>
7457 036740 000012          10.                    ;10. WORDS
7458 036742 000000          0                        ;FILL WITH 0
7459 036744 004037 041326  JSR      R0,@CLAREA     ;CLEAR 10. WORDS, FROM WRFROM+<30.*2>
7460 036750 002564          WRFROM+<30.*2>         ;STARTING FROM WRFROM+<30.*2>
7461 036752 000012          10.                    ;10. WORDS
7462 036754 052525          52525                 ;FILL WITH 52525
7463 036756 004037 041326  JSR      R0,@CLAREA     ;CLEAR 10. WORDS, FROM WRFROM+<40.*2>
7464 036762 002610          WRFROM+<40.*2>         ;STARTING FROM WRFROM+<40.*2>
  
```

```
7465 036764 000012      10.           ;10. WORDS
7466 036766 125252      125252        ;FILL WITH 125252
7467
7468                      ;*FILL LEFT ROTATING ZEROS FROM WRFROM+<50.*2>
7469
7470 036770 010146      MOV    R1,-(SP)      ;;PUSH R1 ON STACK
7471 036772 012700 177776  MOV    #177776,R0    ;:DATA
7472 036776 012705 000020  MOV    #16,R5        ;:COUNT
7473 037002 012701 002634  MOV    #WRFROM+<50.*2>,R1 ;WHERE DATA GOES
7474 037006 000261
7475 037010 010021      1$:  MOV    R0,(R1)+     ;STORE DATA
7476 037012 006100      ROL    R0            ;GET ZERO ONE BIT LEFT
7477 037014 005305      DEC    R5            ;COUNT 16
7478 037016 001374      BNE   1$            ;BRANCH IF 16 NOT DONE
7479
7480                      ;*FILL LEFT ROTATING ONE INTO WRFROM+<65.*2>
7481
7482 037020 000241      CLC
7483 037022 012700 000001  MOV    #1,R0
7484 037026 010021      2$:  MOV    R0,(R1)+
7485 037030 006300      ASL   R0
7486 037032 103375      BCC   2$
7487 037034 012601      MOV    (SP)+,R1     ;;POP STACK INTO R1
7488
7489                      ;*FILL REST OF DATA
7490 037036 004037 041326  JSR    R0,@#CLAREA  ;CLEAR 174. WORDS, FROM WRFROM+<82.*2>
7491 037042 002734      WRFROM+<82.*2>      ;STARTING FROM WRFROM+<82.*2>
7492 037044 000256      174.              ;174. WORDS
7493 037046 000377      377               ;FILL WITH 377
7494 037050 004037 041326  JSR    R0,@#CLAREA  ;CLEAR 2 WORDS, FROM WRFROM+<256.*2>
7495 037054 003470      WRFROM+<256.*2>    ;STARTING FROM WRFROM+<256.*2>
7496 037056 000002      2                 ;2 WORDS
7497 037060 012345      <5*2000>.<7*40>!5 ;FILL WITH <5*2000>!<7*40>!5
7498
7499                      ;*FILL THE WRITE CHECK HEADER AND DATA
7500 037062 004037 043430  JSR    R0,@#RUN     ;SETUP TO RUN FOR DATA COMMAND
7501 037066 000005      5                 ;CYLINDER 5
7502 037070 004      .BYTE 4              ;SECTOR 4
7503 037071 007      .BYTE 7              ;TRACK 7
7504 037072 177400      -256.            ;WORD COUNT = 256.
7505 037074 002470      WRFROM           ;BUS ADDRESS
7506 037076 000000      0                ;DO NOT INHIBIT BUS ADDRESS INCREMENT
7507 037100 014000      ECI.FMT22        ;16 BITS PER WORD FORMAT
7508 037102 002436      WRCHK           ;GET READY TO DO A WRCHK
7509
7510
7511 037104 052777 002000 143166  BIS    #PSEL,@RHCS1 ;SET PORT B
7512                                     ;THAT IS UNIBUS B
7513
7514                      ;*SAVE REGISTERS FOR COMPARISON AFTER WRITE CHECK
7515 037112 004037 041624  JSR    R0,@#SAVER   ;SAVE REGISTERS
7516 037116 002272      RHC           ;RHC IS THE FIRST REGISTER SAVED
7517 037120 004612      SAVER          ;STARTING ADDRESS OF WHERE
7518 037122 000022      18.           ;NUMBER OF REGISTERS
7519
7520 037124 004737 041536  JSR    PC,@#CHECKT  ;CHECK DVA,RDY,MOL,DPR,DRY,VV - 1
```



```

7521 037130 104401 066720 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF ANY OF
7522 037134 000000 HALT ;STOP
7523
7524 037136 013777 004606 143122 MOV @WRP4VEC,@RPVEC ;SET RPO4 VECTOR ADDRESS
7525
7526 ;*SET PORT SELECT
7527 037144 013746 002436 MOV @WRCKEK,-(SP) ;GET READY TO MOVE COMMAND
7528 037150 052716 002101 BIS #GO.IE!PSEL,(SP) ;GET READY TO SET 'GO' AND
7529 037154 012677 143120 MOV (SP)+,@RHCS1 ;GO WITH
7530 037160 011100 MOV @R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
7531 037162 011305 MOV @R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
7532
7533 037164 104413 WAT ;WAIT FOR RDY BIT TO SET
7534 037166 002300 RHCS1 ;WAIT FOR RHCS1 REGISTER
7535 037170 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
7536 037172 001732 986. ;ALLOW 9860 MICRO SECONDS
7537 037174 001502 834. ;RDY MUST SET BETWEEN
7538 037176 013746 002436 MOV @WRCKEK,-(SP) ;SAVE COMMAND
7539 037202 052716 006101 BIS #IE!DVA!PSEL!GO,(SP) ;INCLUDE IE!DVA.PSEL.GO
7540 037206 011637 001124 MOV (SP)+,@SGDDAT ;SAVE FOR PRINTOUT
7541 037212 022600 CMP (SP)+,R0 ;DURING ABOVE OPERATION ONLY IE.DVA!PSEL!GO
7542 037214 001405 BEQ 67$ ;BRANCH IF GOOD
7543 037216 010037 001126 MOV R0,@SBDDAT ;BAD DATA
7544 037222 010137 004600 MOV R1,@REGADR ;FAILING REGISTER RHCS1
7545 037226 104021 ERROR 21 ;DURING ABOVE OPERATION ONLY
7546 037230 012746 010500 67$: MOV #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
7547 037234 011637 001124 MOV (SP)+,@SGDDAT ;SAVE FOR PRINTOUT
7548 037240 022605 CMP (SP)+,R5 ;DURING ABOVE OPERATION ONLY MOL.DPR.VV
7549 037242 001405 BEQ 69$ ;BRANCH IF GOOD
7550 037244 010537 001126 MOV R5,@SBDDAT ;BAD DATA
7551 037250 010337 004600 MOV R3,@REGADR ;FAILING REGISTER RHDS1
7552 037254 104063 ERROR 63 ;DURING ABOVE OPERATION ONLY
7553
7554 ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
7555
7556 037256 004037 041360 JSR R0,@FILLRE ;MOV 0 INTO SAVED RHWC
7557 037262 002272 RHWC ;SAVED REGISTER TO CHANGE
7558 037264 000000 0 ;DATA
7559 037266 004037 041360 JSR R0,@FILLRE ;MOV WRFROM+<256.*2> INTO SAVED RHBA
7560 037272 002274 RHBA ;SAVED REGISTER TO CHANGE
7561 037274 003470 WRFROM+<256.*2> ;DATA
7562 037276 004037 041360 JSR R0,@FILLRE ;MOV 3405 INTO SAVED RHDST
7563 037302 002304 RHDST ;SAVED REGISTER TO CHANGE
7564 037304 003405 3405 ;DATA
7565
7566 ;*COMPARE REGISTERS BEFORE WRITE CHECK HEADER AND DATA
7567 ;*WITH REGISTER AFTER COMMAND
7568 037306 004037 042444 JSR R0,@COMREG ;COMPARE SAVED REGISTERS WITH
7569 037312 004612 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
7570 037314 002354 WC ;TEST DATA STARTING FROM 'RHWC'
7571 037316 000022 18. ;18. REGISTERS TO BE COMPARED
7572 037320 037324 8$ ;RETURN TO 8$ ON ERROR
7573 037322 037330 9$ ;RETURN TO 9$ ON NO ERROR
7574
7575 037324 104076 8$: ERROR 76 ;WHILE USING UNIBUS B
7576 ;WRITE CHECK CAUSED
  
```

CZ
CZ

7577 037326 000207

7578
7579
7580
7581
7582
7583
7584
7585
7586
7587
7588
7589

RTS PC

:AN IMPROPER REGISTER
:CHANGE
:GOOD DATA GIVES WHAT
:SHOULD BE THERE
:RECEIVED DATA GIVES WHAT
:WAS THERE AFTER COMMANDS

037330

98:

```
7590
7591 037330 000004          TST50: SCOPE
7592
7593 037332 005737 004724  TST    @#NOPUSH      ;IS THIS A 220 START ?
7594 037336 001007          BNE    1$            ;SKIP THIS TEST IF SO
7595 037340 005737 000042  TST    @#42          ;MONITOR (ACT 11) RETURN ADDRESS ?
7596 037344 001004          BNE    1$            ;SKIP THIS TEST
7597 037346 005737 001100  TST    @#SPASS       ;FIRST PASS ?
7598 037352 001001          BNE    1$            ;SKIP THIS TEST IF NOT
7599 037354 000402          BR     2$            ;CONTINUE WITH THIS TEST
7600 037356 000137 040554  JMP    TST51 ;      ; JUMP TO NEXT TEST -----)
7601 037362 012706 001000  MOV    #STACK,SP    ;RESET STACK
7602 037366 012737 000050  MOV    #50,@#TSTNM  ;SAVE TEST NUMBER
7603 037374 004737 041456  JSR    PC,@#CLDISK  ;SET R1-RHCS1, R2-RHCS2
7604
7605          ;*FILL SECTOR 0, TRACK 0, CYL 0 WITH ONES
7606          ;*FILL WRITE FROM BUFFER
7607 037400 004037 041326  JSR    R0,@#CLAREA  ;CLEAR 256. WORDS, FROM WRFROM
7608 037404 002470          WRFROM              ;STARTING FROM WRFROM
7609 037406 000400          256.              ;256. WORDS
7610 037410 177777          -1                ;FILL WITH -1
7611
7612          ;*FILL WRITE DATA COMMAND
7613 037412 004037 043430  JSR    R0,@#R1UN    ;SETUP TO RUN FOR DATA COMMAND
7614 037416 000600          0                ;CYLINDER 0
7615 037420          000              ;SECTOR 0
7616 037421          000              ;TRACK 0
7617 037422 177400          -256.            ;WORD COUNT = 256.
7618 037424 002470          WRFROM            ;BUS ADDRESS
7619 037426 000000          0                ;DO NOT INHIBIT BUS ADDRESS INCREMENT
7620 037430 010000          FMT22            ;16 BITS PER WORD FORMAT
7621 037432 002442          WRIDAT           ;GET READY TO DO A WRIDAT
7622
7623 037434 004737 041536  JSR    PC,@#CHECKT  ;CHECK DVA,RDY,MOL,DPR,DRY,VV - 1
7624 037440 104401 066720  TYPE    ,CPHALT    ;CANNOT CONTINUE TESTING IF ANY OF
7625 037444 000000          HALT            ;STOP
7626
7627 037446 013777 004606 142612  MOV    @#RP4VEC,@#RPVEC ;SET RP04 VECTOR ADDRESS
7628
7629 037454 013746 002442  MOV    @#WRIDAT,-(SP) ;GET READY TO MOVE COMMAND
7630 037460 052716 000101  BIS    #GO!IE,(SP)  ;GET READY TO SET 'GO' AND
7631 037464 012677 142610  MOV    (SP)+,@#RHCS1 ;GO WITH
7632
7633          ;*TIME IS NOT CRITICAL
7634 037470 104413          WAT              ;WAIT FOR RDY BIT TO SET
7635 037472 002300          RHCS1           ;WAIT FOR RHCS1 REGISTER
7636 037474 000200          RDY            ;WAIT FOR RDY BIT IN RHCS1 REGISTER
7637 037476 004704          2500.          ;ALLOW 25000 MICRO SECONDS
7638 037500 004704          2500.          ;RDY MUST SET BETWEEN
7639
7640          ;*SAVE REGISTERS FOR COMPARISON AFTER WRITE PROTECT
7641          ;*BUTTON HAS BEEN HIT
7642
7643 037502 004037 041624  JSR    R0,@#SAVER   ;SAVE REGISTERS
7644 037506 002272          RHWC            ;RHWC IS THE FIRST REGISTER SAVED
7645 037510 004612          SAVERE         ;STARTING ADDRESS OF WHERE
```

```

7646 037512 000022          18.          ;NUMBER OF REGISTERS
7647
7648 037514 104401 037522  TYPE      ,68$          ;;TYPE ASCIZ STRING
7649 037520 000407          BR      67$          ;;GET OVER THE ASCIZ
7650 037540 013746 004716  MOV      @#UNIT,-(SP)          ;GET UNIT UNDER TEST
7651 037544 104405          TYPDS
7652 037546 104401 037554  TYPE      ,70$          ;;TYPE ASCIZ STRING
7653 037552 000442          BR      69$          ;;GET OVER THE ASCIZ
7654 037660 000000          HALT
7655          ;*THE ONLY REGISTER THAT SHOULD CHANGE IS RHDS1 - BIT #11
7656          ;*-WRL
7657 037662 004037 042336  JSR      RO,@#CHREG          ;CHANGE BITS IN SAVED REGISTER
7658 037666 002322          RHDS1          ;CHANGE RHDS1 REGISTER
7659 037670 000001          1          ;1 BIT/BITS TO BE CHANGED
7660 037672 000001          1          ;NEW VALUE OF WRL IS 1
7661 037674 004000          WRL          ;CHANGE WRL BIT
7662
7663          ;*COMPARE ALL REGISTERS BEFORE WRITE WAS LOCKED
7664          ;*OUT WITH REGISTER VALUES AFTER WRITE WAS LOCKED OUT
7665 037676 004037 042444  JSR      RO,@#COMREG          ;COMPARE SAVED REGISTERS WITH
7666 037702 004612          SAVERE          ;GOOD DATA SAVED IN 'SAVERE'
7667 037704 002354          WC          ;TEST DATA STARTING FROM 'RHWC'
7668 037706 000022          18.          ;18. REGISTERS TO BE COMPARED
7669 037710 037714          3$          ;RETURN TO 3$ ON ERROR
7670 037712 037720          4$          ;RETURN TO 4$ ON NO ERROR
7671
7672 037714 104041          3$:  ERROR  41          ;LOCKING OUT WRITE BY
7673 037716 000207          RTS      PC          ;WRITE LOCK BUTTON CAUSED
7674          ;IMPROPER REGISTER CHANGE
7675          ;GOOD DATA GIVES WHAT SHOULD
7676          ;BE THERE
7677          ;RECEIVED DATA GIVES WHAT
7678          ;WAS THERE AFTER WRITE
7679          ;WAS LOCKED OUT BY BUTTON
7680
7681          ;*NOW A WRITE WILL BE ATTEMPTED WITH WRITE LOCKED
7682          ;*OUT BY BUTTON
7683          ;*FILL WRITE FROM BUFFER WITH 377
7684
7685 037720 013737 037720 001110 4$:  MOV      @#4$,@#SLPERR          ;SCOPE LOOP STARTS FROM HERE
7686
7687 037726 004737 041456  JSR      PC,@#CLDISK          ;SET R1-RHCS1, R2-RHCS2
7688 037732 004037 041326  JSR      RO,@#CLAREA          ;CLEAR 256. WORDS, FROM WRFROM
7689 037736 002470          WRFROM          ;STARTING FROM WRFROM
7690 037740 000400          256.          ;256. WORDS
7691 037742 000377          377          ;FILL WITH 377
7692
7693          ;*TRY A ONE WORD WRITE
7694 037744 004037 043430  JSR      RO,@#RUN          ;SETUP TO RUN FOR DATA COMMAND
7695 037750 000000          0          ;CYLINDER 0
7696 037752 000          .BYTE 0          ;SECTOR 0
7697 037753 000          .BYTE 0          ;TRACK 0
7698 037754 177777          -1.          ;WORD COUNT - 1.
7699 037756 002470          WRFROM          ;BUS ADDRESS
7700 037760 000000          0          ;DO NOT INHIBIT BUS ADDRESS INCREMENT
7701 037762 010000          FMT22          ;16 BITS PER WORD FORMAT
  
```

```

7702 037764 002442 WRIDAT ;GET READY TO DO A WRIDAT
7703
7704 ;*SAVE REGISTERS
7705 037766 004037 041624 JSR RO,@#SAVER ;SAVE REGISTERS
7706 037772 002272 RHW C ;RHW C IS THE FIRST REGISTER SAVED
7707 037774 004612 SAVERE ;STARTING ADDRESS OF WHERE
7708 037776 000022 18. ;NUMBER OF REGISTERS
7709
7710 040000 013777 004606 142260 MOV @#RP4VEC,@#RPVEC ;SET RP04 VECTOR ADDRESS
7711
7712 040006 013746 002442 MOV @#WRIDAT,-(SP) ;GET READY TO MOVE COMMAND
7713 040012 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET 'GO' AND
7714 040016 012677 142256 MOV (SP)+,@#RHCS1 ;GO WITH
7715 ;*TIME IS NOT CRITICAL
7716 040022 104413 WAT ;WAIT FOR RDY BIT TO SET
7717 040024 002300 RHCS1 ;WAIT FOR RHCS1 REGISTER
7718 040026 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
7719 040030 004704 2500. ;ALLOW 25000 MICRO SECONDS
7720 040032 004704 2500. ;RDY MUST SET BETWEEN
7721
7722 ;*CHANGE SAVED REGISTERS TO EXPECTED VALUE
7723
7724 040034 017737 142232 004612 MOV @#RHW C,@#SAVERE ;RHW C IS UNPREDICTABLE
7725 040042 017737 142226 004614 MOV @#RHBA,@#SAVERE+2 ;RHBA IS UNPREDICTABLE
7726 040050 017746 142222 MOV @#RHCS2,-(SP) ;GET RHCS2
7727 040054 042716 177477 BIC #*(IR!OR),(SP) ;KEEP IR AND OR
7728 040060 042737 000300 004616 BIC #IR!OR,@#SAVERE+4 ;CLEAR SAVED IR OR
7729 040066 052637 004616 BIS (SP)+,@#SAVERE+4 ;SET OR IR AS REQUIRED
7730 040072 004037 042336 JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
7731 040076 002300 RHCS1 ;CHANGE RHCS1 REGISTER
7732 040100 000002 2 ;2 BIT/BITS TO BE CHANGED
7733 040102 000001 1 ;NEW VALUE OF SC IS 1
7734 040104 100000 SC ;CHANGE SC BIT
7735 040106 000001 1 ;NEW VALUE OF TRE IS 1
7736 040110 040000 TRE ;CHANGE TRE BIT
7737 040112 004037 041360 JSR RO,@#FILLRE ;MOV 1 INTO SAVED RHDST
7738 040116 002304 RHDST ;SAVED REGISTER TO CHANGE
7739 040120 000001 1 ;DATA
7740 040122 004037 042336 JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
7741 040126 002302 RHER1 ;CHANGE RHER1 REGISTER
7742 040130 000001 1 ;1 BIT/BITS TO BE CHANGED
7743 040132 000001 1 ;NEW VALUE OF WLE IS 1
7744 040134 004000 WLE ;CHANGE WLE BIT
7745 040136 004037 042336 JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
7746 040142 002322 RHDS1 ;CHANGE RHDS1 REGISTER
7747 040144 000002 2 ;2 BIT/BITS TO BE CHANGED
7748 040146 000001 1 ;NEW VALUE OF ATA IS 1
7749 040150 100000 ATA ;CHANGE ATA BIT
7750 040152 000001 1 ;NEW VALUE OF ERR IS 1
7751 040154 040000 ERR ;CHANGE ERR BIT
7752 040156 053737 004740 004636 BIS @#ATTENT,@#SAVERE+24 ;SET APPROPRIATE 'ATA' BITS
7753
7754 ;*COMPARE REGISTERS BEFORE WRITE WAS ATTEMPTED
7755 ;*WITH REGISTERS AFTER ATTEMPT
7756
7757 040164 004037 042444 JSR RO,@#COMREG ;COMPARE SAVED REGISTERS WITH

```

```

7758 040170 004612 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
7759 040172 002354 WC ;TEST DATA STARTING FROM 'RHWC'
7760 040174 000022 18. ;18. REGISTERS TO BE COMPARED
7761 040176 040202 5$ ;RETURN TO 5$ ON ERROR
7762 040200 040206 6$ ;RETURN TO 6$ ON NO ERROR
7763
7764 040202 104042 5$: ERROR 42 ;ATTEMPTING TO WRITE WITH
7765 040204 000207 RTS PC ;WRITE LOCKED OUT
7766 ;CAUSED IMPROPER REGISTER
7767 ;CHANGE
7768 ;GOOD DATA GIVES WHAT SHOULD
7769 ;BE THERE
7770 ;RECEIVED DATA GIVES WHAT WAS
7771 ;THERE AFTER ATTEMPT
7772
7773 ;*NOW A READ WILL BE DONE TO DETERMIN THAT
7774 ;*READS CAN BE DONE WITH WRITE LOCKED OUT AND
7775 ;*THAT NO DATA ON DISK GOT CHANGED, BUT FIRST CLEAR ERROR
7776
7777 040206 6$:
7778 040206 004737 041456 JSR PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
7779
7780 ;*FILL READ INTO BUFFER WITH 0
7781 040212 004037 041326 JSR RO,@#CLAREA ;CLEAR 256. WORDS, FROM REINTO
7782 040216 003534 REINTO ;STARTING FROM REINTO
7783 040220 000400 256. ;256. WORDS
7784 040222 000000 0 ;FILL WITH 0
7785
7786 ;*FILL WRITE FROM BUFFER WITH EXPECTED DATA FROM READ
7787 040224 004037 041326 JSR RO,@#CLAREA ;CLEAR 256. WORDS, FROM WRFROM
7788 040230 002470 WRFROM ;STARTING FROM WRFROM
7789 040232 000400 256. ;256. WORDS
7790 040234 177777 -1 ;FILL WITH -1
7791
7792 ;*FILL COMMAND
7793 040236 004037 043430 JSR RO,@#RUN ;SETUP TO RUN FOR DATA COMMAND
7794 040242 000000 0 ;CYLINDER 0
7795 040244 000 ;SECTOR 0
7796 040245 000 ;TRACK 0
7797 040246 177400 -256. ;WORD COUNT = 256.
7798 040250 003534 REINTO ;BUS ADDRESS
7799 040252 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
7800 040254 014000 ECI!FMT22 ;16 BITS PER WORD FORMAT
7801 040256 002446 READAT ;GET READY TO DO A READAT
7802
7803 040260 013777 004606 142000 MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
7804
7805 040266 013746 002446 MOV @#READAT,-(SP) ;GET READY TO MOVE COMMAND
7806 040272 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET 'GO' AND
7807 040276 012677 141776 MOV (SP)+,@RHCS1 ;GO WITH
7808
7809 040302 104413 WAT ;WAIT FOR RDY BIT TO SET
7810 040304 002300 RHCS1 ;WAIT FOR RHCS1 REGISTER
7811 040306 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
7812 040310 001614 908. ;ALLOW 9080 MICRO SECONDS
7813 040312 001507 839. ;RDY MUST SET BETWEEN

```

```
7814
7815
7816 040314 004037 043474
7817 040320 002470
7818 040322 003534
7819 040324 000400
7820 040326 040332
7821 040330 040334
7822
7823 040332 104043 7$: ERROR 43
7824
7825
7826
7827
7828
7829
7830
7831
7832
7833
7834
7835
7836 040334 8$:
7837 040334 004037 041624 JSR RO,@#SAVER ;SAVE REGISTERS
7838 040340 002272 RHWC ;RHWC IS THE FIRST REGISTER SAVED
7839 040342 004612 SAVERE ;STARTING ADDRESS OF WHERE
7840 040344 000022 18. ;NUMBER OF REGISTERS
7841
7842 040346 ST20:
7843 040346 104401 040354 TYPE ,65$ ;;TYPE ASCIZ STRING
7844 040352 000407 BR 64$ ;;GET OVER THE ASCIZ
7845 040372 013746 004716 MOV @#UNIT,-(SP) ;GET UNIT UNDER TEST
7846 040376 104405 TYPDS
7847 040400 104401 040406 TYPE ,67$ ;;TYPE ASCIZ STRING
7848 040404 000440 BR 66$ ;;GET OVER THE ASCIZ
7849 040506 000000 HALT
7850
7851
7852 040510 004037 042336 ;*THE ONLY BIT THAT SHOULD CHANGE IS WRL-BIT #11 IN RHDS1
7853 040514 002322 JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
7854 040516 000001 RHDS1 ;CHANGE RHDS1 REGISTER
7855 040520 000000 1 ;1 BIT/BITS TO BE CHANGED
7856 040522 004000 0 ;NEW VALUE OF WRL IS 0
7857
7858 ;*COMPARE ALL REGISTERS BEFORE WRITE LOCK WAS UNLOCKED
7859 ;*WITH REGISTERS AFTER WRITE WAS UNLOCKED
7860
7861 040524 004037 042444 JSR RO,@#COMREG ;COMPARE SAVED REGISTERS WITH
7862 040530 004612 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
7863 040532 002354 WC ;TEST DATA STARTING FROM 'RHWC'
7864 040534 000022 18. ;18. REGISTERS TO BE COMPARED
7865 040536 040542 9$ ;RETURN TO 9$ ON ERROR
7866 040540 040546 10$ ;RETURN TO 10$ ON NO ERROR
7867
7868 040542 104044 9$: ERROR 44 ;UNLOCKING WRITES BY WRITE
7869 040544 000207 RTS PC ;LOCK BUTTON CAUSED AN ERROR
```

7870 :GOOD DATA GIVES WHAT SHOULD
7871 :BE THERE
7872 :RECEIVED DATA GIVES WHAT WAS
7873 :THERE AFTER WRITES WERE
7874 :UNLOCKED
7875 :ON THIS ERROR NO LOOPING IS RECOMMENDED
7876 :JUST A HALT ON ERROR WILL DO THE SAME
7877 :THING AS ONLY THE REGISTERS ARE READ
7878 040546 012737 177777 047274 10\$: MOV #-1,@PRITEM :CLEAR PREVIOUS ITEM NUMBER


```

7879
7880
7881 040554 000004          TST51: SCOPE
7882 040556 012737 000001 001212  MOV #1,$TIMES      ;;DO 1 ITERATION
7883 040564 012737 000000 177776  MOV #0,$PS        ;REINSTATE PS TO 0
7884
7885 040572 104401 040600          TYPE ,65$         ;;TYPE ASCIZ STRING
7886 040576 000424          BR 64$           ;;GET OVER THE ASCIZ
7887 040650 013746 004716          MOV @#UNIT,-(SP)  ;GET READY TO TYPE UNIT NUMBER
7888 040654 104405          TYPDS
7889 040656 104401 040664          TYPE ,67$         ;;TYPE ASCIZ STRING
7890 040662 000402          BR 66$           ;;GET OVER THE ASCIZ
7891 040670 033746 001112          MOV @#ERTTL,-(SP) ;GET READY TO TYPE NUMBER OF ERRORS
7892 040674 104405          TYPDS
7893 040676 005037 001112          CLR @#ERTTL      ;CLEAR TOTAL NUMBER OF ERRORS
7894 040702 005037 001102          CLR @#TSTNM      ;CLEAR TEST NUMBER
7895 040706 005737 004726          TST @#SELECT     ;STARTING FROM 200 ?
7896 040712 001413          BEQ 3$           ;CHECK NEXT DRIVE IF SO
7897
7898
7899 040714 005237 001100          INC @#$PASS      ;INCREASE PASS COUNT
7900 040720 104401 041111          TYPE ,SENDMG     ;TYPE 'END PASS #'
7901 040724 013746 001100          MOV @#$PASS,-(SP)
7902 040730 104405          TYPDS
7903 040732 104401 041106          TYPE ,SENULL
7904 040736 000137 010034          JMP @#TST4       ;JUMP TEST 4 ----->
7905
7906 040742 012737 177777 047274 3$: MOV #-1,@#PRITEM  ;CLEAR PREVIOUS ITEM NUMBER
7907 040750 005337 004720          DEC @#NUNITS     ;NO. OF UNITS PRESENT
7908 040754 001413          BEQ $EOP         ;BRANCH IF ALL DRIVES COMPLETE
7909 040756 013700 004716          MOV @#UNIT,R0   ;UNIT UNDER TEST
7910 040762 012701 004676          MOV #UNITS,R1   ;TABLE POINTER
7911 040766 022100          1$: CMP (R1)+,R0 ;IS THIS UNIT JUST TESTED ?
7912 040770 001401          BEQ 2$          ;BRANCH IF YES
7913 040772 000775          BR 1$           ;BRANCH IF NO
7914 040774 011137 004716          2$: MOV (R1),@#UNIT ;MAKE THIS NEXT UNIT
7915 041000 000137 010034          JMP @#TST4       ;TEST THE NEXT DRIVE ----->
7916
  
```

```

7917                                     .SBTTL
7918                                     .SBTTL **SUBROUTINES**
7919                                     .SBTTL
7920
7921
7922 041004 000004                               SCOPE
7923 041006 005037 001102                       CLR $STNM          ;;ZERO THE TEST NUMBER
7924 041012 005037 001212                       CLR $TIMES        ;;ZERO THE NUMBER OF ITERATIONS
7925 041016 005237 001100                       INC $PASS         ;;INCREMENT THE PASS NUMBER
7926 041022 042737 100000 001100               BIC #100000,$PASS ;;DON'T ALLOW A NEG. NUMBER
7927 041030 005327                               DEC (PC)+         ;;LOOP?
7928 041032 000001                               $EOPCT: .WORD 1
7929 041034 003022                               BGT $DOAGN        ;;YES
7930 041036 012737                               MOV (PC)+,@(PC)+ ;;RESTORE COUNTER
7931 041040 000001                               $ENDCT: .WORD 1
7932 041042 041032                               $EOPCT
7933 041044 104401 041111                       TYPE ,SENDMG      ;;TYPE 'END PASS #'
7934 041050 013746 001100                       MOV $PASS,-(SP)   ;;SAVE $PASS FOR TYPEOUT
7935 041054 104405                               TYPDS             ;;GO TYPE--DECIMAL ASCII WITH SIGN
7936 041056 104401 041106                       TYPE ,SENULL      ;;TYPE A NULL CHARACTER
7937 041062 013700 000042                       $GET42: MOV @#42,R0 ;;GET MONITOR ADDRESS
7938 041066 001405                               BEQ $DOAGN        ;;BRANCH IF NO MONITOR
7939 041070 000005                               RESET            ;;CLEAR THE WORLD
7940 041072 004710                               $ENDAD: JSR PC,(R0) ;;GO TO MONITOR
7941 041074 000240                               NOP              ;;SAVE ROOM
7942 041076 000240                               NOP              ;;FOR
7943 041100 000240                               NOP              ;;ACT11
7944 041102 000137                               JMP @(PC)+       ;;RETURN
7945 041104 006312                               $RTNAD: .WORD TST1
7946 041106 377 377 000                          $ENULL: .BYTE -1,-1,0 ;;NULL CHARACTER STRING
7947 041111 015 042412 042116                       $ENDMG: .ASCIIZ <15><12>/END PASS #/
7948 041116 050040 051501 020123
7949 041124 000043
7950
    
```

```

7951
7952
7953           .SBTTL  JAM CURRENT CYLINDER ROUTINE
7954
7955           ;*THIS ROUTINE WILL CHANGE THE CURRENT CYLINDER REGISTER - 'RHCC'
7956           ;*BY GIVING A 'SEEK' COMMAND FOLLOWED BY AN INIT WHICH WILL LOAD
7957           ;*'RHCC' WITH THE DESIRED CYLINDER VALUE.  THE ROUTINE THEN CHECKS
7958           ;*THAT THE LOADED VALUE IS CORRECT.
7959           ;*
7960           ;*CALL IS:
7961           ;*      JSR      R0,@#MAKECYL
7962           ;*      XC
7963           ;*      XC
7964           ;*      XC
7965           ;*      XC
7966           ;*      XC
7967           ;*      XC
7968           ;*      XC
7969           ;*      XC
7970           ;*      XC
7971           ;*      XC
7972           ;*      XC
7973           ;*      XC
7974           ;*      XC
7975           ;*      XC
7976           ;*      XC
7977           ;*      XC
7978           ;*      XC
7979           ;*      XC
7980           ;*      XC
7981           ;*      XC
7982           ;*      XC
7983           ;*      XC
7984           ;*      XC
7985           ;*      XC
7986           ;*      XC
7987           ;*      XC
7988           ;*      XC
7989           ;*      XC
7990           ;*      XC
7991           ;*      XC
7992           ;*      XC
7993           ;*      XC
7994           ;*      XC
7995           ;*      XC
7996           ;*      XC
7997           ;*      XC
7998           ;*      XC
7999           ;*      XC
8000           ;*      XC
8001           ;*      XC
8002           ;*      XC
  
```

```

8003
8004      ;*THIS FILLS MEMORY WITH GIVEN DATA
8005      ;*USED CHIEFLY FOR HEADER INFORMATION
8006      ;*CALL IS
8007      ;*      JSR      RO,@#FLHEAD      ;FILL HEADER
8008      ;*      LOC      ;LOCATION WHERE SAVED
8009      ;*      XN      ;NUMBER OF WORDS
8010      ;*      XD1     ;DATA REPEATED XN TIMES
8011      ;*      XD2     ;DATA REPEATED XN TIMES
8012      ;*
8013      ;*
8014      ;*
8015
8016      041302      FLHEAD:
8017      041302      010146      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
8018      041304      010246      MOV      R2,-(SP)      ;;PUSH R2 ON STACK
8019      041306      012001      MOV      (R0)+,R1      ;R1 HAS ADDRESS OF WHERE TO SAVE
8020      041310      012002      MOV      (R0)+,R2      ;R2 HAS NUMBER OF WORDS
8021
8022      ;*NOW FILL DATA
8023
8024      041312      012021      1$:      MOV      (R0)+,(R1)+      ;SAVE DATA
8025      041314      005302      DEC      R2      ;DECREMENT COUNT
8026      041316      001375      BNE     1$      ;BRANCH IF INCOMPLETE
8027      041320      012602      MOV      (SP)+,R2      ;;POP STACK INTO R2
8028      041322      012601      MOV      (SP)+,R1      ;;POP STACK INTO R1
8029      041324      000200      RTS      R0
8030
8031
8032
8033      ;*THIS CLEARS ANY BLOCK OF MEMORY.
8034      ;*FILLING IT WITH ANY DATA
8035      ;*CALL IS
8036      ;*      JSR      RO,@#CLAREA
8037      ;*      F      ;FROM
8038      ;*      N      ;NUMBER OF WORDS
8039      ;*      D      ;DATA TO BE FILLED
8040      ;*
8041      ;*R1 WILL HAVE STARTING ADDRESS OF BLOCK TO BE FILLED
8042      ;*R2 WILL HAVE NUMBER OF WORDS
8043      ;*R3 WILL HAVE DATA
8044
8045      041326      CLAREA:
8046      041326      010146      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
8047      041330      010246      MOV      R2,-(SP)      ;;PUSH R2 ON STACK
8048      041332      010346      MOV      R3,-(SP)      ;;PUSH R3 ON STACK
8049      041334      012001      MOV      (R0)+,R1      ;FROM
8050      041336      012002      MOV      (R0)+,R2      ;NUMBER
8051      041340      012003      MOV      (R0)+,R3      ;DATA
8052      041342      010321      1$:      MOV      R3,(R1)+      ;MOVE DATA
8053      041344      005302      DEC      R2      ;COUNT
8054      041346      001375      BNE     1$      ;BRANCH IF NOT COMPLETE
8055      041350      012603      MOV      (SP)+,R3      ;;POP STACK INTO R3
8056      041352      012602      MOV      (SP)+,R2      ;;POP STACK INTO R2
8057      041354      012601      MOV      (SP)+,R1      ;;POP STACK INTO R1
8058      041356      000200      RTS      R0      ;RETURN TO MAIN PROGRAM

```

8059
8060
8061
8062
8063
8064
8065
8066
8067
8068
8069
8070
8071
8072
8073
8074
8075
8076
8077
8078
8079
8080
8081
8082
8083
8084
8085

041360
041360 010146
041362 010246
041364 012001
041366 012002
041370 162701 002272
041374 010261 004612
041400 012602
041402 012601
041404 000200

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

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

```

8086 ;*THIS SUBROUTINE SETS UP FOR SEARCH
8087 ;*CALL IS
8088 ;* JSR R0,@#SRCH
8089 ;* C ;CYLINDER
8090 ;*.BYTE S ;SECTOR
8091 ;*.BYTE T ;TRACK
8092
8093 041406 012077 140700 SRCH: MOV (R0)+,@RHCA ;SET DESIRED CYLINDER ADDRESS
8094 041412 012077 140666 MOV (R0)+,@RHDST ;SET DESIRED SECTOR/TRACK ADDRESS
8095 041416 013777 002434 140654 MOV @#SERCH,@RHCS1 ;GET READY FOR SEARCH
8096 ;WITH 30 IN RHCS1
8097 041424 000200 RTS R0
8098
8099
8100
8101
8102
8103
8104
8105 ;*THIS SUBROUTINE SETS UP FOR SEEK COMMANDS
8106 ;*CALL IS
8107 ;* JSR R0,@#SEEKCY
8108 ;* C ;CYLINDER
8109 ;*
8110
8111 041426 012077 140660 SEEKCY: MOV (R0)+,@RHCA ;SET DESIRED CYLINDER ADDRESS
8112 041432 013777 002452 140640 MOV @#SEECOM,@RHCS1 ;MOV 4 INTO RHCS1
8113 041440 000200 RTS R0 ;RETURN TO MAIN PROGRAM
  
```

```
8114
8115      ;*THIS SUBROUTINE SETS UP FOR OFFSET COMMANDS
8116      ;*CALL IS
8117      ;*   JSR   RO,@#OFFSET
8118      ;*   0           ;MICRO INCHES OFSET
8119
8120 041442 052077 140642      OFFSET: BIS   (RO)+,@RHOF      ;SET OFSET REGISTER
8121 041446 013777 002454 140624  MOV   @#OFFSEC,@RHCS1 ;MOV14 INTO RHCS1
8122 041454 000200          RTS   RO           ;RETURN TO MAIN PROGRAM
8123
8124
8125 041456 013701 002300      CLDISK: MOV   @#RHCS1,    R1      ;R1 WILL BE CONTROL AND STATUS1
8126 041462 013702 002276      MOV   @#RHCS2,    R2      ;R2 WILL BE CONTROL AND STATUS2
8127 041466 013703 002322      MOV   @#RHDS1,    R3      ;R3 WILL BE DISK STATUS REGISTER1
8128 041472 013704 002302      MOV   @#RHER1,    R4      ;R4 WILL BE ERROR REGISTER #1
8129
8130 041476 012712 000040      MOV   #CLR,@R2      ;CLEAR ALL REG.
8131 041502 013712 004716      MOV   @#UNIT,@R2   ;REINSTATE UNIT NO.
8132 041506 005011          CLR   @R1           ;CLEAR FUNCTION BITS
8133 041510 000207          RTS   PC
```

```
8134
8135
8136
8137
8138 ;*THIS CHECKS THAT DEVICE AVAILABLE (DVA) AND READY (RDY) IN RHCS1 1
8139 ;*AND CHECKS MEDIUM ON LINE (MOL), DEVICE PRESENT (DPR), DEVICE READY
8140 ;* (DRY) IN RHDS1 = 1
8141
8142 ;*IT ALSO CHECKS THAT THERE ARE NO BITS STUCK AT 1 IN RHDS1
8143
8144
8145 041512 000000 PCJSR: 0 ;PC OF JSR
8146
8147 041514 011637 041512 CHECK: MOV (SP),@#PCJSR ;SAVE PC OF JSR+4
8148 041520 162737 000004 041512 SUB #4,@#PCJSR ;GET PC OF JSR
8149 041526 011346 MOV @R3,-(SP) ;GET RHDS1
8150 041530 052716 000100 BIS #VV,(SP) ;DONT CHECK VV BIT
8151 041534 000406 BR CHECKC ;GOTO COMMON CHECK ROUTINE
8152
8153 041536 011637 041512 CHECKT: MOV (SP),@#PCJSR ;SAVE PC OF JSR+4
8154 041542 162737 000004 041512 SUB #4,@#PCJSR ;GET PC OF JSR
8155 041550 011346 MOV @R3,-(SP) ;GET RHDS1 & DO VV CHECK AT 3$
8156
8157 041552 011146 CHECKC: MOV @R1,-(SP) ;GET CS1
8158 041554 042716 173577 BIC #173577,(SP) ;CLEAR UNWANTED BITS
8159 041560 022726 004200 CMP #DVA!RDY,(SP)+ ;RHCS1 SHOULD HAVE DEVICE AVAILABLE
8160 ;AND BE READY
8161 041564 001403 BEQ 3$ ;BRANCH IF IT DOES
8162 041566 011137 001122 MOV @R1,@#SBDADR ;BAD DATA REGISTER (RHCS1)
8163 041572 104061 ERROR 61 ;RHCS1 DID NOT HAVE DEVICE
8164 ;AVAILABLE RIGHT AT THE START
8165 ;ALL OTHER BITS SHOULD BE 0
8166
8167 041574 042716 102000 3$: BIC #ATA!LBT,(SP) ;CLEAR UNWANTED BITS
8168 041600 022726 010700 CMP #MOL!DPR!DRY!VV,(SP)+ ;RHDS1 SHOULD HAVE THESE SET
8169 041604 001404 BEQ 7$ ;BRANCH IF GOOD
8170 041606 011337 001122 MOV @R3,@#SBDADR ;BAD DATA IN REGISTER (RHDS1)
8171 041612 104062 ERROR 62 ;RHDS1 HAS SOME BITS OTHER
8172 ;THAN MOL, DRY, DPR,VV SET
8173 ;ALL OTHER BITS SHOULD BE 0
8174 041614 000207 RTS PC ;RETURN TO TEST AND HALT/CONTINUE
8175 ;DEPENDING ON WHETHER THIS IS A
8176 ;'FATAL' ERROR
8177
8178 041616 062716 000006 7$: ADD #6,(SP) ;ADJUST STACK TO JUMP OVER HALT IN TEST
8179 041622 000207 RTS PC ;RETURN TO THE TEST AND CONTINUE
8180
```


8181
8182
8183
8184
8185
8186
8187
8188
8189
8190
8191
8192 041624
8193 041624 010146
8194 041626 010246
8195 041630 010346
8196 041632 012001
8197 041634 012002
8198 041636 012003
8199 041640 013122
8200 041642 005303
8201 041644 001375
8202 041646 012603
8203 041650 012602
8204 041652 012601
8205 041654 000200
8206
8207
8208
8209
8210
8211
8212
8213
8214
8215
8216
8217
8218
8219
8220
8221
8222
8223
8224
8225
8226
8227 041656 005077 140462
8228 041662 017737 140462 041714
8229 041670 017737 140440 004664
8230 041676 017737 140434 004662
8231 041704 000002
8232
8233
8234
8235
8236

```
      ;*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
```

```

8237      : *      WAT
8238      : *      A      ; ABSOLUTE REGISTER ADDRESS
8239      : *      B      ; BIT WAITED FOR
8240      : *      TA     ; TIME ALLOWED GIVEN IN 10 MICROSEC
8241      : *      TO     ; TOLERANCE PLUS/MINUS IN 10 MICROSEC
8242      : *
8243      : *R1-WILL HAVE TIME ALLOWED IN 10 MICRO SECONDS
8244      : *R2-WILL HAVE TOLERANCE PLUS/MINUS IN 10 MICRO SECONDS
8245      : *MINIMUM TIME THAT CAN BE MEASURED IS ABOUT 12 MICRO SECONDS
8246      : *FOR THE SLOWEST PROCESSOR
8247
8248 041706 000000      WAITPC: 0      ; WAT PC
8249 041710 000000      WAITRE: 0      ; WAIT ON REGISTER ADDRESS
8250 041712 000000      WAITBT: 0      ; WAIT ON BIT
8251 041714 000000      WAITTM: 0      ; WAITED TIME
8252 041716 005037 041714      WAIT.P: CLR @WAITTM      ; CLEAR WAITED TIME
8253 041722 005077 140420      CLR @PCLBUF ; CLEAR COUNT SET BUFFER
8254 041726 012777 000021 140410      MOV #GO!BIT4,@PCLCSR ; COUNT UP, 100 KHZ, START CLOCK
8255 041734 010046      MOV R0,-(SP)      ; PUSH R0 ON STACK
8256 041736 010146      MOV R1,-(SP)      ; PUSH R1 ON STACK
8257 041740 010246      MOV R2,-(SP)      ; PUSH R2 ON STACK
8258 041742 010346      MOV R3,-(SP)      ; PUSH R3 ON STACK
8259 041744 016600 000010      MOV 10(SP),R0      ; R0 HAS ADDRESS OF NEXT LOCATION
8260 041750 010037 041706      MOV R0,@WAITPC      ; NOW WAITPC HAS WAT PC + 2
8261 041754 162737 000002 041706      SUB #2,@WAITPC      ; WAT PC IS IN WAITPC
8262 041762 013037 041710      MOV @ (R0)+,@WAITRE ; WAIT ON REGISTER ADDRESS
8263 041766 012037 041712      MOV (R0)+,@WAITBT  ; WAIT ON BIT
8264 041772 012001      MOV (R0)+,R1      ; R1 HAS TIME IN 10 MSEC
8265 041774 012002      MOV (R0)+,R2      ; R2 HAS TOLERANCE IN 10 MSEC
8266 041776 010066 000010      MOV R0,10(SP)     ; RESTORE RETURN ON STACK
8267
8268      ; *THIS SECTION WAITS FOR BIT, THROUGH TWO COUNT DOWNS
8269 042002 013703 042154      MOV @TIMCNT,R3      ; R3 IS A TEMPORARY COUNTER
8270 042006 033777 041712 177674 1$: BIT @WAITBT,@WAITRE ; IS REQUIRED BIT THERE
8271 042014 001025      BNE 4$              ; BRANCH IF YES
8272 042016 005303      DEC R3              ; COUNT IF REQUIRED BIT NOT THERE
8273 042020 001372      BNE 1$
8274 042022 013703 042154      MOV @TIMCNT,R3      ; TEMPORARY COUNTER
8275 042026 033777 041712 177654 2$: BIT @WAITBT,@WAITRE ; IS REQUIRED BIT THERE
8276 042034 001015      BNE 4$              ; BRANCH IF YES
8277 042036 005303      DEC R3              ; COUNT IF REQUIRED BIT NOT THERE
8278 042040 001372      BNE 2$
8279 042042 017737 177642 001126      MOV @WAITRE,@$BDDAT ; REGISTER CONTENTS FOR TYPEOUT
8280 042050 032777 000100 140222      BIT #IE,@RHCS1     ; DID ANY INTERRUPT OCCUR
8281 042056 001402      BEQ 3$              ; BRANCH IF YES
8282 042060 104001      ERROR 1            ; RPO4 DID NOT INTERRUPT
8283 042062 000427      BR 7$              ; OUT
8284 042064 104002      3$: ERROR 2       ; RPO4 INTERRUPTED BUT WAITED
8285      ; ON BIT DID NOT OCCUR
8286      ; EVEN AFTER TWO COUNT DOWNS
8287      ; FROM 177777 TO 0
8288 042066 000425      BR 7$              ; OUT
8289
8290      ; *NOW TIME AND TOLERANCE WILL BE CHECKED
8291 042070 017737 177614 001126 4$: MOV @WAITRE,@$BDDAT ; REGISTER CONTENTS FOR TYPEOUT
8292 042076 032777 000100 140174      BIT #IE,@RHCS1     ; DID ANY INTERRUPT OCCUR

```

```
8293 042104 001402          BEQ      5$          ;BRANCH IF YES
8294 042106 104003          ERROR    3          ;INTERRUPT DID NOT OCCUR EVEN
8295                                     ;AFTER ONE BNE AND ONE MOV
8296                                     ;OF THE WAITED ON BIT SETTING
8297 042110 000414          BR       7$          ;OUT
8298 042112 160201          SUB      R2,R1      ;R1 NOW HAS LOWER LIMIT OF TIME
8299 042114 023701 041714  CMP      @#WAITTM,R1 ;FOR GOOD RESULTS, WAITTM
8300                                     ;MUST BE GREATER OR EQUAL
8301                                     ;TORI
8302 042120 103002          BHIS    6$          ;BRANCH IF GOOD
8303 042122 104004          ERROR    4          ;BIT DID OCCUR BUT TIME
8304                                     ;TAKEN IS BELOW LOWER LIMIT
8305 042124 000406          BR       7$          ;OUT
8306
8307 042126 060202          6$:  ADD    R2,R2      ;DOUBLE TOLERANCE
8308 042130 060201          ADD    R2,R1      ;R1 NOW HAS UPPER LIMIT OF TIME
8309 042132 020137 041714  CMP    R1,@#WAITTM ;FOR GOOD RESULTS, WAITTM
8310                                     ;MUST BE LESS OR EQUAL TO R1
8311 042136 103001          BHIS    7$          ;BRANCH IF GOOD
8312 042140 104004          ERROR    4          ;BIT DID OCCUR BUT TIME TAKEN
8313                                     ;IS ABOVE UPPER LIMIT
8314 042142          7$:
8315 042142 012603          MOV    (SP)+,R3    ;:POP STACK INTO R3
8316 042144 012602          MOV    (SP)+,R2    ;:POP STACK INTO R2
8317 042146 012601          MOV    (SP)+,R1    ;:POP STACK INTO R1
8318 042150 012600          MOV    (SP)+,R0    ;:POP STACK INTO R0
8319 042152 000002          RTI                    ;RETURN TO MAIN TEST
8320
8321
8322
8323
8324
8325
8326          ;*THIS IS A WAIT LOOP WHEN NO P-CLOCK IS AVAILABLE
8327          ;*NO TIMING IS DONE
8328          ;*CALL IS
8329          ;*   WAT
8330          ;*   A      ;ABSOLUTE REGISTER ADDRESS
8331          ;*   B      ;BIT WAITE) FOR
8332          ;*   TA     ;TIME-NOT USED HERE
8333          ;*   TO     ;TIME-NOT USED HERE
8334          ;*R3-IS A TEMPORARY COUNTER
8335 042154 177777          TIMCNT: 177777          ;COUNT FOR WAIT LOOP
8336
8337          WAIT.T:
8338 042156 010046          MOV    R0,-(SP)    ;:PUSH R0 ON STACK
8339 042160 010346          MOV    R3,-(SP)    ;:PUSH R3 ON STACK
8340 042162 016600 000004  MOV    4(SP),R0    ;R0 HAS ADDRESS OF NEXT LOCATION
8341 042166 010037 041706  MOV    R0,@#WAITPC ;WAT PC +2 IS IN WAITPC
8342 042172 162737 000002 041706  SUB    #2,@#WAITPC ;WAT PC IS IN WAITPC
8343 042200 013037 041710  MOV    @#WAITRE    ;WAIT ON REGISTER ADDRESS
8344 042204 012037 041712  MOV    (R0)+,@#WAITBT ;WAIT ON BIT
8345 042210 022020          CMP    (R0)+,(R0)+ ;DUMP NEXT TWO WORDS-TA, TO
8346 042212 010066 000004  MOV    R0,4(SP)    ;RESTORE RETURN ON STACK
8347
8348          ;*THIS HAS THE TWO COUNT DOWNS FROM 177777
```

```

8349 042216 013703 042154      MOV      @#TIMCNT,R3      ;R3 HAS TEMPORARY COUNT
8350 042222 033777 041712 177460 1$:  BIT      @#WAITBT,@WAITRE ;IS REQUIRED BIT THERE
8351 042230 001025                BNE      4$              ;BRANCH IF YES
8352 042232 005303                DEC      R3              ;COUNT IF REQUIRED BIT NOT THERE
8353 042234 001372                BNE      1$              ;
8354 042236 013703 042154      MOV      @#TIMCNT,R3      ;SECOND COUNT DOWN FROM 177777
8355 042242 033777 041712 177440 2$:  BIT      @#WAITBT,@WAITRE ;IS REQUIRED BIT THERE
8356 042250 001015                BNE      4$              ;BRANCH IF YES
8357 042252 005303                DEC      R3              ;COUNT IF REQUIRED BIT NOT THERE
8358 042254 001372                BNE      2$              ;
8359 042256 017737 177426 001126  MOV      @WAITRE,@#SBDAT ;REGISTER CONTENTS FOR TYPEOUT
8360 042264 032777 000100 140006  BIT      #IE,@RHCS1      ;DID ANY INTERRUPT OCCUR
8361 042272 001402                BEQ      3$              ;BRANCH IF YES
8362 042274 104001                ERROR    1              ;RP04 DID NOT INTERRUPT
8363                                ;BIT DID NOT OCCUR
8364 042276 000414                BR       5$              ;OUT
8365 042300 104002                3$:  ERROR    2              ;RP04 INTERRUPTED BUT
8366                                ;WAITED ON BIT DID NOT OCCUR
8367                                ;EVEN AFTER TWO COUNT DOWNS
8368                                ;FROM 177777 TO 0
8369 042302 000412                BR       5$              ;OUT
8370
8371                                ;*BIT DID SET SO CHECK IF INTERRUPT OCCURED
8372 042304 000240                4$:  NOP                    ;ALLOW TIME FOR INTERRUPT
8373 042306 032777 000100 137764  BIT      #IE,@RHCS1      ;DID ANY INTERRUPT OCCUR
8374 042314 001405                BEQ      5$              ;BRANCH IF YES
8375 042316 017737 177366 001126  MOV      @WAITRE,@#SBDAT ;REGISTER CONTENTS FOR TYPEOUT
8376 042324 104003                ERROR    3              ;INTERRUPT DID NOT OCCUR
8377                                ;EVEN AFTER ONE BNE OF
8378                                ;THE WAITED ON BIT OCCURING
8379 042326 000400                BR       5$              ;OUT
8380 042330                5$:
8381 042330 012603                MOV      (SP)+,R3        ;;POP STACK INTO R3
8382 042332 012600                MOV      (SP)+,R0        ;;POP STACK INTO R0
8383 042334 000002                RTI                    ;RETURN TO MAIN TEST

```

```

8384 ;*THIS CHANGES REGISTER SAVED VALUE
8385 ;*CALL IS
8386 ;*   JSR      RO,@CHREG
8387 ;*   R          ;REGISTER TO BE CHANGED
8388 ;*   N          ;NUMBER OF BITS TO BE CHANGED
8389 ;*   NEW        ;NEW VALUE OF BIT MUST BE 0 OR 1
8390 ;*   P          ;POSITION OF BIT TO BE CHANGED
8391 ;*NEW AND P WILL BE REPEATED N NUMBER OF TIMES
8392 CHREG:
8393   MOV      R1,-(SP)      ;;PUSH R1 ON STACK
8394   MOV      R2,-(SP)      ;;PUSH R2 ON STACK
8395   MOV      (R0)+,R1      ;R1 HAS ADDRESS OF ADDRESS OF REGISTER
8396   MOV      (R0)+,R2      ;R2 HAS NUMBER OF CHANGES
8397   SUB      #RHC,R1      ;R1 HAS OFSET OF REQUIRED REGISTER
8398   1$:     TST      (R0)+   ;IS A BIC OR A BIS TO BE DONE
8399         BEQ      2$      ;BRANCH IF A BIC IS REQUIRED
8400   2$:     BIS      (R0)+,SAVERE(R1) ;SET REQUIRED BIT
8401         BR      3$      ;BRANCH TO DECREMENT COUNT
8402   3$:     BIC      (R0)+,SAVERE(R1) ;CLEAR REQUIRED BIT
8403         DEC      R2      ;DECREMENT NUMBER OF CHANGES
8404         BNE      1$      ;BRANCH IF NOT COMPLETE
8405         MOV      (SP)+,R2  ;;POP STACK INTO R2
8406         MOV      (SP)+,R1  ;;POP STACK INTO R1
8407         RTS      R0      ;RETURN TO MAIN PROGRAM
8408
8409
8410
8411
8412
8413

```

```

8414 ;*THIS FILLS A BLOCK WITH INCREMENTAL DATA
8415 ;*CALL IS
8416 ;*   JSR      RO,@FILL
8417 ;*   F          ;FROM
8418 ;*   N          ;NUMBER OF WORDS
8419 ;*   S          ;STARTING VALUE OF DATA
8420 ;*   I          ;INCREMENT DATA BY
8421
8422 FILL:
8423   MOV      R1,-(SP)      ;;PUSH R1 ON STACK
8424   MOV      R2,-(SP)      ;;PUSH R2 ON STACK
8425   MOV      R3,-(SP)      ;;PUSH R3 ON STACK
8426   MOV      R4,-(SP)      ;;PUSH R4 ON STACK
8427   MOV      (R0)+,R1      ;R1 HAS ADDRESS WHERE DATA IS TO GO
8428   MOV      (R0)+,R2      ;R2 HAS NUMBER OF WORDS TO BE FILLED
8429   MOV      (R0)+,R3      ;STARTING VALUE OF DATA
8430   MOV      (R0)+,R4      ;R4 HAS INCREMENT
8431   ;*NOW DATA WILL BE FILLED
8432   1$:     MOV      R3,(R1)+  ;FILL DATA
8433         ADD      R4,R3      ;GET NEXT VALUE OF DATA
8434         DEC      R2      ;DECREMENT COUNT
8435         BNE      1$      ;BRANCH IF ALL NOT DONE
8436         MOV      (SP)+,R4  ;;POP STACK INTO R4
8437         MOV      (SP)+,R3  ;;POP STACK INTO R3
8438         MOV      (SP)+,R2  ;;POP STACK INTO R2
8439         MOV      (SP)+,R1  ;;POP STACK INTO R1

```

CZRJICO, RPO4/5/6 FCTNL CTLR1
CZRJIC.P11 27-JUL-78 13:05

MACY11 30A(1052) 27-JUL-78 13:06 PAGE 188
JAM CURRENT CYLINDER ROUTINE

F 15

SEQ 0187

8440 042442 000200
8441
8442
8443

RTS RO

;RETURN TO MAIN PROGRAM

```
8444
8445
8446
8447      ;*THIS IS A SUBROUTINE TO COMPARE REGISTERS
8448      ;*GOOD DATA IS ALREADY SAVED IN 'SAVERE'
8449      ;*TEST DATA IS IN THE REGISTERS
8450      ;*CALL IS
8451      ;*      JSR      R0,@#COMREG
8452      ;*      SAVERE      ;GOOD DATA
8453      ;*      RHCS1      ;ADDRESS OF ADDRESS TEST DATA
8454      ;*      N.          ;RETURN FOR ERROR
8455      ;*      RG          ;RETURN FOR GOOD COMPARISON
8456      ;*ON RETURN WITH ERROR '$GDDAT' HAS GOOD DATA, '$BDDAT' HAS BAD DATA
8457      ;*'REGADR' HAS REGISTER ADDRESS
8458
8459      COMREG:
8460      042444 010146      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
8461      042446 010246      MOV      R2,-(SP)      ;;PUSH R2 ON STACK
8462      042450 010346      MOV      R3,-(SP)      ;;PUSH R3 ON STACK
8463      042452 010446      MOV      R4,-(SP)      ;;PUSH R4 ON STACK
8464      042454 010546      MOV      R5,-(SP)      ;;PUSH R5 ON STACK
8465      042456 012001      MOV      (R0)+,R1      ;R1 HAS ADDRESS OF GOOD DATA
8466      042460 012002      MOV      (R0)+,R2      ;R2 HAS ADDRESS OF ADDRESS OF TEST DATA
8467      042462 012003      MOV      (R0)+,R3      ;R3 HAS NUMBER OF WORDS
8468      042464 012004      MOV      (R0)+,R4      ;R4 HAS RETURN FOR ERROR
8469      042466 011000      MOV      (R0),R0      ;R0 HAS RETURN ON NO ERROR
8470
8471      042470 004737 043370      JSR      PC,@#PUTREG      ;SAVE REGISTERS
8472      042474 113737 004637 002401      MOV      @#SAVERE+25,@#AS+1;MAKE UPPER BYTE OF R HAS SAME
8473      042502 012705 177776      MOV      #-2,R5          ;PRESET R5 TO -2
8474
8475      042506 062705 000002      1$:      ADD      #2,R5          ;INCREMENT TO INDEX
8476      042512 022122      CMP      (R1)+,(R2)+      ;COMPARE REGISTER CONTENTS
8477      042514 001420      BEQ      2$              ;BRANCH IF GOOD
8478      042516 014137 001124      MOV      -(R1),@#$GDDAT      ;SAVE GOOD DATA
8479      042522 014237 001126      MOV      -(R2),@#$BDDAT      ;SAVE BAD DATA
8480      042526 016537 002272 004600      MOV      RHW(R5),@#REGADR      ;SAVE ADDRESS OF FAILING REGISTER
8481      042534 004714      JSR      PC,@R4          ;RETURN TO MAIN PROGRAM
8482
8483      042536 022122      CMP      (R1)+,(R2)+      ;UNDO -(R1) AND -(R2) FOR ERRORS
8484      042540 017746 136374      MOV      @SWR,-(SP)      ;GET SWITCH SETTING
8485      042544 042716 177177      BIC      #^C60,(SP)      ;KEEP ONLY SWITCH 7 AND 8
8486      042550 022726 000200      CMP      #SW07,(SP)+      ;IS 7 SET AND 8 DOWN
8487      042554 001402      BEQ      3$              ;BRANCH OUT IF YES
8488      042556 005303      2$:      DEC      R3              ;ARE ALL COMPARES DONE
8489      042560 001352      BNE      1$              ;BRANCH IF NOT COMPLETE
8490
8491      042562      3$:
8492      042562 012605      MOV      (SP)+,R5      ;;POP STACK INTO R5
8493      042564 012604      MOV      (SP)+,R4      ;;POP STACK INTO R4
8494      042566 012603      MOV      (SP)+,R3      ;;POP STACK INTO R3
8495      042570 012602      MOV      (SP)+,R2      ;;POP STACK INTO R2
8496      042572 012601      MOV      (SP)+,R1      ;;POP STACK INTO R1
8497      042574 000200      RTS      R0              ;RETURN TO MAIN PROGRAM
8498      042576 000000      4$:      .WORD    0              ;TEMP STORAGE
```

8499
8500
8501
8502
8503
8504
8505
8506
8507
8508
8509
8510
8511
8512
8513
8514
8515
8516
8517
8518
8519
8520
8521
8522
8523
8524
8525
8526
8527
8528
8529
8530
8531
8532
8533
8534
8535
8536
8537
8538
8539
8540
8541
8542
8543
8544
8545
8546
8547
8548
8549
8550
8551
8552

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

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

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

047274

8553
8554
8555
8556
8557
8558
8559
8560
8561
8562
8563
8564
8565
8566
8567
8568
8569
8570
8571
8572
8573
8574
8575
8576
8577
8578
8579
8580
8581
8582
8583
8584
8585
8586
8587
8588
8589
8590
8591
8592
8593
8594
8595
8596
8597
8598
8599
8600
8601
8602
8603
8604
8605
8606
8607

;*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
;COMMAND ADDRESS
```

COM

```
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
```

```
8608
8609
8610      ;*THIS IS A SUBROUTINE TO COMPARE TWO BLOCKS IN MEMORY
8611
8612      ;*R1 HAS GOOD DATA BUFFER ADDRESS
8613      ;*R2 HAS TEST DATA BUFFER ADDRESS
8614      ;*R5 HAS ADDRESS OF RETURN ON ERROR
8615      ;*R3 HAS NUMBER OF WORDS TO BE COMPARED
8616      ;*R4 HAS ONE MORE THAN NUMBER OF WORDS TO BE COMPARED
8617
8618      ;*CALL IS:
8619      ;*      JSR      R0,@#COMPAR
8620      ;*      G
8621      ;*      T
8622      ;*      N
8623      ;*      RE
8624      ;*      RG
8625
8626
8627
8628      COMPAR:
8629      043474 010146      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
8630      043476 010246      MOV      R2,-(SP)      ;;PUSH R2 ON STACK
8631      043500 010346      MOV      R3,-(SP)      ;;PUSH R3 ON STACK
8632      043502 010446      MOV      R4,-(SP)      ;;PUSH R4 ON STACK
8633      043504 010546      MOV      R5,-(SP)      ;;PUSH R5 ON STACK
8634      043506 012001      MOV      (R0)+,R1      ;;ADDRESS OF GOOD DATA BUFFER
8635      043510 012002      MOV      (R0)+,R2      ;;ADDRESS OF TEST DATA BUFFER
8636      043512 012003      MOV      (R0)+,R3      ;;NO OF WORDS TO BE COMPARED
8637      043514 012005      MOV      (R0)+,R5      ;;RETURN ON ERROR
8638      043516 011000      MOV      (R0),R0       ;;RETURN ON NO ERROR
8639      043520 010304      MOV      R3,R4
8640      043522 005204      INC      R4
8641      043524 010437 004602      1$:      MOV      R4,@#ERWORD   ;;FOR ERROR WORD NO
8642      043530 022122      CMP      (R1)+,(R2)+   ;;COMPARE GOOD WITH TEST DATA
8643      043532 001417      BEQ      2$           ;;BRANCH IF GOOD
8644
8645      043534 014137 001124      MOV      -(R1),@#$GDDAT ;;GOOD DATA
8646      043540 014237 001126      MOV      -(R2),@#$BDDAT ;;BAD DATA
8647      043544 160337 004602      SUB      R3,@#ERWORD   ;;ERROR WORD NO.
8648      043550 004715      JSR      PC,@R5        ;;RETURN TO PRINT ERROR
8649      043552 022122      CMP      (R1)+,(R2)+   ;;UNDO -(R1) AND -(R2) FOR ERRORS
8650      043554 017746 135360      MOV      @SWR,-(SP)    ;;GET SWITCH SETTING
8651      043560 042716 177177      BIC      #^C600,(SP)   ;;KEEP ONLY SWITCH 7 AND 8
8652      043564 022726 000200      CMP      #SW07,(SP)+  ;;IS 7 SET AND 8 RESET
8653      043570 001402      BEQ      3$           ;;BRANCH OUT IF YES
8654      043572 005303      2$:      DEC      R3
8655      043574 001353      BNE      1$           ;;COUNT
8656      043576      3$:      BNE      1$           ;;BRANCH IF ALL NOT DEVICE
8657      043576 012605      MOV      (SP)+,R5      ;;POP STACK INTO R5
8658      043600 012604      MOV      (SP)+,R4      ;;POP STACK INTO R4
8659      043602 012603      MOV      (SP)+,R3      ;;POP STACK INTO R3
8660      043604 012602      MOV      (SP)+,R2      ;;POP STACK INTO R2
8661      043606 012601      MOV      (SP)+,R1      ;;POP STACK INTO R1
8662      043610 000200      RTS      R0           ;;RETURN TO MAIN PROGRAM
```

```
8663
8664
8665          ;* THIS ROUTINE WILL ALLOW THE CHANGE OF THE BASE
8666          ;* ADDRESS FROM 176700 TO ANY TYPED VALUE
8667
8668 043612    BASECH:
8669 043612 104401 043620      TYPE      ,65$          ;;TYPE ASCIZ STRING
8670 043616 000425      BR          64$          ;;GET OVER THE ASCIZ
8671 043672 013746 002300    MOV      @#RHCS1,-(SP)  ;;GET READY TO TYPE OLD BASE
8672 043676 104402      TYPOC
8673 043700 104401 043706    TYPE      ,67$          ;;TYPE ASCIZ STRING
8674 043704 000425      BR          66$          ;;GET OVER THE ASCIZ
8675 043760 004737 045666    JSR      PC,@#STKINT    ;;INITIALIZE THE TTY KEYBOARD
8676 043764 104412      RDOCT
8677 043766 012700 002270    MOV      #RHDB,R0      ;;GET STARTING ADDRESS OF REGISTERS
8678 043772 012701 000026    MOV      #22,R1       ;;NUMBER OF REGISTERS
8679 043776 012737 044576 000004  MOV      #ADTIMO,@#4   ;;SET TRAP CATCHER TO CHECK THIS ADDRESS
8680 044004 021637 002300    CMP      @SP,@#RHCS1   ;;NEW ADDRESS?
8681 044010 001407      BEQ      1$           ;;NO, OLD ONE JUST RETYPED.
8682 044012 005776 000000    TST      @0(SP)       ;;OK, SO ACCESS THIS NEW ADDRESS
8683 044016 163716 002300    SUB      @#RHCS1,@SP   ;;GET THE ADDRESS OFFSET
8684 044022 061620      2$: ADD      @SP,(R0)+     ;;AND PLUG IT IN.
8685 044024 005301      DEC      R1           ;;ONE LESS REGISTER TO GO
8686 044026 001375      BNE      2$          ;;BUT WE'RE NOT DONE YET.
8687 044030      1$:
8688 044030 104401 044036    TYPE      ,69$          ;;TYPE ASCIZ STRING
8689 044034 000417      BR          68$          ;;GET OVER THE ASCIZ
8690 044074 013746 002266    MOV      @#RPVEC,-(SP) ;;GET READY TO TYPE OLD VECTOR ADDRESS
8691 044100 104402      TYPOC
8692 044102 104401 044110    TYPE      ,71$          ;;TYPE ASCIZ STRING
8693 044106 000437      BR          70$          ;;GET OVER THE ASCIZ
8694 044206 104412      RDOCT
8695 044210 012637 002266    MOV      (SP)+,@#RPVEC ;;SETUP VECTOR ADDRESS
8696 044214 104401 044222    TYPE      ,73$          ;;TYPE ASCIZ STRING
8697 044220 000416      BR          72$          ;;GET OVER THE ASCIZ
8698 044256 013746 002300    MOV      @#RHCS1,-(SP)
8699 044262 104402      TYPOC
8700 044264 104401 044272    TYPE      ,75$          ;;TYPE ASCIZ STRING
8701 044270 000416      BR          74$          ;;GET OVER THE ASCIZ
8702 044326 013746 002266    MOV      @#RPVEC,-(SP)
8703 044332 104402      TYPOC
8704 044334 104401 044342    TYPE      ,77$          ;;TYPE ASCIZ STRING
8705 044340 000417      BR          76$          ;;GET OVER THE ASCIZ
8706 044400 104401 044406    TYPE      ,79$          ;;TYPE ASCIZ STRING
8707 044404 000402      BR          78$          ;;GET OVER THE ASCIZ
8708 044412 104401 044420    TYPE      ,81$          ;;TYPE ASCIZ STRING
8709 044416 000424      BR          80$          ;;GET OVER THE ASCIZ
8710 044470 104401 044476    TYPE      ,83$          ;;TYPE ASCIZ STRING
8711 044474 000426      BR          82$          ;;GET OVER THE ASCIZ
8712 044552 012746 000200    MOV      #RA,-(SP)
8713 044556 104402      TYPOC
8714 044560 104401 044566    TYPE      ,85$          ;;TYPE ASCIZ STRING
8715 044564 000402      BR          84$          ;;GET OVER THE ASCIZ
8716 044572 000137 005012    JMP      @#BEGIN      ;;RESTART, TO RUN ALL DRIVES
8717 044576
8718 044576 104401 044604    ADTIMO: TYPE      ,65$          ;;TYPE ASCIZ STRING
```

8719	044602	000426	BR	64\$::GET OVER THE ASCIZ
8720	044660	022626	CMP	(SP)+, (SP)+	::RESTORE THE STACK
8721	044662	000137	JMP	@BASECH	::AND DO THE WHOLE THING AGAIN!
8722					
8723					

8724
8725
8726
8727
8728
8729
8730
8731
8732
8733
8734

044666
044666 104401 044674
044672 000411
044716 104402
044720 012777 044666 135340
044726 000000

RPVECT:

TYPE .65\$
BR 64\$
TYPOC
MOV #RPVECT,@RPVEC
HALT

::TYPE ASCIZ STRING
::GET OVER THE ASCIZ
:TYPE FROM PC
:RESTORE TRAP RPO4 VECTOR
:CHANGE TO CONTINUE

CZ
CZ

```

8735 044730 104407          (KSWR
8736 044732 032777 040000 134200 1$: BIT #BIT14,@SWR      ;;TEST FOR CHANGE IN SOFT-SWR
8737 044740 001111          $OVER          ;;LOOP ON PRESENT TEST?
8738 044742 000416          $XTSTR: BR 6$          ;;YES IF SW14=1
8739 044744 013746 000004  MOV @ERRVEC,-(SP)    ;;IF RUNNING ON THE 'XOR' TESTER CHANGE
8740 044750 012737 044770 000004  MOV #5$,@ERRVEC    ;;SAVE THE CONTENTS OF THE ERROR VECTOR
8741 044756 005737 177060  TST @177060        ;;SET FOR TIMEOUT
8742 044762 012637 000004  MOV (SP)+,@ERRVEC  ;;TIME OUT ON XOR?
8743 044766 000463          BR $SVLAD          ;;RESTORE THE ERROR VECTOR
8744 044770 022626          5$: (MP (SP)+,(SP)+  ;;GO TO THE NEXT TEST
8745 044772 012637 000004  MOV (SP)+,@ERRVEC  ;;CLEAR THE STACK AFTER A TIME OUT
8746 044776 000423          BR 7$            ;;RESTORE THE ERROR VECTOR
8747 045000 032777 000400 134132  BIT #BIT08,@SWR    ;;LOOP ON THE PRESENT TEST
8748 045006 001404          BEQ ?$           ;;LOOP ON SPEC. TEST?
8749 045010 127737 134124 001102  (MPB @SWR,$STNM    ;;BR IF NO
8750 045016 001462          BEQ $OVER        ;;ON THE RIGHT TEST? SWR<7:0>
8751 045020 105737 001103          2$: TSTB $ERFLG    ;;BR IF YES
8752 045024 001421          BEQ 3$           ;;HAS AN ERROR OCCURRED?
8753 045026 123737 001115 001103  (MPB $ERMAX,$ERFLG  ;;BR IF NO
8754 045034 101015          BHI 3$           ;;MAX. ERRORS FOR THIS TEST OCCURRED?
8755 045036 032777 001000 134074  BIT #BIT09,@SWR    ;;BR IF NO
8756 045044 001404          BEQ 4$           ;;LOOP ON ERROR?
8757 045046 013737 001110 001106  7$: MOV $LPERR,$LPADR  ;;BR IF NO
8758 045054 000443          BR $OVER         ;;SET LOOP ADDRESS TO LAST SCOPE
8759 045056 105037 001103          4$: CLRB $ERFLG    ;;ZERO THE ERROR FLAG
8760 045062 005037 001212          CLR $TIMES       ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
8761 045066 000415          BR 1$           ;;ESCAPE TO THE NEXT TEST
8762 045070 032777 004000 134042  3$: BIT #BIT11,@SWR  ;;INHIBIT ITERATIONS?
8763 045076 001011          BNE 1$           ;;BR IF YES
8764 045100 005737 001100          TST $PASS        ;;IF FIRST PASS OF PROGRAM
8765 045104 001406          BEQ 1$           ;;INHIBIT ITERATIONS
8766 045106 005237 001104          INC $ICNT        ;;INCREMENT ITERATION COUNT
8767 045112 023737 001212 001104  (MP $TIMES,$ICNT   ;;CHECK THE NUMBER OF ITERATIONS MADE
8768 045120 002021          BGE $OVER        ;;BR IF MORE ITERATION REQUIRED
8769 045122 012737 000001 001104  1$: MOV #1,$ICNT    ;;REINITIALIZE THE ITERATION COUNTER
8770 045130 013737 045200 001212  MOV $MXCNT,$TIMES  ;;SET NUMBER OF ITERATIONS TO DO
8771 045136 105237 001102          $SVLAD: INCB $STNM  ;;COUNT TEST NUMBERS
8772 045142 011637 001106          MOV (SP),$LPADR   ;;SAVE SCOPE LOOP ADDRESS
8773 045146 011637 001110          MOV (SP),$LPERR   ;;SAVE ERROR LOOP ADDRESS
8774 045152 005037 001214          CLR $ESCAPE      ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
8775 045156 112737 000001 001115  MOVB #1,$ERMAX    ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
8776 045164 013777 001102 133750  $OVER: MOV $STNM,@DISPLAY  ;;DISPLAY TEST NUMBER
8777 045172 013716 001106          MOV $LPADR,(SP)  ;;FUDGE RETURN ADDRESS
8778 045176 000002          RTI           ;;FIXES PS
8779 045200 000004          $MXCNT: 4      ;;MAX. NUMBER OF ITERATIONS
  
```

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

8780	045202	010046			MOV	R0,-(SP)	::PUSH R0 ON STACK
8781	045204	010146			MOV	R1,-(SP)	::PUSH R1 ON STACK
8782	045206	010246			MOV	R2,-(SP)	::PUSH R2 ON STACK
8783	045210	010346			MOV	R3,-(SP)	::PUSH R3 ON STACK
8784	045212	010546			MOV	R5,-(SP)	::PUSH R5 ON STACK
8785	045214	012746	020200		MOV	#20200,-(SP)	::SET BLANK SWITCH AND SIGN
8786	045220	016605	000020		MOV	20(SP),R5	::GET THE INPUT NUMBER
8787	045224	100004			BPL	1\$::BR IF INPUT IS POS.
8788	045226	005405			NEG	R5	::MAKE THE BINARY NUMBER POS.
8789	045230	112766	000055	000001	MOVB	#'-,1(SP)	::MAKE THE ASCII NUMBER NEG.
8790	045236	005000			CLR	R0	::ZERO THE CONSTANTS INDEX
8791	045240	0127C3	045416		MOV	#\$DBLK,R3	::SETUP THE OUTPUT POINTER
8792	045244	112723	000040		MOVB	#' ,(R3)+	::SET THE FIRST CHARACTER TO A BLANK
8793	045250	005002			CLR	R2	::CLEAR THE BCD NUMBER
8794	045252	016001	045406		MOV	\$DTBL(R0),R1	::GET THE CONSTANT
8795	045256	160105			SUB	R1,R5	::FORM THIS BCD DIGIT
8796	045260	002402			BLT	4\$::BR IF DONE
8797	045262	005202			INC	R2	::INCREASE THE BCD DIGIT BY 1
8798	045264	000774			BR	3\$	
8799	045266	060105			ADD	R1,R5	::ADD BACK THE CONSTANT
8800	045270	005702			TST	R2	::CHECK IF BCD DIGIT=0
8801	045272	001002			BNE	5\$::FALL THROUGH IF 0
8802	045274	105716			TSTB	(SP)	::STILL DOING LEADING 0'S?
8803	045276	100407			BMI	7\$::BR IF YES
8804	045300	106316			ASLB	(SP)	::MSD?
8805	045302	103003			BCC	6\$::BR IF NO
8806	045304	116663	000001	177777	MOVB	1(SP),-1(R3)	::YES--SET THE SIGN
8807	045312	052702	000060		BIS	#'0,R2	::MAKE THE BCD DIGIT ASCII
8808	045316	052702	000040		BIS	#' ,R2	::MAKE IT A SPACE IF NOT ALREADY A DIGIT
8809	045322	110223			MOVB	R2,(R3)+	::PUT THIS CHARACTER IN THE OUTPUT BUFFER
8810	045324	005720			TST	(R0)+	::JUST INCREMENTING
8811	045326	020027	000010		CMP	R0,#10	::CHECK THE TABLE INDEX
8812	045332	002746			BLT	2\$::GO DO THE NEXT DIGIT
8813	045334	003002			BGT	8\$::GO TO EXIT
8814	045336	010502			MOV	R5,R2	::GET THE LSD
8815	045340	000764			BR	6\$::GO CHANGE TO ASCII
8816	045342	105726			TSTB	(SP)+	::WAS THE LSD THE FIRST NON-ZERO?
8817	045344	100003			BPL	9\$::BR IF NO
8818	045346	116663	177777	177776	MOVB	-1(SP),-2(R3)	::YES--SET THE SIGN FOR TYPING
8819	045354	105013			CLRB	(R3)	::SET THE TERMINATOR
8820	045356	012605			MOV	(SP)+,R5	::POP STACK INTO R5
8821	045360	012603			MOV	(SP)+,R3	::POP STACK INTO R3
8822	045362	012602			MOV	(SP)+,R2	::POP STACK INTO R2
8823	045364	012601			MOV	(SP)+,R1	::POP STACK INTO R1
8824	045366	012600			MOV	(SP)+,R0	::POP STACK INTO R0
8825	045370	104401	045416		TYPE	,\$DBLK	::NOW TYPE THE NUMBER
8826	045374	016666	000002	000004	MOV	2(SP),4(SP)	::ADJUST THE STACK
8827	045402	012616			MOV	(SP)+,(SP)	
8828	045404	000002			RTI		::RETURN TO USER
8829	045406	023420					
8830	045410	001750					
8831	045412	000144					
8832	045414	000012					

\$DTBL: 10000.
1000.
100.
10.

8833	045426	105737	001157		\$TYPE: TSTB	\$TPFLG	:: IS THERE A TERMINAL?	
8834	045432	100002			BPL	1\$:: BR IF YES	
8835	045434	000000			HALT		:: HALT HERE IF NO TERMINAL	
8836	045436	000407			BR	3\$:: LEAVE	
8837	045440	010046		1\$:	MOV	RO, -(SP)	:: SAVE RO	
8838	045442	017600	000002		MOV	@2(SP), RO	:: GET ADDRESS OF ASCIZ STRING	
8839	045446	112046		2\$:	MOVB	(RO)+, -(SP)	:: PUSH CHARACTER TO BE TYPED ONTO STACK	
8840	045450	001005			BNE	4\$:: BR IF IT ISN'T THE TERMINATOR	
8841	045452	005726			TST	(SP)+	:: IF TERMINATOR POP IT OFF THE STACK	
8842	045454	012600		60\$:	MOV	(SP)+, RO	:: RESTORE RO	
8843	045456	062716	000002	3\$:	ADD	#2, (SP)	:: ADJUST RETURN PC	
8844	045462	000002			RTI		:: RETURN	
8845	045464	122716	000011	4\$:	CMPB	#HT, (SP)	:: BRANCH IF <HT>	
8846	045470	001430			BEQ	8\$		
8847	045472	122716	000200		CMPB	#CRLF, (SP)	:: BRANCH IF NOT <CRLF>	
8848	045476	001006			BNE	5\$		
8849	045500	005726			TST	(SP)+	:: POP <CR><LF> EQUIV	
8850	045502	104401			TYPE		:: TYPE A CR AND LF	
8851	045504	001223			\$CRLF			
8852	045506	105037	045642		CLRB	\$CHARCNT	:: CLEAR CHARACTER COUNT	
8853	045512	000755			BR	2\$:: GET NEXT CHARACTER	
8854	045514	004737	045576	5\$:	JSR	PC, \$TYPEC	:: GO TYPE THIS CHARACTER	
8855	045520	123726	001156	6\$:	CMPB	\$FILLC, (SP)+	:: IS IT TIME FOR FILLER CHARS.?	
8856	045524	001350			BNE	2\$:: IF NO GO GET NEXT CHAR.	
8857	045526	013746	001154		MOV	\$NULL, -(SP)	:: GET # OF FILLER CHARS. NEEDED	
8858	045532	105366	000001	7\$:	DECB	1(SP)	:: DOES A NULL NEED TO BE TYPED?	
8859	045536	002770			BLT	6\$:: BR IF NO--GO POP THE NULL OFF OF STACK	
8860	045540	004737	045576		JSR	PC, \$TYPEC	:: GO TYPE A NULL	
8861	045544	105337	045642		DECB	\$CHARCNT	:: DO NOT COUNT AS A COUNT	
8862	045550	000770			BR	7\$:: LOOP	
8863	045552	112716	000040	8\$:	MOVB	#' , (SP)	:: REPLACE TAB WITH SPACE	
8864	045556	004737	045576	9\$:	JSR	PC, \$TYPEC	:: TYPE A SPACE	
8865	045562	132737	000007	045642	BITB	#7, \$CHARCNT	:: BRANCH IF NOT AT	
8866	045570	001372			BNE	9\$:: TAB STOP	
8867	045572	005726			TST	(SP)+	:: POP SPACE OFF STACK	
8868	045574	000724			BR	2\$:: GET NEXT CHARACTER	
8869	045576	105777	133346		\$TYPEC: TSTB	@\$TPS	:: WAIT UNTIL PRINTER IS READY	
8870	045602	100375			BPL	\$TYPEC		
8871	045604	116677	000002	133340	MOVB	2(SP), @\$TPB	:: LOAD CHAR TO BE TYPED INTO DATA REG.	
8872	045612	122766	000015	000002	CMPB	#CR, 2(SP)	:: IS CHARACTER A CARRIAGE RETURN?	
8873	045620	001003			BNE	1\$:: BRANCH IF NO	
8874	045622	105037	045642		CLRB	\$CHARCNT	:: YES--CLEAR CHARACTER COUNT	
8875	045626	000406			BR	\$TYPEX	:: EXIT	
8876	045630	122766	000012	000002	1\$:	CMPB	#LF, 2(SP)	:: IS CHARACTER A LINE FEED?
8877	045636	001402			BEQ	\$TYPEX	:: BRANCH IF YES	
8878	045640	105227			INCB	(PC)+	:: COUNT THE CHARACTER	
8879	045642	000000			\$CHARCNT: .WORD	0	:: CHARACTER COUNT STORAGE	
8880	045644	000207			\$TYPEX: RTS	PC		

8881	045646	000000			\$TKCNT: .WORD	0	::NUMBER OF ITEMS IN QUEUE
8882	045650	000000			\$TKQIN: .WORD	0	::INPUT POINTER
8883	045652	000000			\$TKQOUT: .WORD	0	::OUTPUT POINTER
8884	045666	005037	045646		\$TKINT: CLR	\$TKCNT	::CLEAR COUNT OF ITEMS IN QUEUE
8885	045672	012737	045654	045650	MOV	#\$TKQSRRT,\$TKQIN	::MOVE THE STARTING ADDRESS OF THE
8886	045700	013737	045650	045652	MOV	\$TKQIN,\$TKQOUT	::QUEUE INTO THE INPUT & OUTPUT POINTERS.
8887	045706	012737	045736	000060	MOV	#\$TKSRV,@TKVEC	::INITIALIZE THE KEYBOARD VECTOR
8888	045714	012737	000200	000062	MOV	#200,@TKVEC+2	::'BR' LEVEL 4
8889	045722	005777	133220		TST	@TKB	::CLEAR DONE FLAG
8890	045726	012777	000100	133210	MOV	#100,@STKS	::ENABLE TTY KEYBOARD INTERRUPT
8891	045734	000207			RTS	PC	::RETURN TO CALLER
8892	045736	117746	133204		\$TKSRV: MOV	@TKB,-(SP)	::PICKUP THE CHARACTER
8893	045742	042716	177600		BIC	#\$C177,(SP)	::STRIP THE JUNK
8894	045746	021627	000003		CM	(SP),#3	::IS IT A CONTROL C?
8895	045752	001007			BNE	1\$::BRANCH IF NO
8896	045754	104401	046725		TYPE	,\$CNTLC	::TYPE A CONTROL-C (^C)
8897	045760	004737	045666		JSR	PC,\$TKINT	::INIT THE KEYBOARD
8898	045764	005726			TST	(SP)+	::CLEAN UP STACK
8899	045766	000137	042602		JMP	OPERSEL	::CONTROL C RESTART
8900	045772	021627	000007		1\$: CMP	(SP),#7	::IS IT A CONTROL G?
8901	045776	001004			BNE	2\$::BRANCH IF NO
8902	046000	022737	000176	001140	CM	#\$SWREG,SWR	::IS SOFT-SWR SELECTED?
8903	046006	001500			BEQ	6\$::GO TO SWR CHANGE
8904	046010	022737	000011	045646	CM	#9,\$TKCNT	::IS THE QUEUE FULL?
8905	046016	001004			BNE	3\$::BRANCH IF NO
8906	046020	104401	001216		TYPE	,\$BELL	::RING THE TTY BELL
8907	046024	005726			TST	(SP)+	::CLEAN CHARACTER OFF OF STACK
8908	046026	000451			BR	5\$::EXIT
8909	046030	021627	000023		3\$: CM	(SP),#23	::IS IT A CONTROL-S?
8910	046034	001021			BNE	32\$::BRANCH IF NO
8911	046036	005077	133102		CLR	@STKS	::DISABLE TTY KEYBOARD INTERRUPTS
8912	046042	005726			TST	(SP)+	::CLEAN CHAR OFF STACK
8913	046044	105777	133074		31\$: TSTB	@STKS	::WAIT FOR A CHAR
8914	046050	100375			BPL	31\$::LOOP UNTIL ITS THERE
8915	046052	117746	133070		MOV	@TKB,-(SP)	::GET THE CHARACTER
8916	046056	042716	177600		BIC	#\$C177,(SP)	::MAKE IT 7-BIT ASCII
8917	046062	022627	000021		CM	(SP)+,#21	::IS IT A CONTROL-Q?
8918	046066	001366			BNE	31\$::BRANCH IF NO
8919	046070	012777	000100	133046	MOV	#100,@STKS	::REENABLE TTY KEYBOARD INTERRUPTS
8920	046076	000002			RTI		::RETURN
8921	046100	005237	045646		32\$: INC	\$TKCNT	::COUNT THIS CHARACTER
8922	046104	021627	000140		CM	(SP),#140	::IS IT UPPER CASE?
8923	046110	002405			RLT	4\$::BRANCH IF YES
8924	046112	021627	000175		CM	(SP),#175	::IS IT A SPECIAL CHAR?
8925	046116	003002			BGT	4\$::BRANCH IF YES
8926	046120	042716	000040		BIC	#40,(SP)	::MAKE IT UPPER CASE
8927	046124	112677	177520		4\$: MOV	(SP)+,@TKQIN	::AND PUT IT IN QUEUE
8928	046130	005237	045650		INC	\$TKQIN	::UPDATE THE POINTER
8929	046134	023727	045650	045665	CM	\$TKQIN,\$TKQEND	::GO OFF THE END?
8930	046142	001003			BNE	5\$::BRANCH IF NO
8931	046144	012737	045654	045650	MOV	#\$TKQSRRT,\$TKQIN	::RESET THE POINTER
8932	046152	000002			5\$: RTI		::RETURN
8933	046154	022737	000176	001140	\$CKSWR: CM	#\$SWREG,SWR	::IS THE SOFT-SWR SELECTED
8934	046162	001124			BNE	15\$::EXIT IF NOT
8935	046164	105777	132754		TSTB	@STKS	::IS A CHAR WAITING?
8936	046170	100121			BPL	15\$::IF NOT, EXIT

8937	046172	117746	132750			MOVB	@STKB, -(SP)	::YES
8938	046176	042716	177600			BIC	#^C177, (SP)	::MAKE IT 7-BIT ASCII
8939	046202	021627	000007			CMP	(SP), #7	::IS IT A CONTROL-G?
8940	046206	001300				BNE	2\$::IF NOT, PUT IT IN THE TTY QUEUE
8941	046210	123727	001134	000001	6\$:	CMPB	\$AUTOB, #1	::ARE WE RUNNING IN AUTO-MODE?
8942	046216	001674				BEQ	2\$::BRANCH IF YES
8943	046220	005726				TST	(SP)+	::CLEAR CONTROL-G OFF STACK
8944	046222	004737	045666			JSR	PC, \$TKINT	::FLUSH THE TTY INPUT QUEUE
8945	046226	005077	132712			CLR	@STKS	::DISABLE TTY KEYBOARD INTERRUPTS
8946	046232	112737	000001	001135		MOVB	#1, \$INIAG	::SET INTERRUPT MODE INDICATOR
8947	046240	104401	046737			TYPE	,\$CNTLG	::ECHO THE CONTROL-G (^G)
8948	046244	104401	046744			\$GTSWR: TYPE	,\$MSWR	::TYPE CURRENT CONTENTS
8949	046250	013746	000176			MOV	\$WREG, -(SP)	::SAVE SWREG FOR TYPEOUT
8950	046254	104402				TYPOC		::GO TYPE--OCTAL ASCII(ALL DIGITS)
8951	046256	104401	046755			TYPE	,\$MNEW	::PROMPT FOR NEW SWR
8952	046262	005046			19\$:	CLR	-(SP)	::CLEAR COUNTER
8953	046264	005046				CLR	-(SP)	::THE NEW SWR
8954	046266	105777	132652		7\$:	TSTB	@STKS	::CHAR THERE?
8955	046272	100375				BPL	7\$::IF NO! TRY AGAIN
8956	046274	117746	132646			MOVB	@STKB, -(SP)	::PICK UP CHAR
8957	046300	042716	177600			BIC	#^C177, (SP)	::MAKE IT 7-BIT ASCII
8958	046304	021627	000003			CMP	(SP), #3	::IS IT A CONTROL-C?
8959	046310	001015				BNE	9\$::BRANCH IF NOT
8960	046312	104401	046725			TYPE	,\$CNTLC	::YES, ECHO CONTROL-C (^C)
8961	046316	062706	000006			ADD	#6, SP	::CLEAN UP STACK
8962	046322	123727	001135	000001		CMPB	\$INTAG, #1	::REENABLE TTY KEYBOARD INTERRUPTS?
8963	046330	001003				BNE	8\$::BRANCH IF NO
8964	046332	012777	000100	132604		MOV	#100, @STKS	::ALLOW TTY KEYBOARD INTERRUPTS
8965	046340	000137	042602		8\$:	JMP	OPERSEL	::CONTROL-C RESTART
8966	046344	021627	000025		9\$:	CMP	(SP), #25	::IS IT A CONTROL-U?
8967	046350	001005				BNE	10\$::BRANCH IF NOT
8968	046352	104401	046732			TYPE	,\$CNTLU	::YES, ECHO CONTROL-U (^U)
8969	046356	062706	000006		20\$:	ADD	#6, SP	::IGNORE PREVIOUS INPUT
8970	046362	000737				BR	19\$::LET'S TRY IT AGAIN
8971	046364	021627	000015		10\$:	CMP	(SP), #15	::IS IT A <CR>?
8972	046370	001022				BNE	16\$::BRANCH IF NO
8973	046372	005766	000004			TST	4(SP)	::YES, IS IT THE FIRST CHAR?
8974	046376	001403				BEQ	11\$::BRANCH IF YES
8975	046400	016677	000002	132532		MOV	2(SP), @SWR	::SAVE NEW SWR
8976	046406	062706	000006		11\$:	ADD	#6, SP	::CLEAR UP STACK
8977	046412	104401	001223		14\$:	TYPE	,\$CRLF	::ECHO <CR> AND <LF>
8978	046416	123727	001135	000001		CMPB	\$INTAG, #1	::RE-ENABLE TTY KBD INTERRUPTS?
8979	046424	001003				BNE	15\$::BRANCH IF NOT
8980	046426	012777	000100	132510		MOV	#100, @STKS	::RE-ENABLE TTY KBD INTERRUPTS
8981	046434	000002			15\$:	RTI		::RETURN
8982	046436	004737	045576		16\$:	JSR	PC, \$TYPEC	::ECHO CHAR
8983	046442	021627	000060			CMP	(SP), #60	::CHAR < 0?
8984	046446	002420				BLT	18\$::BRANCH IF YES
8985	046450	021627	000067			CMP	(SP), #67	::CHAR > 7?
8986	046454	003015				BGT	18\$::BRANCH IF YES
8987	046456	042726	000060			BIC	#60, (SP)+	::STRIP-OFF ASCII
8988	046462	005766	000002			TST	2(SP)	::IS THIS THE FIRST CHAR
8989	046466	001403				BEQ	17\$::BRANCH IF YES
8990	046470	006316				ASL	(SP)	::NO, SHIFT PRESENT
8991	046472	006316				ASL	(SP)	::CHAR OVER TO MAKE
8992	046474	006316				ASL	(SP)	::ROOM FOR NEW ONE.

```

8993 046476 005266 000002 17$: INC 2(SP) ::KEEP COUNT OF CHAR
8994 046502 056616 177776 BIS -2(SP),(SP) ::SET IN NEW CHAR
8995 046506 000667 BR 7$ ::GET THE NEXT ONE
8996 046510 104401 001222 18$: TYPE $QUES ::TYPE ?<CR><LF>
8997 046514 000720 BR 20$ ::SIMULATE CONTROL-U
8998 046516 011646 $RDCHR: MOV (SP),-(SP) ::PUSH DOWN THE PC AND
8999 046520 016666 000004 000002 MOV 4(SP),2(SP) ::THE PS
9000 046526 005066 000004 CLR 4(SP) ::GET READY FOR A CHARACTER
9001 046532 005046 CLR -(SP) ::PUT NEW PS ON STACK
9002 046534 012746 046542 MOV #64$,-(SP) ::PUT NEW PC ON STACK
9003 046540 000002 RTI ::POP NEW PC AND PS
9004 046542 005737 045646 1$: TST $TKCNT ::WAIT ON A CHARACTER
9005 046546 001775 BEQ 1$ DEC $TKCNT ::DECREMENT THE COUNTER
9006 046550 005337 045646 MOV @TKQOUT,4(SP) ::GET ONE CHARACTER
9007 046554 117766 177072 000004 INC $TKQOUT ::UPDATE THE POINTER
9008 046562 005237 045652 CMP $TKQOUT,#$TKQEND ::DID IT GO OFF OF THE END?
9009 046566 023727 045652 045665 BNE 2$ ::BRANCH IF NO
9010 046574 001003 MOV #$TKQSRT,$TKQOUT ::RESE; THE POINTER
9011 046576 012737 045654 045652 2$: RTI ::RETURN
9012 046604 000002 $RDLIN: MOV R3,-(SP) ::SAVE R3
9013 046606 010346 1$: MOV #$TTYIN,R3 ::GET ADDRESS
9014 046610 012703 046714 2$: CMP #$TTYIN+9.,R3 ::BUFFER FULL?
9015 046614 022703 046725 BLOS 4$ ::BR IF YES
9016 046620 101405 RDCHR ::GO READ ONE CHARACTER FROM THE TTY
9017 046622 104410 MOV (SP)+,(R3) ::GET CHARACTER
9018 046624 112613 10$: CMPB #177,(R3) ::IS IT A RUBOUT
9019 046626 122713 000177 BNE 3$ ::SKIP IF NOT
9020 046632 001003 4$: TYPE $QUES ::TYPE A '?'
9021 046634 104401 001222 BR 1$ ::CLEAR THE BUFFER AND LOOP
9022 046640 000763 3$: MOV (R3),9$ ::ECHO THE CHARACTER
9023 046642 111337 046712 TYPE 9$
9024 046646 104401 046712 CMPB #15,(R3)+ ::CHECK FOR RETURN
9025 046652 122723 000015 BNE 2$ ::LOOP IF NOT RETURN
9026 046656 001356 CLR -1(R3) ::CLEAR RETURN (THE 15)
9027 046660 105063 177777 TYPE $LF ::TYPE A LINE FEED
9028 046664 104401 001224 MOV (SP)+,R3 ::RESTORE R3
9029 046670 012603 MOV (SP),-(SP) ::ADJUST THE STACK AND PUT ADDRESS OF THE
9030 046672 011646 MOV 4(SP),2(SP) :: FIRST ASCII CHARACTER ON IT
9031 046674 016666 000004 000002 MOV #$TTYIN,4(SP)
9032 046702 012766 046714 000004 RTI ::RETURN
9033 046710 000002 9$: .BYTE 0 ::STORAGE FOR ASCII CHAR. TO TYPE
9034 046712 000 .BYTE 0 ::TERMINATOR
9035 046713 000 $CNTLC: .ASCIZ /^C/<15><12> ::CONTROL 'C'
9036 046725 136 006503 000012 $CNTLU: .ASCIZ /^U/<15><12> ::CONTROL 'U'
9037 046732 052536 005015 000 $CNTLG: .ASCIZ /^G/<15><12> ::CONTROL 'G'
9038 046737 136 006507 000012 $MSWR: .ASCIZ <15><12>/SWR = /
9039 046744 005015 053523 020122 $MNEW: .ASCIZ / NEW - /
9040 046752 020075 000
9041 046755 040 047040 053505
9042 046762 036440 000040
9043

```

;FROM THE TTY

```

9044 046766 011646          $RDOCT: MOV      (SP),-(SP)      ;;PROVIDE SPACE FOR THE
9045 046770 016666 000004 000002 MOV      4(SP),2(SP)      ;;INPUT NUMBER
9046 046776 010046          MOV      R0,-(SP)        ;;PUSH R0 ON STACK
9047 047000 010146          MOV      R1,-(SP)        ;;PUSH R1 ON STACK
9048 047002 010246          MOV      R2,-(SP)        ;;PUSH R2 ON STACK
9049 047004 104411          1$:  RDLIN              ;;READ AN ASCII LINE
9050 047006 012600          MOV      (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
9051 047010 010037 047114  MOV      R0,5$           ;;AND SAVE IT
9052 047014 005001          CLR      R1              ;;CLEAR DATA WORD
9053 047016 005002          CLR      R2
9054 047020 112046          2$:  MOVB      (R0)+,-(SP)  ;;PICKUP THIS CHARACTER
9055 047022 001420          BEQ      3$              ;;IF ZERO GET OUT
9056 047024 122716 000060  CMPB     #'0,(SP)        ;;MAKE SURE THIS CHARACTER
9057 047030 003026          BGT      4$              ;;IS AN OCTAL DIGIT
9058 047032 122716 000067  CMPB     #'7,(SP)
9059 047036 002423          BLT      4$
9060 047040 006301          ASL      R1              ;;*2
9061 047042 006102          ROL      R2
9062 047044 006301          ASL      R1              ;;*4
9063 047046 006102          ROL      R2
9064 047050 006301          ASL      R1              ;;*8
9065 047052 006102          ROL      R2
9066 047054 042716 177770  BIC      #'C7,(SP)      ;;STRIP THE ASCII JUNK
9067 047060 062601          ADD      (SP)+,R1        ;;ADD IN THIS DIGIT
9068 047062 000756          BR       2$              ;;LOOP
9069 047064 005726          3$:  TST      (SP)+        ;;CLEAN TERMINATOR FROM STACK
9070 047066 010166 000012  MOV      R1,12(SP)      ;;SAVE THE RESULT
9071 047072 010237 047124  MOV      R2,$HIOCT
9072 047076 012602          MOV      (SP)+,R2        ;;POP STACK INTO R2
9073 047100 012601          MOV      (SP)+,R1        ;;POP STACK INTO R1
9074 047102 012600          MOV      (SP)+,R0        ;;POP STACK INTO R0
9075 047104 000002          RTI
9076 047106 005726          4$:  TST      (SP)+        ;;CLEAN PARTIAL FROM STACK
9077 047110 105010          CLRB     (R0)           ;;SET A TERMINATOR
9078 047112 104401          TYPE
9079 047114 000000          5$:  .WORD     0           ;;TYPE UP THRU THE BAD CHAR.
9080 047116 104401 001222  TYPE     ,$QUES        ;; '?' 'CR' & 'LF'
9081 047122 000730          BR       1$              ;;TRY AGAIN
9082 047124 000000          $HIOCT: .WORD     0     ;;HIGH ORDER BITS GO HERE
  
```

9083	047126	104407				CKSWR		:::TEST FOR CHANGE IN SOFT-SWR
9084	047130	012737	177777	004734		MOV	#-1,@WERFLG\$:::SET ERROR FLAG
9085	047136	105237	001103		7\$:	INCB	\$ERFLG	:::SET THE ERROR FLAG
9086	047142	001775				BEQ	7\$:::DON'T LET THE FLAG GO TO ZERO
9087	047144	013777	001102	131770		MOV	\$STNM,@DISPLAY	:::DISPLAY TEST NUMBER AND ERROR FLAG
9088	047152	032777	002000	131760		BIT	#BIT10,@SWR	:::BELL ON ERROR?
9089	047160	001402				BEQ	1\$:::NO - SKIP
9090	047162	104401	001216			TYPE	,\$BELL	:::RING BELL
9091	047166	005237	001112		1\$:	INC	\$ERTTL	:::COUNT THE NUMBER OF ERRORS
9092	047172	011637	001116			MOV	(SP),\$ERRPC	:::GET ADDRESS OF ERROR INSTRUCTION
9093	047176	162737	000002	001116		SUB	#2,\$ERRPC	
9094	047204	117737	131706	001114		MOVB	@\$ERRPC,\$ITEMB	:::STRIP AND SAVE THE ERROR ITEM CODE
9095	047212	032777	020000	131720		BIT	#BIT13,@SWR	:::SKIP TYPEOUT IF SET
9096	047220	001004				BNE	20\$:::SKIP TYPEOUTS
9097	047222	004737	047276			JSR	PL,\$ERRTYP	:::GO TO USER ERROR ROUTINE
9098	047226	104401	001223			TYPE	,\$CRLF	
9099	047232	005777	131702		2\$:	TST	@SWR	:::HALT ON ERROR
9100	047236	100002				BPL	3\$:::SKIP IF CONTINUE
9101	047240	000000				HALT		:::HALT ON ERROR!
9102	047242	104407				CKSWR		:::TEST FOR CHANGE IN SOFT-SWR
9103	047244	032777	001000	131666	3\$:	BIT	#BIT09,@SWR	:::LOOP ON ERROR SWITCH SET?
9104	047252	001402				BEQ	4\$:::BR IF NO
9105	047254	013716	001110			MOV	\$LPERR,(SP)	:::FUDGE RETURN FOR LOOPING
9106	047260	005737	001214		4\$:	TST	\$ESCAPF	:::CHECK FOR AN ESCAPE ADDRESS
9107	047264	001402				BEQ	5\$:::BR IF NONE
9108	047266	013716	001214			MOV	\$ESCAPE,(SP)	:::FUDGE RETURN ADDRESS FOR ESCAPE
9109	047272	000002				RTI		:::RETURN
9110								

```
9111
9112          ;*:*****
9113
9114          .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
9115
9116          ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
9117          ;*ERROR IS TO BE REPORTED.  IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
9118          ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
9119          ;*IT IS A COPY OF THE $ERRTYP SUBROUTINE FROM SYSMAC.
9120          ;*WITH ONLY MINOR CHANGES
9121          ;*FIRST IF SWITCH 6 IS SET AND SWITCH 8 RESET THEN
9122          ;*ALL REGISTER CONTENTS WILL BE TYPED BEFOR REPORTING THE ERROR
9123          ;*SECOND IF THE CURRENT ERROR HAS THE SAME ITEM NUMBER
9124          ;*AS THE PREVIOUS ERROR THEN ONLY THE DATA WILL BE TYPED
9125          ;*AND NOT THE ERROR MESSAGE AND HEADER.
9126
9127 047274 000000          PRITEM: 0          ;PREVIOUS ITEM NO. LOCATION
9128
9129 047276 017746 131636  $ERRTYP: MOV @SWR,-(SP)          ;GET SWITCH SETTING
9130 047302 042716 177277          BIC #^C500,(SP)          ;KEEP ONLY SWITCH 8 AND 6
9131 047306 022726 000100          CMP #SW06,(SP)+          ;IS 6 SET AND 8 RESET
9132 047312 001001          BNE 1$          ;IF NOT BRANCH
9133 047314 000402          BR 2$          ;BRANCH IF SW 6 IS SET AND 8 RESET
9134 047316 000137 050236 1$: JMP @#TYPERR          ;JUMP IF SW 8 IS SET
9135                                     ;OR IF SW 8 IS RESET AND SW 6 IS RESET
9136
9136 047322          2$:
9137 047322 104401 047330          TYPE ,65$          ;;TYPE ASCIZ STRING
9138 047326 000406          BR 64$          ;;GET OVER THE ASCIZ
9139 047344 013746 002354          MOV @#WC,-(SP)          ;GET READY TO TYPE RHWC CONTENTS
9140 047350 104402          TYP0C
9141 047352 104401 047360          TYPE ,67$          ;;TYPE ASCIZ STRING
9142 047356 000406          BR 66$          ;;GET OVER THE ASCIZ
9143 047374 013746 002356          MOV @#BA,-(SP)          ;GET READY TO TYPE RHBA CONTENTS
9144 047400 104402          TYP0C
9145 047402 104401 047410          TYPE ,69$          ;;TYPE ASCIZ STRING
9146 047406 000406          BR 68$          ;;GET OVER THE ASCIZ
9147 047424 013746 002360          MOV @#CS2,-(SP)          ;GET READY TO TYPE RHCS2 CONTENTS
9148 047430 104402          TYP0C
9149 047432 104401 047440          TYPE ,71$          ;;TYPE ASCIZ STRING
9150 047436 000406          BR 70$          ;;GET OVER THE ASCIZ
9151 047454 013746 002362          MOV @#CS1,-(SP)          ;GET READY TO TYPE RHCS1 CONTENTS
9152 047460 104402          TYP0C
9153 047462 104401 047470          TYPE ,73$          ;;TYPE ASCIZ STRING
9154 047466 000406          BR 72$          ;;GET OVER THE ASCIZ
9155 047504 013746 002404          MOV @#DS1,-(SP)          ;GET READY TO TYPE RHDS1 CONTENTS
9156 047510 104402          TYP0C
9157 047512 104401 047520          TYPE ,75$          ;;TYPE ASCIZ STRING
9158 047516 000406          BR 74$          ;;GET OVER THE ASCIZ
9159 047534 013746 002364          MOV @#ER1,-(SP)          ;GET READY TO TYPE RHER1 CONTENTS
9160 047540 104402          TYP0C
9161 047542 104401 047550          TYPE ,77$          ;;TYPE ASCIZ STRING
9162 047546 000406          BR 76$          ;;GET OVER THE ASCIZ
9163 047564 013746 002370          MOV @#ER2,-(SP)          ;GET READY TO TYPE RHER2 CONTENTS
9164 047570 104402          TYP0C
9165 047572 104401 047600          TYPE ,79$          ;;TYPE ASCIZ STRING
9166 047576 000406          BR 78$          ;;GET OVER THE ASCIZ
```

```

9167 047614 013746 002376      MOV      @MER3,-(SP)      ;GET READY TO TYPE RHER3 CONTENTS
9168 047620 104402              TYPCC
9169 047622 104401 047630      TYPE      ,81$          ;;TYPE ASCIZ STRING
9170 047626 000406              BR        80$           ;;GET OVER THE ASCIZ
9171 047644 013746 002366      MOV      @WDST,-(SP)    ;GET READY TO TYPE RHDST CONTENTS
9172 047650 104402              TYPCC
9173 047652 104401 047660      TYPE      ,83$          ;;TYPE ASCIZ STRING
9174 047656 000406              BR        82$           ;;GET OVER THE ASCIZ
9175 047674 013746 002374      MOV      @WCA,-(SP)    ;GET READY TO TYPE RHCA CONTENTS
9176 047700 104402              TYPCC
9177 047702 104401 047710      TYPE      ,85$          ;;TYPE ASCIZ STRING
9178 047706 000406              BR        84$           ;;GET OVER THE ASCIZ
9179 047724 013746 002400      MOV      @WAS,-(SP)    ;GET READY TO TYPE RHAS CONTENTS
9180 047730 104402              TYPCC
9181 047732 104401 047740      TYPE      ,87$          ;;TYPE ASCIZ STRING
9182 047736 000406              BR        86$           ;;GET OVER THE ASCIZ
9183 047754 013746 002372      MOV      @WOF,-(SP)    ;GET READY TO TYPE RHOF CONTENTS
9184 047760 104402              TYPCC
9185 047762 104401 047770      TYPE      ,89$          ;;TYPE ASCIZ STRING
9186 047766 000406              BR        88$           ;;GET OVER THE ASCIZ
9187 050004 013746 002402      MOV      @WMR,-(SP)    ;GET READY TO TYPE RHMR CONTENTS
9188 050010 104402              TYPCC
9189 050012 104401 050020      TYPE      ,91$          ;;TYPE ASCIZ STRING
9190 050016 000406              BR        90$           ;;GET OVER THE ASCIZ
9191 050034 013746 002420      MOV      @WLA,-(SP)    ;GET READY TO TYPE RHLA CONTENTS
9192 050040 104402              TYPCC
9193 050042 104401 050050      TYPE      ,93$          ;;TYPE ASCIZ STRING
9194 050046 000406              BR        92$           ;;GET OVER THE ASCIZ
9195 050064 013746 002416      MOV      @WCC,-(SP)    ;GET READY TO TYPE RHCC CONTENTS
9196 050070 104402              TYPCC
9197 050072 104401 050100      TYPE      ,95$          ;;TYPE ASCIZ STRING
9198 050076 000406              BR        94$           ;;GET OVER THE ASCIZ
9199 050114 013746 002412      MOV      @WEC1,-(SP)   ;GET READY TO TYPE RHEC1 CONTENTS
9200 050120 104402              TYPCC
9201 050122 104401 050130      TYPE      ,97$          ;;TYPE ASCIZ STRING
9202 050126 000406              BR        96$           ;;GET OVER THE ASCIZ
9203 050144 013746 002414      MOV      @WEC2,-(SP)   ;GET READY TO TYPE RHEC2 CONTENTS
9204 050150 104402              TYPCC
9205 050152 104401 050160      TYPE      ,99$          ;;TYPE ASCIZ STRING
9206 050156 000406              BR        98$           ;;GET OVER THE ASCIZ
9207 050174 013746 002406      MOV      @WDT,-(SP)    ;GET READY TO TYPE RHDT CONTENTS
9208 050200 104402              TYPCC
9209 050202 104401 050210      TYPE      ,101$         ;;TYPE ASCIZ STRING
9210 050206 000406              BR        100$          ;;GET OVER THE ASCIZ
9211 050224 013746 002410      MOV      @WSN,-(SP)    ;GET READY TO TYPE RHSN CONTENTS
9212 050230 104402              TYPCC
9213 050232 005037 047274      CLR      @WPRITEM      ;CLEAR PREVIOUS ERROR ITEM
9214 050236              TYPERR:
9215 050236 104401 001223      TYPE      ,$CRLF        ;'CARRIAGE RETURN' & 'LINE FEED'
9216 050242 010046              MOV      RO,-(SP)      ;SAVE RO
9217 050244 005000              CLR      RO            ;PICKUP THE ITEM INDEX
9218 050246 153700 001114      BISB    @W$ITEMB,RO
9219 050252 001004              BNE     1$             ;IF ITEM NUMBER IS ZERO, JUST
9220                                ;TYPE THE PC OF THE ERROR
9221 050254 013746 001116      MOV      $ERRPC,-(SP)  ;SAVE $ERRPC FOR TYPEOUT
9222                                ;ERROR ADDRESS
  
```

```
9223 050260 104402          TYPOC          :GO TYPE--OCTAL ASCII(ALL DIGITS)
9224 050262 000454          BR            10$          :GET OUT
9225 050264 005300          1$: DEC        RO          :ADJUST THE INDEX SO THAT IT WILL
9226 050266 006300          ASL          RO          :      WORK FOR THE ERROR TABLE
9227 050270 006300          ASL          RO
9228 050272 006300          ASL          RO
9229 050274 062700 001226    ADD          #SERRTB,RO    :FORM TABLE POINTER
9230 050300 020037 047274    CMP          RO,@#PRITEM  :WAS PREVIOUS ERROR SAME
9231 050304 001002          BNE          13$         :BRANCH IF NOT
9232 050306 022020          CMP          (RO)+,(RO)+ :POP RO OVER EM AND DH
9233 050310 000420          BR            5$
9234 050312 010037 047274    13$: MOV        RO,@#PRITEM :SAVE NEW ERROR ITEM
9235 050316 012037 050326    MOV        (RO)+,2$      :PICKUP 'ERROR MESSAGE' POINTER
9236 050322 001404          BEQ         3$           :SKIP TYPEOUT IF NO POINTER
9237 050324 104401          TYPE        :TYPE THE 'ERROR MESSAGE'
9238 050326 000000          .WORD      0            :'ERROR MESSAGE' POINTER GOES HERE
9239 050330 104401 001223    TYPE        ,SCLRF       :'CARRIAGE RETURN' & 'LINE FEED'
9240 050334 012037 050344    3$: MOV        (RO)+,4$    :PICKUP 'DATA HEADER' POINTER
9241 050340 001404          BEQ         5$           :SKIP TYPEOUT IF 0
9242 050342 104401          TYPE        :TYPE THE 'DATA HEADER'
9243 050344 000000          .WORD      0            :'DATA HEADER' POINTER GOES HERE
9244 050346 104401 001223    TYPE        ,SCLRF       :'CARRIAGE RETURN' & 'LINE FEED'
9245 050352 010146          5$: MOV        R1,-(SP)    :SAVE R1
9246 050354 012001          MOV        (RO)+,R1      :PICKUP 'DATA TABLE' POINTER
9247 050356 001415          BEQ         9$           :BR IF NO DATA TO BE TYPED
9248 050360 012000          MOV        (RO)+,RO      :PICKUP 'DATA FORMAT' POINTER
9249 050362 105720          6$: TSTB      (RO)+       :'OCTAL' OR 'DECIMAL'
9250 050364 001003          BNE          7$         :BR IF DECIMAL
9251 050366 013146          MOV        @ (R1)+,-(SP) :SAVE @ (R1)+ FOR TYPEOUT
9252 050370 104402          TYPOC          :GO TYPE--OCTAL ASCII(ALL DIGITS)
9253 050372 000402          BR            8$
9254 050374          7$:
9255 050374 013146          MOV        @ (R1)+,-(SP) :SAVE @ (R1)+ FOR TYPEOUT
9256 050376 104405          TYPDS        :GO TYPE--DECIMAL ASCII WITH SIGN
9257 050400 005711          8$: TST        (R1)       :IS THERE ANOTHER NUMBER?
9258 050402 001403          BEQ         9$         :BR IF NO
9259 050404 104401 050420    TYPE        ,11$        :TYPE TWO(2) SPACES
9260 050410 000764          BR            6$         :LOOP
9261
9262 050412 012601          9$: MOV        (SP)+,R1    :RESTORE R1
9263 050414 012600          10$: MOV       (SP)+,RO   :'CARRIAGE RETURN' & 'LINE FEED'
9264 050416 000207          RTS         PC          :RETURN
9265 050420 020040 000          11$: .ASCIZ    / /        :TWO(2) SPACES
9266 050424
9267
```



```
9268
9269 050424 017646 000000 $TYPOS: MOV @ (SP), (SP) ;; PICKUP THE MODE
9270 050430 116637 000001 050647 MOVB 1 (SP), $OFILL ;; LOAD ZERO FILL SWITCH
9271 050436 112637 050651 MOVB (SP)+, $OMODE+1 ;; NUMBER OF DIGITS TO TYPE
9272 050442 062716 000002 ADD #2, (SP) ;; ADJUST RETURN ADDRESS
9273 050446 000406 BR $TYPON
9274 050450 112737 000001 050647 $TYPOC: MOVB #1, $OFILL ;; SET THE ZERO FILL SWITCH
9275 050456 112737 000006 050651 MOVB #6, $OMODE+1 ;; SET FOR SIX(6) DIGITS
9276 050464 112737 000005 050646 $TYPON: MOVB #5, $OCNT ;; SET THE ITERATION COUNT
9277 050472 010346 MOV R3, -(SP) ;; SAVE R3
9278 050474 010446 MOV R4, -(SP) ;; SAVE R4
9279 050476 010546 MOV R5, -(SP) ;; SAVE R5
9280 050500 113704 050651 MOVB $OMODE+1, R4 ;; GET THE NUMBER OF DIGITS TO TYPE
9281 050504 005404 NEG R4
9282 050506 062704 000006 ADD #6, R4 ;; SUBTRACT IT FOR MAX. ALLOWED
9283 050512 110437 050650 MOVB R4, $OMODE ;; SAVE IT FOR USE
9284 050516 113704 050647 MOVB $OFILL, R4 ;; GET THE ZERO FILL SWITCH
9285 050522 016605 000012 MOV 12 (SP), R5 ;; PICKUP THE INPUT NUMBER
9286 050526 005003 CLR R3 ;; CLEAR THE OUTPUT WORD
9287 050530 006105 1$: ROL R5 ;; ROTATE MSB INTO 'C'
9288 050532 000404 BR 3$ ;; GO DO MSB
9289 050534 006105 2$: ROL R5 ;; FORM THIS DIGIT
9290 050536 006105 ROL R5
9291 050540 006105 ROL R5
9292 050542 010503 MOV R5, R3
9293 050544 006103 3$: ROL R3 ;; GET LSB OF THIS DIGIT
9294 050546 105337 050650 DECB $OMODE ;; TYPE THIS DIGIT?
9295 050552 100016 BPL 7$ ;; BR IF NO
9296 050554 042703 177770 BIC #177770, R3 ;; GET RID OF JUNK
9297 050560 001002 BNE 4$ ;; TEST FOR 0
9298 050562 005704 TST R4 ;; SUPPRESS THIS 0?
9299 050564 001403 BEQ 5$ ;; BR IF YES
9300 050566 005204 4$: INC R4 ;; DON'T SUPPRESS ANYMORE 0'S
9301 050570 052703 000060 BIS #'0, R3 ;; MAKE THIS DIGIT ASCII
9302 050574 052703 000040 5$: BIS #' , R3 ;; MAKE ASCII IF NOT ALREADY
9303 050600 110337 050644 MOVB R3, 8$ ;; SAVE FOR TYPING
9304 050604 104401 050644 TYPE , 8$ ;; GO TYPE THIS DIGIT
9305 050610 105337 050646 7$: DECB $OCNT ;; COUNT BY 1
9306 050614 003347 BGT 2$ ;; BR IF MORE TO DO
9307 050616 002402 BLT 6$ ;; BR IF DONE
9308 050620 005204 INC R4 ;; INSURE LAST DIGIT ISN'T A BLANK
9309 050622 000744 BR 2$ ;; GO DO THE LAST DIGIT
9310 050624 012605 6$: MOV (SP)+, R5 ;; RESTORE R5
9311 050626 012604 MOV (SP)+, R4 ;; RESTORE R4
9312 050630 012603 MOV (SP)+, R3 ;; RESTORE R3
9313 050632 016666 000002 000004 MOV 2 (SP), 4 (SP) ;; SET THE STACK FOR RETURNING
9314 050640 012616 MOV (SP)+, (SP)
9315 050642 000002 RTI ;; RETURN
9316 050644 000 .BYTE 0 ;; STORAGE FOR ASCII DIGIT
9317 050645 000 .BYTE 0 ;; TERMINATOR FOR TYPE ROUTINE
9318 050646 000 $OCNT: .BYTE 0 ;; OCTAL DIGIT COUNTER
9319 050647 000 $OFILL: .BYTE 0 ;; ZERO FILL SWITCH
9320 050650 000000 $OMODE: .WORD 0 ;; NUMBER OF DIGITS TO TYPE
```

9321	050652	010046			\$TRAP:	MOV	RO,-(SP)	::SAVE RO
9322	050654	016600	000002			MOV	2(SP),RO	::GET TRAP ADDRESS
9323	050660	005740				TST	-(RO)	::BACKUP BY 2
9324	050662	111000				MOVB	(RO),RO	::GET RIGHT BYTE OF TRAP
9325	050664	006300				ASL	RO	::POSITION FOR INDEXING
9326	050666	016000	050706			MOV	\$TRPAD(RO),RO	::INDEX TO TABLE
9327	050672	000200				RTS	RO	::GO TO ROUTINE
9328	050674	011646			\$TRAP2:	MOV	(SP),-(SP)	::MOVE THE PC DOWN
9329	050676	016666	000004	000002		MOV	4(SP),2(SP)	::MOVE THE PSW DOWN
9330	050704	000002				RTI		::RESTORE THE PSW
9331	050706	050674			\$TRPAD:	.WORD	\$TRAP2	
9332	050710	045426				\$TYPE	::CALL=TYPE	TRAP+1(104401) TTY TYPEOUT ROUTINE
9333	050712	050450				\$TYPOC	::CALL=TYPOC	TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
9334	050714	050424				\$TYPOS	::CALL=TYPOS	TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)
9335	050716	050464				\$TYPON	::CALL=TYPON	TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)
9336	050720	045202				\$TYPDS	::CALL=TYPDS	TRAP+5(104405) TYPE DECIMAL NUMBER (WITH SIGN)
9337	050722	046244				\$GTSWR	::CALL=GTSWR	TRAP+6(104406) GET SOFT-SWR SETTING
9338	050724	046154				\$CKSWR	::CALL=CKSWR	TRAP+7(104407) TEST FOR CHANGE IN SOFT-SWR
9339	050726	046516				\$RDCHR	::CALL=RDCHR	TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
9340	050730	046606				\$RDLIN	::CALL=RDLIN	TRAP+11(104411) TTY TYPEIN STRING ROUTINE
9341	050732	046766				\$RDOCT	::CALL=RDOCT	TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
9342	050734	042156				WAIT.T	::CALL WAT	TRAP+13(104413) DONT ADD ABOVE THIS TRAP
9343								

```

9344 050736 012737 051102 000024 $PWRDN: MOV    #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
9345 050744 012737 000340 000026      MOV    #340,@#PWRVEC+2 ;;PRIO:7
9346 050752 010046          MOV    R0,-(SP)      ;;PUSH R0 ON STACK
9347 050754 010146          MOV    R1,-(SP)      ;;PUSH R1 ON STACK
9348 050756 010246          MOV    R2,-(SP)      ;;PUSH R2 ON STACK
9349 050760 010346          MOV    R3,-(SP)      ;;PUSH R3 ON STACK
9350 050762 010446          MOV    R4,-(SP)      ;;PUSH R4 ON STACK
9351 050764 010546          MOV    R5,-(SP)      ;;PUSH R5 ON STACK
9352 050766 017746 130146      MOV    @SWR,-(SP)    ;;PUSH @SWR ON STACK
9353 050772 010637 051106      MOV    SP,$SAVR6    ;;SAVE SP
9354 050776 012737 051010 000024      MOV    #$PWRUP,@#PWRVEC ;;SET UP VECTOR
9355 051004 000000          HALT
9356 051006 000776          BR     .-2          ;;HANG UP
9357 051010 012737 051102 000024 $PWRUP: MOV    #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
9358 051016 013706 051106      MOV    $SAVR6,SP    ;;GET SP
9359 051022 005037 051106      CLR    $SAVR6      ;;WAIT LOOP FOR THE TTY
9360 051026 005237 051106      1$:   INC    $SAVR6  ;;WAIT FOR THE INC
9361 051032 001375          BNF    1$          ;;OF WORD
9362 051034 012677 130100      MOV    (SP)+,@SWR   ;;POP STACK INTO @SWR
9363 051040 012605          MOV    (SP)+,R5    ;;POP STACK INTO R5
9364 051042 012604          MOV    (SP)+,R4    ;;POP STACK INTO R4
9365 051044 012603          MOV    (SP)+,R3    ;;POP STACK INTO R3
9366 051046 012602          MOV    (SP)+,R2    ;;POP STACK INTO R2
9367 051050 012601          MOV    (SP)+,R1    ;;POP STACK INTO R1
9368 051052 012600          MOV    (SP)+,R0    ;;POP STACK INTO R0
9369 051054 012737 050736 000024      MOV    #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
9370 051062 012737 000340 000026      MOV    #340,@#PWRVEC+2 ;;PRIO:7
9371 051070 104401          TYPE   $POWER      ;;REPORT THE POWER FAILURE
9372 051072 051110          $PWRMG: .WORD    $POWER ;;POWER FAIL MESSAGE POINTER
9373 051074 012716          MOV    (PC)+,(SP)  ;;RESTART AT BEGIN
9374 051076 005012          $PWRAD: .WORD    BEGIN  ;;RESTART ADDRESS
9375 051100 000002          RTI
9376 051102 000000          $ILLUP: HALT      ;;THE POWER UP SEQUENCE WAS STARTED
9377 051104 000776          BR     .-2          ;; BEFORE THE POWER DOWN WAS COMPLETE
9378 051106 000000          $SAVR6: 0          ;;PUT THE SP HERE
9379 051110 005015 047520 042527 $POWER: .ASCIIZ <15><12>'POWER'
9380 051116 000122
  
```

9381
9382
9383
9384
9385
9386
9387
9388
9389
9390
9391
9392
9393
9394
9395
9396
9397
9398
9399
9400
9401
9402
9403
9404
9405
9406
9407
9408
9409
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

051120 050122 032060 042040
051126 042111 047040 052117
051134 044440 052116 051105
051142 052522 052120 000
051147 111 052116 051105
051154 052522 052120 042440
051162 040516 046102 020105
051170 044502 020124 047504
051176 047127 041040 052125
051204 042440 050130 041505
051212 042524 020104 044502
051220 020124 044504 020104
051226 047516 020124 042523
051234 000124
051236 050122 032060 042040
051244 042111 047040 052117
051252 044440 052116 051105
051260 052522 052120 053440
051266 042510 020116 054105
051274 042520 052103 042105
051302 041040 052111 042040
051310 042111 051440 052105
051316 000
051317 105 050130 041505
051324 042524 020104 044502
051332 020124 044504 020104
051340 042523 020124 052502
051346 020124 044524 042515
051354 044440 020123 047111
051362 042440 051122 051117
051370 024040 044524 042515
051376 044440 020116 030061
051404 046440 041511 047522
051412 042523 026103 042040
051420 041505 046511 046101
051426 000051
051430 044122 051501 042040
051436 042517 020123 047516
051444 020124 046103 040505
051452 020122 054502 046440
051460 053117 047111 020107
051466 047111 040440 046114
051474 047440 042516 000123
051502 047514 042101 047111
051510 020107 044122 051105
051516 020061 047506 020122
051524 046101 020114 047125

*
*ERROR AND MESSAGE TABLE CONDIMENTS
*

EM1: .ASCIZ /RPO4 DID NOT INTERRUPT/

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

EM3: .ASCIZ /RPO4 DID NOT INTERRUPT WHEN EXPECTED BIT DID SET/

EM4: .ASCIZ /EXPECTED BIT DID SET BUT TIME IS IN ERROR (TIME IN '0 MICROSEC, DECIMAL

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

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

9437	051532	052111	020123	044504	
9438	051540	020104	047516	020124	
9439	051546	042523	020124	047101	
9440	051554	020131	044122	051501	
9441	051562	041040	052111	000123	
9442	051570	047516	020116	054105	EM7: .ASCIZ /NON EXISTENT REGISTER, PROGRAM ABORTED./
9443	051576	051511	042524	052116	
9444	051604	051040	043505	051511	
9445	051612	042524	026122	050040	
9446	051620	047522	051107	046501	
9447	051626	040440	047502	052122	
9448	051634	042105	000056		
9449					
9450	051640	052123	050117	042520	EM10: .ASCIZ /STOPPED DRIVE HAS MOL BIT IN RHDS1 SET/
9451	051646	020104	051104	053111	
9452	051654	020105	040510	020123	
9453	051662	047515	020114	044502	
9454	051670	020124	047111	051040	
9455	051676	042110	030523	051440	
9456	051704	052105	000		
9457	051707	127	052111	020110	EM11: .ASCIZ /WITH SPINDLE POWERED DOWN RHCS2 SHOULD ONLY HAVE JNIT NO. AND IR SET/
9458	051714	050123	047111	046104	
9459	051722	020105	047520	042527	
9460	051730	042522	020104	047504	
9461	051736	047127	051040	041510	
9462	051744	031123	051440	047510	
9463	051752	046125	020104	047117	
9464	051760	054514	044040	053101	
9465	051766	020105	047125	052111	
9466	051774	047040	027117	040440	
9467	052002	042116	044440	020122	
9468	052010	042523	000124		
9469	052014	043101	042524	020122	EM12: .ASCIZ /AFTER SPINDLE POWERED UP, NO PACK ACKN. RHDS1 SHOULD HAVE MOL=1, VV=0/
9470	052022	050123	047111	046104	
9471	052030	020105	047520	042527	
9472	052036	042522	020104	050125	
9473	052044	020054	047516	050040	
9474	052052	041501	020113	041501	
9475	052060	047113	020056	044122	
9476	052066	051504	020061	044123	
9477	052074	052517	042114	044040	
9478	052102	053101	020105	047515	
9479	052110	036514	026061	053040	
9480	052116	036526	000060		
9481	052122	044527	044124	051440	EM13: .ASCIZ /WITH SPINDLE POWERED UP, NO INTIALIZE, RHCS1 SHOULD HAVE GO=0, DVA=1, R
9482	052130	044520	042116	042514	
9483	052136	050040	053517	051105	
9484	052144	042105	052440	026120	
9485	052152	047040	020117	047111	
9486	052160	044524	046101	055111	
9487	052166	026105	051040	041510	
9488	052174	030523	051440	047510	
9489	052202	046125	020104	040510	
9490	052210	042526	043440	036517	
9491	052216	026060	042040	040526	
9492	052224	030475	020054	042122	

9493	052232	036531	026061	044440	
9494	052240	036505	000060		
9495	052244	043101	042524	020122	EM14: .ASCIZ /AFTER SPINDLE POWERED UP, RHCC SHOULD BE-0/
9496	052252	050123	047111	046104	
9497	052260	020105	047520	042527	
9498	052266	042522	020104	050125	
9499	052274	020054	044122	041503	
9500	052302	051440	047510	046125	
9501	052310	020104	042502	030075	
9502	052316	000			
9503	052317	120	041501	020113	EM15: .ASCII /PACK ACKNOWLEDGE COMMAND CAUSED AN ERROR/<15><12>
9504	052324	041501	047113	053517	
9505	052332	042514	043504	020105	
9506	052340	047503	046515	047101	
9507	052346	020104	040503	051525	
9508	052354	042105	040440	020116	
9509	052362	051105	047522	006522	
9510	052370	012			
9511	052371	107	047517	020104	.ASCIZ /GOOD DATA IS BEFORE COMMAND, REC DATA IS AFTER COMMAND/
9512	052376	040504	040524	044440	
9513	052404	020123	042502	047506	
9514	052412	042522	041440	046517	
9515	052420	040515	042116	020054	
9516	052426	042522	020103	040504	
9517	052434	040524	044440	020123	
9518	052442	043101	042524	020122	
9519	052450	047503	046515	047101	
9520	052456	000104			
9521	052460	047516	047455	020120	EM16: .ASCII /NO-OP COMMAND CAUSED AN ERROR/<15><12>
9522	052466	047503	046515	047101	
9523	052474	020104	040503	051525	
9524	052502	042105	040440	020116	
9525	052510	051105	047522	006522	
9526	052516	012			
9527	052517	107	047517	020104	.ASCIZ /GOOD DATA IS BEFORE COMMAND, REC DATA IS AFTER COMMAND/
9528	052524	040504	040524	044440	
9529	052532	020123	042502	047506	
9530	052540	042522	041440	046517	
9531	052546	040515	042116	020054	
9532	052554	042522	020103	040504	
9533	052562	040524	044440	020123	
9534	052570	043101	042524	020122	
9535	052576	047503	046515	047101	
9536	052604	000104			
9537	052606	051104	053111	020105	EM17: .ASCII /DRIVE CLEAR COMMAND CAUSED AN ERROR/<15><12>
9538	052614	046103	040505	020122	
9539	052622	047503	046515	047101	
9540	052630	020104	040503	051525	
9541	052636	042105	040440	020116	
9542	052644	051105	047522	006522	
9543	052652	012			
9544	052653	107	047517	020104	.ASCIZ /GOOD DATA GIVES SHOULD BE, REC DATA GIVES AFTER COMMAND/
9545	052660	040504	040524	043440	
9546	052666	053111	051505	051440	
9547	052674	047510	046125	020104	
9548	052702	042502	020054	042522	

9549	052710	020103	040504	040524	
9550	052716	043440	053111	051505	
9551	052724	040440	052106	051105	
9552	052732	041440	046517	040515	
9553	052740	042116	000		
9554					
9555	052743	122	040505	026504	EM20: .ASCII /READ-IN COMMAND CAUSED AN ERROR/<15><12>
9556	052750	047111	041440	046517	
9557	052756	040515	042116	041440	
9558	052764	052501	042523	020104	
9559	052772	047101	042440	051122	
9560	053000	051117	005015		
9561	053004	047507	042117	042040	.ASCIZ /GOOD DATA GIVES SHOULD BE, REC DATA GIVES REG. CONT. AFTER COMMAND/
9562	053012	052101	020101	044507	
9563	053020	042526	020123	044123	
9564	053026	052517	042114	041040	
9565	053034	026105	051040	041505	
9566	053042	042040	052101	020101	
9567	053050	044507	042526	020123	
9568	053056	042522	027107	041440	
9569	053064	047117	027124	040440	
9570	053072	052106	051105	041440	
9571	053100	046517	040515	042116	
9572	053106	000			
9573	053107	122	041510	030523	EM21: .ASCIZ /RHCS1 CONTENTS DURING COMMAND WAS IN ERROR/
9574	053114	041440	047117	042524	
9575	053122	052116	020123	052504	
9576	053130	044522	043516	041440	
9577	053136	046517	040515	042116	
9578	053144	053440	051501	044440	
9579	053152	020116	051105	047522	
9580	053160	000122			
9581	053162	044122	051504	020061	EM22: .ASCIZ /RHDS1 CONTENTS DURING COMMAND WAS IN ERROR/
9582	053170	047503	052116	047105	
9583	053176	051524	042040	051125	
9584	053204	047111	020107	047503	
9585	053212	046515	047101	020104	
9586	053220	040527	020123	047111	
9587	053226	042440	051122	051117	
9588	053234	000			
9589	053235	125	046116	040517	EM23: .ASCII /UNLOAD COMMAND CAUSED AN ERROR/<15><12>
9590	053242	020104	047503	046515	
9591	053250	047101	020104	040503	
9592	053256	051525	042105	040440	
9593	053264	020116	051105	047522	
9594	053272	006522	012		
9595	053275	107	047517	020104	.ASCIZ /GOOD DATA GIVES SHOULD BE, REC DATA GIVES REG. CONT. AFTER COMMAND/
9596	053302	040504	040524	043440	
9597	053310	053111	051505	051440	
9598	053316	047510	046125	020104	
9599	053324	042502	020054	042522	
9600	053332	020103	040504	040524	
9601	053340	043440	053111	051505	
9602	053346	051040	043505	020056	
9603	053354	047503	052116	020056	
9604	053362	043101	042524	020122	

9605	053370	047503	046515	047101
9606	053376	000104		
9607	053400	043117	051506	052105
9608	053406	041440	046517	040515
9609	053414	042116	041440	052501
9610	053422	042523	020104	047101
9611	053430	042440	051122	051117
9612	053436	005015		
9613	053440	047507	042117	042040
9614	053446	052101	020101	044507
9615	053454	042526	020123	044123
9616	053462	052517	042114	041040
9617	053470	026105	051040	041505
9618	053476	042040	052101	020101
9619	053504	044507	042526	020123
9620	053512	042522	027107	041440
9621	053520	047117	027124	040440
9622	053526	052106	051105	041440
9623	053534	046517	040515	042116
9624	053542	000		
9625	053543	122	052105	051125
9626	053550	020116	047524	041440
9627	053556	047105	042524	020122
9628	053564	044514	042516	041440
9629	053572	046517	040515	042116
9630	053600	041440	052501	042523
9631	053606	020104	047101	042440
9632	053614	051122	051117	005015
9633	053622	047507	042117	042040
9634	053630	052101	020101	044507
9635	053636	042526	020123	044123
9636	053644	052517	042114	041040
9637	053652	026105	051040	041505
9638	053660	042040	052101	020101
9639	053666	044507	042526	020123
9640	053674	042522	027107	041440
9641	053702	047117	027124	040440
9642	053710	052106	051105	041440
9643	053716	046517	040515	042116
9644	053724	000		
9645	053725	065	030060	047440
9646	053732	043106	042523	020124
9647	053740	047503	046515	047101
9648	053746	051504	047440	042516
9649	053754	040440	052106	051105
9650	053762	052040	042510	047440
9651	053770	044124	051105	041440
9652	053776	052501	042523	020104
9653	054004	047101	042440	051122
9654	054012	051117	000	
9655	054015	127	044522	042524
9656	054022	044040	040505	042504
9657	054030	020122	047101	020104
9658	054036	040504	040524	041440
9659	054044	052501	042523	020104
9660	054052	046511	051120	050117

EM24: .ASCII /OFFSET COMMAND CAUSED AN ERROR/<15><12>

.ASCIZ /GOOD DATA GIVES SHOULD BE, REC DATA GIVES REG. CONT. AFTER COMMAND/

EM25: .ASCII /RETURN TO CENTER LINE COMMAND CAUSED AN ERROR/<15><12>

.ASCIZ /GOOD DATA GIVES SHOULD BE, REC DATA GIVES REG. CONT. AFTER COMMAND/

EM26: .ASCIZ /500 OFFSET COMMANDS ONE AFTER THE OTHER CAUSED AN ERROR/

EM27: .ASCII /WRITE HEADER AND DATA CAUSED IMPROPER REGISTER CHANGE/<15><12>

9661	054060	051105	051040	043505
9662	054066	051511	042524	020122
9663	054074	044103	047101	042507
9664	054102	005015		
9665	054104	047507	042117	042040
9666	054112	052101	020101	044507
9667	054120	042526	020123	044127
9668	054126	052101	051440	047510
9669	054134	046125	020104	042502
9670	054142	052040	042510	042522
9671	054150	005015		
9672	054152	042522	042503	053111
9673	054160	042105	042040	052101
9674	054166	020101	044507	042526
9675	054174	020123	044127	052101
9676	054202	053440	051501	052040
9677	054210	042510	042522	040440
9678	054216	052106	051105	041440
9679	054224	046517	040515	042116
9680	054232	000		
9681				
9682	054233	127	044522	042524
9683	054240	044040	040505	042504
9684	054246	020122	047101	020104
9685	054254	040504	040524	041440
9686	054262	040510	043516	042105
9687	054270	053440	044522	042524
9688	054276	043040	047522	020115
9689	054304	052502	043106	051105
9690	054312	000		
9691	054313	122	040505	020104
9692	054320	042510	042101	051105
9693	054326	040440	042116	042040
9694	054334	052101	020101	040503
9695	054342	051525	042105	044440
9696	054350	050115	047522	042520
9697	054356	020122	042522	044507
9698	054364	052123	051105	041440
9699	054372	040510	043516	006505
9700	054400	012		
9701	054401	107	047517	020104
9702	054406	040504	040524	043440
9703	054414	053111	051505	053440
9704	054422	040510	020124	044123
9705	054430	052517	042114	041040
9706	054436	020105	044124	051105
9707	054444	006505	012	
9708	054447	122	041505	044505
9709	054454	042526	020104	040504
9710	054462	040524	043440	053111
9711	054470	051505	053440	040510
9712	054476	020124	040527	020123
9713	054504	044124	051105	020105
9714	054512	043101	042524	020122
9715	054520	047503	046515	047101
9716	054526	000104		

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

.ASCIZ /RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND/

EM30: .ASCIZ /WRITE HEADER AND DATA CHANGED WRITE FROM BUFFER/

EM31: .ASCII /READ HEADFR AND DATA CAUSED IMPROPER REGISTER CHANGE/<15><12>

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

.ASCIZ /RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND/

9717	054530	051127	052111	020105	EM32: .ASCIZ /WRITE HEADER DATA, FOLLOWED BY READ HEADER AND DATA, CAUSED DATA ERROR/
9718	054536	042510	042101	051105	
9719	054544	042040	052101	026101	
9720	054552	043040	046117	047514	
9721	054560	042527	020104	054502	
9722	054566	051040	040505	020104	
9723	054574	042510	042101	051105	
9724	054602	040440	042116	042040	
9725	054610	052101	026101	041440	
9726	054616	052501	042523	020104	
9727	054624	040504	040524	042440	
9728	054632	051122	051117	000	
9729	054637	122	040505	020104	EM3: .ASCII /READ DATA CAUSED IMPROPER REGISTER CHANGE/<15><12>
9730	054644	040504	040524	041440	
9731	054652	052501	042523	020104	
9732	054660	046511	051120	050117	
9733	054666	051105	051040	043505	
9734	054674	051511	042524	020122	
9735	054702	044103	047101	042507	
9736	054710	005015			
9737	054712	047507	042117	042040	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
9738	054720	052101	020101	044507	
9739	054726	042526	020123	044127	
9740	054734	052101	051440	047510	
9741	054742	046125	020104	042502	
9742	054750	052040	042510	042522	
9743	054756	005015			
9744	054760	042522	042503	053111	.ASCIZ /RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND/
9745	054766	042105	042040	052101	
9746	054774	020101	044507	042526	
9747	055002	020123	044127	052101	
9748	055010	053440	051501	052040	
9749	055016	042510	042522	040440	
9750	055024	052106	051105	041440	
9751	055032	046517	040515	042116	
9752	055040	000			
9753	055041	122	040505	020104	EM34: .ASCIZ /READ DATA INCORRECT/
9754	055046	040504	040524	044440	
9755	055054	041516	051117	042522	
9756	055062	052103	000		
9757	055065	127	044522	042524	EM35: .ASCII /WRITE DATA COMMAND CAUSED IMPROPER REGISTER CHANGE/<15><12>
9758	055072	042040	052101	020101	
9759	055100	047503	046515	047101	
9760	055106	020104	040503	051525	
9761	055114	042105	044440	050115	
9762	055122	047522	042520	020122	
9763	055130	042522	044507	052123	
9764	055136	051105	041440	040510	
9765	055144	043516	006505	012	
9766	055151	107	047517	020104	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
9767	055156	040504	040524	043440	
9768	055164	053111	051505	053440	
9769	055172	040510	020124	044123	
9770	055200	052517	042114	041040	
9771	055206	020105	044124	051105	
9772	055214	006505	012		

9773	055217	122	041505	044505		.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER COMMAND/
9774	055224	042526	020104	040504		
9775	055232	040524	043440	053111		
9776	055240	051505	051040	043505		
9777	055246	051511	042524	020122		
9778	055254	047503	052116	047105		
9779	055262	051524	040440	052106		
9780	055270	051105	041440	046517		
9781	055276	040515	042116	000		
9782	055303	127	044522	042524	EM36:	.ASCIZ /WRITE DATA COMMAND CHANGED WRITE FROM BUFFER/
9783	055310	042040	052101	020101		
9784	055316	047503	046515	047101		
9785	055324	020104	044103	047101		
9786	055332	042507	020104	051127		
9787	055340	052111	020105	051106		
9788	055346	046517	041040	043125		
9789	055354	042506	000122			
9790	055360	042523	045505	041440	EM37:	.ASCII /SEEK COMMAND CAUSED IMPROPER REGISTER CHANGE/<15><12>
9791	055366	046517	040515	042116		
9792	055374	041440	052501	042523		
9793	055402	020104	046511	051120		
9794	055410	050117	051105	051040		
9795	055416	043505	051511	042524		
9796	055424	020122	044103	047101		
9797	055432	042507	005015			
9798	055436	047507	042117	042040		.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
9799	055444	052101	020101	044507		
9800	055452	042526	020123	044127		
9801	055460	052101	051440	047510		
9802	055466	046125	020104	042502		
9803	055474	052040	042510	042522		
9804	055502	005015				
9805	055504	042522	042503	053111		.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER SEEK COMMAND/
9806	055512	042105	042040	052101		
9807	055520	020101	044507	042526		
9808	055526	020123	042522	044507		
9809	055534	052123	051105	041440		
9810	055542	047117	042524	052116		
9811	055550	020123	043101	042524		
9812	055556	020122	042523	045505		
9813	055564	041440	046517	040515		
9814	055572	042116	000			
9815						
9816	055575	127	044522	042524	EM40:	.ASCII /WRITE CHECK CAUSED IMPROPER REGISTER CHANGE/<15><12>
9817	055602	041440	042510	045503		
9818	055610	041440	052501	042523		
9819	055616	020104	046511	051120		
9820	055624	050117	051105	051040		
9821	055632	043505	051511	042524		
9822	055640	020122	044103	047101		
9823	055646	042507	005015			
9824	055652	047507	042117	042040		.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
9825	055660	052101	020101	044507		
9826	055666	042526	020123	044127		
9827	055674	052101	051440	047510		
9828	055702	046125	020104	042502		

9829	055710	052040	042510	042522	
9830	055716	005015			
9831	055720	042522	042503	053111	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER COMMAND/
9832	055726	042105	042040	052101	
9833	055734	020101	044507	042526	
9834	055742	020123	042522	044507	
9835	055750	052123	051105	041440	
9836	055756	047117	042524	052116	
9837	055764	020123	043101	042524	
9838	055772	020122	047503	046515	
9839	056000	047101	000104		
9840	056004	047514	045503	047111	EM41: .ASCII /LOCKING OUT WRITE BY WRITE LOCK BUTTON CAUSED IMPROPER REGISTER CHANGE/
9841	056012	020107	052517	020124	
9842	056020	051127	052111	020105	
9843	056026	054502	053440	044522	
9844	056034	042524	046040	041517	
9845	056042	020113	052502	052124	
9846	056050	047117	041440	052501	
9847	056056	042523	020104	046511	
9848	056064	051120	050117	051105	
9849	056072	051040	043505	051511	
9850	056100	042524	020122	044103	
9851	056106	047101	042507	005015	
9852	056114	047507	042117	042040	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
9853	056122	052101	020101	044507	
9854	056130	042526	020123	044127	
9855	056136	052101	051440	047510	
9856	056144	046125	020104	042502	
9857	056152	052040	042510	042522	
9858	056160	005015			
9859	056162	042522	042503	053111	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER WRITES WERE LOCKED OUT/
9860	056170	042105	042040	052101	
9861	056176	020101	044507	042526	
9862	056204	020123	042522	044507	
9863	056212	052123	051105	041440	
9864	056220	047117	042524	052116	
9865	056226	020123	043101	042524	
9866	056234	020122	051127	052111	
9867	056242	051505	053440	051105	
9868	056250	020105	047514	045503	
9869	056256	042105	047440	052125	
9870	056264	000			
9871	056265	101	052124	046505	EM42: .ASCII /ATTEMPTING TO WRITE WITH WRITES LOCKED OUT CAUSED IMPROPER REGISTER CHA
9872	056272	052120	047111	020107	
9873	056300	047524	053440	044522	
9874	056306	042524	053440	052111	
9875	056314	020110	051127	052111	
9876	056322	051505	046040	041517	
9877	056330	042513	020104	052517	
9878	056336	020124	040503	051525	
9879	056344	042105	044440	050115	
9880	056352	047522	042520	020122	
9881	056360	042522	044507	052123	
9882	056366	051105	041440	040510	
9883	056374	043516	006505	012	
9884	056401	107	047517	020104	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>

9885	056406	040504	040524	043440	
9886	056414	053111	051505	053440	
9887	056422	040510	020124	044123	
9888	056430	052517	042114	041040	
9889	056436	020105	044124	051105	
9890	056444	006505	012		
9891	056447	122	041505	044505	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ATTEMPTED WRITE/
9892	056454	042526	020104	040504	
9893	056462	040524	043440	053111	
9894	056470	051505	051040	043505	
9895	056476	051511	042524	020122	
9896	056504	047503	052116	047105	
9897	056512	051524	040440	052106	
9898	056520	051105	040440	052124	
9899	056526	046505	052120	042105	
9900	056534	053440	044522	042524	
9901	056542	000			
9902	056543	127	044522	044524	EM43: .ASCII /WRITING WITH WRITES LOCKED OUT CHANGED DISK DATA/<'5><12>
9903	056550	043516	053440	052111	
9904	056556	020110	051127	052111	
9905	056564	051505	046040	041517	
9906	056572	042513	020104	052517	
9907	056600	020124	044103	047101	
9908	056606	042507	020104	044504	
9909	056614	045523	042040	052101	
9910	056622	006501	012		
9911	056625	107	047517	020104	.ASCII /GOOD DATA GIVES WHAT WAS ON DISK BEFORE WRITE (WITH WRITE LOCKED OUT)/<
9912	056632	040504	040524	043440	
9913	056640	053111	051505	053440	
9914	056646	040510	020124	040527	
9915	056654	020123	047117	042040	
9916	056662	051511	020113	042502	
9917	056670	047506	042522	053440	
9918	056676	044522	042524	024040	
9919	056704	044527	044124	053440	
9920	056712	044522	042524	046040	
9921	056720	041517	042513	020104	
9922	056726	052517	024524	005015	
9923	056734	040527	020123	052101	.ASCII /WAS ATTEMPTED/<15><12>
9924	056742	042524	050115	042524	
9925	056750	006504	012		
9926	056753	122	041505	044505	.ASCII /RECEIVED DATA GIVES WHAT WAS READ BACK AFTER WRITE/<15><12>
9927	056760	042526	020104	040504	
9928	056766	040524	043440	053111	
9929	056774	051505	053440	040510	
9930	057002	020124	040527	020123	
9931	057010	042522	042101	041040	
9932	057016	041501	020113	043101	
9933	057024	042524	020122	051127	
9934	057032	052111	006505	012	
9935	057037	050	044527	044124	.ASCIZ /(WITH WRITE LOCKED OUT) WAS ATTEMPTED/
9936	057044	053440	044522	042524	
9937	057052	046040	041517	042513	
9938	057060	020104	052517	024524	
9939	057066	053440	051501	040440	
9940	057074	052124	046505	052120	

9941	057102	042105	000		
9942	057105	105	040516	046102	EM44: .ASCII /ENABLING WRITES BY WRITE LOCK BUTTON CAUSED IMPROPER REGISTER CHANGE/<1
9943	057112	047111	020107	051127	
9944	057120	052111	051505	041040	
9945	057126	020131	051127	052111	
9946	057134	020105	047514	045503	
9947	057142	041040	052125	047524	
9948	057150	020116	040503	051525	
9949	057156	042105	044440	050115	
9950	057164	047522	042520	020122	
9951	057172	042522	044507	052123	
9952	057200	051105	041440	040510	
9953	057206	043516	006505	012	
9954	057213	107	047517	020104	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
9955	057220	040504	040524	043440	
9956	057226	053111	051505	053440	
9957	057234	040510	020124	044123	
9958	057242	052517	042114	041040	
9959	057250	020105	044124	051105	
9960	057256	006505	012		
9961	057261	122	041505	044505	.ASCII /RECEIVED DATA GIVES REGISTER CONTENTS AFTER WRITE LOCK BUTTON/<15><12>
9962	057266	042526	020104	040504	
9963	057274	040524	043440	053111	
9964	057302	051505	051040	043505	
9965	057310	051511	042524	020122	
9966	057316	047503	052116	047105	
9967	057324	051524	040440	052106	
9968	057332	051105	053440	044522	
9969	057340	042524	046040	041517	
9970	057346	020113	052502	052124	
9971	057354	047117	005015		
9972	057360	047105	041101	042514	.ASCIIZ /ENABLED WRITES/
9973	057366	020104	051127	052111	
9974	057374	051505	000		
9975	057377	124	040522	051516	EM45: .ASCII /TRANSFERRING ON LAST BLOCK - CYLINDER 410, SECTOR 21, TRACK 18/<15><12>
9976	057404	042506	051122	047111	
9977	057412	020107	047117	046040	
9978	057420	051501	020124	046102	
9979	057426	041517	020113	020055	
9980	057434	054503	044514	042116	
9981	057442	051105	032040	030061	
9982	057450	020054	042523	052103	
9983	057456	051117	031040	026061	
9984	057464	052040	040522	045503	
9985	057472	030440	006470	012	
9986	057477	103	052501	042523	.ASCII /CAUSED IMPROPER REGISTER CHANGE/<15><12>
9987	057504	020104	046511	051120	
9988	057512	050117	051105	051040	
9989	057520	043505	051511	042524	
9990	057526	020122	044103	047101	
9991	057534	042507	005015		
9992	057540	047507	042117	042040	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
9993	057546	052101	020101	044507	
9994	057554	042526	020123	044127	
9995	057562	052101	051440	047510	
9996	057570	046125	020104	042502	

9997	057576	052040	042510	042522	
9998	057604	005015			
9999	057606	042522	042503	053111	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER TRANSFER/
10000	057614	042105	042040	052101	
10001	057622	020101	044507	042526	
10002	057630	020123	042522	044507	
10003	057636	052123	051105	041440	
10004	057644	047117	042524	052116	
10005	057652	020123	043101	042524	
10006	057660	020122	051124	047101	
10007	057666	043123	051105	000	
10008	057673	104	052101	020101	EM46: .ASCII /DATA READ FROM LAST BLOCK - CYLINDER 410, SECTOR 21, TRACK 18/<15><12>
10009	057700	042522	042101	043040	
10010	057706	047522	020115	040514	
10011	057714	052123	041040	047514	
10012	057722	045503	026440	041440	
10013	057730	046131	047111	042504	
10014	057736	020122	030464	026060	
10015	057744	051440	041505	047524	
10016	057752	020122	030462	020054	
10017	057760	051124	041501	020113	
10018	057766	034061	005015		
10019	057772	051511	044440	020116	.ASCIZ /IS IN ERROR/
10020	060000	051105	047522	000122	
10021	060006	051124	047101	043123	EM47: .ASCII /TRANSFERRING DATA FROM NONEXISTANT SECTOR CAUSED IMPROPER /<15><12>
10022	060014	051105	044522	043516	
10023	060022	042040	052101	020101	
10024	060030	051106	046517	047040	
10025	060036	047117	054105	051511	
10026	060044	040524	052116	051440	
10027	060052	041505	047524	020122	
10028	060060	040503	051525	042105	
10029	060066	044440	050115	047522	
10030	060074	042520	020122	005015	
10031	060102	042522	044507	052123	.ASCII /REGISTER CHANGE, GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
10032	060110	051105	041440	040510	
10033	060116	043516	026105	043440	
10034	060124	047517	020104	040504	
10035	060132	040524	043440	053111	
10036	060140	051505	053440	040510	
10037	060146	020124	044123	052517	
10038	060154	042114	041040	020105	
10039	060162	044124	051105	006505	
10040	060170	012			
10041	060171	122	041505	044505	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ATTEMPTED TRANSFER/
10042	060176	042526	020104	040504	
10043	060204	040524	043440	053111	
10044	060212	051505	051040	043505	
10045	060220	051511	042524	020122	
10046	060226	047503	052116	047105	
10047	060234	051524	040440	052106	
10048	060242	051105	040440	052124	
10049	060250	046505	052120	042105	
10050	060256	052040	040522	051516	
10051	060264	042506	000122		
10052					

10053	060270	051124	047101	043123	EM50: .ASCII /TRANSFERRING FROM NONEXISTANT SECTOR CAUSED DATA ERROR/<15><12>
10054	060276	051105	044522	043516	
10055	060304	043040	047522	020115	
10056	060312	047516	042516	044530	
10057	060320	052123	047101	020124	
10058	060326	042523	052103	051117	
10059	060334	041440	052501	042523	
10060	060342	020104	040504	040524	
10061	060350	042440	051122	051117	
10062	060356	005015			
10063	060360	047507	042117	042040	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
10064	060366	052101	020101	044507	
10065	060374	042526	020123	044127	
10066	060402	052101	051440	047510	
10067	060410	046125	020104	042502	
10068	060416	052040	042510	042522	
10069	060424	005015			
10070	060426	040502	020104	040504	.ASCIZ /BAD DATA GIVES WHAT WAS IN BUFFER AFTER TRANSFER/
10071	060434	040524	043440	053111	
10072	060442	051505	053440	040510	
10073	060450	020124	040527	020123	
10074	060456	047111	041040	043125	
10075	060464	042506	020122	043101	
10076	060472	042524	020122	051124	
10077	060500	047101	043123	051105	
10078	060506	000			
10079	060507	107	053111	047111	EM51: .ASCII /GIVING ILLEGAL FUNCTION CAUSED IMPROPER REGISTER CHANGE/<15><12>
10080	060514	020107	046111	042514	
10081	060522	040507	020114	052506	
10082	060530	041516	044524	047117	
10083	060536	041440	052501	042523	
10084	060544	020104	046511	051120	
10085	060552	050117	051105	051040	
10086	060560	043505	051511	042524	
10087	060566	020122	044103	047101	
10088	060574	042507	005015		
10089	060600	047507	042117	042040	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
10090	060606	052101	020101	044507	
10091	060614	042526	020123	044127	
10092	060622	052101	051440	047510	
10093	060630	046125	020104	042502	
10094	060636	052040	042510	042522	
10095	060644	005015			
10096	060646	042522	042503	053111	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ILLEGAL FUNCTION IS GIVEN/
10097	060654	042105	042040	052101	
10098	060662	020101	044507	042526	
10099	060670	020123	042522	044507	
10100	060676	052123	051105	041440	
10101	060704	047117	042524	052116	
10102	060712	020123	043101	042524	
10103	060720	020122	046111	042514	
10104	060726	040507	020114	052506	
10105	060734	041516	044524	047117	
10106	060742	044440	020123	044507	
10107	060750	042526	000116		
10108	060754	051127	052111	020105	EM52: .ASCII /WRITE DATA COMMAND ON NONEXISTANT SECTOR CAUSED IMPROPER REG. CHANGE/<1

10109	060762	040504	040524	041440
10110	060770	046517	040515	042116
10111	060776	047440	020116	047516
10112	061004	042516	044530	052123
10113	061012	047101	020124	042523
10114	061020	052103	051117	041440
10115	061026	052501	042523	020104
10116	061034	046511	051120	050117
10117	061042	051105	051040	043505
10118	061050	020056	044103	047101
10119	061056	042507	005015	
10120	061062	047507	042117	042040
10121	061070	052101	020101	044507
10122	061076	042526	020123	044127
10123	061104	052101	051440	047510
10124	061112	046125	020104	042502
10125	061120	052040	042510	042522
10126	061126	005015		
10127	061130	042522	042503	053111
10128	061136	042105	042040	052101
10129	061144	020101	044507	042526
10130	061152	020123	042522	044507
10131	061160	052123	051105	041440
10132	061166	047117	042524	052116
10133	061174	020123	043101	042524
10134	061202	020122	052101	042524
10135	061210	050115	042524	020104
10136	061216	051127	052111	020105
10137	061224	040504	040524	000
10138	061231	122	040505	020104
10139	061236	042510	042101	051105
10140	061244	040440	042116	042040
10141	061252	052101	020101	043101
10142	061260	042524	020122	020101
10143	061266	042523	051101	044103
10144	061274	041440	052501	042523
10145	061302	020104	040504	040524
10146	061310	042440	051122	051117
10147	061316	000		
10148	061317	101	052124	046505
10149	061324	052120	047111	020107
10150	061332	047503	046515	047101
10151	061340	020104	044527	044124
10152	061346	044440	053116	046101
10153	061354	042111	040440	042104
10154	061362	042522	051523	041440
10155	061370	052501	042523	020104
10156	061376	046511	051120	050117
10157	061404	051105	051040	043505
10158	061412	051511	042524	020122
10159	061420	044103	047101	042507
10160	061426	005015		
10161	061430	047507	042117	042040
10162	061436	052101	020101	044507
10163	061444	042526	020123	044127
10164	061452	052101	051440	047510

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

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

EM53: .ASCIZ /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>

10165	061460	046125	020104	042502	
10166	061466	052040	042510	042522	
10167	061474	005015			
10168	061476	042522	042503	053111	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER OPERATION/
10169	061504	042105	042040	052101	
10170	061512	020101	044507	042526	
10171	061520	020123	042522	044507	
10172	061526	052123	051105	041440	
10173	061534	047117	042524	052116	
10174	061542	020123	043101	042524	
10175	061550	020122	050117	051105	
10176	061556	052101	047511	000116	
10177	061564	051127	052111	047111	EM55: .ASCII /WRITING OR READING WITH EXPECTED ADDRESS OVERFLOW ERROR/<15><12>
10178	061572	020107	051117	051040	
10179	061600	040505	044504	043516	
10180	061606	053440	052111	020110	
10181	061614	054105	042520	052103	
10182	061622	042105	040440	042104	
10183	061630	042522	051523	047440	
10184	061636	042526	043122	047514	
10185	061644	020127	051105	047522	
10186	061652	006522	012		
10187	061655	103	052501	042523	.ASCII /CAUSED IMPROPER REGISTER CHANGE/<15><12>
10188	061662	020104	046511	051120	
10189	061670	050117	051105	051040	
10190	061676	043505	051511	042524	
10191	061704	020122	044103	047101	
10192	061712	042507	005015		
10193	061716	047507	042117	042040	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
10194	061724	052101	020101	044507	
10195	061732	042526	020123	044127	
10196	061740	052101	051440	047510	
10197	061746	046125	020104	042502	
10198	061754	052040	042510	042522	
10199	061762	005015			
10200	061764	042522	042503	053111	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER OPERATION/
10201	061772	042105	042040	052101	
10202	062000	020101	044507	042526	
10203	062006	020123	042522	044507	
10204	062014	052123	051105	041440	
10205	062022	047117	042524	052116	
10206	062030	020123	043101	042524	
10207	062036	020122	050117	051105	
10208	062044	052101	047511	000116	
10209	062052	040504	040524	051040	EM56: .ASCII /DATA READ WITH AN EXPECTED ADDRESS OVERFLOW ERROR IS INCORRECT/<15><12>
10210	062060	040505	020104	044527	
10211	062066	044124	040440	020116	
10212	062074	054105	042520	052103	
10213	062102	042105	040440	042104	
10214	062110	042522	051523	047440	
10215	062116	042526	043122	047514	
10216	062124	020127	051105	047522	
10217	062132	020122	051511	044440	
10218	062140	041516	051117	042522	
10219	062146	052103	005015		
10220	062152	047527	042122	047040	.ASCII /WORD NO. 1 TO 260 SHOULD BE READ, WORD NO 261 TO 266 SHOULD/<15><12>

10221	062160	027117	030440	052040	
10222	062166	020117	033062	020060	
10223	062174	044123	052517	042114	
10224	062202	041040	020105	042522	
10225	062210	042101	020054	047527	
10226	062216	042122	047040	020117	
10227	062224	033062	020061	047524	
10228	062232	031040	033066	051440	
10229	062240	047510	046125	006504	
10230	062246	012			
10231	062247	102	020105	044103	.ASCIZ /BE CHANGED/
10232	062254	047101	042507	000104	
10233	062262	052101	042524	050115	EM57: .ASCII /ATTEMPTING DATA COMMAND WITH WRONG FORMAT BIT CAUSED/<15><12>
10234	062270	044524	043516	042040	
10235	062276	052101	020101	047503	
10236	062304	046515	047101	020104	
10237	062312	044527	044124	053440	
10238	062320	047522	043516	043040	
10239	062326	051117	040515	020124	
10240	062334	044502	020124	040503	
10241	062342	051525	042105	005015	
10242	062350	046511	051120	050117	.ASCII /IMPROPER REGISTER CHANGE/<15><12>
10243	062356	051105	051040	043505	
10244	062364	051511	042524	020122	
10245	062372	044103	047101	042507	
10246	062400	005015			
10247	062402	047507	042117	042040	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
10248	062410	052101	020101	044507	
10249	062416	042526	020123	044127	
10250	062424	052101	051440	047510	
10251	062432	046125	020104	042502	
10252	062440	052040	042510	042522	
10253	062446	005015			
10254	062450	042522	042503	053111	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ATTEMPTED DATA TRANSFER/
10255	062456	042105	042040	052101	
10256	062464	020101	044507	042526	
10257	062472	020123	042522	044507	
10258	062500	052123	051105	041440	
10259	062506	047117	042524	052116	
10260	062514	020123	043101	042524	
10261	062522	020122	052101	042524	
10262	062530	050115	042524	020104	
10263	062536	040504	040524	052040	
10264	062544	040522	051516	042506	
10265	062552	000122			
10266					
10267	062554	052101	042524	050115	EM60: .ASCII /ATTEMPTING TO MODIFY REGISTER DURING AN OPERATION CAUSED IMPROPER/<15><
10268	062562	044524	043516	052040	
10269	062570	020117	047515	044504	
10270	062576	054506	051040	043505	
10271	062604	051511	042524	020122	
10272	062612	052504	044522	043516	
10273	062620	040440	020116	050117	
10274	062626	051105	052101	047511	
10275	062634	020116	040503	051525	
10276	062642	042105	044440	050115	

10277	062650	047522	042520	006522	
10278	062656	012			
10279	062657	122	043505	051511	.ASCII /REGISTER CHANGE. GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
10280	062664	042524	020122	044103	
10281	062672	047101	042507	020056	
10282	062700	047507	042117	042040	
10283	062706	052101	020101	044507	
10284	062714	042526	020123	044127	
10285	062722	052101	051440	047510	
10286	062730	046125	020104	042502	
10287	062736	052040	042510	042522	
10288	062744	005015			
10289	062746	042522	042503	053111	.ASCII /RECEIVED DATA GIVES REGISTER CONTENTS AFTER OPERATION WAS ATTEMPTED/<15
10290	062754	042105	042040	052101	
10291	062762	020101	044507	042526	
10292	062770	020123	042522	044507	
10293	062776	052123	051105	041440	
10294	063004	047117	042524	052116	
10295	063012	020123	043101	042524	
10296	063020	020122	050117	051105	
10297	063026	052101	047511	020116	
10298	063034	040527	020123	052101	
10299	063042	042524	050115	042524	
10300	063050	006504	012		
10301	063053	115	042117	043111	.ASCIIZ /MODIFYING REG GIVES ADDRESS OF REGISTER BEING MODIFIED WHICH CAUSED ERR
10302	063060	044531	043516	051040	
10303	063066	043505	043440	053111	
10304	063074	051505	040440	042104	
10305	063102	042522	051523	047440	
10306	063110	020106	042522	044507	
10307	063116	052123	051105	041040	
10308	063124	044505	043516	046440	
10309	063132	042117	043111	042511	
10310	063140	020104	044127	041511	
10311	063146	020110	040503	051525	
10312	063154	042105	042440	051122	
10313	063162	051117	000		
10314	063165	122	041510	030523	EM61: .ASCIIZ /RHCS1 HAS SOME INCORRECT STATUS BITS - 1, OR - 0/
10315	063172	044040	051501	051440	
10316	063200	046517	020105	047111	
10317	063206	047503	051122	041505	
10318	063214	020124	052123	052101	
10319	063222	051525	041040	052111	
10320	063230	020123	020075	026061	
10321	063236	047440	020122	020075	
10322	063244	000060			
10323	063246	044122	051504	020061	EM62: .ASCIIZ /RHDS1 HAS SOME INCORRECT STATUS BITS - 1, OR - 0/
10324	063254	040510	020123	047523	
10325	063262	042515	044440	041516	
10326	063270	051117	042522	052103	
10327	063276	051440	040524	052524	
10328	063304	020123	044502	051524	
10329	063312	036440	030440	020054	
10330	063320	051117	036440	030040	
10331	063326	000			
10332	063327	122	042110	030523	EM63: .ASCIIZ /RHDS1 CONTENTS DURING COMMAND WERE IN ERROR/

10333	063334	041440	047117	042524	
10334	063342	052116	020123	052504	
10335	063350	044522	043516	041440	
10336	063356	046517	040515	042116	
10337	063364	053440	051105	020105	
10338	063372	047111	042440	051122	
10339	063400	051117	000		
10340	063403	122	041505	046101	EM64: .ASCII /RECALIBRATE COMMAND CAUSED IMPROPER REGISTER CHANGE/<15><12>
10341	063410	041111	040522	042524	
10342	063416	041440	046517	040515	
10343	063424	042116	041440	052501	
10344	063432	042523	020104	046511	
10345	063440	051120	050117	051105	
10346	063446	051040	043505	051511	
10347	063454	042524	020122	044103	
10348	063462	047101	042507	005015	
10349	063470	047507	042117	042040	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
10350	063476	052101	020101	044507	
10351	063504	042526	020123	044127	
10352	063512	052101	051440	047510	
10353	063520	046125	020104	042502	
10354	063526	052040	042510	042522	
10355	063534	005015			
10356	063536	042522	042503	053111	.ASCIIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER COMMAND/
10357	063544	042105	042040	052101	
10358	063552	020101	044507	042526	
10359	063560	020123	042522	044507	
10360	063566	052123	051105	041440	
10361	063574	047117	042524	052116	
10362	063602	020123	043101	042524	
10363	063610	020122	047503	046515	
10364	063616	047101	000104		
10365	063622	047111	042524	051122	EM65: .ASCIIZ /INTERRUPT FAILING/
10366	063630	050125	020124	040506	
10367	063636	046111	047111	000107	
10368	063644	042510	042101	051105	EM66: .ASCII /HEADER AND DATA COMMAND FOR HEAD SELECTION TEST CAUSED/<15><12>
10369	063652	040440	042116	042040	
10370	063660	052101	020101	047503	
10371	063666	046515	047101	020104	
10372	063674	047506	020122	042510	
10373	063702	042101	051440	046105	
10374	063710	041505	044524	047117	
10375	063716	052040	051505	020124	
10376	063724	040503	051525	042105	
10377	063732	005015			
10378	063734	051105	047522	020122	.ASCII /ERROR - RHDST GIVES TRACK BEING WRITTEN OR READ ON CYL 0, SCTR 0/<15><1
10379	063742	020055	044122	051504	
10380	063750	020124	044507	042526	
10381	063756	020123	051124	041501	
10382	063764	020113	042502	047111	
10383	063772	020107	051127	052111	
10384	064000	042524	020116	051117	
10385	064006	051040	040505	020104	
10386	064014	047117	041440	046131	
10387	064022	030040	020054	041523	
10388	064030	051124	030040	005015	

10389	064036	042522	042101	044040	EM67: .ASCII /READ HEADER AND DATA ERROR IN HEAD SELECTION TEST./<12><15>
10390	064044	040505	042504	020122	
10391	064052	047101	020104	040504	
10392	064060	040524	042440	051122	
10393	064066	051117	044440	020116	
10394	064074	042510	042101	051440	
10395	064102	046105	041505	044524	
10396	064110	047117	052040	051505	
10397	064116	027124	006412		
10398	064122	044506	051522	020124	.ASCII /FIRST FOUR WORD NUMBERS ARE THE HEADER./<12><15>
10399	064130	047506	051125	053440	
10400	064136	051117	020104	052516	
10401	064144	041115	051105	020123	
10402	064152	051101	020105	044124	
10403	064160	020105	042510	042101	
10404	064166	051105	005056	015	
10405	064173	127	051117	020104	.ASCII /WORD NUMBERS 5 TO 260 ARE DATA WORDS./<12><15>
10406	064200	052516	041115	051105	
10407	064206	020123	020065	047524	
10408	064214	031040	030066	040440	
10409	064222	042522	042040	052101	
10410	064230	020101	047527	042122	
10411	064236	026123	006412		
10412	064242	047101	020104	047111	.ASCII /AND IN DATA WORDS BITS 4,5,6,7,8 GIVE TRACK NUMBER./
10413	064250	042040	052101	020101	
10414	064256	047527	042122	020123	
10415	064264	044502	051524	032040	
10416	064272	032454	033054	033454	
10417	064300	034054	043440	053111	
10418	064306	020105	051124	041501	
10419	064314	020113	052516	041115	
10420	064322	051105	000056		
10421					
10422	064326	042522	042101	044040	EM70: .ASCII /READ HEADER AND DATA ERROR IN/<15><12>
10423	064334	040505	042504	020122	
10424	064342	047101	020104	040504	
10425	064350	040524	042440	051122	
10426	064356	051117	044440	006516	
10427	064364	012			
10428	064365	104	043111	042506	.ASCII /DIFFERENCE LINE TEST/<15><12>
10429	064372	042522	041516	020105	
10430	064400	044514	042516	052040	
10431	064406	051505	006524	012	
10432	064413	127	051117	020104	.ASCII /WORD NOS 1-4 GIVE HEADER/<15><12>
10433	064420	047516	020123	026461	
10434	064426	020064	044507	042526	
10435	064434	044040	040505	042504	
10436	064442	006522	012		
10437	064445	127	051117	020104	.ASCII /WORD NOS 5-260 GIVE DATA WHICH IS THE CYLINDER ADDRESS/
10438	064452	047516	020123	026465	
10439	064460	033062	020060	044507	
10440	064466	042526	042040	052101	
10441	064474	020101	044127	041511	
10442	064502	020110	051511	052040	
10443	064510	042510	041440	046131	
10444	064516	047111	042504	020122	

10613	066372	042522	044507	052123	
10614	066400	051105	042040	052101	
10615	066406	000101			
10616	066410	047125	052111	042040	EM101: .ASCIZ /UNIT DID NOT GO OFFLINE WHEN ADDRESS PLUG REMOVED/
10617	066416	042111	047040	052117	
10618	066424	043440	020117	043117	
10619	066432	046106	047111	020105	
10620	066440	044127	047105	040440	
10621	066446	042104	042522	051523	
10622	066454	050040	052514	020107	
10623	066462	042522	047515	042526	
10624	066470	000104			
10625	066472	047125	052111	047040	EM102: .ASCIZ /UNIT NOT AVAILABLE AFTER ADDRESS PLUG REPLACED/
10626	066500	052117	040440	040526	
10627	066506	046111	041101	042514	
10628	066514	040440	052106	051105	
10629	066522	040440	042104	042522	
10630	066530	051523	050040	052514	
10631	066536	020107	042522	046120	
10632	066544	041501	042105	000	
10633	066551	122	043505	051511	EM103: .ASCIZ /REGISTER CONTENTS INCORRECT BEFORE A DIAG MODE SEEK/
10634	066556	042524	020122	047503	
10635	066564	052116	047105	051524	
10636	066572	044440	041516	051117	
10637	066600	042522	052103	041040	
10638	066606	043105	051117	020105	
10639	066614	020101	044504	043501	
10640	066622	046440	042117	020105	
10641	066630	042523	045505	000	
10642	066635	122	043505	051511	EM104: .ASCIZ /REGISTER CONTENTS INCORRECT AFTER A DIAG MODE SEEK/
10643	066642	042524	020122	047503	
10644	066650	052116	047105	051524	
10645	066656	044440	041516	051117	
10646	066664	042522	052103	040440	
10647	066672	052106	051105	040440	
10648	066700	042040	040511	020107	
10649	066706	047515	042504	051440	
10650	066714	042505	000113		

10819	070522	042514	046107	005015							
10820	070530	020040	020040	020040		.ASCIZ /	NO	ADDRESS DATA	DATA	FUNCTN/	
10821	070536	020040	047516	020040							
10822	070544	020040	020040	042101							
10823	070552	051104	051505	020123							
10824	070560	040504	040524	020040							
10825	070566	020040	040504	040524							
10826	070574	020040	020040	052506							
10827	070602	041516	047124	000							
10828											
10829	070607	120	020103	020040	DH60:	.ASCII /PC	TEST	REG	GOOD	RECVD	MODFING/<15><12>
10830	070614	020040	052040	051505							
10831	070622	020124	020040	051040							
10832	070630	043505	020040	020040							
10833	070636	043440	047517	020104							
10834	070644	020040	051040	041505							
10835	070652	042126	020040	046440							
10836	070660	042117	044506	043516							
10837	070666	005015									
10838	070670	020040	020040	020040		.ASCIZ /	NO	ADDRESS DATA	DATA	REG/	
10839	070676	020040	047516	020040							
10840	070704	020040	020040	042101							
10841	070712	051104	051505	020123							
10842	070720	040504	040524	020040							
10843	070726	020040	040504	040524							
10844	070734	020040	020040	042522							
10845	070742	000107									
10846	070744	041520	020040	020040	DH61:	.ASCII /PC	TEST	PC OF	RHCS1/<15><12>		
10847	070752	020040	042524	052123							
10848	070760	020040	020040	041520							
10849	070766	047440	020106	020040							
10850	070774	044122	051503	006461							
10851	071002	012									
10852	071003	040	020040	020040		.ASCIZ /	NO	JSR	WAS/		
10853	071010	020040	047040	020117							
10854	071016	020040	020040	045040							
10855	071024	051123	020040	020040							
10856	071032	053440	051501	000							
10857	071037	120	020103	020040	DH62:	.ASCII /PC	TEST	PC OF	RHDS1/<15><12>		
10858	071044	020040	052040	051505							
10859	071052	020124	020040	050040							
10860	071060	020103	043117	020040							
10861	071066	051040	042110	030523							
10862	071074	005015									
10863	071076	020040	020040	020040		.ASCIZ /	NO	JSR	WAS/		
10864	071104	020040	047516	020040							
10865	071112	020040	020040	051512							
10866	071120	020122	020040	020040							
10867	071126	040527	000123								
10868	071132	041520	020040	020040	DH65:	.ASCII /PC	TEST	CONT	CONT	CONT	/<15><12>
10869	071140	020040	042524	052123							
10870	071146	020040	020040	047503							
10871	071154	052116	020040	020040							
10872	071162	047503	052116	020040							
10873	071170	020040	047503	052116							
10874	071176	020040	020040	005015							

10875	071204	020040	020040	020040		.ASCIZ /	NO	RHCS1	RHAS	RHDS1	/
10876	071212	020040	047516	020040							
10877	071220	020040	020040	044122							
10878	071226	051503	020061	020040							
10879	071234	044122	051501	020040							
10880	071242	020040	044122	051504							
10881	071250	020061	020040	000							
10882	071255	120	020103	020040	DH66:	.ASCIZ /PC	TEST	RHDST	RHER1	RHER2	RHER3 RHCS1 RHCS2/
10883	071262	020040	052040	051505							
10884	071270	020124	020040	051040							
10885	071276	042110	052123	020040							
10886	071304	051040	042510	030522							
10887	071312	020040	051040	042510							
10888	071320	031122	020040	051040							
10889	071326	042510	031522	020040							
10890	071334	051040	041510	030523							
10891	071342	020040	051040	041510							
10892	071350	031123	000								
10893											
10894	071353	120	020103	020040	DH77:	.ASCII /PC	TEST	PC OF	REG.	GOOD	BAD/<15><12>
10895	071360	020040	052040	051505							
10896	071366	020124	020040	050040							
10897	071374	020103	043117	020040							
10898	071402	051040	043505	020056							
10899	071410	020040	043440	047517							
10900	071416	020104	020040	041040							
10901	071424	042101	005015								
10902	071430	020040	020040	020040		.ASCIZ /	NO	JSR	ADDR	DATA	DATA/
10903	071436	020040	047516	020040							
10904	071444	020040	020040	051512							
10905	071452	020122	020040	020040							
10906	071460	042101	051104	020040							
10907	071466	020040	040504	040524							
10908	071474	020040	020040	040504							
10909	071502	040524	000								
10910											
10911		071506				.EVEN					
10912											
10913	071506	001116	004604	041706	DT1:	.WORD	\$ERRPC	TSTNM	WAITPC	WAITBT	WAITRE \$BDDAT,CS1,0
10914	071514	041712	041710	001126							
10915	071522	002362	000000								
10916	071526	001116	004604	041706	DT4:	.WORD	\$ERRPC	TSTNM	WAITPC	WAITBT	WAITRE \$BDDAT,WAITTM,0
10917	071534	041712	041710	001126							
10918	071542	041714	000000								
10919	071546	001116	004604	004600	DT5:	.WORD	\$ERRPC	TSTNM	REGADR	\$GDDAT \$BDDAT,0	
10920	071554	001124	001126	000000							
10921	071562	001116	004604	004600	DT6:	.WORD	\$ERRPC	TSTNM	REGADR	\$BDDAT,0	
10922	071570	001126	000000								
10923	071574	001116	004604	001200	DT7:	.WORD	\$ERRPC	TSTNM	\$TMP1,0		
10924	071602	000000									
10925											
10926	071604	001116	004604	001122	DT10:	.WORD	\$ERRPC	TSTNM	\$BDADR,CS1,CS2,DS1,ER1,0		
10927	071612	002362	002360	002404							
10928	071620	002364	000000								
10929	071624	001116	004604	002362	DT26:	.WORD	\$ERRPC	TSTNM	CS1,CS2,DS1,ER1,ER2,ER3,0		
10930	071632	002360	002404	002364							

10931	071640	002370	002376	000000				
10932	071646	001116	004604	004602	DT30:	.WORD	\$ERRPC,TSTNM,ERWORD,\$GDDAT,\$BDDAT,0	
10933	071654	001124	001126	000000				
10934								
10935	071662	001116	004604	004600	DT51:	.WORD	\$ERRPC,TSTNM,REGADR,\$GDDAT,\$BDDAT,ILLEGL,0	
10936	071670	001124	001126	002464				
10937	071676	000000						
10938								
10939	071700	001116	004604	004600	DT60:	.WORD	\$ERRPC,TSTNM,REGADR,\$GDDAT,\$BDDAT,\$BDADR,0	
10940	071706	001124	001126	001122				
10941	071714	000000						
10942	071716	001116	004604	041512	DT61:	.WORD	\$ERRPC,TSTNM,PCJSR,\$BDADR,0	
10943	071724	001122	000000					
10944	071730	001116	004604	041512	DT62:	.WORD	\$ERRPC,TSTNM,PCJSR,\$BDADR,0	
10945	071736	001122	000000					
10946	071742	001116	004604	002362	DT65:	.WORD	\$ERRPC,TSTNM,CS1,AS,DS1,0	
10947	071750	002400	002404	000000				
10948	071756	001116	004604	002364	DT66:	.WORD	\$ERRPC,TSTNM,ER1,ER2,ER3,CS1,CS2,0	
10949	071764	002370	002376	002362				
10950	071772	002360	000000					
10951								
10952	071776	001116	004604	041512	DT77:	.WORD	\$ERRPC,TSTNM,PCJSR,REGADR,\$GDDAT,\$BDDAT,0	
10953	072004	004600	001124	001126				
10954	072012	000000						
10955								
10956	072014	000	000	000	DF1:	.BYTE	0,0,0,0,0,0,0	
10957	072017	000	000	000				
10958	072022	000						
10959	072023	000	000	000	DF4:	.BYTE	0,0,0,0,0,1,0	
10960	072026	000	000	001				
10961	072031	000						
10962	072032	000	000	000	DF5:	.BYTE	0,0,0,0,0	
10963	072035	000	000					
10964	072037	000	000	000	DF6:	.BYTE	0,0,0,0	
10965	072042	000						
10966	072043	000	000	000	DF7:	.BYTE	0,0,0	
10967								
10968	072046	000	000	000	DF10:	.BYTE	0,0,0,0,0,0,0,0	
10969	072051	000	000	000				
10970	072054	000						
10971								
10972	072055	000	000	000	DF26:	.BYTE	0,0,0,0,0,0,0,0,0	
10973	072060	000	000	000				
10974	072063	000	000					
10975								
10976	072065	000	000	000	DF30:	.BYTE	0,0,0,0,0	
10977	072070	000	000					
10978								
10979	072072	000	000	000	DF51:	.BYTE	0,0,0,0,0,0,0	
10980	072075	000	000	000				
10981								
10982	072100	000	000	000	DF60:	.BYTE	0,0,0,0,0,0,0	
10983	072103	000	000	000				
10984	072106	000	000	000	DF61:	.BYTE	0,0,0,0	
10985	072111	000						
10986	072112	000	000	000	DF62:	.BYTE	0,0,0,0	

10987	072115	000						
10988	072116	000	000	000	DF65:	.BYTE	0,0,0,0,0	
10989	072121	000	000					
10990	072123	000	000	000	DF66:	.BYTE	0,0,0,0,0,0,0,0	
10991	072126	000	000	000				
10992	072131	000	000					
10993								
10994	072133	000	000	000	DF77:	.BYTE	0,0,0,0,0,0	
10995	072136	000	000	000				
10996								
10997	072142				.EVEN			
10998								
10999								
11000	000001				.END			

CZRJICO, RP04/5/6 FCTNL CTLR1		MACY11 30A(1052) 27-JUL-78 13:06 PAGE 243												SEQ 0241
CZRJIC.P11 27-JUL-78 13:05		CROSS REFERENCE TABLE -- USER SYMBOLS												
		768	853	865	879	893	905	913	920	930	939	950	973	994
DF5	072032	1012	1028	1046	1057	1071	1083	1108	1121	1143	1183	1202	1214	1239
		1282	1294	1360	1370	1388	1408	1427	10962#					
DF51	072072	1169	10979#											
DF6	072037	778	10964#											
DF60	072100	1256	10982#											
DF61	072106	1265	10984#											
DF62	072112	1274	10986#											
DF65	072116	1306	10988#											
DF66	072123	1324	10990#											
DF7	072043	786	10966#											
DF77	072133	1418	10994#											
DH1	067160	715	727	739	10686#									
DH10	067756	791	805	819	835	10755#								
DH26	070135	954	1432	1444	1456	1468	10775#							
DH30	070334	978	1000	1016	1033	1094	1128	1152	1188	1224	1333	1344	1375	1394
		10797#												
DH4	067337	752	10706#											
DH5	067500	763	848	860	874	888	900	911	918	928	937	948	971	992
		1010	1026	1044	1055	1069	1081	1106	1119	1141	1181	1200	1212	1237
		1280	1292	1355	1368	1386	1406	1422	10723#					
DH51	070450	1163	10812#											
DH6	067617	774	10737#											
DH60	070607	1250	10829#											
DH61	070744	1260	10846#											
DH62	071037	1269	10857#											
DH65	071132	1300	10868#											
DH66	071255	1316	10882#											
DH7	067716	783	10748#											
DH77	071353	1412	10894#											
DIGB =	000004	1539#												
DISPLA	001142	679#	1929*	1936*	8776*	9087*								
DISPRE	000174	630#	1936											
DLT =	100000	1512#												
DL64 =	000020	1541#												
DMD =	000001	1574#	2645	2656	2692	2846	3588	3599	3731	3742	7972			
DOG	017352	3579	3594#											
DPR	000400	1545#	2456	2592	2745	3677	3797	3968	3997	4156	4240	4413	4519	4630
		4755	4875	4995	5106	5232	5352	5488	5629	5796	5920	6085	6211	6385
		6503	6616	6782	7204	7369	7546	8168						
DRY	000200	1544#	2458	2592	2749	3659	3786	4003	4138	4222	4307	4320	6297	6605
		6746	6771	6922	7266	7439	8168							
DST	002366	1773#	9171											
DSWR =	177570	629#	678	1928										
DS1	002404	1780#	2745	2749	7975	7987	9155	10926	10929	10946				
DT	002406	1781#	2329*	3277	9207									
DTE =	010000	1567#												
DTSY =	001000	1580#												
DT1	071506	72*	733	744	10913#									
DT10	071604	798	812	826	842	10926#								
DT26	071624	961	1438	1450	1462	1474	10929#							
DT30	071646	982	1001	1017	1034	1095	1129	1153	1189	1225	1334	1345	1376	1395
		10932#												
DT4	071526	757	10916#											
DT5	071546	767	852	864	878	892	904	912	919	929	938	949	972	993
		1011	1027	1045	1056	1070	1082	1107	1120	1142	1182	1201	1213	1238

EM42	056265	1074	9871#											
EM43	056543	1086	9902#											
EM44	057105	1099	9942#											
EM45	057377	1111	9975#											
EM46	057673	1124	10008#											
EM47	060006	1133	10021#											
EM5	051430	761	9426#											
EM50	060270	1146	10053#											
EM51	060507	1157	10079#											
EM52	060754	1173	10108#											
EM53	061231	1186	10138#											
EM54	061317	1193	10148#											
EM55	061564	1205	10177#											
EM56	062052	1217	10209#											
EM57	062262	1229	10233#											
EM6	051502	771	9433#											
EM60	062554	1242	10267#											
EM61	063165	1259	10314#											
EM62	063246	1268	10323#											
EM63	063327	1278	10332#											
EM64	063403	1286	10340#											
EM65	063622	1299	10365#											
EM66	063644	1310	10368#											
EM67	064036	1326	10389#											
EM7	051570	781	9442#											
EM70	064326	1337	10422#											
EM71	064534	1349	10447#											
EM72	064775	10476#												
EM73	065225	1373	10504#											
EM74	065277	1362	1380	10512#										
EM75	065543	1391	10541#											
EM76	065646	1399	10553#											
EM77	066103	1411	10581#											
ERFLGS	004734	1867#	9084*											
ERR =	040000	1551#	2776	2831	4310	7751	7975	7987						
ERRVEC =	000004	629#	1926	1927*	1937*	2037*	2045*	2074*	8739	8740*	8742*	8745*		
ERWORD	004602	1826#	8641*	8647*	10932									
ER1	002364	1772#	9159	10926	10929	10948								
ER2	002370	1774#	9163	10929	10948									
ER3	002376	1777#	9167	10929	10948									
EXT1 -	000001	1609#												
EXT10 =	000010	1612#												
EXT2 =	000002	1610#												
EXT20 =	000020	1613#												
EXT4 =	000004	1611#												
EXT40 -	000040	1614#												
FEN =	000200	1635#												
FER =	000020	1559#												
FILL	042402	8422#												
FILLRE	041360	3206	3209	3246	3249	3252	3255	3258	3261	3268	3273	3282	3285	3418
		3421	3528	3531	3687	3690	3694	3697	3806	4261	4423	4426	4429	4529
		4532	4535	4640	4643	4646	4765	4768	4771	4885	4888	4891	5005	5008
		5011	5116	5119	5122	5242	5245	5248	5362	5365	5368	5517	5520	5523
		5639	5642	5645	5806	5809	5812	5930	5933	5936	6095	6098	6101	6104
		6107	6220	6223	6226	6395	6398	6402	6405	6408	6513	6516	6519	6626
		6792	7042	7045	7048	7109	7112	7115	7214	7217	7220	7379	7382	7385

OF	002372	1775#	9183											
OFREV =	000200	1653#												
OFSET	041442	4300	8120#											
OFSETC	002454	1807#	4119	4130	4141	4302	8121							
OFSTVL	004610	1833#	4078*	4079*	4098	4281*	4282	4284*	4285					
OF100 =	000004	1648#												
OF200 =	000010	1649#												
OF25 =	000001	1646#												
OF400 =	000020	1650#												
OF50 =	000002	1647#												
OF800 =	000040	1651#												
OPERSE	042602	8522#	8899	8965										
OPI =	020000	1568#												
OR =	000200	1504#	7727	7728										
PAR =	000010	1558#	3292											
PCJSR	041512	7966*	7967*	8145#	8147*	8148*	8153*	8154*	10942	10944	10952			
PCLBUF	002346	1747#	8253*											
PCLCSR	002344	1746#	8227*	8254*										
PCLCTR	002350	1748#	8228											
PGE =	002000	1507#												
PIP =	020000	1550#	3677	3797	3968	3997	4156	4240	6616	6782				
PIRQ =	177772	629#												
PIRQVE=	000240	629#												
PKACK	002460	1809#	2491	2502	2562	2573	2584	2875	2985	3874	4059			
PLU -	020000	1641#												
PRE	000020	1681#												
PRITEM	047274	2011*	2543*	2950*	3465*	4048*	7878*	7906*	8524*	9127#	9213*	9230	9234*	
PROG =	001000	1546#												
PRO =	000000	629#												
PR1 =	000040	629#												
PR2 =	000100	629#												
PR3 =	000140	629#												
PR4 =	000200	629#												
PR5 =	000240	629#												
PR6 =	000300	629#												
PR7 =	000340	629#												
PS =	177776	629#	1941*	2360*	2389*	2413*	7883*	8523*						
PSEL =	002000	1529#	5452	5468	5481	5595	5611	5622	7511	7528	7539			
PSU =	000001	1678#												
PSW =	177776	629#												
PUTREG	043370	2723	2744	2959	7974	7986	8471	8565#						
PWRVEC=	000024	629#	1919*	1920*	9344*	9345*	9354*	9357*	9369*	9370*				
RA	000200	639#	8712											
RDCHR =	104410	9017	9339#											
RDLIN =	104411	9049	9340#											
RDOCT -	104412	2020	8540	8547	8676	8694	9341#							
RDY =	000200	1526#	2361	2372	2390	2469	2585	2763	3036	3111	3200	3522	3663	3790
		3940	3949	4142	4226	4402	4508	4619	4744	4864	4984	5095	5221	5341
		5477	5618	5785	5909	6074	6200	6374	6492	6609	6709	6775/	6877	7037
		7103	7193	7358	7535	7636	7718	7811	8159					
READAT	002446	1804#	4596	4611	4622	4841	4856	4867	5072	5087	5098	5318	5333	5344
		5594	5610	5621	7801	7805								
READIN	002462	1810#	3384	3397	3405	3494	3507	3515						
RECALI	002426	1796#	3643	3652	3662	3770	3779	3789	4315	6290	6738			
REFOR	002450	1805#	4485	4500	4511	5884	5899	5912	6177	6192	6203	6468	6484	6495
		6695	6703	6863	6871	7082	7097							

REGADR	004600	1825#	2089*	2480*	2590*	2597*	3675*	3682*	3795*	3802*	3963*	3979*	4154*	4161*
		4238*	4245*	4411*	4418*	4517*	4524*	4628*	4635*	4753*	4760*	4873*	4880*	4993*
		5000*	5104*	5111*	5230*	5237*	5350*	5357*	5486*	5493*	5627*	5634*	5794*	5801*
		5918*	5925*	6083*	6090*	6209*	6216*	6383*	6390*	6501*	6508*	6614*	6621*	6780*
		6787*	7202*	7209*	7367*	7374*	7544*	7551*	7997*	8480*	10919	10921	10935	10939
		10952												
REGSAV	047130	9084#												
REGSA1	047136	2035	9085#											
REINTO	003534	1819#	4356	4363	4456	4471	4482	4534	4562	4593	4645	4670	4701	4705
		4796	4812	4816	4838	4890	4916	4945	5036	5051	5069	5121	5147	5178
		5182	5273	5289	5293	5315	5367	5393	5433	5437	5549	5565	5569	5591
		5644	5671	5729	5736	5740	5747	5839	5854	5881	5935	5964	6017	6024
		6028	6035	6134	6149	6174	6225	6253	6434	6465	6518	6544	6665	6692
		6716	6832	6860	6884	7003	7079	7114	7144	7782	7798	7818		
RELEAS	002432	1798#												
RESVEC=	000010	629#												
RETCL	002456	1808#	4203	4214	4225									
RHAS	002316	1728#	2083	2105	3383*	3493*	4192*							
RHBA	002274	1716#	3210	4427	4533	4644	4769	4889	5009	5120	5246	5366	5521	5643
		5810	5934	6099	6224	6399	6517	7046	7113	7218	7383	7560	7725	8598*
RHBAE	002340	1740#	2063											
RHCA	002312	1726#	3259	3419	3529	6102	6406	7969*	8093*	8111*	8595*			
RHCC	002334	1735#	2478	2480	2482	3191	3688	3807	6105	6403	6627	6793	7993	7997
		8229	8571											
RHCS1	002300	1721#	2491*	2504*	2562*	2575*	2760	2817	2877*	2987*	3019*	3032*	3035	3094*
		3107*	3110	3196*	3199	3228	3289	3384*	3399*	3407*	3494*	3509*	3517*	3521
		3643*	3654*	3695	3770*	3781*	3816	3876*	3921*	3934*	3939	4008	4016	4061*
		4119*	4132*	4167	4203*	4216*	4251	4304*	4317*	4394*	4401	4502*	4507	4613*
		4618	4736*	4743	4858*	4863	4976*	4983	5089*	5094	5213*	5220	5335*	5340
		5452*	5469*	5476	5595*	5612*	5617	5777*	5784	5901*	5908	6066*	6073	6194*
		6199	6292*	6364*	6373	6486*	6491	6598*	6636	6705*	6708	6740*	6764*	6802
		6873*	6876	6916*	7033*	7036	7099*	7102	7187*	7192	7262*	7352*	7357	7435*
		7511*	7529*	7534	7631*	7635	7714*	7717	7731	7807*	7810	7971*	7980*	8095*
		8112*	8121*	8125	8280	8292	8360	8373	8604*	8671	8680	8683	8698	
		1717#	2106	2157*	2253*	2780	2833	3219	3236	5501	7726	8126	8601*	
RHCS2	002276	1741#												
RHCS3	002342	1714#	2040	8677										
RHDB	002270	1723#	2722	2958	3250	3422	3532	4430	4536	4647	4772	4892	5012	5123
RHDST	002304	5249	5369	5524	5646	5813	5937	6108	6227	6409	6520	7049	7116	7221
		7386	7563	7738	7970*	8094*	8596*							
RHDS1	002322	1730#	2507	2513	2580	2600	2771	2826	2881	2991	3126	3274	3411	3434
		3544	3658	3698	3785	3811	3857	3879	3998	4065	4137	4172	4221	4256
		4306	4310	4319	6296	6604	6631	6745	6770	6797	6921	7265	7438	7658
		7746	7853	8127										
RHDT	002324	1731#	2158	2160	2163	2165	2168	2170	2183	2254	2256	2263	2265	2284
		2290	2292	2295	2297	2300	2302	2327	2329					
RHEC1	002330	1733#	3283											
RHEC2	002332	1734#	3286											
RHER1	002302	1722#	2109	3247	7741	8128								
RHER2	002306	1724#	3253											
RHER3	002314	1727#	2785	2838	3262									
RHLA	002336	1736#	8230											
RHMR	002320	1729#	2645*	2656*	2692*	2843	3269	3588*	3599*	3731*	3742*	7972*		
RHOF	002310	1725#	3256	3425	3535	3691	4262	8120*	8602*					
RHSN	002326	1732#	2280	2326	2328									
RHWC	002272	1715#	2496	2567	2697	3024	3062	3099	3162	3207	3350	3389	3475	3499

UPE = 020000	1510#													
US1 = 000001	1497#													
US2 = 000002	1498#													
US4 = 000004	1499#													
UWR = 000010	1680#													
VUF = 000002	1679#													
VU30 = 010000	1640#													
VV = 000100	1543#	2460	2508	2516	2581	2592	2603	2778	2882	2992	3412	3437	3547	
	3677	3797	3859	3880	3970	3971	3999	4000	4066	4156	4240	4413	4519	
	4630	4755	4875	4995	5106	5232	5352	5488	5629	5796	5920	6085	6211	
	6385	6503	6616	6782	7204	7369	7546	8150	8168					
WAITBT 041712	8250#	8263*	8270	8275	8344*	8350	8355	10913	10916					
WAITPC 041706	8248#	8260*	8261*	8341*	8342*	10913	10916							
WAITRE 041710	8249#	8262*	8270	8275	8279	8291	8343*	8350	8355	8359	8375	10913	10916	
WAITTM 041714	8228*	8251#	8252*	8299	8309	10916								
WAIT.P 041716	8252#													
WAIT.T 042156	8337#	9342												
WAT = 104413	2506	2579	2880	2990	3034	3109	3198	3410	3520	3657	3784	3878	3938	
	4064	4136	4220	4305	4318	4400	4506	4617	4742	4862	4982	5093	5219	
	5339	5475	5616	5783	5907	6072	6198	6295	6372	6490	6603	6707	6744	
	6769	6875	6920	7035	7101	7191	7264	7356	7437	7533	7634	7716	7809	
	9342#													
WC 002354	1766#	2523	2611	2798	2856	3044	3137	3300	3445	3555	3706	3825	4028	
	4182	4268	4438	4544	4655	4780	4900	5020	5131	5257	5377	5532	5654	
	5821	5945	6116	6234	6417	6528	6644	6810	7057	7125	7229	7394	7570	
	7667	7759	7863	8570	9139									
WCE = 040000	1511#													
WCF = 000040	1560#													
WCU = 000001	1628#													
WLE = 004000	1566#	7744												
WRCHDT 002440	1801#	7169	7185	7196										
WRCHK 002436	1800#	7336	7350	7361	7508	7527	7538							
WRFROM 002470	1818#	4336	4346	4374	4428	4457	4561	4582	4669	4688	4692	4716	4770	
	4797	4823	4827	4915	4936	4956	5010	5037	5058	5146	5165	5169	5193	
	5247	5274	5300	5304	5392	5420	5424	5448	5522	5550	5576	5580	5670	
	5695	5704	5710	5719	5758	5811	5840	5861	5870	5963	5983	5992	5998	
	6007	6046	6100	6135	6155	6164	6252	6306	6316	6323	6333	6344	6400	
	6442	6450	6454	6543	6671	6678	6682	6715	6839	6846	6850	6883	6929	
	6940	6944	6948	6952	6956	6965	6985	6989	6996	7013	7047	7143	7166	
	7219	7275	7279	7283	7287	7291	7300	7319	7323	7333	7384	7448	7452	
	7456	7460	7464	7473	7491	7495	7505	7561	7608	7618	7689	7699	7788	
	7817													
WRIDAT 002442	1802#	4719	4734	4747	4959	4974	4987	5196	5211	5224	5451	5467	5480	
	7621	7629	7702	7712										
WRIFOR 002444	1803#	4377	4392	4405	5761	5775	5788	6049	6064	6077	6347	6362	6377	
	7016	7031												
WRL = 004000	1548#	7661	7856											
WRU = 000400	1636#													
WSU = 000004	1630#													
XE2 007462	2120	2143#												
\$AUTOB 001134	675#	1980*	8941	9043										
\$BDADR 001122	670#	2432*	2440*	2462*	2472*	8162*	8170*	10926	10939	10942	10944			
\$BDDAT 001126	672#	2087*	2482*	2589*	2596*	3674*	3681*	3794*	3801*	3962*	3978*	4153*	4160*	
	4237*	4244*	4410*	4417*	4516*	4523*	4627*	4634*	4752*	4759*	4872*	4879*	4992*	
	4999*	5103*	5110*	5229*	5236*	5349*	5356*	5485*	5492*	5626*	5633*	5793*	5800*	
	5917*	5924*	6082*	6089*	6208*	6215*	6382*	6389*	6500*	6507*	6613*	6620*	6779*	

\$TKS	001144	680#	8881	8890*	8911*	8913	8919*	8935	8945*	8954	8964*	8980*		
\$TKSRV	045736	8887	8892#											
\$TMP0	001176	695#	2153*	2154*										
\$TMP1	001200	696#	2047*	2363*	2364*	2392*	2393*	6300*	6573*	10923				
\$TMP2	001202	697#												
\$TMP3	001204	698#												
\$TMP4	001206	699#												
\$TMP5	001210	700#												
\$STN =	000052	609#	2017	2027	2031	2032#	2033	2078	2079#	2081	2086	2095	2096#	2097
		2205	2210	2211#	2213	2334	2335#	2338	2350	2351#	2352	2369	2373	2379
		2380#	2381	2398	2409	2410#	2422	2548	2549#	2556	2634	2636	2641	2642#
		2643	2649	2652	2653#	2654	2662	2663#	2671	2674	2687	2754	2889	2890#
		2898	2905	2915	2916	3011	3012#	3152	3153#	3316	3317#	3327	3468	3469#
		3584	3585#	3586	3595	3596#	3597	3607	3608#	3727	3728#	3729	3738	3739#
		3740	3749	3750#	3837	3838#	3846	3847	4070	4071#	4072	4288	4291	4292#
		4294	4324	4328	4329#	4574	4575#	4680	4681#	4928	4929#	5157	5158#	5406
		5407#	5408	5413	5688	5689#	5976	5977#	6266	6267#	6730	6731#	6900	6901#
		7245	7246#	7410	7411#	7416	7591	7592#	7600	7601	7881	7882#		
\$TPB	001152	683#	8871*	8881										
\$TPFLG	001157	687#	8833	8881										
\$TPS	001150	682#	8869	8881										
\$TRAP	050652	1917	9321#											
\$TRAP2	050674	9328#	9331											
\$TRP =	000014	9331#	9333#	9334#	9335#	9336#	9337#	9338#	9339#	9340#	9341#	9342#	9343#	
\$TRPAD	050706	9326	9331#											
\$TSTNM	001102	660#	7894*	7923*	8735	8749	8771*	8776	8780	9087	9110			
\$TTYIN	046714	9014	9015	9032	9036#									
\$TYPBN=	***** U	9337												
\$TYPDS	045202	8780#	9336											
\$TYPE	045426	8833#	9331	9332										
\$TYPEC	045576	8854	8860	8864	8869#	8870	8982							
\$TYPEX	045644	8875	8877	8880#										
\$TYPOC	050450	9274#	9333											
\$TYPON	050464	9273	9276#	9335										
\$TYPOS	050424	9269#	9334											
\$XTSTR	044742	8738#												
\$GET4=	000000	7939#												
\$OFILL	050647	9270*	9274*	9284	9319#									
\$LOCAT=	***** U	8736	9097											
.	= 072142	630#	633#	634#	635#	638#	642#	658#	659#	707	1817#	1818#	1819#	1836#
		1855#	1911	1924	1925	1959#	1964#	1966#	1971#	2054#	2067#	2072#	2130#	2132#
		2179#	2183#	2187#	2242#	2276#	2314#	2318#	2322#	2343#	2345#	2349#	2429#	2926#
		2940#	2946#	2998#	3000#	3863#	4047#	5512#	7650#	7654#	7845#	7891#	7946	7950
		8536#	8545#	8690#	8706#	8712#	8720#	8779	8780	8833#	8881	8884#	9036#	9043
		9083	9110	9139#	9143#	9175#	9179#	9183#	9187#	9191#	9195#	9207#	9211#	9266#
		9356	9377	10911#	10997#									

. ABS. 072142 000

ERRORS DETECTED: 0

CZRJIC,DSKW:CZRJIC.SEQ/SOL/NL:TOC:MC:ME:CND-SYSMAC.SMI [400,1066],CZRJIC.P11[400,1424]
 RUN-TIME: 46 52 2 SECONDS
 RUN-TIME RATIO: 267/102 2.6

CZRJICO, RP04/5/6 FCTNL CTLR1
CZRJIC.P11 27-JUL-78 13:05

MACY11 30A(1052) 27-JUL-78 13:06 ^{K 4} PAGE 258
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0256

CORE USED: 38K (75 PAGES)

DOCUMENT PAGES: 256