

# RP04/5/6

DISKLESS CONTROLLER 1  
CZRJGB0

AH-9210B-MC  
COPYRIGHT © 74-77  
FICHE 1 OF 2

JAN 1978  
**digital**  
MADE IN USA

This microfiche card contains 120 frames of data, organized in a 10x12 grid. Each frame displays a small, high-contrast image of a document page. The text within these frames is too small to be legible, but they appear to be technical diagrams, tables, or code listings. The overall layout is consistent with standard microfiche formatting for data storage and retrieval.

# RP04/5/6

DISKLESS CONTROLLER 1  
CZRJGB0

AH-9210B-MC

COPYRIGHT © 74-77

FICHE 2 OF 2

JAN 1978

**digital**

MADE IN USA

B01

EOF1CZR00000411

00010000

780105

IDENTIFICATION

IHDR1CZRJGBSEG

00010000

780105  
SEQ 0001

PRODUCT CODE: AC-92088-MC  
PRODUCT NAME: CZRJGB0 RPO4/5/6 DISKLESS CONTROLLER TEST - PART 1  
PRODUCT DATE: DECEMBER 1977  
MAINTAINER: DIAGNOSTIC ENGINEERING

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this manual.

The software described in this document is furnished to the purchaser under a license for use on a single computer system and can be copied (with inclusion of Digital's copyright notice) only for use in such system, except as may otherwise be provided in writing by Digital.

Digital Equipment Corporation assumes no responsibility for the use or reliability of its software on equipment that is not supplied by Digital.

Copyright (C) 1974, 1977 Digital Equipment Corporation

503 \*\*\*ERROR TABLE, BIT DEFINITIONS & STARTING ADDRESSES\*\*\*

504 OPERATIONAL SWITCH SETTINGS  
 505 BASIC DEFINITIONS  
 506 TRAP CATCHER  
 507 ACT11 HOOKS  
 508 STARTING ADDRESS  
 509 MEMORY MANAGEMENT DEFINITIONS  
 510 COMMON TAGS  
 511 ERROR POINTER TABLE  
 512 REGISTER ADDRESSES

513 \*\*\*DIAGNOSTIC CODE\*\*\*

514 SETUP TESTS  
 515 INITIALIZE THE COMMON TAGS  
 516 GET VALUE FOR SOFTWARE SWITCH REGISTER  
 517 T1 REFERENCE EACH REGISTER  
 518 T2 RHCS2-CONTROL AND STATUS 2  
 519 T3 PARTIAL TEST OF RHAS FOR UNIT NUMBERS PRESENT  
 520 T4 TEST FOR DRIVES PRESENT USING RHAS AND RHCS2  
 521 T5 TYPE SERIAL NUMBER AND DRIVE TYPE  
 522 T6 CHECK MOL TO BE LOW  
 523 REGISTER TESTS  
 524 T7 RHCS2 - CONTROL AND STATUS 2  
 525 T10 RHCS1 - CONTROL AND STATUS 1 REGISTER  
 526 T11 RHCS1 - BIT #13 - MCPE  
 527 T12 RHWC - WORD COUNT REGISTER  
 528 T13 RHBA - UNIBUS ADDRESS REGISTER  
 529 T14 RHER1 - ERROR REGISTER #1  
 530 T15 RHMN - MAINTENANCE REGISTER  
 531 T16 RHDST - DESIRED SECTOR/TRACK ADDRESS  
 532 T17 RHER2 - ERROR REGISTER #2  
 533 T20 RHOF - MARGIN/OFFSET REGISTER  
 534 T21 RHCA - DESIRED CYLINDER REGISTER  
 535 T22 RHER3 - ERROR REGISTER #3  
 536 T23 CONTROL AND STATUS 2 (RHCS 2) - 'NED'  
 537 T24 CONTROL AND STATUS 2 (RHCS2) - 'CLR'  
 538 T25 PACK ACKNOWLEDGE COMMAND TEST  
 539 T26 UNIBUS INIT TEST  
 540 SILO TESTS  
 541 T27 SILO TST 1  
 542 T30 SILO TEST 2  
 543 T31 SILO TEST 3  
 544 T32 SILO TEST 4  
 545 T33 SILO TEST 5  
 546 MORE REGISTER TESTS  
 547 T34 TEST ODD BYTE INSTRUCTION ON RHCS1 - RH11  
 548 T35 TEST ODD BYTE INSTRUCTION ON RHCS1 - RH70  
 549 T36 TEST ODD BYTE INSTRUCTION ON RHCS2  
 550 T37 ODD BYTE TEST ON RHWC  
 551 T40 TEST ODD BYTE INSTRUCTION ON RHBA  
 552 DCL COMMAND TESTS  
 553 T41 TEST ILF BIT #0 IN REG. RHER1  
 554 T42 READ IN PRESET

503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554

2841	T43	NO OPERATION FUNCTION TEST
2967	T44	DRIVE CLEAR
3120	T45	SEEK COMMAND TEST
3372	T46	UNLOAD COMMAND TEST
3493	T47	OFFSET COMMAND TEST
3626	T50	RETURN TO CENTER LINE COMMAND TEST
3747	T51	RECALIBRATE COMMAND TEST
3867	T52	RELEASE COMMAND TEST
3929	T53	MAKE CURRENT CYLINDER = 0
3931	T54	LOOK AHEAD REGISTER
4128	T55	MAKE CURRENT CYLINDER = 0
4132	READ/WRITE ADDRESSING VIA RHMR	
4135	T56	WRITE HEADER AND DATA 1
4226	T57	WRITE HEADER AND DATA 2
4327	T60	WRITE HEADER AND DATA 3
4432	T61	PROGRAM ERROR RHCS2 #10
4510	T62	READ HEADER AND DATA 1
4518	T63	READ HEADER AND DATA 2
4525	T64	READ HEADER AND DATA 3
4532	T65	WRITE DATA
4539	T66	READ DATA
4638	T67	WRITE CHECK HEADER AND DATA
4758	T70	WRITE CHECK DATA
4876	ERROR BIT FUNCTIONAL TESTS	
4880	T71	ATTENTION WITH ERROR TEST
4994	T72	BUS ADDRESS INHIBIT
5092	T73	RHCS2 - BIT # 11 - NEM
5169	T74	WRITE CHECK ERROR
5323	T75	ERROR REGISTER #1-BIT 4 -FORMAT ERROR
5345	T76	ERROR REGISTER #1-BIT 4 -FORMAT ERROR
5430	T77	RHER1 - BIT #2 - REG. MODIFICATION REFUSED
5510	T100	MAKE CURRENT CYLINDER = 1
5513	T101	ERROR REG1 - BIT #7 - HEADER COMPARE ERROR
5517	T102	MAKE CURRENT CYLINDER = 0
5520	T103	ERROR REG1 - BIT #7 - HEADER COMPARE ERROR
5524	T104	MAKE CURRENT CYLINDER = 1
5527	T105	ERROR REG.1 - BIT #7 - HEADER COMPARE ERROR
5532	T106	MAKE CURRENT CYLINDER = 0
5535	T107	ERROR REG.1 - BIT #7 - HEADER COMPARE ERROR
5539	T110	RHER1 - BIT #8 - CRC ERROR (READING)
5544	T111	RHER1 - BIT 8 - CRC ERROR (WRITING)
5558	T112	MAKE CURRENT CYLINDER = 814.
5564	T113	RHDS1 (BIT #10) - LAST SECTOR TRANSFERRED, 'LST'
5590	T114	MAKE CURRENT CYLINDER = 410.
5594	T115	RHDS1 (BIT #10) - LAST SECTOR TRANSFERRED, 'LST'
5616	T116	ERROR REGISTER 1 - BIT #9 AOE
5781	T117	MAKE CURRENT CYLINDER = 0
5784	T120	ERROR REGISTER 1 - BIT #10 'IAE'
5872	T121	ERROR REGISTER 1- BIT #10 'IAE'
5964	T122	ERROR REGISTER 1- BIT #10 'IAE'
6056	T123	END OF DRIVE

\*\*\*SUBROUTINES\*\*\*

END OF PASS ROUTINE  
 SAVE REGISTERS ROUTINE

6087  
 6088  
 6091  
 6133

CZRJGBD.RP04/5/6 DSKLS CTRLRI MACY11 30(1046) 08-NOV-77 08:54  
CZRJGB.P11 08-NOV-77 08:30 TABLE OF CONTENTS

SEQ 0004

6157	FLOAT 1 AND 0
6216	CLEAR MEMORY ROUTINE
6243	LOCAL TRAPS
6260	CLEAR DISK ROUTINE
6273	CHECK DISK STATUS ROUTINES
6407	SAVE ROUTINE
6429	WRITE CHECK ROUTINE
6469	COMPARE ROUTINE
6551	CRC GENERATION ROUTINE
6845	JAM CURRENT CYLINDER ROUTINE
6882	ECC GENERATION AND COMPARISON ROUTINE
7196	RH BASE ADDRESS CHANGE ROUTINE
7249	DISK SIMULATION
8225	SYSMAC LIBRARY ROUTINES
8227	SCOPE HANDLER ROUTINE
8228	CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
8229	TYPE ROUTINE
8230	TTY INPUT ROUTINE
8232	READ AN OCTAL NUMBER FROM THE TTY
8233	ERROR HANDLER ROUTINE
8234	ERROR MESSAGE TYPEOUT ROUTINE
8235	BINARY TO OCTAL (ASCII) AND TYPE
8236	TRAP DECODER
8237	TRAP TABLE
8238	POWER DOWN AND UP ROUTINES

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
- 9.0 PROGRAM DESCRIPTION

60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116

117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172

## 1.0 ABSTRACT

THE DIAGNOSTIC IS USED TO TEST RP04/5/6 DEVICE CONTROL LOGIC CONNECTED TO EITHER AN RH11 OR RH70 DISK DRIVE CONTROLLER

THIS DIAGNOSTIC TESTS THE RH11 AND DCL OF AN RJPO4/5/6 SUBSYSTEM. IT DOES NOT USE THE DISK SURFACE OR ANY SIGNALS FROM THE MDLI. IT REQUIRES THAT THE DCL CABLE BE PLUGGED INTO THE MDLI OR BE APPROPRIATELY TERMINATED. IF THE DISK IS POWERED UP, IT IS REQUIRED TO GET THE DISK TO THE "HEADS UNLOADED" POSITION. AFTER A SUCCESSFUL RUN (WITH NO ERRORS) OF THIS DIAGNOSTIC IT CAN BE ASSERTED THAT, "THAT PART OF THE DCL THAT HANDLES DATA OR DATA ASSOCIATED LOGIC IS WORKING PROPERLY". THIS IMPLIES THAT THE PART OF THE LOGIC WHICH HANDLES MECHANICAL COMMANDS OR ITS ASSOCIATED LOGIC IS NOT TESTED IN THIS DIAGNOSTIC. ALL DATA COMMANDS USE THE MAINTENANCE REGISTER IN THE WRAPAROUND MODE.

THE DIAGNOSTIC DOES NOT DO ANY TESTING OF THE RH70 CONTROLLER WHEN IT IS USED ON AN RWPO4/5/6 SYSTEM TO TEST RP04/5/6 DISK DRIVES CONNECTED TO THAT TYPE OF CONTROLLER. IT IS ASSUMED THAT THE RH70 SPECIFIC CONTROLLER DIAGNOSTIC HAVE BEEN SUCCESSFULLY RUN TO COMPLETION BEFORE THIS DIAGNOSTIC IS RUN.

## 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/RH70 CONTROLLER, AND DISK CONTROL LOGIC (DCL), THE CABLE FROM THE DCL CAN BE CONNECTED TO THE MDLI, BUT IF NOT THAT CABLE MUST BE PROPERLY TERMINATED.

## 2.2 STORAGE

THIS PROGRAM REQUIRES 16K WORDS OF MEMORY.

## 2.3 PRELIMINARY PROGRAMS

THIS CAN BE THE FIRST PROGRAM RUN ON AN RJPO4/5/6 SYSTEM BUT THE CONTROLLER DIAGNOSTICS MUST BE RUN FIRST IN THE CASE OF AN RWPO4/5/6 SYSTEM.

## 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 FRON PAEL, OR IN THE SOFTWARE SWITCH REGISTER IF THE OPERATOR SO DESIRES. SE PARAGRAPH 5.1 FOR A DESCRIPTION OF SOFTWARE SWITCH REGISTER OPERATION.



173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230

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

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), THE PROCESSOR MAY BE HALTED AND RESTARTED FROM ADDRESS 210. THE NEW PARAMETERS WILL NOT BE CHANGED UNLESS THE PROGRAM IS AGAIN RESTARTED FROM ADDRESS 204. IF ALL UNITS ARE TO BE CHECKED, THE PROCESSOR NEED NOT BE TOUCHED. THE PROGRAM WILL AUTOMATICALLY RESTART AT 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.

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 PASSES DO NOT NEED ANY OPERATOR INTERVENTION.

5.0 OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

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

'SWR = NNNNNN NEW ='

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

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

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

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

SWITCH 14 - LOOP ON TEST  
WHEN THIS SWITCH IS SET THE PROGRAM WILL BEGIN TO LOOP  
ON THE CURRENT TEST BEING EXECUTED. FOR EXAMPLE IF THIS  
SWITCH IS SET WHEN THE PROGRAM IS IN TEST 10 THEN THE  
PROGRAM WILL KEEP EXECUTING ALL OF TEST 10 REPEATEDLY.  
ONE WAY TO BE SURE THAT THE PROGRAM IS IN THE EXPECTED  
TEST IS TO SET THIS SWITCH DURING AN ERROR PRINTOUT OR  
DURING A PROGRAM HALT.

229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284

285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340

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 CONTROFLER SELECT  
THIS SWICH 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: ALSO SEE SECTION 8.3

SWITCH 8 - LOOP ON TEST IN SWR <7:0>  
THIS IS A SPECIAL SWITCH. WHEN SET SWITCHES 0 THRU 7  
HAVE ONE MEANING AND WHEN RESET SWITCHES 0 THRU 7 HAVE  
ANOTHER MEANING. THIS MEANS THAT ANY SETTING OF SWITCH  
0 THRU 7 MUST BE DONE WITH SWITCH 8 IN THE APPROPRIATE  
POSITION. WHEN THIS SWITCH IS SET THEN SWITCHES 0 THRU  
7 GIVE THE TEST NUMBER TO BE LOOPED ON. FOR EXAMPLE  
WITH SWITCH 8 SET AND SWITCH 3 SET THE PROGRAM WILL LOOP

341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396

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: ALSO 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 - ECC TEST-COMPARE END RESULTS ONLY 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 ON ECC TESTS (TEST 120 THRU TEST 134) INSTEAD OF COMPARING CONTENTS OF THE POSITION AND PATTERN REGISTERS AFTER EVERY CLOCK, COMPARES WILL BE DONE ONLY AT THE END OF ALL THE CLOCKS.

5.2 SUB-ROUTINE ABSTRACTS

SEE SECTION 9 "SUBROUTINES"

397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452

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 RPO4/5/6 REGISTERS OR GOOD/RECEIVED DATA. IF THE ERROR OCCURRED IN A SUBROUTINE, THE ADDRESS OF THE SUBROUTINE CALL IS ALSO GIVEN. REFER TO THE PROGRAM LISTING AT THE STATED ADDRESS TO DETERMINE THE CAUSE OF THE ERROR.

6.1 'FATAL' ERRORS

IN THE EVENT THAT THE DISK DRIVE BECOMES UNAVAILABLE TO THE CONTROLLER, POWERS DOWN, OR CERTAIN CRITICAL STATUS BITS CANNOT BE CLEARED PRIOR TO THE START OF A TEST SEQUENCE - THIS INFORMATION WILL BE COMMUNICATED TO THE OPERATOR. IN ADDITION THE TTY BELL WILL RING AND THE PROGRAM WILL HALT.

IT IS SUGGESTED THAT IF THIS HAPPENS THE OPERATOR LOAD ADDRESS 200 (210) AND RESTART THE PROGRAM AS A FIRST ATTEMPT TO SOLVE THE PROBLEM. IF THE FAILURE CONTINUES TO OCCUR, 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.

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

7.0 RESTRICTIONS

IF THERE IS A DRIVE CONNECTED THEN THE OPERATOR MUST HAVE THE DRIVE PORT SWITCH LOCKED EITHER ON PORT A OR PORT B BUT NEVER LEAVE IT IN THE PROGRAMMABLE STATE. IF THERE IS NO DRIVE CONNECTED THEN THE CABLE NORMALLY GOING FROM THE DCL TO THE MDLI MUST BE PROPERLY TERMINATED.

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

8.0 MISCELLANEOUS

8.1 EXECUTION TIME

THE FIRST PASS OF THE PROGRAM WILL TAKE 30 SECONDS PER DRIVE. SUBSEQUENT PASSES WILL TAKE 1 MINUTE.

8.2 STACK POINTER

THE STACK IS INITIALLY SET TO 1000

453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487

8.3 OPERATOR SELECTABLE SCOPE LOOPS

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

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

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

8.4 PROGRAM REVISION HISTORY

9.0 PROGRAM DESCRIPTION

THE FOLLOWING SECTIONS DESCRIBE EACH TEST AND SUBROUTINES IN DETAIL AND CAN ALSO BE USED AS AN INDEX TO THE LISTING. THE LEFT MOST COLUMN IS THE LINE NUMBER WITHIN THE LISTING WHERE THAT ITEM WILL BE FOUND.

;\*DRIVE MUST BE LOCKED ON PORT A OR PORT B

491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502

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

;\*NOTE: MACROS BEGINNING WITH ".S" ARE SUPPLIED BY AN EXTERNAL SYSMAC.SML SYSTEM MACRO PACKAGE WHICH MUST BE MADE AVAILABLE TO THE SOURCE PROGRAM AT ASSEMBLY TIME.  
;\*

CZRJGB0.RP04/5/6 DSKLS CTRLRI  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 11  
BASIC DEFINITIONS

NO1

SEQ 0013

503

504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521

000200 000200 004240  
000204 000137 046240  
000210 000137 004230

.SBTTL STARTING ADDRESS

.=200

RA: JMP @#BEGIN ;NORMAL START  
ADDMOD: JMP @#BASECH ;MODIFY ADDRESSES  
JMP @#BEGIN2 ;SELECT DRIVE START

;\*\*STARTING ADDRESS 200 FOR NORMAL STARTS  
;\*\*THIS WILL TEST ALL DRIVES ON THE SYSTEM A SINGLE DRIVE AT A TIME

;\*\*STARTING ADDRESS 204 FOR PARAMETER MODIFICATION,  
;\*\*RESTART AUTOMATICALLY FROM 200 AFTER PARAMETER MODIFICATION.

;\*\*STARTING ADDRESS 210 WILL TEST ONLY ONE SPECIFIED DRIVE



C02

CZRJGBO, RPO4/5/6 DSKLS CTRLRI  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 13  
MEMORY MANAGEMENT DEFINITIONS

SEG 0015

522  
523

001110

. =1110

; ?

524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579

Line No.	Code 1	Code 2	Item Label	Register	Description
528	001226	057454	;*ITEM1	EM1	;WRONG DATA IN READING OR WRITING HARDWARE REGISTER ;PC ;REG. ADDR. ;GOOD DATA ;RECEIVED DATA ;\$ERRPC, REGADR, \$GDDAT, \$BDDAT ;0,0,0,0,0
529	001230	062727		DH1	
533	001232	067306	;*ITEM1	DT1	;WRONG DATA IN READING OR WRITING HARDWARE REGISTER ;PC ;REG. ADDR. ;GOOD DATA ;RECEIVED DATA ;\$ERRPC, REGADR, \$GDDAT, \$BDDAT ;0,0,0,0,0
534	001234	070024		DF1	
541	001236	057537	;*ITEM2	EM2	;ERROR ON DATA COMMAND  ;PC ;PC OF JSR ;TEST NO ;WORD NO. ;GOOD DATA ;CONTENTS OF RHCS1 ;CONTENTS OF RHDS1 ;CONTENTS OF RHER1 ;\$ERRPC, PCJSR, \$STSTNM, ERWORD, \$GDDAT, CS1, DS1, ER1 ;0,0,0,1,0,0,0,0
542	001240	066064		DH33	
551	001242	067666	;*ITEM2	DT33	;ERROR ON DATA COMMAND  ;PC ;PC OF JSR ;TEST NO ;WORD NO. ;GOOD DATA ;CONTENTS OF RHCS1 ;CONTENTS OF RHDS1 ;CONTENTS OF RHER1 ;\$ERRPC, PCJSR, \$STSTNM, ERWORD, \$GDDAT, CS1, DS1, ER1 ;0,0,0,1,0,0,0,0
552	001244	070174		DF33	
556	001246	057537	;*ITEM3	EM2	;ERROR ON DATA COMMAND  ;PC ;PC OF JSR ;TEST NO ;WORD NO. ;GOOD DATA ;BAD DATA ;CONTENTS OF RHCS1 ;CONTENTS OF RHDS1 ;CONTENTS OF RHER1 ;\$ERRPC, PCJSR, \$STSTNM, ERWORD, \$GDDAT, \$BDDAT, CS1, DS1, ER1 ;0,0,0,1,0,0,0,0,0
558	001250	065641		DH32	
568	001252	067642	;*ITEM3	DT32	;ERROR ON DATA COMMAND  ;PC ;PC OF JSR ;TEST NO ;WORD NO. ;GOOD DATA ;BAD DATA ;CONTENTS OF RHCS1 ;CONTENTS OF RHDS1 ;CONTENTS OF RHER1 ;\$ERRPC, PCJSR, \$STSTNM, ERWORD, \$GDDAT, \$BDDAT, CS1, DS1, ER1 ;0,0,0,1,0,0,0,0,0
569	001254	070163		DF32	
573	001256	057537	;*ITEM4	EM2	;ERROR ON DATA COMMAND  ;PC ;TEST NO ;WORD NO. ;GOOD DATA ;BAD DATA
575	001260	065436		DH31	

580					:CONTENTS OF RHCS1
581					:CONTENTS OF RHDS1
582					:CONTENTS OF RHER1
583					
584	001262	067620	DT31		;\$ERRPC,\$STSTNM,ERWORD,\$GDDAT,\$BDDAT,CS1,DS1,ER1
585	001264	070153	DF31		:0,0,1,0,0,0,0,0,
586					
587					
588					
589				;*ITEM5	
590	001266	000000	0		:
591	001270	000000	0		:
592	001272	067620	DT31		;\$ERRPC,\$STSTNM,ERWORD,\$GDDAT,\$BDDAT,CS1,DS1,ER1
593	001274	070153	DF31		:0,0,1,0,0,0,0,0,
594					
595					
596				;*ITEM6	
597	001276	057566	EM6		;\$ERRPC,\$STSTNM,ERWORD,\$GDDAT,\$BDDAT,CS1,DS1,ER1
598					:0,0,1,0,0,0,0,0,
599	001300	065641	DH32		;
600					:PC
601					:PC OF JSR
602					:TEST NO
603					:WORD NO.
604					:GOOD DATA
605					:BAD DATA
606					:CONTENTS OF RHCS1
607					:CONTENTS OF RHDS1
608					:CONTENTS OF RHER1
609	001302	067642	DT32		;\$ERRPC,PCJSR,\$STSTNM,ERWORD,\$GDDAT,\$BDDAT,CS1,DS1,ER1
610	001304	070163	DF32		:0,0,0,1,0,0,0,0,0,
611					
612					
613					
614				;*ITEM7	
615	001306	057566	EM6		;\$ERRPC,\$STSTNM,ERWORD,\$GDDAT,\$BDDAT,CS1,DS1,ER1
616	001310	063052	DH2		:0,0,1,0,0,0,0,0,
617					;
618					:PC
619					:TEST NO
620					:WORD NO.
621					:GOOD DATA
622	001312	067334	DT3		:BAD DATA
623	001314	070035	DF3		;\$ERRPC,\$STSTNM,ERWORD,\$GDDAT,\$BDDAT
624					:0,0,1,0,0,
625					
626				;*ITEM10	
627	001316	000000	0		:
628	001320	000000	0		:
629	001322	067334	DT3		;\$ERRPC,\$STSTNM,ERWORD,\$GDDAT,\$BDDAT
630	001324	070035	DF3		:0,0,1,0,0,
631					
632				;*ITEM11	
633	001326	057625	EM11		;\$ERRPC,\$STSTNM,ERWORD,\$GDDAT,\$BDDAT
634	001330	063175	DH11		:0,0,1,0,0,
635					;
					:CONTROLLER OR DRIVE STATUS
					:PC
					:TEST NO

636				: FAILING REG. ADDR
637				: CONTENTS OF RHCS1
638				: CONTENTS OF RHCS2
639				: CONTENTS OF RHDS1
640				: CONTENTS OF RHER1
641	001332	067350	DT11	: \$ERRPC, \$TSTNM, \$BDADR, CS1, CS2, DS1, ER1
642	001334	070042	DF11	: 0,0,0,0,0,0
643				
644				
645				
646	001336	057625	; *ITEM12 EM11	: WRONG DATA FROM SILO
647				
648	001340	062727	DH1	: PC
649				: REG. ADDR
650				: GOOD DATA
651				: RECEIVED DATA
652	001342	067306	DT1	: \$ERRPC, REGADR, \$GDDAT, \$BDDAT
653	001344	070024	DF1	: 0,0,0,0
654				
655				
656				
657	001346	000000	; *ITEM13 0	
658	001350	000000	0	
659	001352	067306	DT1	: \$ERRPC, TSTNM, REGADR, \$GDDAT, \$BDDAT
660	001354	070024	DF1	: 0,0,0,0,0
661				
662				
663				
664	001356	057660	; *ITEM14 EM14	: REGISTER FAILED
665	001360	063355	DH14	: PC
666				: FAILING REG. ADDR
667				: CONTENTS OF FAILING REG.
668				: CONTENTS OF RHCS1
669				: CONTENTS OF RHCS2
670				: CONTENTS OF RHDS1
671				: CONTENTS OF RHER1
672	001362	067370	DT14	: \$ERRPC, \$BDADR, \$BDDAT, CS1, CS2, DS1, ER1
673	001364	070051	DF14	: 0,0,0,0,0,0,0
674				
675				
676				
677	001366	057700	; *ITEM15 EM15	: SPECIFIED REG. NON EXISTANT SO ABORT
678				: PROGRAM
679	001370	063555	DH15	: PC
680				: ADDR. OF REG
681	001372	067412	DT15	: \$ERRPC, TEMP1
682	001374	070061	DF15	: 0,0
683				
684				
685				
686	001376	057751	; *ITEM16 EM16	: WAIT LOOP FAILED
687	001400	063606	DH16	: PC
688				: WAT PC
689				: BIT WANTED
690				: REG. ADR.
691				: REG. CONT.

692	001402	067422	DT16	:SERRPC, \$TMP3, \$TMP1, \$TMP0, \$BDDAT
693	001404	070064	DF16	:0,0,0,0
694				
695				
696			:*ITEM17	
697	001406	057772	EM17	:WRITE CHECK FAILING
698	001410	063751	DH17	:PC
699				:TEST NO
700				:CONTENTS OF RHBA
701				:CONTENTS OF RHDB
702				:CONTENTS OF RHWC
703				:CONTENTS OF RHCS1
704				:CONTENTS OF RHCS2
705	001412	067440	DT17	:SERRPC, \$TSTNM, \$BA, DB, WC, CS1, CS2
706	001414	070071	DF17	:0,0,0,0,0,0,0
707				
708				
709			:*ITEM20	
710	001416	060016	EM20	:REGISTER FAILING
711	001420	064134	DH20	:PC
712				:TST NO
713				:CONTENTS OF RHER1
714				:CONTENTS OF RHER2
715				:CONTENTS OF RHER3
716				:CONTENTS OF RHAS
717				:CONTENTS OF RHDS1
718	001422	067460	DT20	:SERRPC, TSTNM, ER1, ER2, ER3, AS, DS1
719	001424	070100	DF20	:0,0,0,0,0,0,0
720				
721			:*ITEM21	
722				
723	001426	060037	EM21	:INTERRUPT FAILING
724	001430	064317	DH21	:PC
725				:TEST NO
726				:CONTENTS OF RHCS1
727				:CONTENTS OF RHAS
728				:CONTENTS OF RHDS1
729	001432	067500	DT21	:SERRPC, TSTNM, CS1, AS, DS1
730	001434	070107	DF21	:0,0,0,0,0
731				
732				
733			:*ITEM22	
734	001436	060061	EM22	:ERROR IN DRIVE PRESENT -
735				:LOOKING AT RHAS AND RHCS2-NED(BIT#12)
736				:DRIVES PRESENT DO NOT AGREE
737				:NOTE: ON DUAL PORT SYSTEM
738				:DRIVE ON OTHER PORT WILL NOT GIVE NED
739				:HENCE THERE WILL BE A MISSMATCH
740	001440	064437	DH22	:PC
741				:TEST NO
742				:RHAS UNIT (RHER1 BITS SET)
743				:RHCS2 UNIT ('NED' BIT TEST)
744	001442	067514	DT22	:SERRPC, TSTNM
745	001444	070114	DF22	:0,0
746				
747				

Line	Code	Address	Item	Description
748			;*ITEM23	
749	001446	000000		: NO LONGER USED DUE TO SPECIAL 'NED'
750	001450	000000		: TEST TABLE TYPE OUT ROUTINE
751	001452	000000		
752	001454	000000		
753				
754				
755			;*ITEM 24	
756	001456	060541	EM24	: LOOK AHEAD REGISTER AT THE
757				: BEGINNING OF A SECTOR IS IN
758				: ERROR
759	001460	064542	DH24	: PC
760				: RHDST
761				: BAD RHLA
762				: GOOD RHLA
763				: SECTOR NO
764				: SECTOR CLOCK
765	001462	067522	DT24	: \$ERRPC,DST,\$BDDAT,\$TMP1,\$TMP2,\$TMP3
766	001464	070120	DF24	: 0,0,0,0,0
767				
768			;*ITEM 25	
769	001466	060634	EM25	: LOOK AHEAD REGISTER IS
770				: IN ERROR
771				
772	001470	064542	DH24	: PC
773				: RHDST
774				: BAD RHLA
775				: GOOD RHLA
776				: SECTOR NO
777				: SECTOR CLOCK
778	001472	067522	DT24	: \$ERRPC,DST,\$BDDAT,\$TMP1,\$TMP2,\$TMP3
779	001474	070120	DF24	: 0,0,0,0,0
780			;*ITEM26	
781	001476	057625	EM11	: CONTROLLER OR DRIVE STATUS
782				
783	001500	064725	DH26	: PC
784				: PC OF JSR
785				: FAILING REGISTER ADDRESS
786				: CONTENTS OF RHCS1
787				: CONTENTS OF RHCS2
788				: CONTENTS OF RHDS1
789				: CONTENTS OF RHER1
790				
791	001502	067542	DT26	: \$ERRPC,PCJSR,\$BDADR,CS1,CS2,DS1,ER1
792	001504	070127	DF26	: 0,0,0,0,0,0,
793				
794				
795			;*ITEM27	
796				
797	001506	057454	EM1	: ERROR IN READING OR WRITING HARDWARE REGISTER
798				
799	001510	065130	DH27	: PC
800				: PC OF JSR
801				: TEST NUMBER
802				: FAILING REGISTER
803				: GOOD DATA

0004  
0005  
0006  
0007  
0008  
0009  
0010  
0011  
0012  
0013  
0014  
0015  
0016  
0017  
0018  
0019  
0020  
0021  
0022  
0023  
0024  
0025  
0026  
0027  
0028  
0029  
0030  
0031  
0032  
0033  
0034  
0035  
0036  
0037  
0038  
0039  
0040  
0041  
0042  
0043  
0044  
0045  
0046  
0047  
0048  
0049  
0050  
0051  
0052  
0053  
0054  
0055  
0056  
0057  
0058  
0059  
0060  
0061  
0062  
0063  
0064  
0065  
0066  
0067  
0068  
0069  
0070  
0071  
0072  
0073  
0074  
0075  
0076  
0077  
0078  
0079  
0080  
0081  
0082  
0083  
0084  
0085  
0086  
0087  
0088  
0089  
0090  
0091  
0092  
0093  
0094  
0095  
0096  
0097  
0098  
0099  
0100

001512 067564  
001514 070137

DT27  
DF27

;RECEIVED DATA  
;SERRPC,PCJSR,TSTNM,REGADR,\$GDDAT,\$BDDAT  
;0,0,0,0,0,0

;\*ITEM30

001516 060674  
001520 065273

EM30  
DH30

;CURRENT CYLINDER DOES NOT REFLECT DESIRED CYLINDER REG.  
;PC  
;PC OF JSR  
;REGISTER ADDRESS  
;GOOD DATA  
;BAD DATA

001522 067602  
001524 070145

DT30  
DF30

;SERRPC,PCJSR,REGADR,\$GDDAT,\$BDDAT  
;0,0,0,0,0

;\*ITEM31

001526 061015

EM31

;ECC GENERATED IS INCORRECT  
;EVERY WORD IN THIS SECTOR IS GIVEN IN "DATA USED"

001530 066267

DH34

;PC  
;TEST NUMBER  
;GOOD ECC1  
;GOOD ECC2  
;WRITTEN ECC1  
;WRITTEN ECC2  
;DATA USED

001532 067710

DT34

;SERRPC,TSTNM,GECC1,GECC2,WECC1,WECC2,DISK

001534 070204

DF34

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

;\*ITEM32

001536 061140

EM32

;ON READ COMMAND AFTER DATA AND ECC HAVE BEEN READ  
;ECC REGISTER OR RHER1 IS IN ERROR  
;ONLY LOWER 11 BITS OF PATTERN REGISTER  
;CAN BE READ  
;THIS SHUOLD MATCH LOWER 11 BITS OF ECC1

001540 066452

DH35

;PC  
;TEST NUMBER  
;GOOD ECC1  
;GOOD ECC2  
;PATTERN REGISTER  
;RHER1

001542 067730

DT35

;SERRPC,TSTNM,GECC1,GECC2,ECC2,ER1

001544 070213

DF35

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

860					
861			;*ITEM33		
862	001546	061427	EM33		: HIGH COUNT BIT NOT HIGH AFTER 38859 CLOCKS
863	001550	066655	DH36		: PC
864					: PC OF JSR
865					: TEST NUMBER
866					: RHMR
867					: POSITION REG.
868					: PATTERN REGISTER
869					
870	001552	067752	DT36		: SERRPC, PCJSR, TSTNM, MR, EC1, EC2
871					
872	001554	070223	DF36		: 0,0,0,0,0,0
873					
874			;*ITEM34		
875	001556	061501	EM34		: ZERO DETECT BIT NOT HIGH WHEN THE
876					: 32 BIT ECC REGISTER HAS ITS 21 BITS
877					: OF ZEROS
878					: ERROR PRINTOUT WILL CONTINUE TILL
879					: ZERO DETECT BIT IS HIGH
880	001560	066655	DH36		: PC
881					: PC OF JSR
882					: TEST NUMBER
883					: RHMR
884					: POSITION REG.
885					: PATTERN REGISTER
886					
887	001562	067752	DT36		: SERRPC, PCJSR, TSTNM, MR, EC1, EC2
888					
889	001564	070223	DF36		: 0,0,0,0,0,0
890					
891					
892					
893			;*ITEM35		
894	001566	061574	EM35		: POSITION REGISTER OR 11 BITS OF
895					: PATTERN REGISTER INCORRECT
896					: LOWER 11 BITS OF PATTERN REGISTER
897					: SHOULD MATCH LOWER 11 BITS OF GOOD ECC1
898					: DATA ENVELOPE AND N-CODE ZEROS ARE IN DECIMAL
899					
900	001570	067014	DH37		: PC
901					: TEST NUMBER
902					: ECC POSITION
903					: GOOD POSITION
904					: GOOD ECC1
905					: GOOD ECC2
906					: ECC PATTERN
907					: DATA ENVELOPE
908					: N-CODE ZEROS
909					
910	001572	067770	DT37		: SERRPC, TSTNM, EC1, POSITI, GECC1, GECC2, EC2, DATENV, ZCODE
911					
912	001574	070231	DF37		: 0,0,0,0,0,0,0,0,0,0
913					
914					
915					



```

916                                     ;*ITEM36
917 001576 062073                       EM36      ; ON A READ COMMAND WITH NON CORRECTABLE
918                                     ; ERROR INSERTED DCK AND ECH SHOULD BE SET
919 001600 066452                       DH35      ; PC
920                                     ; TEST NUMBER
921                                     ; GOOD ECC1
922                                     ; GOOD ECC2
923                                     ; PATTERN REGISTER
924                                     ; POSITION REGISTER
925                                     ; RHER1
926
927 001602 067730                       DT35      ; $ERRPC, TSTNM, GECC1, GECC2, EC2, EC1, ER1
928
929 001604 070213                       DF35      ; 0,0,0,0,0,0,0
930
931                                     ;*ITEM37
932 001606 062204                       EM37      ; PGE ERROR
933 001610 063175                       DH11      ; PC
934                                     ; TEST NO
935                                     ; FAILING REG. ADDR
936                                     ; CONTENTS OF RHCS1
937                                     ; CONTENTS OF RHCS2
938                                     ; CONTENTS OF RHDS1
939                                     ; CONTENTS OF RHER1
940 001612 067350                       DT11      ; $ERRPC, $TSTNM, $BDADR, CS1, CS2, DS1, ER1
941 001614 070042                       DF11      ; 0,0,0,0,0,0
942
943                                     ;*ITEM40
944 001616 062341                       EM40      ; RHWC DID NOT = 0 AFTER A READ OR
945                                     ; WRITE HEADER AND DATA
946 001620 067232                       DH40      ; PC
947                                     ; TEST NO
948                                     ; CONTENTS OF RHWC
949 001622 070014                       DT40      ; $ERRPC, TSTNM, $BDDAT
950 001624 070242                       DF40      ; 0,0,0
951
952                                     ;*RH11/RH70 REGISTERS
953
954
955
956                                     ;*WORD COUNT REGISTER (RHWC)
957                                     ;*EACH BIT IS CALLED BY BIT NUMBER
958
959
960
961                                     ;*BUS ADDRESS REGISTER (RHBA)
962                                     ;*EACH BIT IS CALLED BY BIT NUMBER
963
964
965
966                                     ;*CONTROL AND STATUS REGISTER 2 (RHCS2)
967
968 000001                               US1=      1      ; UNIT SELECT (BIT #0)
969 000002                               US2=      2      ; UNIT SELECT (BIT #1)
970 000004                               US4=      4      ; UNIT SELECT (BIT #2)
971 000010                               BAI=     10     ; BUS ADDRESS INCREMENT INHIBIT (BIT #3)

```

```

972      000020      PAT=      20      ; INVERT PARITY ON MASS BUS TO EVEN (BIT #4)
973      000040      CLR=      40      ; CLEAR (BIT #5)
974      000100      IR=       100     ; INPUT READY (BIT #6)
975      000200      OR=       200     ; OUTPUT READY (BIT #7)
976      000400      MPE=     400     ; MASS BUS PARITY ERROR (BIT #8)
977      001000      MXF=    1000    ; MISSED TRANSFER ERROR (BIT #9)
978      002000      PGE=    2000    ; PROGRAM ERROR (BIT #10)
979      004000      NEM=    4000    ; NON EXISTANT MEMORY (BIT #11)
980      010000      NED=   10000    ; NON EXISTANT DRIVE (BIT #12)
981      020000      UPE=   20000    ; UNIBUS PARITY ERROR (BIT #13)
982      040000      WCE=   40000    ; WRITE CHECK ERROR (BIT #14)
983      100000      DLT=  100000    ; DATA LATE (BIT #15)
984
985      ; *DATA BUFFER REGISTER (RHDB)
986      ; *EACH BIT IS CALLED BY BIT NUMBER
987
988
989
990
991      ; *RPO4 REGISTERS
992
993
994      ; *CONTROL AND STATUS 1 REGISTER. (#00)
995
996      000001      GO=       1      ; GO (BIT #0)
997      000100      IE=      100     ; INTERRUPT ENABLE (BIT #6)
998      000200      RDY=     200     ; READY (BIT #7)
999      000400      A16=    400     ; HIGH ORDER UNIBUS BITS (BIT #8)
1000     001000      A17=   1000    ; HIGH ORDER UNIBUS BITS (BIT #9)
1001     002000      PSEL=   2000    ; PORT SELECT (BIT #10)
1002     004000      DVA=    4000    ; DEVICE AVAILABLE (BIT #11)
1003     020000      MCPE=  20000    ; MASSBUSS PARITY ERROR (BIT #13)
1004     040000      TRE=   40000    ; TRANSFER ERROR (BIT #14)
1005     100000      SC=  100000    ; SPECIAL CONDITION (BIT #15)
1006
1007      ; *STATUS REGISTER (RHDS1) (#01)
1008
1009     000001      DFS=       1      ; DRIVE FORWARD 5"/SEC. (BIT #0)
1010     000002      DFF20=      2      ; DRIVE FORWARD 20"/SEC. (BIT #1)
1011     000004      DIGB=       4      ; DRIVE TO INNER GAVRD BAND (BIT #2)
1012     000010      GRV=      10      ; GO REVERSE (BIT #3)
1013     000020      DL64=     20      ; DIFFERENCE LESS THAN 64 (BIT #4)
1014     000040      DE1=     40      ; DIFFERENCE EQUALS 1 (BIT #5)
1015     000100      VV=     100      ; VOLUME VALID (BIT #6)
1016     000200      DRY=     200      ; DRIVE READY (BIT #7)
1017     000400      DPR=     400      ; DRIVE PRESENT (BIT #8)
1018     001000      PROG=    1000     ; PROGRAMABLE (BIT #9)
1019     002000      LST=    2000     ; LAST SECTOR TRANSFERRED (BIT #10)
1020     004000      WRL=    4000     ; WRITE LOCK (BIT #11)
1021     010000      MOL=   10000     ; MEDIUM ON-LINE (BIT #12)
1022     020000      PIP=   20000     ; POSITIONING OPERATION IN PROGRESS (BIT #13)
1023     040000      ERR=   40000     ; COMPOSIT ERROR. (BIT #14)
1024     100000      ATA=  100000     ; ATTENTION ACTIVE (BIT #15)
1025
1026      ; *ERROR REGISTER #01 (RHER1) (#02)
1027     000001      ILF=       1      ; ILLEGAL FUNCTION (BIT #0)

```

1028	000002	ILR=	2	; ILLEGAL REGISTER (BIT #1)
1029	000004	RMR=	4	; REGISTER MODIFICATION REFUSED (BIT #2)
1030	000010	PAR=	10	; PARITY ERROR (BIT #3)
1031	000020	FER=	20	; FORMAT ERROR (BIT #4)
1032	000040	WCF=	40	; WRITE LOCK FAIL (BIT #5)
1033	000100	ECH=	100	; ECC HARD ERROR (BIT #6)
1034	000200	HCE=	200	; HEADER COMPARE ERROR (BIT #7)
1035	000400	HCRC=	400	; HEADER CRC ERROR (BIT #8)
1036	001000	AOE=	1000	; ADDRESS OVERFLOW ERROR (BIT #9)
1037	002000	IAE=	2000	; INVALID ADDRESS ERROR (BIT #10)
1038	004000	WLE=	4000	; WRITE LOCK ERROR (BIT #11)
1039	010000	DTE=	10000	; DRIVE TIMING ERROR (BIT #12)
1040	020000	OPI=	20000	; OPERATION INCOMPLETE (BIT #13)
1041	040000	UNS=	40000	; DRIVE UNSAFE (BIT #14)
1042	100000	DCK=	100000	; DATA CHECK ERROR (BIT 15)
1043				
1044				; *MAINTAINABILITY REGISTER (RHMR) (#03)
1045				
1046	000001	DMD=	1	; DIAGINOSTIC MODE (BIT #0)
1047	000002	MCLK=	2	; MAINTAINABILITY CLOCK (BIT #1)
1048	000004	MINX=	4	; MAINTAINABILITY INDEX (BIT #2)
1049	000010	MSTCK=	10	; MAINTAINABILITY SECTOR CLOCK (BIT #3)
1050	000020	MRD=	20	; MAINTAINABILITY READ (BIT #4)
1051	000040	MWR=	40	; MAINTAINABILITY WRITE (BIT #5)
1052	000200	DENVL=	200	; DATA ENVELOPE (BIT #7)
1053	000400	ZER=	400	; ZERO DETECT (BIT #8)
1054	001000	DTSY=	1000	; MAINTAINABILITY SYNC DETECTED (BIT #9)
1055				
1056				; *ATTENTION SUMMARY PSEUDO-REGISTER (RHAS) (#04)
1057				
1058	000001	AT0=	1	; DEVICE 0 (BIT #0)
1059	000002	AT1=	2	; DEVICE 1 (BIT #1)
1060	000004	AT2=	4	; DEVICE 2 (BIT #2)
1061	000010	AT3=	10	; DEVICE 3 (BIT #3)
1062	000020	AT4=	20	; DEVICE 4 (BIT #4)
1063	000040	AT5=	40	; DEVICE 5 (BIT #5)
1064	000100	AT6=	100	; DEVICE 6 (BIT #6)
1065	000200	AT7=	200	; DEVICE 7 (BIT #7)
1066				
1067				
1068				
1069				
1070				
1071				; *DESIRED SECTOR/TRACK ADDRESS REGISTER (RHDST) (#1)
1072				; *EACH BIT IS CALLED BY BIT NUMBER
1073				
1074				
1075				
1076				
1077				
1078				; *DRIVE TYPE REGISTER (RHDT) (#06)
1079				; *EACH BIT IS CALLED BY BIT NUMBER
1080				
1081				
1082				
1083				

Address	Hex Value	Field Name	Field Description
; *LOOK-AHEAD REGISTER (RHLA) (#07)			
1084		EXT1=	1
1085		EXT2=	2
1086		EXT4=	4
1087	000001	EXT10=	10
1088	000002	EXT20=	20
1089	000004	EXT40=	40
1090	000010	SC1=	100
1091	000020	SC2=	200
1092	000040	SC4=	400
1093	000100	SC10=	1000
1094	000200	SC20=	2000
1095	000400	TRK1=	4000
1096	001000	TRK2=	10000
1097	002000	TRK4=	20000
1098	004000	TRK10=	40000
1099	010000	TRK20=	100000
1100	020000		
1101	040000		
1102	100000		
1103			
; *RP04 ERROR REGISTER #2 (RHER2) (#10)			
1106	000001	WCU=	1
1107	000002	CSF=	2
1108	000004	WSU=	4
1109	000010	CSU=	10
1110	000020	MSE=	20
1111	000040	TDF=	40
1112	000100	TUF=	100
1113	000200	FEN=	200
1114	000400	WRU=	400
1115	001000	MHS=	1000
1116	002000	NHS=	2000
1117	004000	IXE=	4000
1118	010000	VU30=	10000
1119	020000	PLU=	20000
1120	100000	ACU=	100000
1121			
; *RP05/6 ERROR REGISTER #2 (RHER2) (#10)			
1124	000001	WCU=	1
1125	000002	CSF=	2
1126	000004	WSU=	4
1127	000010	CSU=	10
1128	000020	RAW=	20
1129	000040	TDF=	40
1130	000100	TUF=	100
1131	000200	ABS=	200
1132	000400	WRU=	400
1133	001000	MHS=	1000
1134	002000	NHS=	2000
1135	004000	IXE=	4000
1136	020000	PLU=	20000
1137			
; *OFFSET REGISTER (RHOF) (#11)			

; EXTENSION 1 (BIT #0)  
; EXTENSION 2 (BIT #1)  
; EXTENSION 3 (BIT #2)  
; EXTENSION 4 (BIT #3)  
; EXTENSION 5 (BIT #4)  
; EXTENSION 6 (BIT #5)  
; SECTOR COUNT FIELD 0 (BIT #6)  
; SECTOR COUNT FIELD 1 (BIT #7)  
; SECTOR COUNT FIELD 2 (BIT #8)  
; SECTOR COUNT FIELD 3 (BIT #9)  
; SECTOR COUNT FIELD 4 (BIT #10)  
; TRACK FIELD 1 (BIT #11)  
; TRACK FIELD 2 (BIT #12)  
; TRACK FIELD 3 (BIT #13)  
; TRACK FIELD 4 (BIT #14)  
; TRACK FIELD 5 (BIT #15)  
; WRITE CURRENT UNSAFE (BIT #0)  
; CURRENT SINK FAILURE (BIT #1)  
; WRITE SELECT UNSAFE (BIT #2)  
; CURRENT SWITCH UNSAFE (BIT #3)  
; MOTOR SEQUENCE ERROR (BIT #4)  
; TRANSITIONS DETECTOR FAILURE (BIT #5)  
; TRANSITIONS UNSAFE (BIT #6)  
; FAILSAFE ENABLED (BIT #7)  
; WRITE READY UNSAFE (BIT #8)  
; MULTIPLE HEAD SELECT (BIT #9)  
; NO HEAD SELECTION (BIT #10)  
; INDEX ERROR (BIT #11)  
; 30VOLT UNSAFE (BIT #12)  
; PLO UNSAFE (BIT #13)  
; ACUNSAFE (BIT #15)  
; WRITE CURENT UNSAFE  
; CURRENT SINK FAILURE  
; CURENT SELECT UNSAFE  
; CURRENT SWITCH UNSAFE  
; READ AND WRITE  
; TRANSITIONS DETECTOR FAILURE  
; TRANSITIONS UNSAFE  
; ABNORMAL STOP  
; WRITE READY UNSAFE  
; MULTIPLE HEAD SELECT  
; NO HEAD SELECTION  
; INDEX ERROR  
; PLO UNSAFE

1140	000001	OF25=	1	; OFFSET 25 MICRO INCHES (BIT #0)
1141	000002	OF50=	2	; OFFSET 50 MICRO INCHES (BIT #1)
1142	000004	OF100=	4	; OFFSET 100 MICRO INCHES (BIT #2)
1143	000010	OF200=	10	; OFFSET 200 MICRO INCHES (BIT #3)
1144	000020	OF400=	20	; OFFSET 400 MICRO INCHES (BIT #4)
1145	000040	OF800=	40	; OFFSET 800 MICRO INCHES (BIT #5)
1146				
1147	000200	OFREV=	200	; OFFSET NEGATIVE (REVERSE) (BIT #7)
1148	002000	HCI=	2000	; HEADER COMPARE INHIBIT (BIT #10)
1149	004000	ECI=	4000	; ERROR CORRECTION CODE INHIBIT (BIT #11)
1150	010000	FMT22=	10000	; FORMAT BIT (BIT #12)
1151				
1152				
1153				
1154				
1155				
1156				
1157				
1158				
1159				
1160				
1161				
1162				
1163				
1164				
1165				
1166				
1167				
1168				
1169				
1170				
1171				
1172				
1173				
1174				
1175				
1176				
1177				
1178				
1179				
1180	000001	PSU=	1	; PACK SPEED UNSAFE (BIT #0)
1181	000002	VUF=	2	; VELOCITY UNSAFE (BIT #1)
1182	000010	UWR=	10	; ANY UNSAFE EXCEPT READ/WRITE (BIT #3)
1183	000020	PRE=	20	; DISK PACK ROTATION ERROR (BIT #4)
1184	000040	ACL=	40	; AC LOW (BIT #5)
1185	000100	DCL=	100	; DC LOW (BIT #6)
1186	040000	SKI=	40000	; SEEK INCOMPLETE (BIT #14)
1187	100000	OCYL=	100000	; OFF CYLINDER (BIT #15)
1188				
1189				
1190				
1191				
1192				
1193				
1194				
1195				

; \*DESIRED CYLINDER ADDRESS (RHCA) (#12)  
; \*EACH BIT IS CALLED BY BIT NUMBER.

; \*CURRENT CYLINDER ADDRESS (RHCC) (#13)  
; \*EACH BIT IS CALLED BY BIT NUMBER

; \*SERIAL NUMBER REGISTER (RHSN) (#14)  
; \*EACH IS CALLED BY BIT NUMBER

; \*ERROR REGISTER #03 (RHER3) (#15)

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

1196  
1197  
1198  
1199  
1200  
1201  
1202  
1203

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

1204  
1205  
1206  
1207  
1208  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1250  
1251  
1252  
1253  
1254

.SBTTL REGISTER ADDRESSES

;\*RP04 VECTOR ADDRESS

001626 000254

RPVEC: 254

;RP04 VECTOR ADDRESS

;\*RP04/5/6 DISK I/O REGISTERS LOCATED IN THE RH11 CONTROLLER  
;\*NOTE: THE CONTENTS OF THESE LOCATIONS WILL BE DIFFRENT  
;\* IF THE "CHANGE BASE ADDRESS" ROUTINE IS USED.  
;\* THIS ROUTINE STARTS AT LOCATION TAGED "BASECH"

001630 176722  
001632 176702  
001634 176704  
001636 176710

RHDB: 176722  
RHWC: 176702  
RHBA: 176704  
RHCS2: 176710

;DATA BUFFER SEE NOTE ABOVE  
;WORD COUNT SEE NOTE ABOVE  
;BUS ADDRESS SEE NOTE ABOVE  
;CONTROL AND STATUS 2 SEE NOTE ABOVE

;\*RP04/5/6 DISK I/O REGISTERS LOCATED IN THE DEVICE CONTROL LOGIC (DCL)  
;\*NOTE: THE CONTENTS OF THESE LOCATIONS WILL BE DIFFERENT  
;\* IF THE "CHANGE BASE ADDRESS ROUTINE IS USED.  
;\* THIS ROUTINE STARTS AT LOCATION TAGED "BASECH"

001640 176700  
001642 176714  
001644 176706  
001646 176740  
001650 176732  
001652 176734  
001654 176742  
001656 176716  
001660 176724  
001662 176712  
001664 176726  
001666 176730  
001670 176744  
001672 176746  
001674 176720  
001676 176736

RHCS1: 176700  
RHER1: 176714  
RHUST: 176706  
RHER2: 176740  
RHOF: 176732  
RHCA: 176734  
RHER3: 176742  
RHAS: 176716  
RHMR: 176724  
RHDS1: 176712  
RHDT: 176726  
RHSN: 176730  
RHEC1: 176744  
RHEC2: 176746  
RHLA: 176720  
RHCC: 176736

;CONTROL AND STATUS 1 SEE NOTE ABOVE  
;ERROR #1 SEE NOTE ABOVE  
;DESIRED SECTOR/TRACK ADDRESS SEE NOTE ABOVE  
;ERROR #2 SEE NOTE ABOVE  
;OFFSET SEE NOTE ABOVE  
;DESIRED CYLINDER ADDRESS SEE NOTE ABOVE  
;ERROR #3 SEE NOTE ABOVE  
;ATTENTION SUMMARY SEE NOTE ABOVE  
;MAINTAINABILITY SEE NOTE ABOVE  
;DRIVE STATUS SEE NOTE ABOVE  
;DRIVE TYPE SEE NOTE ABOVE  
;SERIAL NUMBER SEE NOTE ABOVE  
;ECC POSITION SEE NOTE ABOVE  
;ECC PATTERN SEE NOTE ABOVE  
;LOOK-AHEAD SEE NOTE ABOVE  
;CURRENT CYLINDER ADDRESS SEE NOTE ABOVE

;\*ADDITIONAL REGISTERS LOCATED IN THE RH70 CONTROLLER LOGIC

001700 176750  
001702 176752

RHBAE: 176750  
RHCS3: 176752

;BUS ADDRESS EXTENSION REGISTER  
;CONTROL AND STATUS REGISTER #3

```

1255
1256
1257
1258
1259
1260
1261
1262 001704 000000
1263 001706 000000
1264 001710 000000
1265 001712 000000
1266
1267
1268 001714 000000
1269 001716 000000
1270 001720 000000
1271 001722 000000
1272 001724 000000
1273 001726 000000
1274 001730 000000
1275 001732 000000
1276 001734 000000
1277 001736 000000
1278 001740 000000
1279 001742 000000
1280 001744 000000
1281 001746 000000
1282 001750 000000
1283 001752 000000

```

```

;*THE FOLLOWING LOCATIONS ARE RESERVED FOR REGISTER SAVES
;*ANY TIME THERE IS AN ERROR ALL THESE WILL BE FILLED
;*ONLY SOME MAY BE PRINTED BUT ALL WILL BE FILLED TRUE
;*FOR THE TIME JUST AFTER THE "ERROR" ERROR COMMAND

```

```

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

CS1:     0      ;CONTROL AND STATUS 1
ER1:     000000 ;ERROR #1
DST:     000000 ;DESIRED SECTOR/TRACK ADDRESS
ER2:     000000 ;ERROR #2
OF:      000000 ;OFFSET
CA:      000000 ;DESIRED CYLINDER ADDRESS
ER3:     000000 ;ERROR #3
AS:      000000 ;ATTENTION SUMMARY
MR:      000000 ;MAINTAINABILITY
DS1:     000000 ;DRIVE STATUS
DT:      000000 ;DRIVE TYPE
SN:      000000 ;SERIAL NUMBER
EC1:     000000 ;ECC POSITION
EC2:     000000 ;ECC PATTERN
LA:      000000 ;LOOK-AHEAD
CC:      000000 ;CURRENT CYLINDER ADDRESS

```



1284					
1285					
1286					
1287	001754	000010	UNITS:	.BLKW	8.
1288	001774	000000	UNIT:	.WORD	0
1289	001776	000000	NUNIT:	.WORD	0
1290					
1291	002000	000000	NUNIT:	.WORD	0
1292					
1293	002002	000000	SELECT:	.WORD	0
1294	002004	000000	UNITSL:	.WORD	0
1295					
1296	002006	000000	ERFLGS:	0	
1297					
1298	002010	000000	SAVDT:	0	
1299					
1300					
1301	002012	000000	SAVSN:	0	
1302					
1303					
1304					
1305	002014	000000	PCJSR:	0	
1306					
1307	002016	000000	ATTENT:	0	
1308	002020	000000	TOTALAT:	0	
1309					
1310	002022	000000	TMPILL:	0	
1311					
1312	002024	000000	TSECC:	0	
1313					
1314					
1315					
1316	002026	000000	TESDTE:	0	
1317					
1318					
1319					
1320	002030	000000	TAGDTE:	0	
1321					
1322					
1323	002032	000000	TSTNM:	0	
1324					
1325	002034	000000	FIRST:	0	
1326					
1327	002036	000000	RP06:	0	
1328					
1329	002040	000000	RH70:	0	
1330					

\*FLAGS & INTERNAL PROGRAM CONTROL WORDS

; TABLE OF DRIVES PRESENT TO TEST  
; UNIT UNDER TEST  
; NUMBER OF UNITS PRESENT  
; USED TO KEEP TRACK OF UNIT UNDER TEST  
; USED TO DETERMINE IF THERE IS MORE  
; THAN ONE UNIT  
; ALL ONES INDICATE UNIT TO BE SELECTED  
; UNIT NO. SELECTED

; ERROR FLAG

; SAVE DRIVE TYPE REGISTER  
; FOR COMPARISON IN DRIVE CLEAR TEST  
; AND RH INIT TEST  
; SAVE SERIAL NUMBER REGISTER  
; FOR COMPARISON IN DRIVE CLEAR TEST  
; AND RH INIT TEST

; SAVE PC OF JSR WHICH GAVE THE ERROR

; ATTENTION BIT FOR PRESENT UNIT  
; TATAL ATTENTION BITS

; TEMPORARY ILLEGAL FUNCTION

; FLAG TO SAY IF ECC TEST OR NOT  
; WHEN =177777 IT IS AN ECC TEST  
; WHEN =0IT IS NOT AN ECC TEST

; FLAG TO SAY IF DRIVE TIMING ERROR OR NOT  
; WHEN = 177777 IT IS A DTE TEST  
; WHEN = 0 IT IS NOT A DTE TEST

; TEMPORARY TAG USED IN DRIVE TIMING  
; ERROR TEST

; TEST NUMBER

; IF ZERO WILL TYPE HEADER

; IF 0 PROGRAM WILL TREAT DRIVE AS RP04

; IF 1 PROGRAM IS RUNNING ON RH70  
; IF 0 PROGRAM IS ON AN RH11

```

1331
1332
1333
1334
1335
1336
1337 002042
1338 002042 000000
1339 002044 000002
1340 002046 000006
1341 002050 000010
1342 002052 000012
1343 002054 000030
1344 002056 000050
1345 002060 000052
1346 002062 000060
1347 002064 000062
1348 002066 000070
1349 002070 000072
1350 002072 000004
1351 002074 000014
1352 002076 000016
1353 002100 000022
1354 002102 000020
1355 002104 000000
1356
1357
1358
1359
1360
1361 002110 000422
1362 003154 000422
1363
1364
1365
1366
1367
1368
1369 004220 001 002 004
1370 004223 010 020 040
1371 004226 100 200
1372

```

```

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

FUTABL:
NOPERA: 0 ;NO OPERATION
UNLOAD: 2 ;UNLOAD (STAND BY)
RECALI: 6 ;RECALIBRATE
DCLEAR: 10 ;DRIVE CLEAR
RELEAS: 12 ;RELEASE (DUAL-PORT OPERATION)
SERCH: 30 ;SEARCH COMMAND
WRCHK: 50 ;WRITE CHECK DATA
WRCHDT: 52 ;WRITE CHECK HEADER AND DATA
WRIDAT: 60 ;WRITE DATA
WRIFOR: 62 ;WRITE HEADER AND DATA (FORMAT)
READAT: 70 ;READ DATA
REFOR: 72 ;READ HEADER AND DATA
SEECOM: 4 ;SEEK COMMAND
OFSETC: 14 ;OFFSET COMMAND
RETCL: 16 ;RETURNTO CENTERLINE
PKACK: 22 ;PACK ACKNOWLEDGE
READIN: 20 ;READ IN
ILLEGL: .WORD ;COMPUTED ILLEGAL FUNCTION

;*DATA BUFFERS FOR READ WRITE

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

;*TABLE FOR ATTENTION BITS
;*ATTENTION TABLE
ATABLE: .BYTE 1,2,4,10,20,40,100,200

```

```

1373
1374
1375          .SBTTL      ***DIAGNOSTIC CODE***
1376          .SBTTL
1377
1378          .SBTTL      SETUP TESTS
1379
1380 004230 012737 177777 002002 BEGIN2: MOV    #-1, @#SELECT ;SELECT UNIT
1381 004236 000402          BR      START
1382 004240 005037 002002 BEGIN:  CLR    @#SELECT ;DO NOT SELECT UNIT
1383                                     ;NORMAL RUN
1384
1385 004244          START:
1386 004244 000005          BR      RESET
1387
1388
1389
1390 004466 012737 000000 177776 STARTA: MOV    #0, PS ;SET PROCESSOR STATUS TO 0
1391 004474 012777 054132 175124 MOV    @RPVECT, @RPVEC ;THIS IS FOR UNTIMELY DRIVE INTERRUPTS
1392 004502 004737 055172          JSR    PC, @#STKINT ;INITIALIZE THE TTY KEYBOARD
1393 004506 005737 002034          TST    @#FIRST ;IS THIS FIRST TIME ROUND ?
1394 004512 001001          BNE    1$ ;SKIP HEADER IF NOT
1395 004514 000402          BR      2$ ;DO HEADER IF SO
1396
1397 004516 000137 005326          1$:  JMP    @#SND1 ;SKIP OVERALL PROGRAM HEADER
1398 004522          2$:
1399
1400
1401
1402 005326 012737 177777 002034 SND1:  MOV    #-1, @#FIRST ;NEXT TIME DO NOT GIVE HEADER
1403
1404
1405 005364 032777 010000 173546 RH70CK: BIT    #SW12, @SWR ;LOOK TO SEE IF USING RH70
1406 005372 001403          BEQ    3$ ;IF SW12 = 0, SKIP NEXT
1407 005374 012737 000001 002040          MOV    #1, @#RH70 ;IF SW12 = 1, CU IS AN RH70
1408
1409 005402 005737 002002          3$:  TST    @#SELECT ;200 START?
1410 005462 104412          RDOCT
1411 005464 042716 177770          BIC    #177770, (SP) ;ONLY KEEP LAST 3 BITS
1412 005470 011637 001774          MOV    (SP), @#UNIT ;SAVE UNIT TO BE TESTED
1413 005474 012637 002004          MOV    (SP)+, @#UNITSL ;SAVE UNIT TO BE TESTED
1414
1415

```

```

1416
1417
1418 005510 012706 001000          MOV    #STACK, SP      ;SET UP STACK POINTER
1419
1420 005522 012737 056442 000030    MOV    #REGSA1, @#EMTVEC; ERROR VECTOR SO THAT
1421                                     ;NO REGISTERS ARE SAVED
1422 005530 012737 005556 000004    MOV    #2$, @#ERRVEC   ;SET UP FOR BUS TIMEOUT
1423 005536 012700 000024          MOV    #24, RO         ;THERE ARE 24 REG TO TEST
1424 005542 012701 001630          MOV    #RH0B, R1      ;R1 NOW HAS ADDR OF ADDR OF FIRST REG.
1425 005546 013102          1$:  MOV    @ (R1)+, R2     ;READ HARDWARE REG.
1426 005550 005300          DEC    RO             ;COUNT DOWN
1427 005552 001375          BNE   1$             ;BRANCH IF 24 NOT DONE
1428 005554 000454          BR    3$             ;BRANCH IF 24 DONE
1429 005556 012737 000006 000004    2$:  MOV    #ERRVEC+2, @#ERRVEC ;RESTORE TRAP CATCHER
1430 005564 022626          CMP    (SP)+, (SP)+   ;CLEAN STACK
1431 005566 016137 177776 001200    MOV    -2(R1), $TMP1  ;STORE FAILING REG ADDR
1432 005574 104015          ERROR 15             ;REGISTER NON EXISTANT
1433 005576 032777 020000 173334    BIT    #SW13, @SWR    ;INHIBIT ERROR PRINTOUT ?
1434 005604 001036          BNE   4$             ;BRANCH IF YES
1435
1436
1437 005672 012746 000204          MOV    #ADDMOD, -(SP) ;GET READY TO TYPE STARTING ADDRESS
1438                                     ;OF "CHANGE OF BASE ADDRESS" ROUTINE
1439 005676 104402          TYPOC
1440 005700 000000          HALT
1441                                     ;FORCE THE RESTART!
1442 005702 000137 041314          4$:  JMP    @#SEOP         ;GO TO END OF PROGRAM ----->
1443
1444 005706 012737 005762 000004    3$:  MOV    #TRP, @#4     ;INITIALIZE VECTOR
1445 005714 005737 001700          TST   @#RH0AE        ;ADDRESS RPBAE (RH11/RH70?)
1446 005720 005237 002040          INC   @#RH70         ;FOUND AN RH70-SET MASK
1447 005760 000417          BR    RTN
1448 005762 022626          TRP: CMP    (SP)+, (SP)+ ;SET UP THE STACK
1449 006020 012737 056432 000030    RTN: MOV    #SERR0R, @#EMTVEC; RESTORE ERROR VECTOR
1450                                     ;SO THAT REGISTERS ARE SAVED
1451 006026 012737 000006 000004    MOV    #ERRVEC+2, @#ERRVEC ;RESTORE TRAP CATCHER
1452

```

```

1453
1454
1455
1456 006044 012706 001000          MOV    #STACK,SP          ;RESET STACK
1457
1458 006070 013737 010736 006110    MOV    @#PRCS2+12,@#UN
1459 006076 013737 001636 006112    MOV    @#RHCS2,@#UN+2
1460 006104 004537 042266          JSR    RS,@#BITST        ;TEST BITS IN REGISTER
1461 006110 000000          UN:   .WORD 0            ;ONLY THESE BITS TESTED FOR READ/WRITE
1462 006112 000000          .WORD 0            ;ADDRESS OF REG. BEING TESTED
1463 006114 104001          ERROR 1            ;IN CORRECT DATA RECEIVED
1464 006116 000207          RTS    PC           ;RETURN TO BLT3 ROUTINE
1465
1466
1467
1468
1469
1470
1471
1472
1473 006136 013701 001656          MOV    @#RHAS,R1        ;R1 HAS ADDRESS OF RHAS
1474 006142 012711 177777          MOV    #-1,@R1         ;THIS CLEARS RHAS (SURPRISED!)
1475 006146 011137 001126          MOV    @R1,@#SBDDAT    ;TEST DATA
1476 006152 105737 001126          TSTB  @#SBDDAT
1477 006160 005037 001124          CLR   @#SGDDAT         ;GOOD DATA
1478 006164 010137 042264          MOV    R1,@#REGADR     ;FAILING REG. RHAS
1479 006170 104001          ERROR 1            ;RHAS DOES NOT CLEAR
1480                          ;BY WRITING ONES IN

```

# K03

CZRJGB0, RPO4/5/6 DSKLS CTRLRI  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 34  
T3 PARTIAL TEST OF RHAS FOR UNIT NUMBERS PRESENT

SEQ 0036

```

1481
1482
1483
1484 006202 000005          RESET          ; START WITH AN INIT
1485 006212 004737 055172 JSR          PC, @STKINT ; INITILIZE TTY KEYBOARD
1486 006216 032777 020000 BIT          #SW13, @SWR  ; INHIBIT ERROR TYPEOUT?
1487 006224 001147          BNE          4$          ; BRANCH IF YES
1488
1489
1490 006544 013701 001656   4$:  MOV          @RHAS, R1    ; LOAD R1 WITH ADDR. OF RHAS
1491 006550 013702 001636   MOV          @RHCS2, R2   ; LOAD R2 WITH ADDR. OF RHCS2
1492 006554 005012          CLR          @R2          ; CLEAR RHCS2 (ADDRESS UNIT #0)
1493 006556 012700 000010   MOV          #8, R0       ; INITIALIZE DRIVE COUNTER
1494 006562 013704 001642   MOV          @RHER1, R4   ; LOAD R4 WITH ADDR. OF RHER1
1495 006566 012714 177777   1$:  MOV          #-1, @R4  ; MOVE ERRORS INTO RHER1 OF UNIT ADDRESSED
1496 006572 005212          INC          @R2          ; INCREMENT UNIT NO. (RHCS2)
1497 006574 005300          DEC          R0           ; COUNT DOWN DRIVE COUNTER
1498 006576 001373          BNE          1$          ; TEST AND DO NEXT UNIT IF 8 NOT DONE
1499
1500 006600 111137 002020   MOVB         @R1, @TOTALAT ; SAVE ALL RESULTING ATTENTION BITS
1501                                ; (USED IN DRIVE CLEAR TEST)
1502 006604 105037 002021   CLRB         @TOTALAT+1  ; CLEAR UPPER BYTE
1503 006610 105711          TSTB         @R1          ; TEST RHAS FOR ANY DRIVES PRESENT
1504 006612 001402          BEQ          2$          ; NONE RESPONDING - TYPE THE MESSAGE
1505 006614 000137 007166   JMP          XE2          ; SOME THERE - GO FILL "UNITS" TABLE
1506
1507 006620 032777 020000 172312 2$:  BIT          #SW13, @SWR  ; INHIBIT ERROR TYPE OUT?
1508 006626 001402          BEQ          3$          ; TYPE "NO DRIVES" MESSAGE IF NO
1509 006630 000137 007524   JMP          SELTST       ; CHECK FOR SELECTED UNIT START AND LOAD
1510                                ; "UNITS" TABLE WITH DESIRED DRIVE IF SO
1511
1512 006634          3$:
1513
1514 007162 000137 041314   JMP          @SEOP        ; GO OUT----->
1515
1516
1517                                ; *SET UP DRIVES PRESENT TABLE
1518 007166   XE2:
1519
1520 007166 012700 000010   2$:  MOV          #8, R0       ; LOAD "UNITS" TABLE COUNTER
1521 007172 012703 001754   MOV          #UNITS, R3   ; LOAD "UNITS" TABLE POINTER
1522 007176 012723 177777   3$:  MOV          #-1, (R3)+  ; PRESET 1ST TABLE BLOCK TO ALL ONES
1523 007202 005300          DEC          R0           ; COUNT DOWN
1524 007204 001374          BNE          3$          ; PRESET NEXT BLOCK IF 8 NOT DONE
1525
1526 007206 012703 001754   10$: MOV          #UNITS, R3  ; RELOAD THE TABLE POINTER
1527 007212 005005          CLR          R5           ; INITIALIZE UNIT NO. TO 0
1528 007214 005037 001776   CLR          @NOUNIT     ; NO. OF UNITS PRESENT
1529 007220 012700 000010   MOV          #8, R0       ; RELOAD THE TABLE COUNTER
1530 007224 011137 001176   MOV          @R1, @STMPD  ; ADDR OF RHAS INTO TEMPORARY STORAGE
1531 007230 006037 001176   4$:  ROR          @STMPD     ; SET CARRY IF 0 BIT = 1 (UNIT ATTEN.)
1532 007234 103120          BCC          5$          ; CHECK NEXT UNIT IF ONE NOT IN BIT 0
1533
1534 007236 010577 172374   11$: MOV          R5, @RHCS2   ; INSERT UNIT NO. INTO RHCS2 UNIT ADDR.
1535 007242 022777 024020 172414   CMP          #24020, @RHDT ; READ RHDT - IS IT A DUAL PORT RPO4 ?
1536 007250 001503          BEQ          6$          ; YES...TYPE THE UNIT NO.

```

```

1537 007252 022777 020020 172404    CMP    #20020, @RHDT    ; READ RHDT - IS IT A SINGLE PORT RP04 ?
1538 007260 001477                BEQ    6$              ; YES...TYPE THE UNIT NO.
1539
1540
1541
1542 007262 022777 024021 172374    CMP    #24021, @RHDT    ; DUAL PORT RP05 ?
1543 007270 001473                BEQ    6$              ; TYPE UNIT NO. IF SO
1544 007272 022777 020021 172364    CMP    #20021, @RHDT    ; SINGLE PORT RP05 ?
1545 007300 001467                BEQ    6$              ; TYPE NO. IF SO
1546
1547 007302 022777 024022 172354    CMP    #24022, @RHDT    ; READ RHDT - IS IT A DUAL PORT RP06 ?
1548 007310 001463                BEQ    6$              ; YES...TYPE THE UNIT NO.
1549 007312 022777 020022 172344    CMP    #20022, @RHDT    ; READ RHDT - IS IT A SINGLE PORT RP06 ?
1550 007320 001457                BEQ    6$              ; YES...TYPE THE UNIT NO.
1551
1552
1553
1554
1555
1556
1557
1558 007350 010546                MOV    R5, -(SP)       ; PUT THE UNIT NUMBER ON STACK
1559 007352 104405                TYPDS                ; TYPE IT
1560 007376 017746 172262        MOV    @RHDT, -(SP)    ; PUT RHDT ON THE STACK
1561 007402 104402                TYPDS                ; TYPE IT
1562 007456 000407                BR     5$              ; UNIT NOT AN RP04/RP05/RP06 SO TEST NEXT ONE
1563
1564 007460 010523                6$:  MOV    R5, (R3)+     ; LOAD TABLE POSITION AND INCR IT
1565 007462 104401 001223        TYPE    $CRLF          ; CRLF
1566 007466 010546                MOV    R5, -(SP)       ; PUT UNIT NO. ON THE STACK
1567 007470 104405                TYPDS                ; TYPE THE UNIT NO.
1568 007472 005237 001776        INC    @#NOUNIT        ; INCR THE TOTAL NO. OF UNITS
1569
1570 007476 005205                5$:  INC    R5           ; 'RHCS2' UNIT ADDRESS
1571 007500 005300                DEC    R0              ; DRIVE COUNTER DOWN ONE
1572 007502 001252                BNE   4$              ; TEST AND DO NEXT UNIT IF B NOT DONE
1573
1574 007504 013737 001754 001774 12$:  MOV    @#UNITS, @#UNIT ; SET UNIT NO. TO FIRST ONE FOUND/OR 0
1575 007512 013737 001776 002000        MOV    @#NOUNIT, @#NUNIT ; SAVE NO. OF UNITS
1576 007520 005337 002000                DEC    @#NUNIT        ; IF NUNIT = 0 THEN ONLY ONE UNIT
1577
1578
1579 007524 005737 002002 001774  SELTST: TST    @#SELECT    ; STARTING ADDRESS 200 ?
1580 007532 013737 002004                MOV    @#UNITS, @#UNIT ; CHANGE UNIT NUMBER TO SELECTED ONE
1581

```

```

1582
1583
1584
1585 007564 004737 042614 JSR PC,@CLDISK ;FILL UNIT NO.
1586 007570 005037 002016 CLR @ATTENT ;CLEAR
1587
1588 ;*TEST FOR UNIT #0
1589
1590 007574 005737 001774 TST @UNIT ;IS UNIT #0 NEXT IN THE UNITS TABLE ?
1591 007600 001022 BNE 10$ ;IF NOT, TEST THIS UNIT
1592 007602 012700 000041 MOV #41,RO ;IF SO, CHECK THE LOAD MEDIA LOCATION
1593 007606 122710 000011 CMPB #11,(RO) ;WAS IT AN RPO4/5/6 ?
1594 007612 001015 BNE 10$ ;NO...GO AHEAD WITH TESTING UNIT #0
1595 007614 005737 002002 TST @SELECT ;WAS UNIT #0 SELECTED ?
1596 ;(IE. WAS IT A 210 START ?)
1597 007620 001012 BNE 10$ ;IF SO...TEST IT
1598
1599 ;*INCREMENT THE UNITS TABLE TO NEXT DRIVE (IF ANY)
1600 ;*& DECREMENT THE "NOUNITS" PRESENT (TO BE TESTED)
1601
1602 007622 012700 001754 MOV #UNITS,RO ;LOAD THE UNITS TABLE POINTER
1603 007626 005720 TST (RO)+ ;SELECT THE NEXT UNIT IN THE TABLE
1604 ;(DOUBLE INCREMENT THE POINTER, RO)
1605 007630 022710 177777 CMP #-1,(RO) ;IS THERE ANOTHER TABLE ENTRY PRESENT ?
1606 007634 001404 BEQ 10$ ;IF NOT (LOC = -1)...MUST USE UNIT #0
1607 007636 011037 001774 MOV (RO),@UNIT ;SET UP TO BE THE UNIT UNDER TEST
1608 007642 005337 001776 DEC @NOUNITS ;DECREMENT BECAUSE UNIT # 0 WON'T BE TESTED
1609 007646 013700 001774 10$: MOV @UNIT,RO ;RO CONTAINS UNIT NO.
1610
1611
1612 ;*SET UP THE PROPER DEVICE TYPE FLAG
1613
1614 007652 010077 171760 MOV RO,@RHCS2 ;SET UP UNIT ADDRESS
1615 007656 005037 002036 CLR @RPO6 ;CLEAR RPO6 DEVICE TYPE FLAG
1616 007662 022777 024022 171774 CMP #24022,@RHDT ;DUAL PORT RPO6 ?
1617 007670 001405 BEQ 2$ ;YES...SET THE FLAG
1618 007672 022777 020022 171764 CMP #20022,@RHDT ;SINGLE PORT RPO6 ?
1619 007700 001401 BEQ 2$ ;YES..SET FLAG
1620 007702 000403 BR 3$ ;NO...DON'T SET RPO6 FLAG
1621 007704 012737 177777 002036 2$: MOV #-1,@RPO6 ;SET IT
1622
1623 007712 3$: ;ASSUME THE NEXT UNIT IS AN RPO4
1624
1625 007712 116037 004220 002016 MOVB ATABLE(RO),@ATTENT ;SET APPROPRIATE ATTENTION BIT
1626
1627 007756 013746 001774 MOV @UNIT,-(SP) ;UNIT NO. TO STACK
1628 007762 104405 TYPDS ;TYPE DRIVE NO.
1629
1630 010054 022777 024020 171602 CMP #24020,@RHDT ;DUAL PORT RPO4 ?
1631 010062 001425 BEQ 4$ ;TYPE ASCII MSG OUT
1632 010064 022777 020020 171572 CMP #20020,@RHDT ;SINGLE PORT RPO4 ?
1633 010072 001421 BEQ 4$ ;TYPE THE MESSAGE
1634
1635 010074 022777 024021 171562 CMP #24021,@RHDT ;DUAL PORT RPO5 ?
1636 010102 001453 BEQ 5$ ;TYPE MSG
1637 010104 022777 020021 171552 CMP #20021,@RHDT ;SINGLE PORT RPO5 ?

```





CZRJGBD RPO4/5/6 DSKLS CTRLR1  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046)  
TS

08-NOV-77 08:54 PAGE 38  
TYPE SERIAL NUMBER AND DRIVE TYPE

SEQ 0040

1662			
1663	010322	004737	042614
1664	010326	032713	010000
1665	010550	032713	010000
1666	010554	001375	

1\$:

JSR	PC,@#CLDISK	:GIVE INITILIZE
BIT	#MOL,@R3	:CHECK MOL IN RHDS1
BIT	#MOL,@R3	:CHECK MOL IN RHDS1
BNE	IS	:BRANCH IF MOL IS HIGH

C04

CZRJGBD.RP04/5/6 DSKLS CTRLR1  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046)  
T6

08-NOV-77 08:54 PAGE 39  
CHECK MOL TO BE LOW

SEQ 0041

1667  
1668  
1669  
1670  
1671  
1672  
1673  
1674  
1675  
1676  
1677  
1678

010700 004737 042614

.SBTTL REGISTER TESTS  
JSR PC,2#CLDISK ;GIVE INITIALIZE



CZRJGBO.RP04/5/6 DSKLS\_CTRLR1  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 41  
T11 RHCS1 - BIT # 13 - MCPE

E04

SEQ 0043

1724  
1725  
1726  
1727  
1728  
1729  
1730  
1731  
1732  
1733

F04

CZRJGBD.RP04/5/6 DSKLS CTRLRI  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046)  
T13

08-NOV-77 08:54 PAGE 42  
RHBA - UNIBUS ADDRESS REGISTER

SEQ 0044

1734  
1735  
1736  
1737

```

1738
1739
1740
1741
1742 011346 004737 042614 JSR PC, @#CLDISK ; SET UNIT NUMBER AND INIT
1743 011352 013700 001660 MOV @#RHMR, R0 ; R0 HAS MAINTENANCE REG. ADR.
1744 011356 012701 000001 MOV #1, R1 ; R1 HAS DATA
1745 011362 012702 000005 MOV #5, R2 ; R2 HAS COUNT OF NUMBER OF BITS
1746 011366 012710 000001 1$: MOV #DMO, @RO ; SET DIAGNOSTIC MODE BIT
1747 011372 050110 BIS R1, @RO ; SET DATA IN RHMR
1748 011374 010146 MOV R1, -(SP) ; SAVE DATA FOR COMPARES
1749 011376 052716 000401 BIS #DMO!400, (SP) ; INCLUDE BIT 0
1750 011402 011637 001124 MOV (SP), @#SGDDAT ; SAVE FOR ERROR PRINTOUT
1751 011406 022610 CMP (SP)+, @RO ; COMPARE DATA
1752 011410 001405 BEQ 2$ ; BRANCH IF GOOD
1753 011412 011037 001126 MOV @RO, $BDDAT ; BAD DATA
1754 011416 010037 042264 MOV RO, @#REGADR ; FAILING REG. ADR.
1755 011422 104001 ERROR 1 ; MAINTENANCE REGISTER
1756 ; FAILED TO SET INDICATED
1757 ; BITS
1758 011424 000241 2$: CLC ; CLEAR CARRY
1759 011426 006101 ROL R1 ; GET NEXT DATA
1760 011430 052701 000400 BIS #400, R1 ; SET UNUSED BITS
1761 011434 042701 001000 BIC #BIT09, R1 ; CLEAR READ ONLY BIT
1762 011440 005302 DEC R2 ; COUNT
1763 011442 001351 BNE 1$ ; BRANCH IF 5 BITS NOT DONE
1764
1765 ; *NOW FLOAT A 0
1766
1767
1768 011444 012701 000435 MOV #435, R1 ; R1 HAS DATA
1769 011450 012702 000005 MOV #5, R2 ; R2 HAS COUNT BITS
1770 011454 012710 000001 3$: MOV #DMO, @RO ; SET DIAGNOSTIC MODE BITS
1771 011460 050110 BIS R1, @RO ; SET DATA IN RHMR
1772 011462 020110 CMP R1, @RO ; COMPARE DATA
1773 011464 001407 BEQ 4$ ; BRANCH IF GOOD
1774 011466 010137 001124 MOV R1, @#SGDDAT ; GOOD DATA
1775 011472 011037 001126 MOV @RO, @#BDDAT ; BAD DATA
1776 011476 010037 042264 MOV RO, @#REGADR ; FAILING REG. ADR. RHMR
1777 011502 104001 ERROR 1 ; MAINTENANCE REGISTER
1778 ; DOES NOT ALLOW WRITING
1779 ; ZEROS
1780 011504 000261 4$: SEC ; SET CARRY
1781 011506 006101 ROL R1 ; GET NEXT DATA
1782 011510 042701 001340 BIC #BIT05!BIT06!BIT07!BIT09, R1 ; CLEAR READ ONLY BIT
1783 011514 052701 000400 BIS #BIT08, R1 ; SET BIT ZEROED BY ROL
1784 011520 005302 DEC R2 ; COUNT IF 5 BITS DONE
1785 011522 001354 BNE 3$ ; BRANCH IF INCOMPLETE
1786
1787

```

H04

CZRJGBO.RP04/S/6 DSKLS CTRLR1  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 44  
T15 RHMR - MAINTENANCE REGISTER

SEQ 0046

11788  
11789  
11790  
11791  
11792  
11793  
11794  
11795  
11796  
11797  
11798  
11799  
11800  
11801  
11802  
11803  
11804  
11805  
11806  
11807  
11808  
11809  
11810  
11811  
11812  
11813  
11814  
11815  
11816  
11817  
11818  
11819  
11820  
11821  
11822  
11823  
11824  
11825  
11826  
11827  
11828  
11829  
11830  
11831  
11832  
11833  
11834  
11835  
11836  
11837  
11838  
11839  
11840  
11841  
11842  
11843  
11844  
11845  
11846  
11847  
11848  
11849  
11850  
11851  
11852  
11853  
11854  
11855  
11856  
11857  
11858  
11859  
11860  
11861  
11862  
11863  
11864  
11865  
11866  
11867  
11868  
11869  
11870  
11871  
11872  
11873  
11874  
11875  
11876  
11877  
11878  
11879  
11880  
11881  
11882  
11883  
11884  
11885  
11886  
11887  
11888  
11889  
11890  
11891  
11892  
11893  
11894  
11895  
11896  
11897  
11898  
11899  
11900  
11901  
11902  
11903  
11904  
11905  
11906  
11907  
11908  
11909  
11910  
11911  
11912  
11913  
11914  
11915  
11916  
11917  
11918  
11919  
11920  
11921  
11922  
11923  
11924  
11925  
11926  
11927  
11928  
11929  
11930  
11931  
11932  
11933  
11934  
11935  
11936  
11937  
11938  
11939  
11940  
11941  
11942  
11943  
11944  
11945  
11946  
11947  
11948  
11949  
11950  
11951  
11952  
11953  
11954  
11955  
11956  
11957  
11958  
11959  
11960  
11961  
11962  
11963  
11964  
11965  
11966  
11967  
11968  
11969  
11970  
11971  
11972  
11973  
11974  
11975  
11976  
11977  
11978  
11979  
11980  
11981  
11982  
11983  
11984  
11985  
11986  
11987  
11988  
11989  
11990  
11991  
11992  
11993  
11994  
11995  
11996  
11997  
11998  
11999  
12000



104

CZRJGBD.RP04/5/6 DSKLS CTRLR1  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 45  
T17 RHER2 - ERROR REGISTER #2

SEQ 0047

1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1810

J04

CZRJGBD, RPO4/5/6 DSKLS CTRLR1  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 46  
T21 RHCA - DESIRED CYLINDER REGISTER

SEG 0048

1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819  
1820  
1821  
1822  
1823

\*\*\*OF THE TWENTY REGISTERS (4 IN RH11, 16 IN RPO4) ONLY 12 ARE  
\*\*\*CHECKED IN THE ABOVE TESTS  
\*\*\*TWO ARE ALREADY TESTED (SERIAL NO. AND DRIVE TYPE)  
\*\*\*THE OTHER 7 WHICH ARE RHDS1, RHLA, RHCC, RHEC1, RHEC1, RHEC2  
\*\*\*ARE READ ONLY REGISTERS. ONE OR ZERO CANNOT BE WRITTEN

```

1824
1825
1826 012012 012706 001000          MOV    #STACK, SP          ;RESET STACK
1827
1828 012024 004737 042614          JSR    PC, @#CLDISK       ;HERE IT IS USED TO SETUP HARDWARE/
1829                                ;CPU REGISTER CORRESPONDENCE
1830                                ;R1=RHCS1
1831                                ;R2=RHCS2
1832                                ;R3=RHDS1
1833                                ;R4=RHER1
1834 012030 005037 002006          CLR    @#ERFLG$          ;CLEAR ERROR FLAG
1835
1836                                ;*SIMULATED DISK AREA WILL BE USED AS A TEMPORARY
1837                                ;*STORAGE TABLE FOR DRIVES PRESENT DETERMINED FROM 'NED' = 0 IN RHCS2
1838
1839 012034 012701 051416          MOV    #DISK, R1          ;LOAD TABLE POINTER
1840 012040 012700 000010          MOV    #8, R0             ;LOAD TABLE LOCATION COUNTER
1841 012044 012721 177777          1$:   MOV    #-1, (R1)+    ;FILL 8 LOCATIONS WITH -1
1842 012050 005300                   DEC    R0                 ;COUNT DOWN ONE LOCATION
1843 012052 001374                   BNE    1$                 ;BRANCH IF 8 NOT DONE
1844
1845 012054 005012                   CLR    @R2                ;SELECT UNIT NO.0 (U2!U1!U0=0)
1846 012056 012700 000010          MOV    #8, R0             ;RELOAD TABLE LOCATION COUNTER
1847 012062 012701 051416          MOV    #DISK, R1          ;RELOAD THE TABLE POINTER
1848 012066 005714                   2$:   TST    @R4            ;READ A DRIVE REGISTER (RHER1)
1849 012070 032712 010000          BIT    #NED, @R2          ;NON EXISTENT DRIVE BIT = 0 ?
1850 012074 001415                   BEQ    3$                 ;YES...DRIVE PRESENT, CHECK THE TYPE
1851 012076 005300                   7$:   DEC    R0             ;NO...DECREMENT DRIVE COUNT
1852 012100 001454                   BEQ    4$                 ;CHECK RESULTS IF 8 DRIVES DONE
1853
1854 012102 011246                   10$:  MOV    @R2, -(SP)        ;PUT RHCS2 ON THE STACK
1855 012104 042716 177770          BIC    #1C7, (SP)        ;MASK ALL BUT THE UNIT NUMBER
1856 012110 005216                   INC    (SP)               ;INCREMENT THE UNIT NUMBER
1857 012112 013703 001640          MOV    @#RHCS1, R3        ;GET RHCS1 ADDRESS
1858 012116 005203                   INC    R3                 ;ADDRESS UPPER BYTE OF RHCS1
1859 012120 112713 000100          MOVB  #100, @R3           ;SET 'TRE' IN RHCS1
1860                                ;WITHOUT ADDRESSING DRIVE
1861 012124 012612                   MOV    (SP)+, @R2        ;RHCS2 HAS THE INCREMENTED UNIT
1862                                ;WITH 'NED' CLEARED
1863 012126 000757                   BR     2$                 ;TEST FOR NEXT DRIVE
1864
1865
1866                                ;*CHECK THE UNIT TYPE AND BUILD 'NED' DERIVED UNITS TABLE
1867
1868
1869 012130 022777 024020 167526    3$:   CMP    #24020, @RHDT     ;IS THIS A DUAL PORT RPO4 ?
1870 012136 001425                   BEQ    8$                 ;ENTER IN TABLE IF SO
1871 012140 022777 020020 167516    CMP    #20020, @RHDT     ;IS THIS A SINGLE PORT RPO4 ?
1872 012146 001421                   BEQ    8$                 ;ENTER IN TABLE IF SO
1873
1874 012150 022777 024022 167506    CMP    #24022, @RHDT     ;IS THIS A DUAL PORT RPO6 ?
1875 012156 001415                   BEQ    8$                 ;ENTER IN TABLE IF SO
1876 012160 022777 020022 167476    CMP    #20022, @RHDT     ;IS THIS A SINGLE PORT RPO6 ?
1877 012166 001411                   BEQ    8$                 ;ENTER IN TABLE IF SO
1878
1879 012170 022777 024021 167466    CMP    #24021, @RHDT     ;IS THIS A DUAL PORT RPO5 ?

```

```

1880 012176 001405          BEQ      8$          ;ENTER IN TABLE IF SO
1881 012200 022777 020021 167456  CMP      #20021, @RHDT ;IS THIS A SINGLE PORT RPOS ?
1882 012206 001401          BEQ      8$          ;ENTER IN TABLE IF SO
1883
1884 012210 000732          BR       7$          ;NO RPO4 FOUND SO CHECK NEXT UNIT
1885
1886 012212 012746 000010 8$:      MOV      #8, -(SP) ;LOAD MAX NO. OF DRIVES
1887 012216 160016          SUB      RD, (SP)   ;(SP) NOW HAS THE PRESENT DRIVE NO.
1888 012220 012621          MOV      (SP)+, (R1)+ ;LOAD TABLE, INCR TABLE LOCATION &
1889                                     ;RESTORE THE STACK TO WHERE IT WAS
1890 012222 005300          DEC      RD         ;DECREMENT THE DRIVE COUNT
1891 012224 001402          BEQ      4$          ;CHECK RESULTS IF 8 UNITS CHECKED
1892 012226 005212          INC      @R2        ;SELECT NEXT UNIT
1893 012230 000716          BR       2$          ;GO TEST IT
1894
1895
1896                                     ;*COMPARE 'NED' DERIVED UNITS TABLE WITH THAT DERIVED USING RHAS IN T4
1897
1898 012232 004037 043510 4$:      JSR      RD, @#COMPAR ;COMPARE RESULTS
1899 012236 001754          UNITS   ;RHER1/RHAS DERIVED DATA
1900 012240 051416          DISK    ;'NED' TEST DATA
1901 012242 000010          8.     ;NO. OF WORDS TO COMPARE
1902 012244 012252          5$     ;RETURN FOR ERROR HEADER
1903 012246 012300          6$     ;RETURN FOR ERROR DATA
1904 012250 012416          13$    ;RETURN FOR GOOD COMPARISON (NEXT TEST)
1905
1906
1907
1908                                     ;*SPECIAL 'NED'/'RHAS' TABLE TYPE OUT ROUTINE (BYPASSES .SERRTYP AND
1909                                     ;*HENCE IGNORES INHIBIT ERROR TYPEOUT SWITCH)
1910
1911 012252 104022          5$:      ERROR   22
1912 012254 012703 000010  MOV      #8, R3     ;LENGTH OF BOTH UNIT TABLES
1913 012260 012701 001754  MOV      #UNITS, R1 ;ADDRESS OF RHAS/RHER1 UNITS TABLE
1914 012264 012702 051416  MOV      #DISK, R2  ;ADDRESS OF 'NED' RHCS2 UNITS TABLE
1915 012270 012137 001124 14$:     MOV      (R1)+, @#SGDDAT ;LOAD RHAS UNIT NO. INTO "SGDDAT" AND
1916                                     ;INCREMENT THE TABLE LOCATION
1917 012274 012237 001126  MOV      (R2)+, @#SBDDAT ;LOAD 'NED' UNIT NO. INTO "SBDDAT"
1918                                     ;& INCR TABLE LOCATION
1919
1920 012300 032777 020000 166632 6$:      BIT      #SW13, @SWR ;INHIBIT ERROR TYPE OUTS ?
1921 012306 001043          BNE     13$         ;YES...EXIT
1922 012310 022737 177777 001124  CMP      #-1, @#SGDDAT ;DOES RHAS UNIT TABLE LOCATION = -1 ?
1923 012316 001413          BEQ     11$         ;YES...DON'T TYPE IT - CHECK 'NED' TABLE
1924 012320 104401 062716  TYPE     ,SPACE8    ;NO...TAB OVER PC COLUMN
1925 012324 104401 062716  TYPE     ,SPACE8    ;TAB OVER THE TEST NO. COLUMN
1926 012340 104401 062724  TYPE     ,SPACE2    ;SPACE OVER TO THE NEXT COLUMN
1927 012344 000406          BR      12$        ;CHECK THE 'NED' UNIT TABLE
1928
1929 012346 104401 062716 11$:     TYPE     ,SPACE8    ;TAB OVER THE PC COLUMN
1930 012352 104401 062716  TYPE     ,SPACE8    ;TAB OVER THE TEST NO. COLUMN
1931 012356 104401 062716  TYPE     ,SPACE8    ;TAB OVER THE RHAS UNIT COLUMN
1932
1933 012362 022737 177777 001126 12$:     CMP      #-1, @#SBDDAT ;DOES 'NED' UNIT TABLE LOCATION = -1 ?
1934 012370 001404          BEQ     9$          ;YES...DON'T TYPE IT
1935

```

M04

CZRJGBO, RPO4/5/6 DSKLS CTRLRI  
 CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 49  
 T23 CONTROL AND STATUS 2 (RMCS 2) - 'NED'

SEQ 0051

1936	012402	104401	001223	9S:	TYPE	\$CRLF			
1937	012406	005303			DEC	R3			:FOR THE NEXT LINE IN BOTH TABLES
1938	012410	001327			BNE	14S			:COUNT DOWN 2 TABLES LOCATION COUNTER
1939	012412	062706	000014		ADD	#14,SP			:IF NOT = 0 TYPE OUT NEXT 2 LOCATIONS
1940									:ADJUST STACK FOR NO "POP" & RTS FROM "COMPAR"
1941	012416			13S:					
1942									
1943									
1944									
1945									

```

;*IN THE ABOVE TEST BITS 0,1,2, AND BIT 12 ARE TESTED
;*IF THE "DRIVES PRESENT" TYPE OUT DOES NOT AGREE WITH WHAT WAS
;*FOUND USING RHER1 & RHAS, THEN THE ERROR IS IN THE LOGIC
;*FOR BIT12(NED), OR UNIT SELECT(BIT 0 TO 2), OR RHER1, OR RHAS
;*IT IS NOT POSSIBLE BY PROGRAM TO CHECK IF A NON-EXISTENT
;*DRIVE IS REALLY STANDING THERE OR NOT
;*MANUALLY LOAD LOCATION "ERUNIT" WITH A UNIT NUMBER
;*AND RESTART AT LOCATION "ERSTAR" THIS WILL LOOP FOR
;*EVER DOING EXACTLY AS TEST ON THAT ONE UNIT
;*TO GET BACK TO MAIN DIAGNOSTIC HIT HALT SWITCH AND
;*RESTART PROGRAM IN NORMAL MANNER
  
```

```

1965
1966
1967
1968
1969
1970 012420 004737 042614 JSR PC,ARCLDISK ;SET REGISTERS AND CLEAR
1971 012424 005037 002006 CLR ARERFLGS ;CLEAR ANY ERRORS
1972
1973 ;*FILL ALL POSSIBLE BITS WITH ONES
1974
1975 012430 012777 177777 167172 MOV #177777,ARHDB ;BUS ADDRESS REGISTER GETS 177777
1976 012436 012777 177777 167166 MOV #177777,ARHWC ;WORD COUNT REGISTER GETS 177777
1977 012444 012777 177777 167162 MOV #177777,ARHBA ;BUS ADDRESS REGISTER GETS 177777
1978 012452 052777 157010 167156 BIS #157010,ARHCS2 ;CONTROL AND STATUS 2 GETS 157010
1979 012460 012777 001476 167152 MOV #1476,ARHCS1 ;CONTROL AND STATUS REGISTER GETS 1476
1980 012466 012777 177777 167146 MOV #177777,ARHER1 ;ERROR REGISTER1 GETS 177777
1981 012474 012777 017437 167142 MOV #17437,ARHDS1 ;DESIRED SECTOR TRACK
1982 012502 012777 177777 167136 MOV #177777,ARHER2 ;ERROR REGISTER 2
1983 012510 012777 016277 167132 MOV #16277,ARHOF ;OFFSET REGISTER
1984 012516 012777 177777 167126 MOV #177777,ARHCA ;DESIRED CYLINDER
1985 012524 012777 177777 167122 MOV #177777,ARHER3 ;ERROR REGISTER 3
1986 012532 012777 000001 167120 MOV #DMD,ARHMR ;MAINTENANCE REGISTER
1987 012540 012777 177777 167112 MOV #177777,ARHMR ;MAINTENANCE REGISTER
1988
1989 012546 052712 000040 BIS #CLR,AR2 ;CLEAR ALL POSSIBLE BITS
1990 012552 013712 001774 MOV ARUNIT,AR2 ;REINSTATE UNIT NO.
1991 012556 012700 001630 MOV ARHDB,AR0 ;AR0 CONTAINS ADDR. OF ADDR. OF REG.
1992
1993 ;*DATA BUFFER REGISTER
1994
1995 012562 012737 177777 001124 MOV #177777,ARSGDDAT ;GOOD DATA FOR ERROR
1996 012570 011037 042264 MOV AR0,AREGADR ;REGISTER ADDRESS
1997 012574 013037 001126 MOV AR(R0)+,ARSBDDAT ;TEST DATA
1998 012600 023737 001124 001126 CMP ARSGDDAT,ARSBDDAT ;COMPARE GOOD WITH TEST DATA
1999 012606 001401 BEQ 25 ;BRANCH IF GOOD
2000 012610 104001 ERROR 1 ;RHDB DID NOT HAVE ALL ONES
2001 ;AFTER A CLR IN RHCS2
2002
2003 012612 052712 000040 25: BIS #CLR,AR2 ;SET CLEAR AGAIN BECAUSE
2004 ;READING RHDB AFTER CLEARING WILL
2005 ;SET DLT SC AND TRE
2006 012616 013712 001774 MOV ARUNIT,AR2 ;REINSTATE UNIT NO.
2007
2008 ;*WORD COUNT REGISTER
2009
2010
2011 ;*BUS ADDRESS REGISTER
2012
2013
2014 ;*CONTROL AND STATUS 2 REGISTER
2015
2016
2017 ;*CONTROL AND STATUS 1 REGISTER
2018
2019
2020 ;*ERROR 1 REGISTER

```

2021  
2022  
2023  
2024  
2025  
2026  
2027  
2028  
2029  
2030  
2031  
2032  
2033  
2034  
2035  
2036  
2037  
2038  
2039  
2040  
2041  
2042  
2043  
2044  
2045  
2046  
2047  
2048  
2049  
2050  
2051  
2052  
2053  
2054  
2055  
2056  
2057  
2058  
2059  
2060  
2061  
2062  
2063  
2064  
2065  
2066  
2067  
2068  
2069  
2070  
2071  
2072  
2073  
2074  
2075  
2076

; \*DESIRED SECTOR/TRACK REGISTER

; \*ERROR 2 REGISTER

; \*OFFSET REGISTER

; \*DESIRED CYLINDER ADDRESS REGISTER

; \*ERROR 3 REGISTER

; \*ATTENTION SUMMARY REGISTER

```

2040 013164 013037 001126 15$: MOV      2(RO)+, 2#SBDDAT      ; GET RHAS CONTENTS
2041 013170 012737 000000 001124 MOV      #0, 2#SGDDAT      ; GOOD DATA FOR ERROR TYPE OUT
2042 013176 123737 001124 001126 CMPB    2#SGDDAT, 2#SBDDAT ; COMPARE FOR RHAS
2043 013204 001402 BEQ     16$                ; BRANCH IF GOOD
2044 013206 004737 013430 JSR     PC, 2#ERCS2C      ; JUMP TO ERROR FOR CLR (BIT 5)
                                           ; IN RHCS2

```

; \*MAINTAINABILITY REGISTER

; \*DRIVE STATUS REGISTER

```

2052 013240 012737 000600 001124 17$: MOV      #600, 2#SGDDAT    ; GOOD DATA FOR ERROR TYPEOUT
2053 013246 013046 MOV      2(RO)+, -(SP)    ; GET RHDS1
2054 013250 011637 001126 MOV      (SP), 2#SBDDAT  ; TEST DATA
2055 013254 042716 001100 BIC     #VV!PROG, (SP)   ; CLEAR VV AND PROG
2056 013260 022726 000600 CMP      #600, (SP)+     ; COMPARE DATA
2057 013264 001402 BEQ     20$                ; BRANCH IF GOOD
2058 013266 004737 013430 JSR     PC, 2#ERCS2C      ; JUMP TO ERROR FOR CLR (BIT 5)
                                           ; IN RHCS2

```

; \*DRIVE TYPE

; \*SERIAL NUMBER REGISTER

; \*ECC1 POSITION

; \*ECC2 PATTERN

; \*LOOK-AHEAD REGISTER

```

2077
2078 013422 005720      24$:  TST      (RO)+      ;AS THE LOOK-AHEAD REG. CANNOT BE PREDICTED
2079                                     ;AFTER AN INIT IT IS NOT CHECKED
2080
2081                                     ;*CURRENT CYLINDER ADDRESS REGISTER
2082
2083 013424 005720      25$:  TST      (RO)+      ;AS THE CURRENT REG. CANNOT BE PREDICTED
2084                                     ;AFTER A INIT IT IS NOT CHECKED
2085
2086 013426
2087
2088
2089 013430 014037 042264  ERCS2C: MOV      -(RO),  2#REGADR ;FAILING REGISTER ADDRESS
2090 013434 104001          ERROR    1          ;CLR (BIT 5) IN RHCS2 DID
2091                                     ;NOT CLEAR APPROPRIATE BITS
2092                                     ;OR CLEARED EXTRA BITS
2093 013436 005720          TST      (RO)+      ;UNDO -(RO) FOR BAD DATA
2094 013440 000207          RTS      PC          ;RETURN TO TEST ABOVE

```





EOS

CZRJGBD.RP04/5/6 DSKLS CTRLR1  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046)  
T25

08-NOV-77 08:54 PAGE 54  
PACK ACKNOWLEDGE COMMAND TEST

SEQ 0056

2151 013604

2\$:

;CONTINUE WITH THE NEXT TEST

# F05

CZRJGBD.RP04/S/6 DSKLS CTRLR1  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 55  
T25 PACK ACKNOWLEDGE COMMAND TEST

SEQ 0057

```

2152
2153
2154
2155 013606 012706 001000      MOV      #STACK,SP      ;RESET STACK
2156 013620 004737 042614      JSR      PC,@#CLDISK    ;INIT AND SET UP GENERAL CPU/DEVICE
2157                                     ;REGISTER CORRESPONDENCE
2158
2159                                     ;*FILL ALL POSSIBLE REGISTER BITS WITH ONES
2160
2161 013624 012777 177777 166000      MOV      #177777,@RHWC  ;WORD COUNT REGISTER GETS 177777
2162 013632 012777 177777 165774      MOV      #177777,@RHBA  ;BUS ADDRESS REGISTER GETS 177777
2163 013640 052777 157010 165770      BIS      #157010,@RHCS2 ;CONTROL AND STATUS 2 GETS 177430
2164 013646 012777 001476 165764      MOV      #1476,@RHCS1  ;CONTROL AND STATUS REGISTER 1 GETS 21476
2165 013654 012777 177777 165760      MOV      #177777,@RHER1 ;ERROR REGISTER1 GETS 177777
2166 013662 012777 017437 165754      MOV      #17437,@RHDS1  ;DESIRED SECTOR TRACK
2167 013670 012777 177777 165750      MOV      #177777,@RHER2 ;ERROR REGISTER 2
2168 013676 012777 016277 165744      MOV      #16277,@RHOF   ;OFFSET REGISTER
2169 013704 012777 000777 165740      MOV      #777,@RHCA    ;DESIRED CYLINDER
2170 013712 012777 177777 165734      MOV      #177777,@RHER3 ;ERROR REGISTER 3
2171 013720 012777 000001 165732      MOV      #0MD,@RHMR    ;MAINTENANCE REGISTER
2172 013726 012777 177777 165724      MOV      #177777,@RHMR ;MAINTENANCE REGISTER
2173
2174                                     ;*BEFORE RESET SAVE REGISTERS IN READ INTO BUFFER
2175 013734 004037 043306      JSR      RO,@#SAVER     ;SAVE
2176 013740 001632                                     ;FROM
2177 013742 003154                                     ;TO
2178 013744 000021                                     ;NUMBER
2179
2180                                     ;*GIVE RESET AND REINSTATE UNIT NUMBER
2181 RESET
2182 013750 004737 055172      JSR      PC,@#STKINT    ;INITIALIZE TK
2183 013754 053777 001774 165654      BIS      @#UNIT,@RHCS2
2184
2185                                     ;*CHANGE ORIGINAL SAVED REGISTERS TO EXPECTED VALUES AFTER RESET
2186 013762 005037 003156      CLR      @#REINTO+2     ;CLEAR SAVED RHBA
2187 013766 013746 001774      MOV      @#UNIT,-(SP)  ;GET UNIT NUMBER FRO SAVED RHCS2
2188 013772 052716 000100      BIS      #IR,(SP)      ;INCLUDE IR
2189 013776 012637 003160      MOV      (SP)+,@#REINTO+4 ;SAVED RHCS2
2190 014002 012737 004276 003162      MOV      #DVA!RDY!76,@#REINTO+6 ;SAVED RHCS1
2191 014010 005037 003164      CLR      @#REINTO+10   ;SAVED RHER1
2192 014014 005037 003170      CLR      @#REINTO+14   ;SAVED RHER2
2193 014020 012737 116000 003172      MOV      #116000,@#REINTO+16 ;SAVED RHOF
2194 014026 005037 003176      CLR      @#REINTO+22   ;SAVED RHER3
2195 014032 105037 003200      CLRB    @#REINTO+24   ;SAVED RHAS
2196 014036 012737 000400 003202      MOV      #400,@#REINTO+26 ;SAVED RHMR
2197
2198                                     ;*CHANGE RHDS1 WITHOUT CHANGING PROG BIT
2199 014044 013746 003204      MOV      @#REINTO+30,-(SP) ;GET RHDS1
2200 014050 042716 176777      BIC      #1CPRG,(SP)   ;CLEAR EVERYTHING EXCEPT PRG
2201 014054 052716 000700      BIS      #700,(SP)    ;SET EXPECTED BITS - 'DPR', 'DRY' & 'VV'
2202
2203 014060 012637 003204      4$: MOV      (SP)+,@#REINTO+30 ;SAVED RHDS1
2204 014064 005037 003212      CLR      @#REINTO+36   ;SAVED RHEC1
2205 014070 005037 003214      CLR      @#REINTO+40   ;SAVED RHEC2
2206
2207                                     ;*AFTER RESET, SAVE REGISTERS FOR COMPARISONS TO BE DONE

```

```

2208 014074 004037 043306 JSR RO, @SAVER ;SAVE
2209 014100 001632 RHW C ;FROM
2210 014102 002110 WRFROM ;TO
2211 014104 000021 17. ;NUMBER
2212
2213 ;*AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
2214 ;*OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
2215 ;*SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
2216 014106 113737 003201 002135 MOVB @REINTO+25, @WRFROM+25; SAVE UPPER RHAS
2217
2218 ;*COMPARE REGISTERS BEFORE RESET WITH REGISTERS AFTER RESET
2219 014114 004037 043510 JSR RO, @COMPAR ;COMPARE
2220 014120 003154 REINTO ;GOOD BUFFER
2221 014122 002110 WRFROM ;TEST BUFFER
2222 014124 000021 17. ;NUMBER
2223 014126 014134 1$ ;RETURN FOR ERROR
2224 014130 014134 1$ ;SAME
2225 014132 014154 2$ ;RETURN FOR GOOD COMPARISON
2226
2227 014134 013705 047620 1$: MOV @ERWORD, R5 ;GETTING READY TO INDEX
2228 014140 060505 ADD R5, R5 ;DOUBLE ERROR WORD
2229 014142 016537 001630 042264 MOV RHW C-2(R5), @REGADR ;FAILING REGISTER ADDRESS
2230 014150 104001 ERROR 1 ;REGISTER CONTENTS AFTER
2231 ;A RESET THAT IS AN
2232 ;UNIBUS INITIALIZE CAUSED
2233 ;AN IMPROPER REGISTER CHANGE
2234 014152 000207 RTS PC ;RETURN TO COMPARISON
2235 014154 2$: ;RETURN TO POINT ON GOOD COMPARISON

```

```

2236
2237
2238
2239
2240 014156 012706 001000      MOV      #STACK, SP      ;RESET STACK
2241 014202 004737 042614      JSR      PC, CLDISK      ;CLEAR DISK AND LOAD R'S
2242
2243 014206 017700 165416      MOV      @RHDB, R0       ;READ FROM EMPTY SILO
2244 014212 013746 001774      MOV      @#UNIT, -(SP)   ;GET UNIT NO. IN
2245 014216 052716 100100      BIS      #DLT!IR, (SP)   ;GET DATA LATE BIT AND IR
2246 014222 004737 042220      JSR      PC, @#PUTREG    ;SAVE REGISTERS
2247 014226 022637 001712      CMP      (SP)+, @#CS2    ;IS DATA LATE BIT UP?
2248 014232 001403              BEQ      1$              ;IF YES BRANCH
2249 014234 010237 001122      MOV      R2, @#SBDADR    ;IF NOT STORE FAILING REG.
2250 014240 104011              ERROR 11                ;RHCS2 DID NOT HAVE DLT
2251
2252
2253
2254
2255 014242 022737 144200 001714 1$:  CMP      #SC!TRE!RDY!DVA, @#CS1 ;IS SPECIAL CONDITION, TRANSFER ERROR
2256
2257 014250 001403              BEQ      2$              ;AND READY UP?
2258 014252 010137 001122      MOV      R1, @#SBDADR    ;IF YES BRANCH
2259 014256 104011              ERROR 11                ;IF NOT STORE FAILING REG.
2260
2261
2262
2263
2264 014260 012711 040000          2$:  MOV      #TRE, @R1       ;RHCS1 DID NOT HAVE SC, DVA
2265
2266 014264 004737 042220          JSR      PC, @#PUTREG    ;TRE AND RDY. AFTER A
2267 014270 022737 004200 001714  CMP      #RDY!DVA, @#CS1 ;READ FROM EMPTY SILO ONLY
2268
2269 014276 001403              BEQ      3$              ;THESE BITS SHOULD BE UP
2270 014300 010137 001122      MOV      R1, @#SBDADR    ;ALL OTHERS SHOULD BE 0
2271 014304 104011              ERROR 11                ;CLEAR ERROR BITS BY MOVING
2272
2273
2274 014306 013746 001774          3$:  MOV      @#UNIT, -(SP)   ;ONE INTO TRE IN RHCS1
2275 014312 052716 000100      BIS      #IR, (SP)       ;SAVE REGISTERS
2276 014316 022637 001712      CMP      (SP)+, @#CS2    ;ALL BITS BUT RDY AND DVA SHOULD
2277 014324 010237 001122      MOV      R2, @#SBDADR    ;BE 0
2278 014330 104011              ERROR 11                ;BRANCH IF YES
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288

```

```

2279
2280
2281 014334 012706 001000      MOV      #STACK,SP      ;RESET STACK
2282 014360 004737 042614      JSR      PC,CLDISK      ;CLEAR REGISTERS LOAD R'S
2283
2284 014364 013746 001774      MOV      @#UNIT,-(SP)
2285 014370 052716 000100      BIS      #IR,(SP)
2286 014374 004737 042220      JSR      PC,@#PUTREG     ;SAVE REGISTERS
2287 014400 022637 001712      CMP      (SP)+,@#CS2     ;IR SHOULD BE SET "OR" RESET
2288 014404 001403
2289 014406 010237 001122      BEQ      1$
2290 014412 104011      MOV      R2,@#SBDADR
2291      ERROR      11      ;FAILING REGISTER RHCS2
2292      ;RHCS2 DOES NOT HAVE IR
2293      ;SET, UNIT NO. SET AND
2294      ;ALL OTHER BITS 0
2295 014414 005077 165210      CLR      @RHDB           ;LOAD DATA BUFFER (SILO) WITH 0
2296 014420 012777 177777      MOV      #-1,@RHDB      ;LOAD SILO WITH ALL ONES
2297 014426 013737 001636      MOV      @#RHCS2,@#2$
2298 014434 104415      WAT
2299 014436 000000      .WORD
2300 014440 000200      OR
2301 014442 013746 001774      MOV      @#UNIT,-(SP)
2302 014446 052716 000300      BIS      #OR:IR,(SP)
2303 014452 004737 042220      JSR      PC,@#PUTREG     ;SAVE REGISTERS
2304 014456 022637 001712      CMP      (SP)+,@#CS2     ;IR AND "OR" SHOULD BE SET
2305 014462 001403
2306 014464 010237 001122      BEQ      4$
2307 014470 104011      MOV      R2,@#SBDADR
2308      ERROR      11      ;SAVE RHCS2 ADDR. FAILING REG.
2309      ;"OR" IN RHCS2 SHOULD BE
2310      ;SET TOGETHER WITH IR AND
2311      ;UNIT NO.
2312 014472 017700 165132      MOV      @RHDB,R0
2313 014476 017705 165126      MOV      @RHDB,R5
2314 014502 022700 000000      CMP      #0,R0
2315 014506 001410      BEQ      5$
2316 014510 005037 001124      CLR      @#SGDDAT
2317      ;GOOD DATA
2318 014514 010037 001126      MOV      R0,@#SBDAT
2319 014520 013737 001630      MOV      @#RHDB,@#REGADR ;BAD DATA
2320 014526 104001      ERROR      1      ;SAVE RHDB FAILING REG.
2321      ;SILO DID NOT HAVE THE FIRST WORD
2322      ;"0" WHEN "OR" WAS SET
2323 014530 022705 177777      CMP      #-1,R5
2324 014536 012737 177777      MOV      #-1,@#SGDDAT
2325 014544 010537 001126      MOV      R5,@#SBDAT
2326 014550 013737 001630      MOV      @#RHDB,@#REGADR ;BAD DATA
2327 014556 104001      ERROR      1      ;SAVE RHDB FAILING REG.
2328      ;SILO DID NOT HAVE THE SECOND
2329      ;WORD OF ALL ONES WHEN "OR"
2330      ;WAS SET

```

```

2326
2327
2328
2329 014602 012700 051416      MOV      #SILOTB,RO      ;TABLE POINTER
2330 014606 012705 000103      MOV      #67,R5        ;COUNTER
2331 014612 005020          1$: CLR      (RO)+        ;CLEAR TOTAL TABLE
2332 014614 005305          DEC      R5            ;COUNT
2333 014616 001375          BNE      1$           ;BRANCH IF NOT COMPLETELY CLEAR
2334 014620 004737 042614      JSR      PC,@#CLDISK   ;CLEAR ALL REG.
2335 014624 005000          CLR      RO
2336 014626 012705 000102      MOV      #66,R5        ;COUNT
2337 014632 010077 164772      MOV      RO,@RHDB      ;LOAD SILO WITH COUNT FROM 0 TO 65
2338 014636 005200          INC      RO            ;NEXT COUNT
2339 014640 005305          DEC      R5            ;IS 66 LOADS DONE?
2340 014642 001373          BNE      2$           ;BRANCH IF NOT.
2341 014644 013746 001774      MOV      @#UNIT,-(SP)
2342 014650 052716 000200      BIS      #OR,(SP)
2343 014654 004737 042220      JSR      PC,@#PUTREG   ;SAVE REGISTERS
2344 014660 022637 001712      CMP      (SP)+,@#CS2  ;"OR" SHOULD BE SET IR RESET
2345 014664 001405          BEQ      3$           ;BRANCH IF YES
2346 014666 010237 001122      MOV      R2,@#SBDADR  ;SAVE RHCS2 ADR. FAILING REG.
2347 014672 104011          ERROR   11           ;"OR" WAS NOT SET, IR WAS NOT
2348 014674 005037 002006      CLR      @#ERFLG$     ;RESET AFTER SILO WAS FULL
2349 014700 012700 051416      MOV      #SILOTB,RO   ;POINTER
2350 014704 012705 000102      MOV      #66,R5        ;COUNTER
2351 014710 017720 164714      MOV      @RHDB,(RO)+  ;READ SILO
2352 014714 005305          DEC      R5            ;COUNT
2353 014716 001374          BNE      4$           ;BRANCH IF 66 NOT DONE
2354 014720 012700 051416      MOV      #SILOTB,RO   ;POINTER
2355 014724 012705 000102      MOV      #66,R5
2356 014730 005046          CLR      -(SP)
2357 014732 021620          5$: CMP      (SP),(RO)+
2358 014734 001425          BEQ      7$           ;BRANCH IF GOOD
2359 014736 014037 001126      MOV      -(RO),@#SBDAT ;BAD DATA
2360 014742 011637 001124      MOV      (SP),@#SGDAT ;GOOD DATA
2361 014746 013737 001630 042264      MOV      @RHDB,@#REGADR ;FAILING REG. RHDB
2362 014754 005737 002006      TST      @#ERFLG$     ;IS THIS FIRST ERROR?
2363 014760 001002          BNE      6$           ;IF NOT BRANCH
2364 014762 104012          ERROR   12           ;THESE TWO ERROR CALLS ARE FOR
2365 014764 000401          BR       64$          ;BRANCH TO AVOID PRINTING NEXT ERROR
2366 014766 104013          6$: ERROR   13           ;THE SAME TYPEOUT. SILO
2367                                     ;HAD A COUNT WRITTEN IN.
2368                                     ;ON READ OUT AN ERROR WAS
2369                                     ;DETECTED. THE TOTAL SILO
2370                                     ;READOUT IS IN LOCATION
2371                                     ;"SILOTB" TO THE NEXT 65
2372                                     ;WORDS.
2373 014770 005720          64$: TST      (RO)+    ;INCREMENT (RO)
2374                                     ;ARE FURTHER COMPARES TO
2375                                     ;BE DONE
2376 014772 017746 164142      MOV      @SWR,-(SP)   ;ONLY KEEP SW7 AND SW8
2377 014776 042716 177577      BIC      #!CSW07!SW08,(SP)
2378 015002 022726 000200      CMP      #SW07,(SP)+ ;TEST SW07
2379 015006 001403          BEQ      10$          ;IF NO MORE COMPARE THEN BRANCH
2380 015010 005216          7$: INC      (SP)     ;NEXT GOOD WORD
2381 015012 005305          DEC      R5            ;COUNT
2382 015014 001346          BNE      5$           ;BRANCH IF 66 NOT COMPLETE

```

K05

CZRJGBD.RP04/5/6 DSKLS CTRLRI  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 60  
T31 SILO TEST 3

SEQ 0062

2382 015016 005726

10\$: TST (SP)+ ;POP STACK





2406							
2407							
2408							
2409							
2410	015166	004737	042614		JSR	PC, @#CLDISK	; CLEAR DISK
2411							
2412	015172	013746	001774		MOV	@#UNIT, -(SP)	; GET UNIT NO.
2413	015176	052716	000100		BIS	@#IR, (SP)	; SET INPUT READY
2414	015202	004737	042220		JSR	PC, @#PUTREG	; SAVE REGISTERS
2415	015206	022637	001712		CMP	(SP)+, @#CS2	; IR SHOULD BE SET "OR" CLEARED
2416	015212	001403			BEQ	1\$	; BRANCH IF GOOD
2417	015214	010237	001122		MOV	R2, @#\$BDADR	; FAILING REGISTER RHCS2
2418	015220	104011			ERROR	11	; RHCS2 DOES NOT HAVE IR SET
2419							; AND ALL OTHER BITS 0
2420	015222	013700	001630		1\$: MOV	@#RHDB, R0	; R0 HAS RHDB ADDRESS
2421	015226	005001			CLR	R1	; DATA
2422	015230	010110			2\$: MOV	R1, @R0	; 0, THEN 1 THEN 2 THEN 3
2423							; IN RHDB
2424	015232	005201			INC	R1	; INCREMENT DATA
2425	015234	022701	000004		CMP	#4, R1	; IS 4 DONE
2426	015240	103373			BHIS	2\$	; BRANCH IF NOT
2427	015242	013737	001636	015252	MOV	@#RHCS2, @#3\$	
2428	015250	104415			WAT		; WAIT FOR "OR"
2429	015252	000000			3\$: .WORD	0	; RHCS2 ADDRESS
2430	015254	000200			OR		; WAIT ON OR.
2431	015256	004737	042614		JSR	PC, @#CLDISK	; CLR IN RHCS2
2432	015262	013746	001774		MOV	@#UNIT, -(SP)	; UNIT NO.
2433	015266	052716	000100		BIS	@#IR, (SP)	
2434	015272	004737	042220		JSR	PC, @#PUTREG	; SAVE REGISTERS
2435	015276	022637	001712		CMP	(SP)+, @#CS2	; IR SHOULD BE SET "0"=0
2436	015302	001403			BEQ	4\$	; BRANCH IF GOOD
2437	015304	010237	001122		MOV	R2, @#\$BDADR	; FAILING REGISTER RHCS2
2438	015310	104011			ERROR	11	; RHCS2 DOES NOT HAVE IR SET
2439							; AND ALL OTHER BITS 0
2440	015312	013700	001630		4\$: MOV	@#RHDB, R0	; R0 HAS RHDB ADDRESS
2441	015316	012710	000004		MOV	#4, @R0	; LOAD 4 IN SILO
2442	015322	011201			MOV	@R2, R1	; SAVE RHCS2
2443	015324	011005			MOV	@R0, R5	; READ THE 4 IN SILO
2444	015326	011003			MOV	@R0, R3	; READ SILO TO GET DLT
2445	015330	011204			MOV	@R2, R4	; SAVE RHCS2
2446	015332	032701	000200		BIT	@OR, R1	; TEST FOR OR IN RHCS2
2447	015336	001424			BEQ	6\$	; IF OR IS NOT SET BRANCH
2448	015340	022705	000004		CMP	#4, R5	; SILO 4 IS NOW COMPARED
2449	015344	001410			BEQ	5\$	
2450	015346	010037	042264		MOV	R0, @#REGADR	; SILO ADDRESS
2451	015352	012737	000004	001124	MOV	#4, @#\$GDDAT	; GOOD DATA
2452	015360	010537	001126		MOV	R5, @#\$BDDAT	; BAD DATA
2453	015364	104001			ERROR	1	; SILO DID NOT CONTAIN WORD
2454							; PUT IN AFTER "OR" WAS UP
2455	015366	005703			5\$: TST	R3	; IS IT ZERO BECAUSE SILO
2456							; IS DESTRUCTIVE READ
2457	015370	001407			BEQ	6\$	; BRANCH IF GOOD
2458	015372	010037	042264		MOV	R0, @#REGADR	; SILO ADDRESS
2459	015376	005037	001124		CLR	@#\$GDDAT	; GOOD DATA
2460	015402	010337	001126		MOV	R3, @#\$BDDAT	; BAD DATA
2461	015406	104001			ERROR	1	; SILO SHOULD BE ZERO

CZRJGBO,RP04/5/6 DSKLS CTRLRI  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 63  
T33 SILO TEST 5

SEQ 0065

2462			
2463			
2464			
2465	015410	032704	100000
2466	015416	013746	001774
2467	015422	052716	100300
2468	015426	012637	001124
2469	015432	010437	001126
2470	015436	010237	042264
2471	015442	104001	

6\$:

BIT	#DLT,R4
MOV	@#UNIT, -(SP)
BIS	#DLT!OR!IR, (SP)
MOV	(SP)+, @#\$GDDAT
MOV	R4, @#\$BDDAT
MOV	R2, @#\$REGADR
ERROR	1

```

;AFTER THE ONE WORD PUT IN
;HAS BEEN TAKEN OUT AS
;SILO IS A DESTRUCTIVE READ
;
;GET UNIT NO
;
;GOOD DATA
;BAD DATA
;RHCS2 ADDRESS
;DATA LATE ERROR

```

.SBTTL MORE REGISTER TESTS

2472  
2473  
2474  
2475  
2476  
2477  
2478  
2479  
2480  
2481  
2482  
2483  
2484  
2485  
2486  
2487  
2488  
2489  
2490  
2491  
2492  
2493  
2494  
2495  
2496  
2497  
2498  
2499  
2500  
2501  
2502  
2503

```

015466 012706 001000      MOV      #STACK,SP      ;RESET STACK
015472 004737 042614      JSR      PC,CLDISK      ;CLEAR DISK REG.

015476 012711 003566      MOV      #3566,R1       ;LOAD RHCS1 WITH ANY NUMBER
015502 010146          MOV      R1,-(SP)       ;GETTING READY TO FORM ODD BYTE
015504 005216          INC      (SP)           ;SP NOW HAS ODD BYTE FOR RHCS1
015506 112736 000005      MOVVB   #5,(SP)+       ;MOVE 5 INTO ODD BYTE FOR RHCS1
015512 011137 001126      MOV      R1,@#SBDDAT    ;TEST DATA
015516 022737 006766 001126      CMP      #2566!DVA!RDY,@#SBDDAT ;RHCS1 SHOULD HAVE 6766
015524 001406          BEQ     IS              ;BRANCH IF GOOD
015526 012737 006766 001124      MOV      #2566!DVA!RDY,@#SGDDAT ;GOOD DATA
015534 010137 042264      MOV      R1,@#REGADR    ;FAILING REGISTER RHCS1
015540 104001          ERROR   1              ;MOVING A NUMBER INTO
                          ;ODD BYTE OF RHCS1 GAVE
                          ;WRONG RESULTS

IS:
015542 112711 000032      MOVVB   #32,R1         ;MOVE INTO EVEN BYTE
015546 011137 001126      MOV      R1,@#SBDDAT    ;TEST DATA
015552 022737 006632 001126      CMP      #2432!DVA!RDY,@#SBDDAT ;RHCS1 SHOULD HAVE 6632
015562 012737 006632 001124      MOV      #2432!DVA!RDY,@#SGDDAT ;GOOD DATA
015570 010137 042264      MOV      R1,@#REGADR    ;FAILING REGISTER RHCS1
015574 104001          ERROR   1              ;MOVING A NUMBER INTO EVEN
                          ;BYTE OF RHCS1 GAVE WRONG
                          ;RESULT

```

```

2504
2505
2506
2507 015612 012706 001000      MOV      #STACK, SP      ;RESET STACK
2508 015616 004737 042614      JSR      PC, CLDISK     ;CLEAR DISK REG.
2509
2510 015622 012711 003566      MOV      #3566, R1      ;LOAD RHCS1 WITH ANY NUMBER
2511 015626 010146              MOV      R1, -(SP)      ;GETTING READY TO FORM ODD BYTE
2512 015630 005216              INC      (SP)           ;SP NOW HAS ODD BYTE FOR RHCS1
2513 015632 112736 000005      MOV      #5, R1         ;MOVE 5 INTO ODD BYTE FOR RHCS1
2514 015636 011137 001126      MOV      R1, R0         ;TEST DATA
2515
2516 015642 022737 004766 001126      CMP      #566!DVA!RDY, R1 ;RHCS1 SHOULD HAVE 4766
2517 015650 001406              BEQ      1$             ;BRANCH IF GOOD
2518 015652 012737 004766 001124      MOV      #566!DVA!RDY, R1 ;GOOD DATA
2519 015660 010137 042264      MOV      R1, R0         ;FAILING REGISTER RHCS1
2520 015664 104001              ERROR    1             ;MOVING A NUMBER INTO
2521                                     ;ODD BYTE OF RHCS1 GAVE
2522                                     ;WRONG RESULTS
2523
2524 015666 112711 000032      1$:  MOV      #32, R1      ;MOVE INTO EVEN BYTE
2525 015672 011137 001126      MOV      R1, R0         ;TEST DATA
2526 015676 022737 004632 001126      CMP      #432!DVA!RDY, R1 ;RHCS1 SHOULD HAVE 4632
2527 015706 012737 004632 001124      MOV      #432!DVA!RDY, R1 ;GOOD DATA
2528 015714 010137 042264      MOV      R1, R0         ;FAILING REGISTER RHCS1
2529 015720 104001              ERROR    1             ;MOVING A NUMBER INTO EVEN
2530                                     ;BYTE OF RHCS1 GAVE WRONG
2531                                     ;RESULTS
2532

```

```

2533
2534
2535
2536 015744 004737 042614 JSR PC,@#CLDISK ;GIVE INIT & SETUP REGISTER CORRES
2537
2538 015750 052712 177000 BIS #177000,(R2) ;LOAD RHCS2
2539 015754 010246 MOV R2,-(SP) ;GETTING READY FOR ODD BYTE
2540 015756 005216 INC (SP) ;SP NOW HAS ODD BYTE FOR RHCS2
2541 015760 105036 CLRB @#(SP)+ ;CLEAR RHCS2 ODD BYTE
2542 015762 013746 001774 MOV @#UNIT,-(SP) ;GET UNIT NO.
2543 015766 052716 000100 BIS #IR,(SP) ;INPUT READY AS IT IS SET
2544 015772 011237 001126 MOV @R2,@#SBDDAT ;TEST DATA
2545 015776 022637 001126 CMP (SP)+,@#SBDDAT ;COMPARE TO SEE THAT
; "CLRB" DID CLEAR
2546
2547 016002 001411 BEQ 1$
2548 016004 013737 001774 001124 MOV @#UNIT,@#SGDDAT
2549 016012 052737 000100 001124 BIS #IR,@#SGDDAT ;GOOD DATA
2550 016020 010237 042264 MOV R2,@#REGADR ;FAILING REGISTER RHCS2
2551 016024 104001 ERROR 1 ;CLEARING ODD BYTE OF RHCS2
; GAVE WRONG RESULTS
2552
2553 016026 013746 001774 1$: MOV @#UNIT,-(SP)
2554 016032 052716 000013 BIS #BAI,(SP)
2555 016036 052712 020000 BIS #UPE,@R2 ;HAVE UPE AND MPE IN RHCS2
; BESIDES UNIT SELECT
2556
2557 016042 112612 MOVB (SP)+,@R2 ;MOVE INTO EVEN BYTE OF RHCS2
2558 016044 013746 001774 MOV @#UNIT,-(SP)
2559 016050 052716 020110 BIS #UPE!IR!BAI,(SP)
2560 016054 011637 001124 MOV (SP),@#SGDDAT ;GOOD DATA
2561 016060 011237 001126 MOV @R2,@#SBDDAT ;TEST DATA
2562 016064 022637 001126 CMP (SP)+,@#SBDDAT ;COMPARE TO SEE THAT MOVB DID
; MOVE EVEN BYTE ONLY
2563
2564 016072 010237 042264 MOV R2,@#REGADR ;FAILING REGISTER RHCS2
2565 016076 104001 ERROR 1 ;MOVING A NUMBER INTO EVEN
; BYTE OF RHCS2 GAVE WRONG
; RESULTS
2566
2567
2568

```

2569								
2570								
2571								
2572	016110	012706	001000					
2573	016114	004737	042614					
2574	016120	013704	001632					
2575	016124	012714	025252					
2576	016130	010446						
2577	016132	005216						
2578	016134	112736	000377					
2579	016140	011437	001126					
2580	016144	022737	177652	001126				
2581	016152	001406						
2582	016154	012737	177652	001124				
2583	016162	010437	042264					
2584	016166	104001						
2585								
2586	016170	112714	000123					
2587	016174	011437	001126		1\$:			
2588	016200	022737	177523	001126				
2589	016210	012737	177523	001124				
2590	016216	010437	042264					
2591	016222	104001						

  

MOV	#STACK, SP	; RESET STACK
JSR	PC CLDISK	; CLEAR DISK REGISTERS
MOV	@RHWC, R4	; R4 NOW IS WORD COUNT REGISTER
MOV	#25252, @R4	; LOAD RHWC
MOV	R4, -(SP)	; GETTING READY TO FORM ODD BYTE
INC	(SP)	; SP NOW HAS ODD BYTE FOR RHWC
MOVB	#377, @ (SP)+	; MOVE 377 INTO ODD BYTE OF RHWC
MOV	@R4, @#\$BDDAT	; TEST DATA
CMP	#177652, @#\$BDDAT	; COMPARE TO SEE IF MOV B DID OK
BEQ	1\$	; BRANCH IF GOOD
MOV	#177652, @#\$GDDAT	; GOOD DATA
MOV	R4, @#\$REGADR	; REGISTER FAILING RHWC
ERROR	1	; MOVING INTO ODD BYTE OF RHWC
		; GAVE WRONG RESULTS
MOVB	#123, @R4	; MOVE INTO EVEN BYTE OF RHWC
MOV	@R4, @#\$BDDAT	; TEST DATA
CMP	#177523, @#\$BDDAT	
MOV	#177523, @#\$GDDAT	; GOOD DATA
MOV	R4, @#\$REGADR	; REGISTER FAILING RHWC
ERROR	1	

F06

CZRJGB0.RP04/5/6 DSKLS CTRLRI  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 68  
T37 ODD BYTE TEST ON RHWC

SEQ 0070

2592						
2593						
2594						
2595						
2596	016226	012706	001000			MOV #STACK, SP ; RESET STACK
2597	016240	004737	042614			JSR PC, CLDISK
2598	016244	013704	001634			MOV @#RHBA, R4 ; R4 HAS ADDRESS OF RHBA
2599	016250	012714	025253			MOV #25253, @R4 ; LOAD RHBA
2600	016254	010446				MOV R4, -(SP) ; GETTING READY FOR ODD BYTE
2601	016256	005216				INC (SP) ; SP HAS ODD BYTE ADR. OF RHBA
2602	016260	112736	000377			MOVB #377, @ (SP)+ ; LOAD ODD BYTE OF RHBA
2603	016264	011437	001126			MOV @R4, @#SBDDAT ; TEST DATA
2604	016270	022737	177652	001126		CMP #177652, @#SBDDAT ; COMPARE MOVB RESULTS
2605	016276	001406				BEQ 1\$ ; BRANCH IF GOOD
2606	016300	012737	177652	001124		MOV #177652, @#SGDDAT ; GOOD DATA
2607	016306	010437	042264			MOV R4, @#REGADR ; FAILING REGISTER RHBA
2608	016312	104001				ERROR 1 ; MOVING INTO ODD BYTE OF
2609						; RHBA GAVE WRONG RESULTS
2610	016314	112714	000125		1\$:	MOVB #125, @R4
2611	016320	011437	001126			MOV @R4, @#SBDDAT ; TEST DATA
2612	016324	022737	177524	001126		CMP #177524, @#SBDDAT
2613	016334	012737	177524	001124		MOV #177524, @#SGDDAT ; GOOD DATA
2614	016342	010437	042264			MOV R4, @#REGADR ; FAILING REGISTER RHBA
2615	016346	104001				ERROR 1 ; MOVING INTO EVEN BYTE OF
2616						; RHBA GAVE WRONG RESULTS



2617  
2618  
2619  
2620  
2621  
2622  
2623  
2624  
2625  
2626  
2627  
2628  
2629  
2630  
2631  
2632  
2633  
2634  
2635  
2636  
2637  
2638  
2639  
2640  
2641  
2642  
2643  
2644  
2645  
2646  
2647  
2648  
2649  
2650  
2651  
2652  
2653  
2654  
2655  
2656  
2657  
2658  
2659  
2660  
2661  
2662  
2663  
2664  
2665  
2666  
2667  
2668  
2669  
2670  
2671  
2672

.SBTTL DCL COMMAND TESTS

\*\*\* FOUR GENERAL REGISTERS WILL BE RESERVED FOR HARDWARE  
\*\*\* R1=RHCS1 CONTROL AND STATUS1  
\*\*\* R2=RHCS2 CONTROL AND STATUS2  
\*\*\* R3=RHDS1 DRIVE STATUS 1  
\*\*\* R4=RHER1 ERROR REGISTER1  
  
\*\*\* WHENEVER ANY OTHER USE IS MADE OF THESE REGISTERS  
\*\*\* APPROPRIATE SAVING MUST BE DONE

\*\*\* ERROR REGISTER #01 (RHER1) TEST  
\*\*\* BIT #1 (ILLEGAL REGISTER) CANNOT BE TESTED ON PDP11 THIS BIT  
\*\*\* IS FOR PDP10 USE ONLY

```

016352 012706 001000      MOV      #STACK, SP      ; RESET STACK
016364 004737 042614      JSR      PC, @#CLDISK   ; CLEAR REGISTERS
016370 012777 000001 163262  MOV      #DMD, @RHMR    ; SET DIAGNOSTIC MODE
016376 005037 002022      CLR      @#TMPILL      ; GET READY TO MAKE ILLEGAL FUNCTION
016402 012700 002042      MOV      @#FUTABL, R0   ; LOAD FUNCTION CODE TABLE START
016406 012705 000021 1$:      MOV      #17, R5        ; COUNTER (16 GOOD FUNCTIONS)
016412 023720 002022      CMP      @#TMPILL, (R0)+ ; IS THIS A LEGAL FUNCTION CODE?
016416 001004          BNE      3$            ; NO - DECR. FUNCT. CODE CTR
016420 062737 000002 002022  ADD      #2, @#TMPILL   ; YES MAKE NEXT FUNCTION CODE
016426 000765          BR       1$           ; TEST NEXT FUNCTION CODE
016430 005305          DEC      R5           ; MAKE NEXT CODE IF 1ST 16
016432 001367          BNE      2$           ; LEGAL FUNCTIONS NOT DONE
016434 032737 000100 002022  BIT      #100, @#TMPILL ; BRANCH IF 16 NOT COMPLETE
016442 001077          BNE      12$          ; ALL BITS UP TO BIT #5 COMPARED?
016444 013737 002022 002104  MOV      @#TMPILL, @#ILLEGL ; YES - EXIT ----->
016452 062737 000002 002022  ADD      #2, @#TMPILL   ; NO - TEST THE ILLEGAL FUNCTION
016460 004737 042614 4$:      JSR      PC, @#CLDISK   ; SET DIAGNOSTIC MODE
016464 012777 000001 163166  MOV      #DMD, @RHMR    ; ILLEGAL FUNCTION ----> RHCS1
016472 013711 002104          MOV      @#ILLEGL, @R1 ; ERROR RETURN POINT
016476 012737 016460          MOV      #4$, @#SLPERR ; ERROR RETURN POINT
016504 004737 042220          JSR      PC, @#PUTREG   ; SAVE REGISTERS
016510 005737 001716          TST     @#ER1         ; THERE SHOULD NOT BE ANY ERROR YET
016514 001403          BEQ     5$           ; CONTINUE IF RHER1 STILL = 0
016516 010437 001122          MOV     R4, @#$BDADR   ; FAILING REGISTER ADDRESS RHER1
016522 104011          ERROR  11           ; ALTHOUGH AN ILLEGAL FUNCTION
                                ; HAS BEEN MOVED INTO RHCS1
                                ; NO ERRORS SHOULD SHOW TILL
                                ; GO IS SET RHER1 SHOULD BE
                                ; ALL ZEROS

```



```

2726
2727
2728 016646 012706 001000      MOV      #STACK,SP      ;RESET STACK
2729
2730 016660 004737 042614      JSR      PC,@#CLDISK    ;INIT AND SET GENERAL REGISTERS
2731
2732      ;*FILL ALL POSSIBLE BITS WITH ONES
2733
2734 016664 012777 177777 162740  MOV      #177777,@RHWC  ;WORD COUNT REGISTER GETS 177777
2735 016672 012777 177777 162734  MOV      #177777,@RHBA  ;BUS ADDRESS REGISTER GETS 177777
2736 016700 012777 017437 162736  MOV      #17437,@RHDST  ;DESIRED SECTOR TRACK GETS 17437
2737 016706 012777 016377 162734  MOV      #16377,@RHOF   ;OFFSET REGISTER GETS 16277
2738 016714 012777 000777 162730  MOV      #777,@RHCA     ;DESIRED CYLINDER GETS 777
2739 016722 012746 001400      MOV      @A16!A17,-(SP) ;GET BIT 9 AND 8
2740 016726 053716 002102      BIS      @#READIN,(SP)
2741 016732 012677 162702      MOV      (SP)+,@RHCSI   ;FILL READ IN PRESET IN RHCSI
2742 016736 012777 000001 162714  MOV      #DMD,@RHMR    ;SET DIAGNOSTIC MODE
2743
2744      ;*THE REGISTERS WILL BE SAVED IN REINTO BUFFER
2745 016744 004037 043306      JSR      RO,@#SAVER     ;SAVE
2746 016750 001632      RHWC      ;FROM
2747 016752 003154      REINTO    ;TO
2748 016754 000021      17.      ;NUMBER SAVED
2749
2750      ;*GIVE READ IN PRESET COMMAND
2751 016756 052777 000001 162654  BIS      #GO,@RHCSI    ;INCLUDE GO TO READ IN PRESET
2752
2753      ;*NOW SAVED REGISTERS WILL BE CHANGED TO EXPECTED VALUE
2754 016764 005037 003166      CLR      @#REINTO+12   ;CLEAR SAVED RHDST
2755 016770 042737 016000 003172  BIC      #FMT22!HCI!ECI,@#REINTO+16 ;CLEAR FMT22,HCI,ECI IN
2756      ;SAVED RHOF
2757 016776 052737 000100 003172  BIS      #VV,@#REINTO+16 ;SET VV IN SAVED RHOF
2758 017004 005037 003174      CLR      @#REINTO+20   ;CLEAR SAVED RHCA
2759
2760      ;*AFTER A READ IN PRESET COMMAND
2761      ;*SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE
2762
2763 017010 004037 043306      JSR      RO,@#SAVER     ;SAVE
2764 017014 001632      RHWC      ;FROM
2765 017016 002110      WRFROM    ;TO
2766 017020 000021      17.      ;NUMBER OF REGISTERS SAVED
2767
2768      ;*AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
2769      ;*OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
2770      ;*SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
2771 017022 113737 003201 002135  MOVB     @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
2772
2773      ;*COMPARE REGISTERS BEFORE READ IN PRESET COMMAND
2774      ;*WITH AFTER COMMAND
2775
2776 017030 004037 043510      JSR      RO,@#COMPAR   ;COMPARE
2777 017034 003154      REINTO    ;GOOD BUFFER
2778 017036 002110      WRFROM    ;TEST BUFFER
2779 017040 000021      17.      ;NUMBER OF REGISTERS
2780 017042 017050      1$      ;RETURN FOR ERROR
2781 017044 017050      1$      ;SAME

```



```

2793
2794
2795
2796
2797 017100 004737 042614
2798 017104 012777 000001 162546
2799 017112 013711 002042
2800 017116 012700 001632
2801 017122 012703 001706
2802 017126 012702 000021
2803 017132 013023 1$:
2804 017134 005302
2805 017136 001375
2806 017140 013737 001662 017160
2807 017146 010137 017166
2808 017152 052711 000001
2809 017156 104415
2810 017160 000000 2$:
2811 017162 000200
2812 017164 104415
2813 017166 000000 3$:
2814
2815 017170 000200
2816
2817
2818
2819
2820 017172 004037 043306
2821 017176 001632
2822 017200 002110
2823 017202 000021
2824
2825
2826
2827
2828 017204 113737 001733 002135
2829
2830
2831
2832
2833
2834 017212 004037 043510
2835 017216 001706
2836 017220 002110
2837 017222 000021
2838 017224 017232
2839 017226 017232
2840 017230 017252
2841
2842 017232 013705 047620 4$:
2843 017236 060505
2844 017240 016537 001630 042264
2845 017246 104001
2846
2847 017250 000207
2848

```

```

; *START WITH CLR IN RHCS2 (BITS)
JSR PC, @#CLDISK ; CLEAR ALL POSSIBLE BITS
MOV #DMD, @RHMR ; SET DIAGNOSTIC MODE
MOV @#NOPERA, @R1 ; PUT NOP OPERATION=0 IN RHCS1
MOV #RHWC, @R0 ; STARTING ADDRESS OF REG
MOV #WC, @R3 ; STARTING ADDRESS OF WHERE SAVED
MOV #RHEC2-RHWC+2/2, @R2 ; NUMBER OF REGISTERS
MOV @ (R0)+, (R3)+ ; SAVE HARDWARE REG
DEC R2 ; COUNT
BNE 1$ ; BRANCH IF NOT COMPLETE
MOV @#RHDS1, @#2$ ; GET ADDRESS OF DRIVE STATUS
MOV R1, @#3$ ; GET ADDRESS OF RHCS1
BIS #GO, @R1 ; GO TO RHCS1
WAT ; WAIT FOR DRY IN RHDS1
.WORD 0 ; ADDRESS OF DRIVE STATUS RHDS1
DRY ; DRY WILL BE WAITED ON
WAT ; WAIT FOR RDY IN RHCS1
.WORD 0 ; ADDRESS OF RHCS1 PUT HERE BY AN
RDY ; EARLIER MOV
; RDY WILL BE WAITED ON

; *AFTER A NO OP COMMAND
; *SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE
JSR @R0, @#SAVER ; SAVE
RHWC ; FROM
WRFROM ; TO
17. ; NUMBER OF REGISTERS SAVED

; *AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
; *OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
; *SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
MOVB @#AS+1, @#WRFROM+25; SAVE UPPER RHAS

; *COMPARE REGISTERS BEFORE NO OP COMMAND
; *WITH AFTER COMMAND
JSR @R0, @#COMPAR ; COMPARE
WC ; GOOD BUFFER
WRFROM ; TEST BUFFER
17. ; NUMBER OF REGISTERS
4$ ; RETURN FOR ERROR
4$ ; SAME
5$ ; RETURN FOR GOOD COMPARISON

MOV @#ERWORD, @R5 ; GETTING READY TO INDEX
ADD @R5, @R5 ; DOUBLE ERROR WORD
MOV @RHWC-2(R5), @#REGADR ; FAILING REG. ADDRESS
ERROR 1 ; NO OP COMMAND CAUSED IMPROPER
RTS PC ; REGISTER CHANGE
; RETURN FOR FURTHER COMPARISONS

```

L06

CZRJGB0 RPO4/5/6 DSKLS CTRLR1  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 74  
T43 NO OPERATION FUNCTION TEST

SEQ 0076

```

2849 017252          5$:                               ;NO ERRORS
2850
2851
2852
2853 017252 012737 017260 001110      MOV    #14$, @#SLPERR      ;SET SCOPE LOOP TO 14$
2854 017260 004737 042614          JSR    PC, @#CLDISK      ;INIT LAST ALL ZERO TEST
2855 017264 012777 000001 162366      MOV    @#DMD, @#RHMR     ;SET DIAGNOSTIC MODE
2856
2857
2858                               ;*NOW START WITH ALL ONES IN ALL POSSIBLE REGISTERS
2859
2860 017272 012700 001632      MOV    @#RHWC, R0        ;ADDRESS OF FIRST REGISTER
2861 017276 012705 000021      MOV    @#RHEC2-RHWC+2/2, R5 ;NO. OF REGISTERS
2862 017302 012730 177676      MOV    #177676, @#(R0)+  ;FILL WITH ALL ONES
2863 017306 013777 001774 162322      MOV    @#UNIT, @#RHCS2   ;REINSTATE UNIT NUMBER UNDER TEST
2864                               ;KEEP INTERRUPT DISABLED
2865 017314 005305      DEC    R5                ;COUNT
2866 017316 001371      BNE    @#B$              ;BRANCH IF INCOMPLETE
2867 017320 013711 002042      MOV    @#NOPERA, @#R1    ;PUT NOP OPERATION =0 IN RHCS1
2868 017324 012700 001632      MOV    @#RHWC, R0        ;STARTING ADDRESS OF REG
2869 017330 012703 001706      MOV    @#WC, R3          ;STARTING ADDRESS OF WHERE SAVED
2870 017334 012702 000021      MOV    @#RHEC2-RHWC+2/2, R2 ;NUMBER OF REGISTERS
2871 017340 013023      MOV    @#(R0)+, @#(R3)+  ;SAVE HARDWARE REG
2872 017342 005302      DEC    R2                ;COUNT
2873 017344 001375      BNE    @#B$              ;BRANCH IF NOT COMPLETE
2874 017346 013737 001662 017366      MOV    @#RHDS1, @#10$    ;GET ADDRESS OF DRIVE STATUS
2875 017354 010137 017374      MOV    R1, @#11$        ;GET ADDRESS OF RHCS1
2876 017360 052711 000001      BIS    @#G0, @#R1        ;GO TO RHCS1
2877 017364 104415      WAT    @#10$             ;WAIT FOR DRY IN RHDS1
2878 017366 000000      .WORD 0                 ;ADDRESS OF DRIVE STATUS RHDS1
2879 017370 000200      DRY    @#10$             ;DRY WILL BE WAITED ON
2880 017372 104415      WAT    @#11$             ;WAIT FOR RDY IN RHCS1
2881 017374 000000      .WORD 0                 ;ADDRESS OF RHCS1 PUT HERE BY AN
2882                               ;EARLIER MOV.
2883 017376 000200      RDY    @#11$             ;RDY WILL BE WAITED ON
2884
2885                               ;*AFTER A NO OP COMMAND
2886                               ;*SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE
2887
2888 017400 004037 043306      JSR    RO, @#SAVER        ;SAVE
2889 017404 001632          RHWC                    ;FROM
2890 017406 002110          WRFROM                  ;TO
2891 017410 000021          17.                     ;NUMBER OF REGISTERS SAVED
2892
2893                               ;*AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
2894                               ;*OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
2895                               ;*SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
2896 017412 113737 001733 002135      MOVB   @#AS+1, @#WRFROM+25;SAVE UPPER RHAS
2897
2898
2899                               ;*COMPARE REGISTERS BEFORE NO OP COMMAND
2900                               ;*WITH AFTER COMMAND
2901
2902 017420 004037 043510      JSR    RO, @#COMPAR      ;COMPARE
2903 017424 001706          WC                      ;GOOD BUFFER
2904 017426 002110          WRFROM                  ;TEST BUFFER

```



```

2918
2919
2920 017462 012706 001000      MOV      #STACK, SP      ;RESET STACK
2921 017474 004737 042614      JSR      PC, @#CLDISK    ;SET REGISTERS AND CLEAR
2922
2923      ;*FILL ALL POSSIBLE BITS WITH ONES
2924
2925 017500 012777 177777 162122      MOV      #177777, @RHOB  ;BUS ADDRESS REGISTER GETS 177777
2926 017506 012777 177777 162116      MOV      #177777, @RHWC  ;WORD COUNT REGISTER GETS 177777
2927 017514 012777 177777 162112      MOV      #177777, @RHBA  ;BUS ADDRESS REGISTER GETS 177777
2928 017522 052777 157010 162106      BIS      #157010, @RHCS2 ;CONTROL AND STATUS 2 GETS 157010
2929 017530 012777 001476 162102      MOV      #1476, @RHCS1  ;CONTROL AND STATUS REGISTER GETS 1476
2930 017536 012777 177777 162076      MOV      #177777, @RHER1 ;ERROR REGISTER1 GETS 177777
2931 017544 012777 017437 162072      MOV      #17437, @RHDST  ;DESIRED SECTOR TRACK
2932 017552 012777 177777 162066      MOV      #177777, @RHER2 ;ERROR REGISTER 2
2933 017560 012777 016277 162062      MOV      #16277, @RHOF   ;OFFSET REGISTER
2934 017566 012777 177777 162056      MOV      #177777, @RHCA  ;DESIRED CYLINDER
2935 017574 012777 177777 162052      MOV      #177777, @RHER3 ;ERROR REGISTER 3
2936 017602 012777 000001 162050      MOV      #DMD, @RHMR    ;MAINTENANCE REGISTER
2937 017610 012777 177777 162042      MOV      #177777, @RHMR ;MAINTENANCE REGISTER
2938
2939
2940      ;*THIS SETS BITS FOR ALL PRESENT DRIVES
2941
2942 017616 013700 002020      MOV      @#TOTALAT, R0   ;GET DRIVE PRESENT
2943 017622 005012          CLR      @R2             ;CLEAR RHCS2 AND CARRY BIT
2944 017624 012705 000010      MOV      #8, R5         ;COUNTER
2945 017630 006000          ROR      R0             ;GET BIT INTO CARRY
2946 017632 103002          BCC     31$            ;BRANCH IF NO UNIT ON THIS BIT
2947 017634 012714 177777      MOV      #-1, @R4       ;MOVE INTO ERROR REGISTER TO SET ATA
2948 017640 005212          INC     @R2            ;INCREMENT RHCS2 - UNIT NO.
2949 017642 005305          DEC     R5             ;COUNT
2950 017644 001401          BEQ     27$            ;BRANCH IF 8 DONE
2951 017646 000770          BR     30$            ;CONTINUE THIS ROUTINE
2952 017650 013746 001774      MOV      @#UNIT, -(SP)  ;
2953 017654 052716 157010      BIS      #157010, (SP)  ;REINSTATE SET BITS
2954 017660 012612          MOV      (SP)+, @R2    ;
2955
2956
2957 017662 012777 000001 161770      MOV      #DMD, @RHMR    ;SET DMD
2958 017670 013711 002050      MOV      @#DCLEAR, @R1 ;DRIVE CLEAR = 10 INTO RHCS1
2959 017674 052711 000001      BIS      #GO, @R1      ;GO
2960 017700 012700 001630      MOV      #RHOB, R0     ;R0 CONTAINS ADDR. OF ADDR. OF REG.
2961
2962
2963      ;*DATA BUFFER REGISTER
2964
2965
2966
2967      ;*WORD COUNT REGISTER
2968
2969
2970
2971      ;*BUS ADDRESS REGISTER
2972
2973

```



```

2974
2975
2976
2977
2978
2979
2980
2981
2982 020042 005737 002000 6$: TST 2#NUNIT ;ARE THERE MORE THAN ONE UNIT
2983 020046 001404 BEQ 32$ ;BRANCH IF ONLY ONE UNIT
2984 020050 012737 104210 001124 MOV #104210,2#SGDDAT;GOOD DATA
2985 020056 000403 BR 33$
2986 020060 012737 004210 001124 32$: MOV #4210,2#SGDDAT ;GOOD DATA
2987 020066 013037 001126 33$: MOV 2(RO)+,2#SBDDAT ;TEST DATA
2988
2989 020072 023737 001124 001126 CMP 2#SGDDAT,2#SBDDAT;COMPARE DATA
2990 020100 001402 BEQ 7$ ;BRANCH IF GOOD
2991 020102 004737 020564 JSR PC,2#ERCLFC ;JUMP TO ERROR FOR CLR BIT 5
2992 ;IN RHCS2
2993
2994 ;*ERROR 1 REGISTER
2995
2996
2997 ;*DESIRED SECTOR/TRACK REGISTER
2998
2999 ;*ERROR 2 REGISTER
3000
3001 ;*OFFSET REGISTER
3002
3003 ;*DESIRED CYLINDER ADDRESS REGISTER
3004
3005 ;*ERROR 3 REGISTER
3006
3007 ;*ATTENTION SUMMARY REGISTER
3008
3009
3010
3011
3012
3013
3014 020312 013737 002020 001124 15$: MOV 2#TOTALAT,2#SGDDAT;SET ALL BITS OF DRIVE PRESENT IN RHAS
3015 020320 043737 002016 001124 BIC 2#ATTENT,2#SGDDAT ;CLEAR ONLY WORKING DRIVE BIT
3016 020326 013037 001126 MOV 2(RO)+,2#SBDDAT ;GET RHAS
3017 020332 123737 001124 001126 CMPB 2#SGDDAT,2#SBDDAT ;COMPARE DATA
3018 020340 001402 BEQ 16$ ;BRANCH IF GOOD
3019 020342 004737 020564 JSR PC,2#ERCLFC ;JUMP TO ERROR FOR CLR (BIT 5) IN RHCS2
3020
3021 ;*MAINTAINABILITY REGISTER
3022
3023
3024 ;*DRIVE STATUS REGISTER
3025
3026 020374 012737 000700 001124 17$: MOV #700,2#SGDDAT ;GOOD DATA FOR PRINTOUT
3027 020402 013046 MOV 2(RO)+,-(SP) ;GET RHDS1
3028 020404 011637 001126 MOV (SP),2#SBDDAT ;TEST DATA
3029 020410 042716 001000 BIC #PROG,(SP) ;CLEAR PROG BIT

```

```

3030 020414 022726 000700      CMP      #700, (SP)+      ;COMPARE DATA
3031 020420 001402              BEQ      20$              ;BRANCH IF GOOD
3032 020422 004737 020564      JSR      PC, @#ERCLFC    ;JUMP TO ERROR FOR DRIVE CLEAR
3033
3034                          ;*DRIVE TYPE
3035
3036                          ;*SERIAL NUMBER REGISTER
3037
3038
3039
3040                          ;*ECC1 POSITION
3041
3042                          ;*ECC2 PATTERN
3043
3044
3045
3046
3047                          ;*LOOK-AHEAD REGISTER
3048
3049
3050 020556 005720      24$:    TST      (R0)+      ;AS THE LOOK-AHEAD REG. CANNOT BE PREDICTED
3051                          ;IT IS NOT CHECKED AFTER AN INIT
3052
3053                          ;*CURRENT CYLINDER ADDRESS REGISTER
3054
3055 020560 005720      25$:    TST      (R0)+      ;AS THE CURRENT CYL REG. CANNOT BE PREDICTED
3056                          ;AFTER AN INIT IT IS NOT CHECKED
3057
3058 020562      26$:
3059
3060 020564 014037 042264      ERCLFC: MOV      -(R0), @#REGADR ;FAILING REGISTER ADDRESS
3061 020570 104001              ERROR      1              ;CLR FUNCTION = 10 IN RHCS1 DID
3062                          ;NOT CLEAR APPROPRIATE BITS
3063                          ;OR CLEARED EXTRA BITS
3064 020572 005720      TST      (R0)+      ;UNDO -(R0) FOR ERROR
3065 020574 000207      RTS      PC          ;RETURN TO ABOVE PROGRAM
3066                          ;OR CLEARED EXTRA BITS
3067 020576 005720      TST      (R0)+      ;UNDO -(R0) FOR BAD DATA
3068

```

```

3069
3070
3071
3072 020614 012706 001000      MOV      #STACK, SP      ;RESET STACK
3073 020626 004737 042614      JSR      PC, @#CLDISK    ;INIT AND SET UP GENERAL REG. CORRES.
3074                                     ;AND UNIT NUMBER
3075 020632 012777 000001 161020  MOV      #DMD, @RHMR     ;SET DIAGNOSTIC MODE BIT
3076                                     ;THIS ENABLES COMMANDS WITHOUT MOL
3077                                     ;AND HOLDS RHLA FROM MOVING
3078 020640 005077 161000      CLR      @RHDS1          ;MAKE DESIRED SECTOR TRACK LEGAL
3079 020644 013777 002072 160766  MOV      @#SEECOM, @RHCS1 ;LOAD SEEK COMMAND INTO CONTROLLER
3080 020652 017746 161020      MOV      @RHCC, -(SP)    ;GET CURRENT CYLINDER
3081
3082
3083                                     ;*FOLLOWING ARE TWO BLOCKS OF CODE TO LOAD RHCA WITH THE PROPER
3084                                     ;*ADDRESS DEPENDING UPON WHETHER THE DRIVE IS AN RPO6 OR RPO4
3085
3086 020656 005737 002036      TST      @#RPO6          ;MOVE DRIVE TYPE FLAG TO ITSELF TO TEST
3087 020662 001416                                     BEQ      11$             ;TREAT THE DRIVE AS AN RPO4
3088
3089                                     ;TREAT THE DRIVE AS AN RPO6
3090 020664 022726 001456      CMP      #814., (SP)+    ;IS CURRENT CYLINDER SAME AS 814. ?
3091 020670 001404                                     BEQ      9$              ;BRANCH IF YES TO MAKE RHCA = 813.
3092 020672 012737 001456 001210  MOV      #814., @#STMP5   ;GET READY TO MAKE RHCA = 814.
3093 020700 000403                                     BR       10$             ;FILL RHCA
3094 020702 012737 001455 001210 9$:  MOV      #813., @#STMP5   ;GET READY TO MAKE RHCA = 813.
3095 020710 013777 001210 160734 10$:  MOV      @#STMP5, @RHCA   ;MAKE DESIRED CYLINDER 814., OR 813.
3096 020716 000415                                     BR       14$             ;SAVE REGISTERS
3097
3098                                     ;TREAT THE DRIVE AS AN RPO4
3099 020720 022726 000632                                     11$:  CMP      #410., (SP)+    ;IS CURRENT CYLINDER SAME AS 410. ?
3100 020724 001404                                     BEQ      12$             ;BRANCH IF YES TO MAKE RHCA = 409.
3101 020726 012737 000632 001210  MOV      #410., @#STMP5   ;GET READY TO MAKE RHCA = 410.
3102 020734 000403                                     BR       13$             ;FILL RHCA
3103 020736 012737 000631 001210 12$:  MOV      #409., @#STMP5   ;GET READY TO MAKE RHCA = 409.
3104 020744 013777 001210 160700 13$:  MOV      @#STMP5, @RHCA   ;MAKE DESIRED CYLINDER 410., OR 409.
3105
3106                                     ;*SAVE REGISTERS FOR COMPARISON AFTER GO
3107
3108 020752 004037 043306                                     14$:  JSR      RO, @#SAVER     ;SAVE
3109 020756 001632                                     RHW     ;FROM
3110 020760 003154                                     REINTO  ;TO
3111 020762 000023                                     19.    ;NUMBER OF REGISTERS SAVED
3112
3113                                     ;*GIVE GO TO COMMAND
3114 020764 052777 000001 160646  BIS      #GO, @RHCS1     ;GO TO COMMAND
3115
3116                                     ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
3117
3118 020772 052737 000001 003162  BIS      #GO, @#REINTO+6 ;SAVED RHCS1
3119 021000 052737 020000 003204  BIS      #PIP, @#REINTO+30 ;SAVED RHDS1
3120 021006 042737 000200 003204  BIC      #DRY, @#REINTO+30 ;SAVED RHDS1
3121
3122
3123                                     ;*AFTER GO HAS BEEN GIVEN FOR SEEK COMMAND
3124                                     ;*SAVE REGISTERS AGAIN SO THAT COMPARISONS CAN

```

```

3125 ;*BE DONE
3126
3127 021014 004037 043306 JSR RO,@#SAVER ;SAVE
3128 021020 001632 RHW C ;FROM
3129 021022 002110 WRFROM ;TO
3130 021024 000023 19. ;NUMBER OF REGISTERS SAVED
3131
3132 ;*AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
3133 ;*OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
3134 ;*SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
3135
3136 021026 113737 003201 002135 MOVB @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
3137
3138 ;*COMPARE REGISTERS BEFORE SEEK COMMAND
3139 ;*WITH CONTENTS AFTER GO IS ISSUED
3140
3141 021034 004037 043510 JSR RO,@#COMPAR ;COMPARE
3142 021040 003154 REINTO ;GOOD BUFFER
3143 021042 002110 WRFROM ;TEST BUFFER
3144 021044 000023 19. ;NUMBER
3145 021046 021054 1$ ;RETURN FOR ERROR
3146 021050 021054 1$ ;SAME
3147 021052 021074 2$ ;RETURN FOR GOOD COMPARISON
3148
3149 021054 013705 047620 1$: MOV @#ERWORD,R5 ;GETTING READY TO INDEX
3150 021060 060505 ADD R5,R5 ;DOUBLE ERROR WORD
3151 021062 016537 001630 042264 MOV RHW C-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
3152
3153 021070 104001 ERROR 1 ;IMPROPER REGISTER CHANGE
3154 ;AFTER SEEK COMMAND
3155 ;WITH GO IS GIVEN
3156 021072 000207 RTS PC ;RETURN TO COMPARISON
3157
3158
3159 ;*NOW GIVE INIT AND GET GO AND PIP DOWN
3160
3161 021074 052712 000040 2$: BIS #CLR,@R2 ;RH INITILIZE
3162 021100 013712 001774 MOV @#UNIT,@R2 ;REINSTATE UNIT NUMBER
3163 021104 012777 000001 160546 MOV #DMD,@RHM R ;SET DIAGNOSTIC MODE BIT
3164 ;THIS ENABLES COMMANDS WITHOUT MOL
3165 ;AND HOLDS RHLA FROM MOVING
3166
3167 ;*CHANGE REGISTERS TO EXPECTED VALUE
3168
3169 021112 042737 000001 003162 BIC #GO,@#REINTO+6 ;SAVED RHCS1
3170 021120 042737 020000 003204 BIC #PIP,@#REINTO+30 ;SAVED RHDS1
3171 021126 052737 000200 003204 BIS #DRY,@#REINTO+30 ;SAVED RHDS1
3172 021134 017737 160534 003216 MOV @RHLA,@#REINTO+42;SAVED RHLA
3173 021142 013737 001210 003220 MOV @#STMP5,@#REINTO+44 ;SAVED RHCC
3174
3175
3176 ;*AFTER INITILIZE SAVE REGISTERS SO THAT
3177 ;*COMPARES CAN BE DONE
3178
3179 021150 004037 043306 JSR RO,@#SAVER ;SAVE
3180 021154 001632 RHW C ;FROM

```

```

3181 021156 002110      WRFROM      ;TO
3182 021160 000023      19.        ;NUMBER OF REGISTERS SAVED
3183
3184
3185
3186 ;*AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
3187 ;*OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
3188 ;*SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
3189 021162 113737 003201 002135  MOVB      @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
3190
3191
3192 ;*COMPARE REGISTERS AFTER INITIALIZE
3193
3194 021170 004037 043510      JSR      RD,@#COMPAR      ;COMPARE
3195 021174 003154      REINTO      ;GOOD BUFFER
3196 021176 002110      WRFROM      ;TEST BUFFER
3197 021200 000023      19.        ;NUMBER OF REGISTERS TO BE
3198 ;COMPARED
3199 021202 021210      3$        ;RETURN POINT FOR ERROR
3200 021204 021210      3$        ;SAME
3201 021206 021230      4$        ;RETURN POINT FOR GOOD COMPARISON
3202
3203 021210 013705 047620      3$:      MOV      @#ERWORD,R5      ;GETTING READY TO INDEX
3204 021214 060505      ADD      R5,R5            ;DOUBLE ERROR WORD
3205 021216 016537 001630 042264  MOV      RHC-2(R5),@#REGADR ;FADING REGISTER ADDRESS
3206 021224 104001      ERROR      1            ;IMPROPER REGISTER
3207 ;CONTENTS AFTER GIVING AN
3208 ;INITILIZE FOLLOWING A
3209 ;SEEK COMMAND
3210 021226 000207      RTS      PC              ;RETURN TO COMPARISON
3211
3212 021230      4$:      ;GOOD COMPARISON
3213
3214 021230 004737 042614      JSR      PC,@#CLDISK     ;INIT AND SET UP GENERAL REG.
3215 ;AND UNIT NUMBER
3216 021234 012777 000001 160416  MOV      #DMD,@#RHMR     ;SET DIAGNOSTIC MODE BIT
3217 ;THIS ENABLES COMMANDS WITHOUT MOL
3218 ;AND HOLDS RHLA FROM MOVING
3219
3220 021242 013777 002072 160370  MOV      @#SEECOM,@#RHCS1 ;LOAD SEEK COMMAND INTO RHC
3221 021250 005077 160376      CLR      @#RHCA         ;DESIRED CYLINDER ADDRESS
3222
3223 ;*SAVE REGISTERS FOR COMPARISON AFTER GO
3224 021254 004037 043306      JSR      RO,@#SAVER      ;SAVE
3225 021260 001632      RHC        ;FROM
3226 021262 003154      REINTO      ;TO
3227 021264 000023      19.        ;NUMBER OF REGISTERS SAVED
3228
3229 ;*GIVE GO TO SEEK COMMAND
3230 021266 052777 000001 160344  BIS      #GO,@#RHCS1     ;GO TO SEEK COMMAND
3231
3232 ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
3233
3234 021274 052737 000001 003162  BIS      #GO,@#REINTO+6  ;SAVED RHCS1
3235 021302 052737 020000 003204  BIS      #PIP,@#REINTO+30 ;SAVED RHDS1
3236 021310 042737 000200 003204  BIC      #DRY,@#REINTO+30 ;SAVED RHDS1

```

```

3237
3238
3239
3240
3241
3242 021316 004037 043306
3243 021322 001632
3244 021324 002110
3245 021326 000023
3246
3247
3248
3249
3250
3251 021330 113737 003201 002135
3252
3253
3254
3255
3256
3257 021336 004037 043510
3258 021342 003154
3259 021344 002110
3260 021346 000023
3261 021350 021356
3262 021352 021356
3263 021354 021376
3264 021356 013705 047620 5$:
3265 021362 060505
3266 021364 016537 001630 042264
3267 021372 104001
3268
3269
3270 021374 000207
3271
3272
3273
3274 021376 052712 000040 6$:
3275 021402 013712 001774
3276 021406 012777 000001 160244
3277
3278
3279
3280
3281
3282 021414 042737 000001 003162
3283 021422 042737 020000 003204
3284 021430 052737 000200 003204
3285 021436 017737 160232 003216
3286 021444 005037 003220
3287
3288
3289
3290
3291
3292 021450 004037 043306

```

```

; *AFTER GO HAS BEEN GIVEN TO SEEK COMMAND
; *SAVE REGISTERS AGAIN SO THAT COMPARISONS CAN
; *BE DONE
JSR      RO, @#SAVER      ; SAVE
RHW C      ; FROM
WRFROM      ; TO
19.          ; NUMBER OF REGISTERS SAVED

; *AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
; *OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
; *SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
MOV B      @#REINTO+25, @#WRFROM+25; SAVE UPPER RHAS

; *COMPARE REGISTERS BEFORE COMMAND
; *WITH CONTENTS AFTER GO IS GIVEN
JSR      RO, @#COMPAR      ; COMPARE
REINTO      ; GOOD BUFFER
WRFROM      ; TEST BUFFER
19.          ; NUMBER
5$         ; RETURN FOR ERROR
5$         ; SAME
6$         ; RETURN FOR GOOD COMPARISON
MOV      @#ERWORD, R5      ; GETTING READY TO INDEX
ADD      R5, R5            ; DOUBLG ERROR WORD
MOV      RHW C-2(R5), @#REGADR ; FAILING REGISTER ADDRESS
ERROR    1                ; IMPROPER REGISTER CHANGE
; AFTER COMMAND
; WITH GO IS GIVEN
; RETURN TO COMPARISON
RTS      PC

; *NOW GIVE INIT AND GET GO AND PIP DOWN
BIS      #CLR, @R2        ; RH INITILIZE
MOV      @#UNIT, @R2      ; REINSTATE UNIT NUMBER
MOV      #DMD, @RHMR      ; SET DIAGNOSTIC MODE BIT
; THIS ENABLES COMMANDS WITHOUT MOL
; AND HOLDS RHLA FROM MOVING

; *CHANGE REGISTERS TO EXPECTED VALUE
BIC      #GO, @#REINTO+6  ; SAVED RHCS1
BIC      #PIP, @#REINTO+30 ; SAVED RHDS1
BIS      #DRY, @#REINTO+30 ; SAVED RHDS1
MOV      @RHLA, @#REINTO+42 ; SAVED RHLA
CLR      @#REINTO+44      ; SAVED RHCC

; *AFTER INITIALIZE SAVE REGISTERS SO THAT
; *COMPARES CAN BE DONE
JSR      RO, @#SAVER      ; SAVE

```

```

3293 021454 001632          RHCW          ;FROM
3294 021456 002110          WRFROM        ;TO
3295 021460 000023          19.           ;NUMBER OF REGISTERS SAVED
3296
3297                          ;*AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
3298                          ;*OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
3299                          ;*SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
3300 021462 113737 003201 002135  MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
3301
3302
3303                          ;*COMPARE REGISTERS AFTER INITIALIZE
3304
3305 021470 004037 043510      JSR     R0,@#COMPAR    ;COMPARE
3306 021474 003154            REINTO   ;GOOD BUFFER
3307 021476 002110            WRFROM   ;TEST BUFFER
3308 021500 000023            19.        ;NUMBER OF REGISTERS TO BE
3309                          ;COMPARED
3310 021502 021510            7$          ;RETURN POINT FOR ERROR
3311 021504 021510            7$          ;SAME
3312 021506 021530            8$          ;RETURN POINT FOR GOOD COMPARISON
3313
3314 021510 013705 047620      7$:      MOV     @#ERWORD,R5    ;GETTING READY TO INDEX
3315 021514 060505            ADD     R5,R5          ;DOUBLE ERROR WORD
3316 021516 016537 001630 042264  MOV     RHCW-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
3317 021524 104001            ERROR    1            ;# CONTENTS AFTER GIVING AN
3318                          ;COMMAND
3319 021526 000207            RTS     PC            ;RETURN TO COMPARISON
3320 021530 8$:              ;GOOD COMPARISON

```

```

3321
3322
3323 021532 012706 001000      MOV    #STACK, SP      ; RESET STACK
3324 021544 004737 042614      JSR    PC, @#CLDISK    ; INIT AND SET UP GENERAL REG.
3325                                ; AND UNIT NUMBER
3326 021550 012777 000001 160102  MOV    #DMD, @RHMR     ; SET DIAGNOSTIC MODE BIT
3327                                ; THIS ENABLES COMMANDS WITHOUT MOL
3328                                ; AND HOLDS RHLA FROM MOVING
3329
3330 021556 013777 002044 160054  MOV    @#UNLOAD, @RHCSI ; LOAD UNLOAD COMMAND INTO RH
3331                                ; *SAVE REGISTERS FOR COMPARISON AFTER GO
3332
3333
3334 021564 004037 043306      JSR    RO, @#SAVER     ; SAVE
3335 021570 001632                RHW    ; FROM
3336 021572 003154                REINTO ; TO
3337 021574 000023                19.      ; NUMBER OF REGISTERS SAVED
3338
3339                                ; *GIVE GO TO UNLOAD COMMAND
3340 021576 052777 000001 160034  BIS    #GO, @RHCSI    ; GO TO UNLOAD COMMAND
3341
3342                                ; *CHANGE SAVED REGISTERS TO EXPECTED VALUES
3343 021604 052737 000001 003162  BIS    #GO, @#REINTO+6 ; SAVED RHCSI
3344 021612 052737 020000 003204  BIS    #PIP, @#REINTO+30 ; SAVED RHDS1
3345 021620 042737 000200 003204  BIC    #DRY, @#REINTO+30 ; SAVED RHDS1
3346
3347 021626 005737 001100      TST    @#SPASS        ; IS THIS FIRST PASS
3348 021632 001053                BNE    SS             ; BRANCH IF NOT FIRST PASS
3349 021634 032777 020000 157276  BIT    #SW13, @SWR    ; INHIBIT ERROR PRINT HIGH?
3350 021642 001047                BNE    SS             ; BRANCH IF SW13 HIGH
3351
3352
3353 021754 013746 001774      MOV    @#UNIT, -(SP)  ; UNIT UNDER TEST
3354 021760 104405                TYPDS
3355
3356                                ; *AFTER GO HAS BEEN GIVEN TO UNLOAD COMMAND
3357                                ; *SAVED REGISTERS AGAIN SO THAT COMPARISONS CAN
3358                                ; *BE DONE
3359
3360 021762 004037 043306      SS:   JSR    RO, @#SAVER     ; SAVE
3361 021766 001632                RHW    ; FROM
3362 021770 002110                WRFROM ; TO
3363 021772 000023                19.      ; NUMBER OF REGISTERS SAVED
3364
3365                                ; *AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
3366                                ; *OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
3367                                ; *SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
3368 021774 113737 003201 002135  MOVB   @#REINTO+25, @#WRFROM+25 ; SAVE UPPER RHAS
3369
3370
3371                                ; *COMPARE REGISTERS BEFORE UNLOAD COMMAND
3372                                ; *WITH AFTER GO
3373
3374 022002 004037 043510      JSR    RO, @#COMPAR    ; COMPARE
3375 022006 003154                REINTO ; GOOD BUFFER
3376 022010 002110                WRFROM ; TEST BUFFER

```



```

3377 022012 000023          19.          ;NUMBER
3378 022014 022022          1$          ;RETURN FOR ERROR
3379 022016 022022          1$          ;SAME
3380 022020 022042          2$          ;RETURN FOR GOOD COMPARISON
3381 022022 013705 047620  1$:          MOV      @#ERWORD,R5 ;GETTING READY TO INDEX
3382 022026 060505          ADD      R5,R5        ;DOUBLE ERROR WORD
3383 022030 016537 001630 042264  MOV      RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
3384
3385 022036 104001          ERROR    1           ;IMPROPER REGISTER CHANGE
3386
3387
3388 022040 000207          RTS      PC          ;AFTER UNLOAD COMMAND
3389
3390
3391
3392 022042 052712 000040          2$:          ;*NOW GIVE INIT AND GET ALL GO AND PIP DOWN
3393 022046 013712 001774          BIS      #CLR,@R2    ;RH INITILIZE
3394 022052 012777 000001 157600  MOV      @#UNIT,@R2  ;REINSTATE UNIT NUMBER
3395
3396
3397
3398 022060 042737 000001 003162  MOV      #DMD,@RHRM ;SET DIAGNOSTIC MODE BIT
3399 022066 042737 020000 003204  ;THIS ENABLES COMMANDS WITHOUT MOL
3400 022074 052737 000200 003204  ;AND HOLDS RHLA FROM MOVING
3401 022102 017737 157566 003216  ;*CHANGE REGISTERS TO EXPECTED VALUE
3402
3403
3404
3405
3406 022110 004037 043306          BIC      #GO,@#REINTO+6 ;SAVED RHCSI
3407 022114 001632          BIC      #PIP,@#REINTO+30 ;SAVED RHDS1
3408 022116 002110          BIS      #DRY,@#REINTO+30 ;SAVED RHDS1
3409 022120 000023          MOV      @RHLA,@#REINTO+42 ;SAVED RHLA
3410
3411
3412
3413
3414
3415 022122 113737 003201 002135  ;*AFTER INITIALIZE SAVE REGISTERS SO THAT
3416
3417
3418
3419
3420
3421
3422
3423
3424
3425
3426
3427 022130 004037 043510          JSR      RD,@#SAVER   ;SAVE
3428 022134 003154          RHWC     ;FROM
3429 022136 002110          WRFROM  ;TO
3430 022140 000023          19.     ;NUMBER OF REGISTERS SAVED
3431
3432
3433
3434
3435
3436
3437
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3889
3890
3891
3892
3893
3894
3895
3896
3897
3898
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
3999
4000
4001
4002
4003
4004
4005
4006
4007
4008
4009
4010
4011
4012
4013
4014
4015
4016
4017
4018
4019
4020
4021
4022
4023
4024
4025
4026
4027
4028
4029
4030
4031
4032
4033
4034
4035
4036
4037
4038
4039
4040
4041
4042
4043
4044
4045
4046
4047
4048
4049
4050
4051
4052
4053
4054
4055
4056
4057
4058
4059
4060
4061
4062
4063
4064
4065
4066
4067
4068
4069
4070
4071
4072
4073
4074
4075
4076
4077
4078
4079
4080
4081
4082
4083
4084
4085
4086
4087
4088
4089
4090
4091
4092
4093
4094
4095
4096
4097
4098
4099
4100
4101
4102
4103
4104
4105
4106
4107
4108
4109
4110
4111
4112
4113
4114
4115
4116
4117
4118
4119
4120
4121
4122
4123
4124
4125
4126
4127
4128
4129
4130
4131
4132
4133
4134
4135
4136
4137
4138
4139
4140
4141
4142
4143
4144
4145
4146
4147
4148
4149
4150
4151
4152
4153
4154
4155
4156
4157
4158
4159
4160
4161
4162
4163
4164
4165
4166
4167
4168
4169
4170
4171
4172
4173
4174
4175
4176
4177
4178
4179
4180
4181
4182
4183
4184
4185
4186
4187
4188
4189
4190
4191
4192
4193
4194
4195
4196
4197
4198
4199
4200
4201
4202
4203
4204
4205
4206
4207
4208
4209
4210
4211
4212
4213
4214
4215
4216
4217
4218
4219
4220
4221
4222
4223
4224
4225
4226
4227
4228
4229
4230
4231
4232
4233
4234
4235
4236
4237
4238
4239
4240
4241
4242
4243
4244
4245
4246
4247
4248
4249
4250
4251
4252
4253
4254
4255
4256
4257
4258
4259
4260
4261
4262
4263
4264
4265
4266
4267
4268
4269
4270
4271
4272
4273
4274
4275
4276
4277
4278
4279
4280
4281
4282
4283
4284
4285
4286
4287
4288
4289
4290
4291
4292
4293
4294
4295
4296
4297
4298
4299
4300
4301
4302
4303
4304
4305
4306
4307
4308
4309
4310
4311
4312
4313
4314
4315
4316
4317
4318
4319
4320
4321
4322
4323
4324
4325
4326
4327
4328
4329
4330
4331
4332
4333
4334
4335
4336
4337
4338
4339
4340
4341
4342
4343
4344
4345
4346
4347
4348
4349
4350
4351
4352
4353
4354
4355
4356
4357
4358
4359
4360
4361
4362
4363
4364
4365
4366
4367
4368
4369
4370
4371
4372
4373
4374
4375
4376
4377
4378
4379
4380
4381
4382
4383
4384
4385
4386
4387
4388
4389
4390
4391
4392
4393
4394
4395
4396
4397
4398
4399
4400
4401
4402
4403
4404
4405
4406
4407
4408
4409
4410
4411
4412
4413
4414
4415
4416
4417
4418
4419
4420
4421
4422
4423
4424
4425
4426
4427
4428
4429
4430
4431
4432
4433
4434
4435
4436
4437
4438
4439
4440
4441
4442
4443
4444
4445
4446
4447
4448
4449
4450
4451
4452
4453
4454
4455
4456
4457
4458
4459
4460
4461
4462
4463
4464
4465
4466
4467
4468
4469
4470
4471
4472
4473
4474
4475
4476
4477
4478
4479
4480
4481
4482
4483
4484
4485
4486
4487
4488
4489
4490
4491
4492
4493
4494
4495
4496
4497
4498
4499
4500
4501
4502
4503
4504
4505
4506
4507
4508
4509
4510
4511
4512
4513
4514
4515
4516
4517
4518
4519
4520
4521
4522
4523
4524
4525
4526
4527
4528
4529
4530
4531
4532
4533
4534
4535
4536
4537
4538
4539
4540
4541
4542
4543
4544
4545
4546
4547
4548
4549
4550
4551
4552
4553
4554
4555
4556
4557
4558
4559
4560
4561
4562
4563
4564
4565
4566
4567
4568
4569
4570
4571
4572
4573
4574
4575
4576
4577
4578
4579
4580
4581
4582
4583
4584
4585
4586
4587
4588
4589
4590
4591
4592
4593
4594
4595
4596
4597
4598
4599
4600
4601
4602
4603
4604
4605
4606
4607
4608
4609
4610
4611
4612
4613
4614
4615
4616
4617
4618
4619
4620
4621
4622
4623
4624
4625
4626
4627
4628
4629
4630
4631
4632
4633
4634
4635
4636
4637
4638
4639
4640
4641
4642
4643
4644
4645
4646
4647
4648
4649
4650
4651
4652
4653
4654
4655
4656
4657
4658
4659
4660
4661
4662
4663
4664
4665
4666
4667
4668
4669
4670
4671
4672
4673
4674
4675
4676
4677
4678
4679
4680
4681
4682
4683
4684
4685
4686
4687
4688
4689
4690
4691
4692
4693
4694
4695
4696
4697
4698
4699
4700
4701
4702
4703
4704
4705
4706
4707
4708
4709
4710
4711
4712
4713
4714
4715
4716
4717
4718
4719
4720
4721
4722
4723
4724
4725
4726
4727
4728
4729
4730
4731
4732
4733
4734
4735
4736
4737
4738
4739
4740
4741
4742
4743
4744
4745
4746
4747
4748
4749
4750
4751
4752
4753
4754
4755
4756
4757
4758
4759
4760
4761
4762
4763
4764
4765
4766
4767
4768
4769
4770
4771
4772
4773
4774
4775
4776
4777
4778
4779
4780
4781
4782
4783
4784
4785
4786
4787
4788
4789
4790
4791
4792
4793
4794
4795
4796
4797
4798
4799
4800
4801
4802
4803
4804
4805
4806
4807
4808
4809
4810
4811
4812
4813
4814
4815
4816
4817
4818
4819
4820
4821
4822
4823
4824
4825
4826
4827
4828
4829
4830
4831
4832
4833
4834
4835
4836
4837
4838
4839
4840
4841
4842
4843
4844
4845
4846
4847
4848
4849
4850
4851
4852
4853
4854
4855
4856
4857
4858
4859
4860
4861
4862
4863
4864
4865
4866
4867
4868
4869
4870
4871
4872
4873
4874
4875
4876
4877
4878
4879
4880
4881
4882
4883
4884
4885
4886
4887
4888
4889
4890
4891
4892
4893
4894
4895
4896
4897
4898
4899
4900
4901
4902
4903
4904
4905
4906
4907
4908
4909
4910
4911
4912
4913
4914
4915
4916
4917
4918
4919
4920
4921
4922
4923
4924
4925
4926
4927
4928
4929
4930
4931
4932
4933
4934
4935
4936
4937
4938
4939
4940
4941
4942
4943
4944
4945
4946
4947
4948
4949
4950
4951
4952
4953
4954
4955
4956
4957
4958
4959
4960
4961
4962
4963
4964
4965
4966
4967
4968
4969
4970
4971
4972
4973
4974
4975
4976
4977
4978
4979
4980
4981
4982
4983
4984
4985
4986
4987
4988
4989
4990
4991
4992
4993
4994
4995
4996
4997
4998
4999
5000

```

K07

CZRJGB0.RP04/5/6 DSKLS CTRLRI  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046)  
T46

08-NOV-77 08:54 PAGE 86  
UNLOAD COMMAND TEST

SEG 0088

3433 022166 000207  
3434 022170  
3435

4\$:

RTS PC

;RETURN TO COMPARISON  
;GOOD COMPARISON

```

3436
3437 022170 032777 000020 156742 BIT #SW4, @SWR ;TEST FOR NO OFFSET OR RTC
3438 022176 001402 BEQ 6$ ;IF = 0, DO THE NEXT TWO TESTS
3439 022204 6$: ;CONTINUE WITH NEXT TWO TESTS
3440
3441
3442
3443 022206 012706 001000 MOV #STACK, SP ;RESET STACK
3444 022220 004737 042614 JSR PC, @CLDISK ;INIT AND SET UP GENERAL REG.
3445 ;AND UNIT NUMBER
3446 022224 012777 000001 157426 MOV #DMD, @RHMR ;SET DIAGNOSTIC MODE BIT
3447 ;THIS ENABLES COMMANDS WITHOUT MOL
3448 ;AND HOLDS RHLA FROM MOVING
3449
3450 ;*GIVE ONE INDEX PULSE TO CLEAR RHLA BEFORE THE START OF THIS TEST
3451 022232 052777 000004 157420 BIS #MINX, @RHMR ;SET INDEX PULSE
3452 022240 042777 000004 157412 BIC #MINX, @RHMR ;CLEAR INDEX
3453
3454 ;*TO ENABLE LOOP ON THIS TEST THE POSITIONER HAS TO
3455 ;*BE BROUGHT TO CENTER LINE
3456
3457 022246 017777 157424 157376 MOV @RHCC, @RHCA ;SET DESIRED CYLINDER TO RHCC
3458 022254 013711 002072 MOV @SEECOM, @R1 ;SEEK COMMAND TO RHCS1
3459 022260 005211 INC @R1 ;GO TO SEEK COMMAND
3460
3461 ;*FOUR SECTOR CLOCKS ARE GIVEN TO TAKE POSITIONER OFF OFFSET POSITION
3462 022262 012700 000004 MOV #4, RO ;COUNTER
3463 022266 012777 000011 157364 5$: MOV #MSTCK!DMD, @RHMR ;SET SECTOR CLOCK
3464 022274 012777 000001 157356 MOV #DMD, @RHMR ;RESET SECTOR CLOCK
3465 022302 005300 DEC RO ;COUNT
3466 022304 001370 BNE 5$ ;BRANCH IF NOT COMPLETE
3467
3468 022306 004737 042614 JSR PC, @CLDISK ;INIT AND SET UP GENERAL REG.
3469 ;AND UNIT NUMBER
3470 022312 012777 000001 157340 MOV #DMD, @RHMR ;SET DIAGNOSTIC MODE BIT
3471 ;THIS ENABLES COMMANDS WITHOUT MOL
3472 ;AND HOLDS RHLA FROM MOVING
3473
3474 022320 013777 002074 157312 MOV @OFSETC, @RHCS1 ;LOAD AN OFFSET BIT
3475 022326 012777 000001 157314 MOV #OF25, @RHOF ;SET AN OFFSET BIT
3476
3477 ;*SAVE REGISTERS FOR COMPARISON AFTER GO
3478 022334 004037 043306 JSR RO, @SAVER ;SAVE
3479 022340 001632 RHW ;FROM
3480 022342 003154 REINTO ;TO
3481 022344 000023 19. ;NUMBER OF REGISTERS SAVED
3482
3483 ;*GIVE GO TO OFFSET COMMAND
3484 022346 052777 000001 157264 BIS #GO, @RHCS1 ;GO TO OFFSET COMMAND
3485
3486 ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
3487 022354 052737 000001 003162 BIS #GO @REINTO+6 ;SAVED RHCS1
3488 022362 052737 020000 003204 BIS #PIP, @REINTO+30 ;SAVED RHDS1
3489 022370 042737 000200 003204 BIC #DRY, @REINTO+30 ;SAVED RHDS1
3490
3491 ;*AFTER GO HAS BEEN GIVEN TO OFFSET COMMAND

```

```

3492 ;*SAVE REGISTERS AGAIN SO THAT COMPARISONS CAN
3493 ;*BE DONE
3494
3495 022376 004037 043306 JSR RO, @#SAVER ;SAVE
3496 022402 001632 RHWC ;FROM
3497 022404 002110 WRFROM ;TO
3498 022406 000023 19. ;NUMBER OF REGISTERS SAVED
3499
3500 ;*AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
3501 ;*OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
3502 ;*SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
3503 022410 113737 003201 002135 MOVB @#REINT0+25, @#WRFROM+25; SAVE UPPER RHAS
3504
3505
3506 ;*COMPARE REGISTERS BEFORE OFFSET COMMAND
3507 ;*WITH AFTER GO
3508
3509 022416 004037 043510 JSR RO, @#COMPAR ;COMPARE
3510 022422 003154 REINT0 ;GOOD BUFFER
3511 022424 002110 WRFROM ;TEST BUFFER
3512 022426 000023 19. ;NUMBER
3513 022430 022436 1$ ;RETURN FOR ERROR
3514 022432 022436 1$ ;SAME
3515 022434 022456 2$ ;RETURN FOR GOOD COMPARISON
3516
3517 022436 013705 047620 1$: MOV @#ERWORD, R5 ;GETTING READY TO INDEX
3518 022442 060505 ADD R5, R5 ;DOUBLE ERROR WORD
3519 022444 016537 001630 042264 MOV RHWC-2(R5), @#REGADR ;FAILING REGISTER ADDRESS
3520 022452 104001 ERROR 1 ;IMPROPER REGISTER CHANGE
3521 ;AFTER OFFSET COMMAND
3522 ;WITH GO IS GIVEN
3523 022454 000207 RTS PC ;RETURN TO COMPARISON
3524
3525 ;*NOW GIVE INIT AND GET ALL GO AND PIP DOWN
3526
3527 022456 052712 000040 2$: BIS #CLR, @R2 ;RH INITILIZE
3528 022462 013712 001774 MOV @#UNIT, @R2 ;REINSTATE UNIT NUMBER
3529 022466 012777 000001 157164 MOV #DMD, @RHR ;SET DIAGNOSTIC MODE BIT
3530 ;THIS ENABLES COMMANDS WITHOUT MOL
3531 ;AND HOLDS RHLA FROM MOVING
3532
3533 ;*CHANGE REGISTERS TO EXPECTED VALUE
3534 022474 042737 000001 003162 BIC #GO, @#REINT0+6 ;SAVED RHCSI
3535 022502 042737 000001 003172 BIC #OF25, @#REINT0+16 ;SAVED RHOF
3536 022510 042737 020000 003204 BIC #PIP, @#REINT0+30 ;SAVED RHDS1
3537 022516 052737 000200 003204 BIS #DRY, @#REINT0+30 ;SAVED RHDS1
3538 022524 017737 157144 003216 MOV @RHLA, @#REINT0+42 ;SAVED RHLA
3539
3540 ;*AFTER INITIALIZE SAVE REGISTERS SO THAT
3541 ;*COMPARES CAN BE DONE
3542
3543 022532 004037 043306 JSR RO, @#SAVER ;SAVE
3544 022536 001632 RHWC ;FROM
3545 022540 002110 WRFROM ;TO
3546 022542 000023 19. ;NUMBER OF REGISTERS SAVED
3547

```

```

3548 ;*AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
3549 ;*OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
3550 ;*SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
3551 022544 113737 003201 002135 MOVB @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
3552
3553 ;*COMPARE REGISTERS AFTER INITIALIZE
3554 022552 004037 043510 JSR RO,@#COMPAR ;COMPARE
3555 022556 003154 REINTO ;GOOD BUFFER
3556 022560 002110 WRFROM ;TEST BUFFER
3557 022562 000023 19. ;NUMBER OF REGISTERS TO BE
3558 ;COMPARED
3559 022564 022572 3$ ;RETURN POINT FOR ERROR
3560 022566 022572 3$ ;SAME
3561 022570 022612 4$ ;RETURN POINT FOR GOOD COMPARISON
3562
3563 022572 013705 047620 3$: MOV @#ERWORD,R5 ;GETTING READY TO INDEX
3564 022576 060505 ADD R5,R5 ;DOUBLE ERROR WORD
3565 022600 016537 001630 042264 MOV RHC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
3566 022606 104001 ERROR 1 ;IMPROPER REGISTER
3567 ;CONTENTS AFTER GIVING AN
3568 ;INITIALIZE FOLLOWING A
3569 ;OFFSET COMMAND
3570 022610 000207 RTS PC ;RETURN TO COMPARISON
3571
3572 022612 4$ ;GOOD COMPARISON

```

```

3573
3574
3575 022614 012706 001000      MOV      #STACK, SP      ;RESET STACK
3576 022626 004737 042614      JSR      PC, @#CLDISK    ;INIT AND SET UP GENERAL REG.
3577                                     ;AND UNIT NUMBER
3578 022632 012777 000001 157020  MOV      #DMD, @RHMR     ;SET DIAGNOSTIC MODE BIT
3579                                     ;THIS ENABLES COMMANDS WITHOUT MOL
3580                                     ;AND HOLDS RHLA FROM MOVING
3581
3582                                     ;*GIVE ONE INDEX PULSE TO CLEAR RHLA BEFORE THE START OF THIS TEST
3583 022640 052777 000004 157012  BIS      #MINX, @RHMR    ;SET INDEX PULSE
3584 022646 042777 000004 157004  BIC      #MINX, @RHMR    ;CLEAR INDEX
3585
3586
3587 022654 013777 002076 156756  MOV      @#RETCL, @RHCS1 ;LOAD RETURN TO CENTER LINE COMMAND INTO RHCS1
3588
3589                                     ;*SAVE REGISTERS FOR COMPARISON AFTER GO
3590 022662 004037 043306      JSR      RO, @#SAVER     ;SAVE
3591 022666 001632                                     ;FROM
3592 022670 003154 REINTO                                     ;TO
3593 022672 000023 19.                                     ;NUMBER OF REGISTERS SAVED
3594
3595                                     ;*GIVE GO TO RETURN TO CENTER LINE COMMAND
3596 022674 052777 000001 156736  BIS      #GO, @RHCS1    ;GO TO RETURN TO CENTER COMMAND
3597
3598
3599                                     ;*FOUR SECTOR CLOCKS ARE GIVEN TO TAKE POSITIONER TO CENTER LINE
3600 022702 012700 000004      MOV      #4, RO         ;COUNTER
3601 022706 012777 000011 156744  SS:     MOV      #MSTCK!DMD, @RHMR ;SET SECTOR CLOCK
3602 022714 012777 000001 156736  MOV      #DMD, @RHMR    ;RESET SECTOR CLOCK
3603 022722 005300      DEC      RO             ;COUNT
3604 022724 001370      BNE     SS              ;BRANCH IF NOT COMPLETE
3605
3606
3607                                     ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
3608 022726 052737 000001 003162  BIS      #GO, @#REINTO+6 ;SAVED RHCS1
3609 022734 052737 020000 003204  BIS      #PIP, @#REINTO+30 ;SAVED RHDS1
3610 022742 042737 000200 003204  BIC      #DRY, @#REINTO+30 ;SAVED RHDS1
3611
3612                                     ;*AFTER GO HAS BEEN GIVEN TO RETURN TO CENTER LINE COMMAND
3613                                     ;*SAVE REGISTERS AGAIN SO THAT COMPARISONS CAN
3614                                     ;*BE DONE
3615 022750 004037 043306      JSR      RO, @#SAVER     ;SAVE
3616 022754 001632                                     ;FROM
3617 022756 002110 WRFROM                                     ;TO
3618 022760 000023 19.                                     ;NUMBER OF REGISTERS SAVED
3619
3620                                     ;*AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
3621                                     ;*OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
3622                                     ;*SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
3623 022762 113737 003201 002135  MOV      @#REINTO+25, @#WRFROM+25;SAVE UPPER RHAS
3624
3625
3626                                     ;*COMPARE REGISTERS BEFORE RETURN TO CENTER LINE COMMAND
3627                                     ;*WITH AFTER GO
3628 022770 004037 043510      JSR      RO, @#COMPAR    ;COMPARE

```

```

3629 022774 003154 REINTO ;GOOD BUFFER
3630 022776 002110 WRFROM ;TEST BUFFER
3631 023000 000023 19. ;NUMBER
3632 023002 023010 1$ ;RETURN FOR ERROR
3633 023004 023010 1$ ;SAME
3634 023006 023030 2$ ;RETURN FOR GOOD COMPARISON
3635
3636 023010 013705 047620 1$: MOV 2#ERWORD,R5 ;GETTING READY TO INDEX
3637 023014 060505 ADD R5,R5 ;DOUBLE ERROR WORD
3638 023016 016537 001630 042264 MOV RHWC-2(R5),2#REGADR ;FAILING REGISTER ADDRESS
3639 023024 104001 ERROR 1 ;IMPROPER REGISTER CHANGE
3640 ;AFTER RETURN TO CENTER LINE COMMAND
3641 ;WITH GO IS GIVEN
3642 023026 000207 RTS PC ;RETURN TO COMPARISON
3643
3644 ;*NOW GIVE INIT AND GET ALL GO AND PIP DOWN
3645
3646 023030 052712 000040 2$: BIS #CLR,2R2 ;RH INITILIZE
3647 023034 013712 001774 MOV 2#UNIT,2R2 ;REINSTATE UNIT NUMBER
3648 023040 012777 000001 156612 MOV #DMD,2RHMR ;SET DIAGNOSTIC MODE BIT
3649 ;THIS ENABLES COMMANDS WITHOUT MOL
3650 ;AND HOLDS RHLA FROM MOVING
3651
3652 ;*CHANGE REGISTERS TO EXPECTED VALUE
3653 023046 042737 000001 003162 BIC #GO,2#REINTO+6 ;SAVED RHCS1
3654 023054 042737 020000 003204 BIC #PIP,2#REINTO+30 ;SAVED RHDS1
3655 023062 052737 000200 003204 BIS #DRY,2#REINTO+30 ;SAVED RHDS1
3656 023070 017737 156600 003216 MOV 2RHLA,2#REINTO+42;SAVED RHLA
3657
3658 ;*AFTER INITIALIZE SAVE REGISTERS SO THAT
3659 ;*COMPARES CAN BE DONE
3660 023076 004037 043306 JSR RD,2#SAVER ;SAVE
3661 023102 001632 RHWC ;FROM
3662 023104 002110 WRFROM ;TO
3663 023106 000023 19. ;NUMBER OF REGISTERS SAVED
3664
3665 ;*AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
3666 ;*OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
3667 ;*SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
3668 023110 113737 003201 002135 MOVB 2#REINTO+25,2#WRFROM+25;SAVE UPPER RHAS
3669
3670 ;*COMPARE REGISTERS AFTER INITIALIZE
3671 JSR RD,2#COMPAR ;COMPARE
3672 023116 004037 043510 REINTO ;GOOD BUFFER
3673 023122 003154 WRFROM ;TEST BUFFER
3674 023124 002110 19. ;NUMBER OF REGISTERS TO BE
3675 023126 000023 ;COMPARED
3676 ;RETURN POINT FOR ERROR
3677 023130 023136 3$ ;SAME
3678 023132 023136 3$ ;RETURN POINT FOR GOOD COMPARISON
3679 023134 023156 4$
3680
3681
3682 023136 013705 047620 3$: MOV 2#ERWORD,R5 ;GETTING READY TO INDEX
3683 023142 060505 ADD R5,R5 ;DOUBLE ERROR WORD
3684 023144 016537 001630 042264 MOV RHWC-2(R5),2#REGADR ;FAILING REGISTER ADDRESS

```

D08

CZRJGBD.RP04/5/6 DSKLS CTRLR1  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046)  
T50

08-NOV-77 08:54 PAGE 92  
RETURN TO CENTER LINE COMMAND TEST

SEQ 0094

3685 023152 104001  
3686  
3687  
3688  
3689 023154 000207  
3690  
3691 023156  
3692

ERROR 1

;IMPROPER REGISTER  
;CONTENTS AFTER GIVING AN  
;INITIALIZE FOLLOWING A RETURN TO  
;CENTER LINE COMMAND  
;RETURN TO COMPARISON

RTS PC

4\$:

;GOOD COMPARISON



```

3693
3694
3695 023160 012706 001000      MOV      #STACK,SP      ;RESET STACK
3696 023172 004737 042614      JSR      PC,@#CLDISK    ;INIT AND SET UP GENERAL REG.
3697                                     ;AND UNIT NUMBER
3698 023176 012777 000001 156454  MOV      #DMD,@RHMR     ;SET DIAGNOSTIC MODE BIT
3699                                     ;THIS ENABLES COMMANDS WITHOUT MOL
3700                                     ;AND HOLDS RHLA FROM MOVING
3701
3702
3703                                     ;*GIVE ONE INDEX PULSE TO CLEAR RHLA BEFORE THE START OF THIS TEST
3704
3705 023204 052777 000004 156446  BIS      #MINX,@RHMR    ;SET INDEX PULSE
3706 023212 042777 000004 156440  BIC      #MINX,@RHMR    ;CLEAR INDEX
3707
3708 023220 013777 002046 156412  MOV      @#RECALI,@RHCS1 ;LOAD RECALIBRATE COMMAND INTO RHCS1
3709
3710                                     ;*SAVE REGISTERS FOR COMPARISON AFTER GO
3711 023226 004037 043306      JSR      RO,@#SAVER     ;SAVE
3712 023232 001632                                     ;FROM
3713 023234 003154 REINTO                                     ;TO
3714 023236 000023 19.                                     ;NUMBER OF REGISTERS SAVED
3715
3716                                     ;*GIVE GO TO RECALIBRATE COMMAND
3717 023240 052777 000001 156372  BIS      #GO,@RHCS1    ;GO TO RECALIBRATE COMMAND
3718
3719
3720                                     ;*FOUR SECTOR CLOCKS ARE GIVEN TO TAKE POSITIONER TO CYLINDER 0
3721 023246 012700 000004      MOV      #4,RO          ;COUNTER
3722 023252 012777 000011 156400 5$:  MOV      #MSTCK!DMD,@RHMR ;SET SECTOR CLOCK
3723 023260 012777 000001 156372  MOV      #DMD,@RHMR     ;RESET SECTOR CLOCK
3724 023266 005300      DEC      RO            ;COUNT
3725 023270 001370      BNE      5$            ;BRANCH IF NOT COMPLETE
3726
3727
3728                                     ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
3729 023272 052737 000001 003162  BIS      #GO,@#REINTO+6 ;SAVED RHCS1
3730 023300 052737 020000 003204  BIS      #PIP,@#REINTO+30 ;SAVED RHDS1
3731 023306 042737 000200 003204  BIC      #DRY,@#REINTO+30 ;SAVED RHDS1
3732
3733                                     ;*AFTER GO HAS BEEN GIVEN TO RECALIBRATE COMMAND
3734                                     ;*SAVE REGISTERS AGAIN SO THAT COMPARISONS CAN
3735                                     ;*BE DONE
3736 023314 004037 043306      JSR      RO,@#SAVER     ;SAVE
3737 023320 001632                                     ;FROM
3738 023322 002110 WRFROM                                     ;TO
3739 023324 000023 19.                                     ;NUMBER OF REGISTERS SAVED
3740
3741                                     ;*AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
3742                                     ;*OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
3743                                     ;*SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
3744 023326 113737 003201 002135  MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
3745
3746
3747                                     ;*COMPARE REGISTERS BEFORE RECALIBRATE COMMAND
3748                                     ;*WITH AFTER GO

```

# F08

CZRJGB0,RP04/5/6 DSKLS CTRLR1 MACY11 30(1046) 08-NOV-77 08:54 PAGE 94  
 CZRJGB.P11 08-NOV-77 08:30 T51 RECALIBRATE COMMAND TEST

SEQ 0096

```

3749 023334 004037 043510 JSR RO, @#COMPAR ; COMPARE
3750 023340 003154 REINTO ; GOOD BUFFER
3751 023342 002110 WRFROM ; TEST BUFFER
3752 023344 000023 19. ; NUMBER
3753 023346 023354 1$ ; RETURN FOR ERROR
3754 023350 023354 1$ ; SAME
3755 023352 023374 2$ ; RETURN FOR GOOD COMPARISON
3756
3757 023354 013705 047620 1$: MOV @#ERWORD, R5 ; GETTING READY TO INDEX
3758 023360 060505 ADD R5, R5 ; DOUBLE ERROR WORD
3759 023362 016537 001630 042264 MOV RHWC-2(R5), @#REGADR ; FAILING REGISTER ADDRESS
3760 023370 104001 ERROR 1 ; IMPROPER REGISTER CHANGE
3761 ; AFTER RECALIBRATE COMMAND
3762 ; WITH GO IS GIVEN
3763 023372 000207 RTS PC ; RETURN TO COMPARISON
3764
3765 ; *NOW GIVE INIT AND GET ALL GO AND PIP DOWN
3766
3767 023374 052712 000040 2$: BIS #CLR, @R2 ; RH INITILIZE
3768 023400 013712 001774 MOV @#UNIT, @R2 ; REINSTATE UNIT NUMBER
3769 023404 012777 000001 156246 MOV #DMD, @RHMR ; SET DIAGNOSTIC MODE BIT
3770 ; THIS ENABLES COMMANDS WITHOUT MOL
3771 ; AND HOLDS RHLA FROM MOVING
3772
3773 ; *CHANGE REGISTERS TO EXPECTED VALUE
3774 023412 042737 000001 003162 BIC #GO, @#REINTO+6 ; SAVED RHCS1
3775 023420 042737 020000 003204 BIC #PIP, @#REINTO+30 ; SAVED RHDS1
3776 023426 052737 000200 003204 BIS #DRY, @#REINTO+30 ; SAVED RHDS1
3777 023434 017737 156234 003216 MOV @RHLA, @#REINTO+42 ; SAVED RHLA
3778
3779 ; *AFTER INITIALIZE SAVE REGISTERS SO THAT
3780 ; *COMPARES CAN BE DONE
3781 023442 004037 043306 JSR RO, @#SAVER ; SAVE
3782 023446 001632 RHWC ; FROM
3783 023450 002110 WRFROM ; TO
3784 023452 000023 19. ; NUMBER OF REGISTERS SAVED
3785
3786 ; *AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
3787 ; *OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
3788 ; *SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
3789 023454 113737 003201 002135 MOVB @#REINTO+25, @#WRFROM+25 ; SAVE UPPER RHAS
3790
3791
3792 ; *COMPARE REGISTERS AFTER INITIALIZE
3793 023462 004037 043510 JSR RO, @#COMPAR ; COMPARE
3794 023466 003154 REINTO ; GOOD BUFFER
3795 023470 002110 WRFROM ; TEST BUFFER
3796 023472 000023 19. ; NUMBER OF REGISTERS TO BE
3797 ; COMPARED
3798 023474 023502 3$ ; RETURN POINT FOR ERROR
3799 023476 023502 3$ ; SAME
3800 023500 023522 4$ ; RETURN POINT FOR GOOD COMPARISON
3801
3802 023502 013705 047620 3$: MOV @#ERWORD, R5 ; GETTING READY TO INDEX
3803 023506 060505 ADD R5, R5 ; DOUBLE ERROR WORD
3804 023510 016537 001630 042264 MOV RHWC-2(R5), @#REGADR ; FAILING REGISTER ADDRESS
  
```

G08

CZRJGBO RPO4/5/6 DSKLS CTRLRI  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046)  
TS1

08-NOV-77 08:54 PAGE 95  
RECALIBRATE COMMAND TEST

SEQ 0097

3805 023516 104001  
3806  
3807  
3808  
3809 023520 000207  
3810  
3811 023522

ERROR 1

; IMPROPER REGISTER  
; CONTENTS AFTER GIVING AN  
; INITIALIZE FOLLOWING A  
; RECALIBRATE COMMAND  
; RETURN TO COMPARISON

RTS PC

4\$:

; GOOD COMPARISON



108

CZRJGBD RPO4/5/6 DSKLS CTRLR1  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 97  
T52 RELEASE COMMAND TEST

SEQ 0099

3868 023676 000207  
3869 023700  
3870

2\$: RTS PC ;RETURN TO COMPARISON

```

3871
3872
3873
3874
3875
3876 023736 012706 001000      MOV      #STACK, SP      ;RESET STACK
3877 023750 004737 042614      JSR      PC, @#CLDISK    ;INIT AND SET UP GENERAL REGISTERS
3878
3879                                     ;*THESE ARE REGULAR SET UPS FOR SEARCH COMMAND
3880 023754 012777 000025 155662  MOV      #21., @RHDS1    ;DESIRED SECTOR/TRACK REGISTER
3881                                     ;TRACK 0 SECTOR 21
3882 023762 005077 155664      CLR      @RHCA           ;DESIRED CYLINDER =0
3883 023766 012777 010000 155654  MOV      #FMT22, @RHOF   ;FORMAT BIT=1 (16 BITS PER WORD)
3884 023774 013711 002054      MOV      @#SERCH, @R1    ;FILL SEARCH COMMAND IN RHCS1
3885
3886                                     ;*NOW SAVE REGISTERS STARTING FROM RHWC IN WRITE FROM BUFFER
3887 024000 004037 043306      JSR      RO, @#SAVER     ;SAVE REGISTERS FOR COMPARISON
3888                                     ;AT THE END OF THE SEARCH
3889 024004 001632      RHWC      ;START SAVING FROM RHWC
3890 024006 003154      REINTO   ;SAVE INTO REINTO
3891 024010 000023      19.      ;NUMBER OF REGISTERS SAVED
3892
3893
3894                                     ;*NOW THE DIAGNOSTIC MODE BIT WILL BE SET
3895                                     ;*AND THE SEARCH OPERATION STARTED
3896
3897 024024 005037 001200      CLR      @#STMP1        ;THIS WILL HAVE THE EXPECTED
3898                                     ;VALUE OF RHLA REGISTER
3899
3900 024030 013700 001660      MOV      @#RHMR, RO      ;NOW RO HAS MAINTENANCE REG. ADDR.
3901 024034 017703 155604      MOV      @RHDS1, R3     ;GET DESIRED SECTOR/TRACK REG.
3902 024040 042703 177400      BIC      #177400, R3     ;GET SECTOR ONLY
3903 024044 010337 053544      MOV      R3, @#SECTR    ;DUPLICATE SECTOR
3904 024050 012710 000001      MOV      #DMD, @RO      ;S
3905 024054 052777 000001 155556  BIS      #GO, @RHCS1     ;GO
3906 024062 052710 000010      BIS      #MSTCK, @RO    ;SET SECTOR CLOCK
3907 024066 042710 000010      BIC      #MSTCK, @RO    ;CLEAR SECTOR CLOCK
3908 024072 000240      NOP                    ;ALLOW TIME BETWEEN SECTOR CLOCKS
3909 024074 052710 000010      BIS      #MSTCK, @RO    ;SET SECTOR CLOCK
3910 024100 042710 000010      BIC      #MSTCK, @RO    ;CLEAR SECTOR CLOCK
3911 024104 000240      NOP                    ;ALLOW TIME BETWEEN SECTOR CLOCKS
3912 024106 052710 000014      BIS      #MINX!MSTCK, @RO ;SET INDEX AND SECTOR CLOCK
3913 024112 012710 000001      MOV      #DMD, @RO      ;RESET INDEX AND SECTOR CLOCK
3914 024116 005703      TST      R3             ;IF SECTOR REQUIRED JUMP OUT
3915 024120 001555      BEQ     11$            ;BRANCH OF SECTOR ZERO REQUIRED
3916
3917                                     ;*AFTER THE INDEX PULSE RHLA WILL BE CHECKED TO BE ZERO
3918                                     ;*AND STMP4 WILL BE SET UP TO COUNT BYTES
3919 024122 012737 001140 001206 1$:  MOV      #608., @#STMP4  ;THERE ARE 608 BYTES PER SECTOR
3920 024130 017737 155540 001126  MOV      @RHLA, @#SBDDAT ;SAVE RHLA
3921 024136 017737 155502 001720  MOV      @RHDS1, @#DST   ;SAVE DESIRED SECTOR TRACK
3922 024144 023737 001200 001126  CMP      @#STMP1, @#SBDDAT ;RHLA SHOULD BE HAVE EXTENSION
3923                                     ;FIELD EQUAL TO ZERO
3924 024152 001414      BEQ     2$             ;BRANCH IF GOOD
3925 024154 013737 053544 001202  MOV      @#SECTR, @#STMP2 ;GET SECTOR SOUGHT
3926 024162 160337 001202      SUB     R3, @#STMP2    ;STMP2 NOW HAS PRESENT SECTOR

```

```

3927 024166 012746 001140
3928 024172 163716 001206
3929 024176 012637 001204
3930 024202 104024
3931
3932
3933
3934
3935
3936
3937
3938
3939
3940
3941 024204 012702 000010
3942 024210 012705 000002
3943 024214 000404
3944 024216 052710 000012
3945 024222 042710 000012
3946 024226 052710 000002
3947 024232 042710 000002
3948 024236 005302
3949 024240 001372
3950 024242 005337 001206
3951 024246 012702 000007
3952 024252 005305
3953 024254 001360
3954
3955
3956
3957
3958
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968 024256 012705 000100
3969
3970 024262 012701 000177
3971
3972 024266 012702 000007
3973 024272 052710 000012
3974 024276 042710 000012
3975 024302 052710 000002
3976 024306 042710 000002
3977 024312 005302
3978 024314 001372
3979 024316 005337 001206
3980 024322 001436
3981 024324 005301
3982

```

```

MOV #608.,-(SP) ;NUMBER OF BYTES PER SECTOR
SUB @#STMP4 (SP) ;(SP)HAS PRESENT BYTE NUMBER
MOV (SP)+,@#STMP3 ;PRESENT BYTE NUMBER
ERROR 24 ;LOOK AHEAD REGISTER AT THE BEGINING OF A
;SECTOR IS IN ERROR

```

```

;#NOW THE 304 WORDS WILL START
;#FOR FIRST BYTE CLOCK WILL BE INDEPENDENT OF
;#SECTOR CLOCK THEN IT WILL COINCIDE FOREVER TILL
;#THE BEGINNING OF NEXT SECTOR

```

```

;#ONE WORD ONLY THAT IS TWO BYTES

```

```

2$: MOV #8.,R2 ;BYTE
MOV #2,R5 ;BYTES PER WORD
BR 4$
3$: BIS #MSTCK!MCLK,@R0 ;SET SECTOR AND CLOCK
BIC #MSTCK!MCLK,@R0 ;CLEAR SECTOR AND CLOCK
4$: BIS #MCLK,@R0 ;SET CLOCK
BIC #MCLK,@R0 ;CLEAR CLOCK
DEC R2 ;BYTE COUNTER
BNE 4$ ;BRANCH IF BYTE NOT COMPLETE
DEC @#STMP4 ;BYTE COUNT DOWN
MOV #7,R2 ;SETUP FOR SECOND BYTE
DEC R5 ;IS WORD COMPLETE?
BNE 3$ ;BRENCH IF NOT COMPLETE
;TO GIVE SECTOR CLOCK AND CLOCK

```

```

;#NOW 303 WORDS ARE LEFT ALL ARE IDENTICAL
;#THAT IS 606 IDENTICAL BYTES WILL BE GIVEN
;#RHLA WILL BE CHECKED STAR TO COUNT AFTER
;#BEGINNING OF SECTOR PULSE
;#AFTER 128 BYTES (2 BYTES ARE ALREADY GIVEN)
;#SO 127 MORE
;#THEN RHLA WILL BE CHECKED AFTER 128 MORE BYTES
;#THEN RHLA WILL BE CHECKED AFTER 256 MORE BYTES
;#THEN THE TOTAL OF 608 BYTES WILL BE COMPLETED
;#AND RHLA WILL BE MADE READY FOR NEXT SECTOR
;#AND RHLA WILL BE CHECKED

```

```

MOV #64.,R5 ;R5 WILL KEEP TRACK WHEN
;EXTENSION FIELD IS TO BE CHECKED
MOV #127.,R1 ;FIRST TIME CHECK EXTENSION FIELD
;AFTER 127 MORE BYTES
5$: MOV #7,R2 ;CLOCKS PER BYTE COUNTER
BIS #MSTCK!MCLK,@R0 ;SET SECTOR CLOCK AND CLOCK
BIC #MSTCK!MCLK,@R0 ;CLEAR SECTOR CLOCK AND CLOCK
6$: BIS #MCLK,@R0 ;SET CLOCK
BIC #MCLK,@R0 ;RESET CLOCK
DEC R2 ;COUNT DOWN CLOCKS PER BYTE
BNE 6$ ;BRANCH IF BYTE NOT COMPLETE
DEC @#STMP4 ;COUNT DOWN BYTES
BEQ 10$ ;BRANCHOUT IF 608 BYTES DONE
DEC R1 ;COUNT DOWN NUMBER OF BYTES
;TO CHECK EXTENSION FIELD

```

```

3983 024326 001357          BNE      5$          ;BRANCH IF EXTENSION FIELD NOT
3984                          ;TO BE CHECKED YET
3985
3986                          ;*NOW THE EXTENSION FIELD OF THE LOOK AHEAD REGISTER
3987                          ;*WILL BE CHECKED
3988
3989 024330 062737 000020 001200  ADD      #20,2#STMP1 ;GET TO THE NEXT EXTENSION
3990 024336 017737 155332 001126  MOV      2#RHLA,2#SBDDAT ;GET RHLA FOR COMPARISON
3991
3992 024344 017737 155274 001720  MOV      2#RHDEST,2#DST ;SAVE DESIRED SECTOR TRACK
3993 024352 023737 001200 001126  CMP      2#STMP1,2#SBDDAT ;CHECK VALUE OF RHLA
3994 024360 001414          BEQ      7$          ;BRANCH IF GOOD
3995 024362 013737 053544 001202  MOV      2#SECTR,2#STMP2 ;GET SECTOR SOUGHT
3996 024370 160337 001202          SUB      R3,2#STMP2 ;STMP2 NOW HAS PRESENT SECTOR
3997 024374 012746 001140          MOV      #608,-(SP) ;NUMBER OF BYTES PER SECTOR
3998 024400 163716 001206          SUB      2#STMP4,(SP) ;(SP) HAS PRESENT BYTE NUMBER
3999 024404 012637 001204          MOV      (SP)+,2#STMP3 ;PRESENT BYTE NUMBER
4000 024410 104025          ERROR    25          ;LOOK AHEAD ERROR IN THE MIDDLE
4001                          ;OF A SECTOR IS IN ERROR
4002
4003 024412 060505          7$:      ADD      R5,R5          ;GET NEXT STEP TO CHECK EXTENSION FIELD
4004 024414 010501          MOV      R5,R1          ;PUT IN COUNTER
4005 024416 000723          BR       5$          ;BRANCH BACK SECTOR
4006                          ;IS NOT COMPLETE
4007 024420 062737 000020 001200 10$:     ADD      #20,2#STMP1
4008 024426 052710 000010          BIS      #MSTCK,2#RO ;THESE TWO INSTRUCTIONS GIVE
4009 024432 042710 000010          BIC      #MSTCK,2#RO ;ONE SECTOR CLOCK EXTRA
4010 024436 000240          NOP                      ;ALLOW TIME BETWEEN SECTOR CLOCK
4011 024440 052710 000010          BIS      #MSTCK,2#RO ;THESE TWO INSTRUCTIONS GIVE
4012 024444 042710 000010          BIC      #MSTCK,2#RO ;ONE SECTOR CLOCK EARLY
4013                          ;BEFORE THE NEXT SECTOR
4014 024450 005303          DEC      R3          ;IS REQUIRED NO OF SECTORS COMPLETE
4015 024452 001223          BNE     1$          ;BRANCH IF NOT
4016
4017                          ;*NOW THE REQUIRED SECTOR IS REACHED
4018                          ;*ONE SECTOR CLOCK WILL BE GIVEN TO GET SECTOR PULSE
4019                          ;*DOWN AND HENCE ATA UP
4020
4021 024454 012702 000010          11$:    MOV      #8,R2          ;8 CLOCKS
4022 024460 052710 000002          12$:    BIS      #MCLK,2#RO ;SET CLOCK
4023 024464 042710 000002          BIC      #MCLK,2#RO ;CLEAR CLOCKS
4024 024470 005302          DEC      R2          ;COUND DOWN
4025 024472 001372          BNE     12$          ;BRANCH IF 8 NOT DONE
4026 024474 052710 000012          BIS      #MSTCK!MCLK,2#RO ;SET SECTOR AND CLOCK
4027 024500 042710 000012          BIC      #MSTCK!MCLK,2#RO ;CLEAR SECTOR AND CLOCK
4028
4029                          ;*NOW ALL REGISTERS WILL BE COMPARED
4030                          ;*SO FILL EXPECTED VALUE INTO SAVED LOCATIONS
4031
4032 024504 052737 100000 003162  BIS      #SC,2#REINTO+6 ;INCLUDE SC IN SAVED RHCS1
4033 024512 053737 002016 003200  BIS      2#ATTENT,2#REINTO+24 ;FILL APPROPRIATE ATTENTION
4034                          ;IN SAVED RHAS
4035 024520 052737 000001 003202  BIS      #DMD,2#REINTO+26 ;SET DMD IN RHMR SAVED
4036 024526 052737 100000 003204  BIS      #ATA,2#REINTO+30 ;SET ATA IN RHDS1 SAVED
4037 024534 013737 001200 003216  MOV      2#STMP1,2#REINTO+42 ;MOVE EXPECTED VALUE
4038                          ;INTO RHLA SAVED

```



```

4039
4040
4041
4042
4043 024542 004037 043306
4044 024546 001632
4045 024550 002110
4046 024552 000023
4047
4048
4049
4050
4051 024554 113737 003201 002135
4052
4053
4054
4055
4056 024562 004037 043510
4057 024566 003154
4058
4059 024570 002110
4060 024572 000022
4061 024574 024602
4062 024576 024602
4063 024600 024622
4064 024602 013705 047620 13$:
4065 024606 060505
4066 024610 016537 001630 042264
4067 024616 104001
4068 024620 000207
4069 024622
4070

```

```

; *AFTER SEARCH COMMAND
; *SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE
JSR RO, @SAVER ; SAVE
RHWC ; FROM
WRFROM ; TO
19. ; NUMBER

; *AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
; *OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
; *SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
MOVB @REINTO+25, @WRFROM+25; SAVE UPPER RHAS

; *COMPARE REGISTERS BEFORE SEARCH WITH AFTER
JSR RO, @COMPAR ; COMPAR
REINTO ; GO BUFFER
WRFROM ; TEST BUFFER
18. ; NUMBER
13$ ; RETURN FOR ERROR
13$ ; SAME
14$ ; RETURN FOR GOOD COMPARISON
MOV @ERWORD, R5 ; GETTING READY TO INDEX
ADD R5, R5 ; DOUBLE ERROR WORD
MOV RHWC-2(R5), @REGADR ; FAILING REG. ADDRESS
ERROR 1 ; CONTENTS OF REGISTER
RTS PC ; CHANGED AT END OF
; SEARCH

```

CZRJGBD.RP04/5/6 DSKLS CTRLR1  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046)  
T54

08-NOV-77 08:54 PAGE 102  
LOOK AHEAD REGISTER

**N08**

SEQ 0104

4071

4072  
4073  
4074  
4075  
4076  
4077  
4078  
4079  
4080  
4081  
4082  
4083  
4084  
4085  
4086  
4087  
4088  
4089  
4090  
4091  
4092  
4093  
4094  
4095  
4096  
4097  
4098  
4099  
4100  
4101  
4102  
4103  
4104  
4105  
4106  
4107  
4108  
4109  
4110  
4111  
4112  
4113  
4114  
4115  
4116  
4117  
4118  
4119  
4120  
4121  
4122  
4123  
4124  
4125  
4126  
4127

.SBTTL READ/WRITE ADDRESSING VIA RHMR

```

MOV #STACK, SP ;RESET STACK
MOV #SECGAP, R0 ;POINTER
MOV #304, R1 ;COUNTER
1$: MOV #-1, (R0)+ ;CLEAR "DISK" AREA TO ALL ONES
DEC R1 ;
BNE 1$ ;
JSR PC, CLDISK ;THIS IS USED TO SET UP GENERAL
;REGISTER CORRESPONDENCE

; *THESE ARE TO SET UP FOR DISKLESS USE ONLY
MOV #FMT22, @#WCYL ;FORMAT 22=16 BIT WORDS AND
;CYLINDER 0
CLR @#WSECTR ;TRACK=0, SECTOR=0
CLR @#WKEY1 ;KEY1=0
CLR @#WKEY2 ;KEY2=0
MOV #256, @#FNWORD ;256 DATAWORDS
JSR R5, @#CRC ;GO TO CALCULATE CRC
WCYL
GCRC

; *THESE ARE REGULAR SETUPS FOR RH11 & "WRFROM" OUTPUT BUFFER
MOV #-260, @RHWC ;256 DATA WORDS 4 HEADER WORDS
MOV #WRFROM, R0 ;BUS ADDRESS TO BE
MOV R0, @RHBA ;BUFFER "WRFROM"
MOV #259, R5 ;COUNTER
MOV #FMT22, (R0)+ ;FORMAT =16 BIT WORD

2$: CLR (R0)+ ;SECTOR=0, TRACK=0, KEYS=0, ALL DATA=0
DEC R5 ;& CYLINDER=0.... SO CLEAR ALL "WRFROM"
BNE 2$ ;CONTINUE IF ALL 259 NOT COMPLETE
CLR @RHDSST ;TRACK=0, SECTOR=0

MOV @WRIFOR, @R1 ;GET READY FOR WRITE HEADER
;AND DATA WITH 62 IN RHCS1
CLR @ERFLGS ;CLEAR ERROR FLAG
MOV #FMT22, @RHOF ;FORMAT BIT=1 16 BIT WORDS
CLR @RHCA ;CYLINDER 0
JSR PC, @COMWHD ;WRITE HEADER AND DATA FROM "WRFROM"
;INTO THE RHMR REGISTER AND BACK INTO
;CORE "DISK" AREA

; *IF THE PROGRAM COMES BACK HERE WITHOUT ERROR
; *PRINT OUTS FROM THE "COMWHD" ROUTINE THAT MEANS
; *ALL HEADER ON DISK IS GOOD IE. ONLY DATA IS
; *TO BE CHECKED TO SEE IF IT IS ZERO
    
```

```

4128
4129
4130
4131
4132
4133 025056 017737 154550 001126      MOV    @RHWC,$BDDAT    ;MOVE WORD COUNTER INTO BAD DATA
4134 025064 001401                      BEQ    5$              ;SHOULD HAVE COUNTED UP TO ZERO
4135 025066 104040                      ERROR  40              ;RHWC DID NOT = 0 AFTER A WRITE
4136                                     ;HEADER AND DATA
4137
4138 025070 005737 002006      5$:  TST    @#ERFLGS      ;HAVE ANY ERRORS OCCURED?
4139 025110 005037 052416      CLR    @#WECC1        ;CLEAR ECC
4140 025114 005037 052420      CLR    @#WECC2
4141
4142                                     ;*REINTO BUFFER IS FILLED WITH EXPECTED DATA OF ALL 0'S
4143
4144 025120 004037 042532      JSR    RO,@#CLAREA    ;CLEAR "REINTO"
4145 025124 003154                      REINTO                ;FROM
4146 025126 004214                      REINTO+(<272.*2>)    ;TO
4147 025130 000000                      .WORD 0               ;FILL WITH ZEROS
4148 025132 005037 002006      CLR    @#ERFLGS      ;CLEAR ERROR FLAG
4149
4150
4151                                     ;*COMPARE "REINTO" BUFFER WITH "DISK" BUFFER
4152 025136 004037 043510      JSR    RO,@#COMPAR    ;CHECK
4153 025142 003154                      REINTO                ;GOOD BUFFER
4154 025144 051416                      DISK                  ;TEST BUFFER
4155 025146 000421                      273.                 ;NUMBER OF WORDS CHECKED
4156 025150 025156                      3$                   ;RETURN POINT FOR ERROR HEADER
4157 025152 025162                      4$                   ;RETURN POINT FOR ERROR DATA
4158 025156 104007      3$:  ERROR  7           ;READ ERROR10 NEXT
4159 025160 000207      RTS    PC             ;RETURN TO "COMPAR"
4160 025162 104010      4$:  ERROR  10        ;WORD NOS 1 TO 256 ARE
4161                                     ;DATA WORDS
4162                                     ;257 AND 258 ARE ECC
4163                                     ;ZEROED OUT
4164                                     ;259 TO 273 TOLERANCE GAP
4165 025164 000207      RTS    PC
4166

```

```

4167
4168 025170 012706 001000      MOV      #STACK, SP      ;RESET STACK
4169
4170 025202 012700 051320      MOV      #SECGAP, R0     ; POINTER
4171 025206 012701 000460      MOV      #304., R1       ; COUNTER
4172 025212 005020      1$: CLR      (R0)+           ; CLEAR SIMULATED "DISK" AREA IN CORE
4173 025214 005301      DEC      R1
4174 025216 001375      BNE     1$
4175 025220 004737 042614      JSR     PC, @#CLDISK     ; THIS IS USED TO SET GENERAL REGISTERS
4176
4177      ; *THESE ARE TO BE SETUP FOR DISKLESS USE ONLY
4178
4179 025224 012737 010000 052636      MOV      #FMT22, @#WCYL  ; FORMAT 22 = 16 BIT WORDS AND
4180      ; CYLINDER 0
4181 025232 012737 000001 052640      MOV      #1, @#WSECTR    ; TRACK=0, SECTOR=1
4182 025240 005037 052642      CLR      @#WKEY1         ; KEY1=0
4183 025244 005037 052644      CLR      @#WKEY2         ; KEY2=0
4184 025250 012737 000400 052676      MOV      #256., @#FNWORD ; 256 DATA WORDS
4185 025256 004537 044022      JSR     R5, @#CRC        ; GO TO CALCULATE CRC
4186 025262 052636
4187 025264 052646
4188
4189      ; *THESE ARE REGULAR SETUPS FOR THE RH11 AND "WRFROM" BUFFER
4190
4191 025266 012777 177374 154336      MOV      #-260., @RHWC   ; 256 DATA WORDS 4 HEADER WORDS
4192 025274 012700 002110      MOV      #WRFROM, R0     ; THESE TWO INSTRUCTIONS GETS
4193 025300 010077 154330      MOV      R0, @RHBA       ; ADDR. OF WRFROM INTO R0 AND
4194      ; BUS ADDRESS REGISTER
4195 025304 012720 010000      MOV      #FMT22, (R0)+   ; FORMAT=16 BIT WORDS
4196      ; CYLINDER=0
4197 025310 012720 000001 2$: MOV      #1, (R0)+       ; TRACK=0, SECTOR=1, KEYS=0
4198 025314 005020      CLR      (R0)+           ; KEY1=0
4199 025316 005020      CLR      (R0)+           ; KEY2=0
4200 025320 012705 000400      MOV      #256., R5       ; COUNTER
4201 025324 012720 177777 3$: MOV      #-1, (R0)+     ; MOVE ALL ONES FOR DATA
4202 025330 005305      DEC      R5
4203 025332 001374      BNE     3$
4204 025334 012777 000001 154302      MOV      #1, @RH DST    ; BRANCH IF DATA NOT COMPLETE
4205      ; TRACK=0 SECTOR=1
4206
4207 025354 013711 002064      MOV      @#WRIFOR, @R1   ; GET READY FOR WRITE HEADER AND
4208      ; DATA WITH 62 IN RHCS1
4209 025360 005037 002006      CLR      @#ERFLG$       ; CLEAR ERROR FLAG
4210 025364 012777 010000 154256      MOV      #FMT22, @RHOF   ; FORMAT BIT=1 (16 BIT WORDS)
4211 025372 005077 154254      CLR      @RHCA          ; CYLINDER =0
4212 025376 004737 052462      JSR     PC, @#COMWHD     ; WRITE HEADER AND DATA INTO "DISK" AREA
4213
4214      ; *IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
4215      ; *FROM THE "COMWHD" ROUTINE THAT MEANS ALL THE HEADER ON "DISK"
4216      ; *IS GOOD IE. ONLY THE DATA IS TO BE CHECKED TO SEE IF IT IS
4217      ; *ALL ONES AND WRITE DATA GAP AND TOLERANCE GAP TO SEE IF
4218      ; *THEY ARE ALL ZEROS, - ECC1 AND ECC2 ARE NOT CHECKED.
4219
4220      ; *RHWC IS CHECKED TO BE = 0
4221
4222 025402 017737 154224 001126      MOV      @RHWC, $BDDAT   ; LOAD WORD COUNTER JUST IN CASE

```

```

4223 025410 001401 BEQ 6$ ; SHOULD BE = 0
4224 025412 104040 ERROR 40 ; RHWIC DOES NOT = 0 AFTER A WRITE
4225 ; HEADER AND DATA IS COMPLETED
4226
4227 025414 005737 002006 6$: TST @#ERFLG$ ; HAVE ANY ERRORS OCCURRED?
4228 025434 005037 052416 CLR @#WECC1 ; CLEAR ECC
4229 025440 005037 052420 CLR @#WECC2 ; CLEAR ECC
4230
4231 ; *FILL "REINTO" BUFFER WITH EXPECTED DATA OF ALL 1'S
4232
4233 025444 004037 042532 JSR R0,@#CLAREA ; FILL REINTO BUFFER
4234 025450 003154 REINTO ; FROM
4235 025452 004152 REINTO+<255.*2> ; TO
4236 025454 177777 .WORD -1 ; DATA
4237 025456 004037 042532 JSR R0,@#CLAREA ; FILL REST
4238 025462 004154 REINTO+<256.*2> ; FROM
4239 025464 004214 REINTO+<272.*2> ; TO
4240 025466 000000 0 ; DATA
4241
4242 025470 005037 002006 CLR @#ERFLG$ ; CLEAR ERROR FLAG
4243
4244 ; *NOW COMPARE "DISK" BUFFER WITH "REINTO" BUFFER IN CORE
4245
4246 025474 004037 043510 JSR R0,@#COMPAR ; CHECK
4247 025500 003154 REINTO ; GOOD BUFFER
4248 025502 051416 DISK ; TEST BUFFER
4249 025504 000421 273. ; NUMBER OF WORDS CHECKED
4250 025506 025514 4$ ; RETURN POINT FOR ERROR HEADER
4251 025510 025520 5$ ; RETURN POINT FOR ERROR DATA
4252 025514 104007 4$: ERROR 7 ; READ ERROR 10 NEXT
4253 025516 000207 RTS PC ; RETURN TO COMPARE
4254 025520 104010 5$: ERROR 10 ; WORD NOS 1 TO 256 ARE
4255 ; DATA WORDS
4256 ; WORD NOS 257 AND 258
4257 ; ARE ECC WHICH HAVE BEEN
4258 ; ZEROED
4259 ; WORD NOS 259
4260 ; IS DATA GAP
4261 ; WORD NOS 260 TO 273
4262 ; ARE TOLERANCE GAP
4263 025522 000207 RTS PC ; RETURN TO COMPARE
4264
4265
4266

```



```

4323
4324 025742 017737 153664 001126      MOV    @RHC, $BDDAT    ; LOAD AND TEST FOR ZERO
4325 025750 001401                      BEQ    6$              ; RHC SHOULD = 0
4326 025752 104040                      ERROR  40              ; RHC DID NOT = 0 AFTER A WRITE
4327                                         ; HEADER AND DATA WAS COMPLETED
4328
4329                                         ; *ONLY ECC1 AND ECC2 ARE NOT CHECKED
4330
4331 025754 005737 002006      6$:   TST    @#ERFLG$      ; HAVE ANY ERRORS OCCURED?
4332 025774 005037 052416      CLR    @#WECC1        ; CLEAR ECC
4333 026000 005037 052420      CLR    @#WECC2        ; CLEAR ECC
4334
4335                                         ; *FILL "REINTO" BUFFER WITH EXPECTED DATA
4336
4337 026004 004037 042532      JSR    RO, @#CLAREA   ; FILL REINTO BUFFER
4338 026010 003154                      REINTO ; FROM
4339 026012 004152                      REINTO+(<255.*2>)    ; TO
4340 026014 052525                      .WORD  52525         ; DATA
4341 026016 004037 042532      JSR    RO, @#CLAREA   ; FILL REST
4342 026022 004154                      REINTO+(<256.*2>)    ; FROM
4343 026024 004214                      REINTO+(<272.*2>)    ; TO
4344 026026 000000                      .WORD  0             ; DATA
4345 026030 005037 002006      CLR    @#ERFLG$      ; CLEAR ERROR FLAG
4346
4347                                         ; *NOW COMPARE "DISK" BUFFER WITH "REINTO" BUFFER IN CORE
4348
4349 026034 004037 043510      JSR    RO, @#COMPAR   ; CHECK
4350 026040 003154                      REINTO ; GOOD BUFFER
4351 026042 051416                      DISK   ; TEST BUFFER
4352 026044 000421                      273.  ; NUMBER OF WORDS CHECKED
4353 026046 026054                      4$    ; RETURN POINT FOR ERROR HEADER
4354 026050 026060                      5$    ; RETURN POINT FOR ERROR DATA
4355 026054 104007      4$:   ERROR  7          ; READ ERROR 10 NEXT
4356 026056 000207      RTS    PC            ; RETURN TO COMPARE
4357 026060 104010      5$:   ERROR  10         ; WORD NOS 1 TO 256 ARE
4358                                         ; DATA WORDS
4359                                         ; WORD NOS 257 AND 258
4360                                         ; ARE ECC WHICH HAVE BEEN
4361                                         ; ZEROED
4362                                         ; WORD NOS 259
4363                                         ; IS DATA GAP
4364                                         ; WORD NOS 260 TO 273
4365                                         ; ARE TOLERANCE GAP
4366 026062 000207      RTS    PC            ; RETURN TO COMPARE
4367
4368
4369
4370

```



```

4371      026066 012706 001000      MOV      #STACK, SP      ;RESET STACK
4372
4373
4374      026100 012700 051320      MOV      #SECGAP, R0     ; POINTER
4375      026104 012701 000460      MOV      #304, R1       ; COUNTER
4376      026110 012720 177777      1$:     MOV      #-1, (R0)+     ; CLEAR DISK AREA TO ALL ONES.
4377      026114 005301                DEC      R1              ;
4378      026116 001374                BNE     1$              ;
4379      026120 004737 042614      JSR      PC, CLDISK     ; THIS IS USED TO SET GENERAL
4380                                     ; REGISTERS
4381
4382                                     ; *THESE ARE TO SET UP FOR DISKLESS USE ONLY
4383
4384      026124 012737 010000 052636  MOV      #FMT22, @#WCYL ; FORMAT 22=16 BITWORDS AND
4385                                     ; CYLINDER 0
4386      026132 005037 052640      CLR      @#WSECTR      ; TRACK=0, SECTOR=0
4387      026136 005037 052642      CLR      @#WKEY1      ; KEY1=0
4388      026142 005037 052644      CLR      @#WKEY2      ; KEY2=0
4389      026146 012737 000400 052676  MOV      #256, @#FNWORD ; 256 DATAWORDS
4390      026154 004537 044022      JSR      R5, @#CRC     ; GO TO CALCULATE CRC
4391      026160 052636                WCYL
4392      026162 052646                GCRC
4393
4394                                     ; *THESE ARE REGULAR SETUPS
4395
4396      026164 012777 177374 153440  MOV      #-260, @RHWC   ; 256 DATA WORDS 4 HEADER WORDS
4397      026172 012700 002110      MOV      #WRFROM, R0   ; FROM BUFFER "WRFROM"
4398      026176 010077 153432      MOV      R0, @RHBA     ; IN BUS ADDRESS
4399      026202 012705 000403      MOV      #259, R5      ; COUNTER
4400      026206 012720 010000      MOV      #FMT22, (R0)+ ; FORMAT =16 BIT WORD
4401                                     ; CYLINDER=0
4402      026212 005020 2$:     CLR      (R0)+         ; SECTOR=0, TRACK=0, KEYS=0, ALL DATA=0
4403      026214 005305                DEC      R5            ; COUNT
4404      026216 001375                BNE     2$            ; BRANCH IF ALL 259 NOT COMPLETE
4405      026220 005077 153420      CLR      @RHDST       ; TRACK=0, SECTOR=0
4406
4407
4408      026236 013711 002064      MOV      @#WRIFOR, @R1 ; GET READY FOR WRITE HEADER
4409                                     ; AND DATA WITH 62 IN RHCS1
4410      026242 005037 002006      CLR      @#ERFLG$     ; CLEAR ERROR FLAG
4411      026246 012777 010000 153374  MOV      #FMT22, @RHOF ; FORMAT BIT=1 16 BIT WORDS
4412      026254 005077 153372      CLR      @RHCA        ; CYLINDER 0
4413      026260 012777 000001 153372  MOV      #DMD, @RHMR  ; SET DIAGNOSTIC MODE
4414      026266 052777 000001 153344  BIS      #GO, @RHCS1  ; GO
4415      026274 000240                NOP
4416      026276 052777 000001 153334  BIS      #GO, @RHCS1  ; THIS GO SHOULD SET PGE
4417
4418      026304 004737 042220      JSR      PC, @#PUTREG  ; SAVE REGISTERS
4419      026310 032737 002000 001714  BIT      #PGE, @#CSI  ; IS PGE SET
4420      026316 001404                BEQ     3$            ; BRANCH IF GOOD
4421      026320 013737 001636 001122  MOV      @#RHCS2, @#SBDADR
4422      026326 104037                ERROR 37              ; PGE DID NOT SET WHEN A WRITE
4423                                     ; WAS ATTEMPTED WITH ONE IN PROGRESS
4424      026330 3$:

```



J09

CZRJGB0,RP04/5/6 DSKLS CTRLR1  
CZRJGB.P11 08-NOV-77 09:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 111  
T61 PROGRAM ERROR RHCS2 #10

SEG 0113

4445  
4446  
4447  
4448  
4449  
4450  
4451  
4452  
4453  
4454

026332 012706 001000

MOV #STACK,SP ;RESET STACK

K09

CZRJGB0 RPO4/5/6 DSKLS CTRLRI  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046)  
T62

08-NOV-77 08:54 PAGE 112  
READ HEADER AND DATA 1

SEQ 0114

4455  
4456 026716 012706 001000  
4457  
4458  
4459  
4460

MOV #STACK,SP ;RESET STACK

L09

CZRJGBD.RP04/5/6 DSKLS CTRLR1  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 113  
T63 READ HEADER AND DATA 2

SEQ 0115

4461  
4462 027302 012706 001000  
4463  
4464  
4465  
4466  
MOV #STACK,SP ;RESET STACK

CZRJG80.RP04/5/6 DSKLS CTRLRI  
CZRJG8.P11 08-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 114  
164 READ HEADER AND DATA 3

M09

SEQ 0116

4467  
4468  
4469  
4470  
4471  
4472

```

4473
4474 030226 012706 001000      MOV      #STACK,SP      ;RESET STACK
4475
4476 030240 004037 042532      JSR      RD,@#CLAREA    ;CLEAR SIMULATED DISK
4477 030244 051416                .WORD   DISK            ;FROM
4478 030246 052414                .WORD   DISK+776       ;TO
4479 030250 177400                .WORD   177400         ;DATA
4480
4481 030252 004037 042532      JSR      RD,@#CLAREA    ;CLEAR READ INTO BUFFER
4482 030256 003154                .WORD   REINTO         ;FROM
4483 030260 004152                .WORD   REINTO+776    ;TO
4484 030262 000000                .WORD   0              ;DATA
4485
4486      ;*THESE ARE TO SETUP FOR DISKLESS USE ONLY
4487
4488 030264 012737 010000 047500    MOV      #FMT22,@#CYL   ;CYLINDER 0 16 BITS PER WORD FORMAT
4489 030272 105037 047503                CLR     @#SECOTR+1     ;TRACK 0
4490 030276 112737 000001 047502    MOV     #1,@#SECOTR    ;SECTOR 1
4491 030304 005037 047504                CLR     @#KEY1         ;KEY1=0
4492 030310 005037 047506                CLR     @#KEY2         ;KEY2=0
4493 030314 012737 000012 047560    MOV     #10,@#DAWORD   ;NO. OF DATA WORDS
4494 030322 005037 047510                CLR     @#X            ;THIS IS A READ COMMAND
4495 030326 004537 044022      JSR      RS,@#CRC       ;GO TO CALCULATE CRC
4496 030332 047500                CYL
4497 030334 051400                WCRC
4498
4499      ;*THESE ARE REGULAR SETUPS
4500
4501 030336 004737 042614      JSR      PC,@#CLDISK    ;SETUP GENERAL REGISTERS
4502 030342 013711 002066      MOV     @#READAT,@R1   ;READ DATA INTO RHCS1=70
4503 030346 012777 177766 151256    MOV     #-10,@#RWC     ;10 DATA WORDS
4504 030354 012777 003154 151252    MOV     @#REINTO,@#RBA ;STARTING ADDRESS OF READ BUFFER
4505 030362 112746 000001                MOV     #1,-(SP)       ;IN LOWER BYTE GET SECTOR 1
4506 030366 112766 000000 000001    MOV     #0,1(SP)       ;GET TRACK0 IN UPPER BYTE
4507 030374 012677 151244      MOV     (SP)+,@#RHDST  ;TRACK/SECTOR IN RHDST
4508 030400 012777 014000 151242    MOV     #FMT22!ECI,@#RHOF ;16 BITS PER WORD
4509                                ;ECC CORRECTION INHIBIT BECAUSE
4510                                ;ECC IS NOT CHECKED HERE
4511 030406 005077 151240      CLR     @#RHCA         ;CYLINDER 0
4512 030424 005037 002006      CLR     @#ERFLGS       ;CLEAR ERROR FLAG
4513 030430 004737 047340      JSR      PC,@#COMHD    ;READ DATA
4514
4515      ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUT
4516      ;*FROM "COMHD" ROUTINE IN MEANS DATA IS TO BE CHECKED
4517
4518      ;*NOW THE DATA READ INTO "REINTO" BUFFER WILL
4519      ;*BE CHECKED, ONLY 10 WORDS SHOULD BE CHANGED
4520      ;*ALL OTHER WORDS SHOULD REMAIN UNCHANGED
4521      ;*THE "WRFROM" BUFFER IS FILLED WITH EXPECTED DATA AND CHECKED
4522
4523 030434 005737 002006      TST     @#ERFLGS       ;HAVE ANY ERRORS OCCURED?
4524 030442 004037 042532      JSR      RD,@#CLAREA    ;CLEAR BUFFER
4525 030446 002110                WRFROM                ;FROM
4526 030450 003106                WRFROM+776            ;TO
4527 030452 000000                0                      ;DATA
4528

```

4529	030454	004037	042532		JSR	RO, @#CLAREA	: FILL EXPECTED DATA
4530	030460	002110			WRFROM		: FROM
4531	030462	002132			WRFROM+22		: TO
4532	030464	177400			177400		: DATA
4533							
4534							
4535							
4536	030466	012700	002110		MOV	#WRFROM, RO	: GOOD DATA
4537	030472	012701	003154		MOV	#REINTO, R1	: DATA READ
4538	030476	012702	000400		MOV	#256., R2	: COUNTER
4539	030502	012737	000401	047620 1\$:	MOV	#257., @#ERWORD	: FOR ERROR WORD NO
4540	030510	022021			CMP	(RO)+, (R1)+	: COMPARE GOOD WITH READ BUFFER
4541	030512	001424			BEQ	2\$	: BRANCH IF GOOD
4542	030514	014037	001124		MOV	-(RO), @#\$GDDAT	: GOOD DATA
4543	030520	014137	001126		MOV	-(R1), @#\$BDDAT	: BAD DATA
4544	030524	160237	047620		SUB	R2, @#ERWORD	: ERROR WORD NO
4545	030530	005737	002006		TST	@#ERFLGS	: ANY ERRORS ALREADY THERE
4546	030534	001002			BNE	3\$	: IF YES BRANCH DO NOT TYPE HEADER
4547	030536	104004			ERROR	4	: ERROR ON READ DATA
4548	030540	000401			BR	4\$	: BRANCH TO AVOID PRINTING NEXT ERROR
4549	030542	104005		3\$:	ERROR	5	: WORD NO 1-10 ARE DATA
4550							: WORDS
4551							: WORD NOS 11-256 HAVE NOT BEEN
4552							: READ AND BUFFER SHOULD BE
4553							: ZERO IF OTHER THAN ZERO
4554							: WRONG NUMBER OF WORDS HAVE
4555							: BEEN READ IN THE DISK NOW
4556							: CONTAINS 177400 ALL 256
4557							: WORDS BUT ONLY 10 WORDS
4558							: SHOULD BE READ IN
4559							
4560	030544	022021		4\$:	CMP	(RO)+, (R1)+	: UNDO -(RO) AND -(R1) FOR ERROR
4561	030546	017746	150366		MOV	@SWR, -(SP)	: GET SWITCH SETTING
4562	030552	042716	177177		BIC	#177177, (SP)	: KEEP ONLY SWITCH 7 AND 8
4563	030556	022726	000200		CMP	#SW07, (SP)+	: IS 7 SET AND 8 RESET
4564	030564	005302		2\$:	DEC	R2	: COUNT
4565	030566	001345			BNE	1\$	: BRANCH IF NOT COMPLETE
4566							
4567							
4568							



```

4569
4570 ;*
4571
4572
4573 ;*
4574 ;*
4575 ;*
4576 ;*
4577
4578 030572 012706 001000 MOV #STACK,SP ;RESET STACK
4579
4580 ;*SET UP "REINTO" FOR WHAT IS TO BE READ
4581 030604 012701 003154 MOV #REINTO,R1 ;STARTING ADDRESS
4582 030610 012721 010000 MOV #FMT22,(R1)+ ;CYLINDER 0 FORMAT 16 BIT WORDS
4583 030614 012721 000401 MOV #401,(R1)+ ;TRACK=1, SECTOR=1
4584 030620 005021 CLR (R1)+ ;KEY1=0
4585 030622 005021 CLR (R1)+ ;KEY2=0
4586 030624 004037 042532 JSR RO,#CLAREA ;FILL "REINTO" BUFFER
4587 030630 003164 .WORD REINTO+(4*2) ;FROM
4588 030632 003232 .WORD REINTO+(23.*2) ;TO
4589 030634 070707 .WORD 070707 ;DATA
4590
4591 030636 012700 177776 MOV #177776,RO ;GETTING READY TO FLOAT 0
4592 030642 012701 003234 MOV #REINTO+(24.*2),R1 ;STARTING ADDRESS WHERE 177776 GOES
4593 030646 010021 1$: MOV RO,(R1)+ ;MOVE IN FLOATING 0
4594 030650 000261 SEC ;SET CARRY
4595 030652 006100 ROL RO ;GET 0 ONE BIT LEFT
4596 030654 103774 BCS 1$ ;BRANCH IF 16 NOT DONE
4597
4598 030656 004037 042532 JSR RO,#CLAREA ;FILL THE REST OF BUFFER WITH 0
4599 030662 003274 .WORD REINTO+(40.*2) ;FROM
4600 030664 004152 .WORD REINTO+776 ;TO
4601 030666 000000 .WORD 0 ;DATA
4602
4603 ;*SET UP SIMULATED DISK WITH WHAT IS TO BE READ
4604
4605 030670 004037 042532 JSR RO,#CLAREA ;FILL "DISK" BUFFER
4606 030674 051416 .WORD DISK ;FROM
4607 030676 051464 .WORD DISK+(19.*2) ;TO
4608 030700 070707 .WORD 070707 ;DATA
4609
4610 030702 012700 177776 MOV #177776,RO ;GETTING READY TO FLOAT ZEROS
4611 030706 012701 051466 MOV #DISK+(20.*2),R1 ;STARTING ADDRESS WHERE 177776 GOES
4612 030712 010021 2$: MOV RO,(R1)+ ;MOVE IN FLOATING 0
4613 030714 000261 SEC ;SET CARRY
4614 030716 006100 ROL RO ;GET 0 ONE BIT LEFT
4615 030720 103774 BCS 2$ ;BRANCH IF 16 NOT DONE
4616
4617 030722 004037 042532 JSR RO,#CLAREA ;FILL THE REST OF BUFFER WITH 177777
4618 030726 051526 .WORD DISK+(36.*2) ;FROM
4619 030730 052414 .WORD DISK+776 ;TO
4620 030732 177777 .WORD 177777 ;DATA
4621
4622 030734 004737 043340 JSR PC,#WRCHHD ;WRITE CHECK HEADER AND DATA
4623 ;CYLINDER 0, TRACK 1, SECTOR 1
4624

```

D10

CZRJGBO, RPO4/5/6 DSKLS CTRLRI  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 118  
T67 WRITE CHECK HEADER AND DATA

SEQ 0120

```

4625                                     ;*IF THE PROGRAM COMES BACK HERE THEN WRITE CHECK
4626                                     ;*HAS BEEN COMPLETED NOW WRITE CHECK ERROR BIT IS TO BE TESTED
4627
4628 030740 013746 001774   MOV      @#UNIT, -(SP)   ;GET UNIT NUMBER
4629 030744 052716 000100   BIS      #IR, (SP)     ;ONLY BIT 6 SHOULD BE SET
4630 030750 004737 042220   JSR      PC, @#PUTREG  ;SAVE REGISTERS
4631 030754 022637 001712   CMP      (SP)+, @#CS2  ;COMPARE RHCS2
4632 030760 001406         BEQ      4$            ;BRANCH IF GOOD
4633 030762 032712 040000   BIT      #WCE, @R2     ;WRITE CHECK ERROR HIGH?
4634 030766 001402         BEQ      3$            ;BRANCH IF ERROR NOT DUE TO "WCE"
4635 030770 104017         ERROR   17            ;RHDB CONTAINS FAILING WORD
4636 030772 000401         BR       4$            ;RHBA CONTAINS ADDRESS+2
4637                                     ;OF THE WORD IN MEMORY FROM
4638                                     ;THE DISK THAT DID NOT COMPARE
4639                                     ;TRE AND SC WILL BE SET DUE TO
4640                                     ;WCE
4641 030774 104017         3$:   ERROR   17            ;WCE CORRECTLY WAS NOT SET BUT SOME
4642                                     ;BITS OTHER THAN IR
4643                                     ;AND UNIT NO. WAS SET
4644
4645                                     ;*NOW CHECK MEMORY TO SEE IF NOTHING GOT DESTROYED
4646                                     ;*FILL "WRFROM" WITH WHAT SHOULD BE IN "REINTO" THEN CHECK
4647
4648 030776 012700 002110   4$:   MOV      #WRFROM, R0 ;STARTING ADDRESS
4649 031002 012720 010000   MOV      #FMT22, (R0)+ ;CYLINDER
4650 031006 012720 000401   MOV      #401, (R0)+   ;TRACK=1, SECTOR=1
4651 031012 005020         CLR      (R0)+         ;KEY1=0
4652 031014 005020         CLR      (R0)+         ;KEY2=0
4653
4654 031016 004037 042532   JSR      R0, @#CLAREA  ;FILL "WRFROM" BUFFER
4655 031022 002120         .WORD   WRFROM+(4*2) ;FROM
4656 031024 002166         .WORD   WRFROM+(23.*2);TO
4657 031026 070707         .WORD   070707       ;DATA
4658
4659 031030 012700 177776   MOV      #177776, R0   ;GETTING READY TO FLOAT 0
4660 031034 012701 002170   MOV      #WRFROM+(24.*2), R1 ;STARTING ADDRESS WHERE 177776 GOES
4661 031040 010021         MOV      R0, (R1)+    ;MOVE IN FLOATING 0
4662 031042 000261         SEC      ;SET CARRY
4663 031044 006100         ROL      R0           ;GET 0 ONE BIT LEFT
4664 031046 103774         BCS     5$           ;BRANCH IF 16 NOT DONE
4665
4666 031050 004037 042532   JSR      R0, @#CLAREA  ;FILL THE REST OF BUFFER WITH 0
4667 031054 002230         .WORD   WRFROM+(40.*2);FROM
4668 031056 003106         .WORD   WRFROM+776   ;TO
4669 031060 000000         .WORD   0            ;DATA
4670
4671                                     ;*NOW THE READ BUFFER WILL BE CHECKED
4672 031062 005037 002006   CLR      @#ERFLG$     ;CLEAR ERROR FLAG
4673
4674 031066 004037 043510   JSR      R0, @#COMPAR  ;CHECK
4675 031072 002110         WRFROM                ;GOOD BUFFER
4676 031074 003154         REINTO                ;TEST BUFFER
4677 031076 000400         256.                ;NUMBER OF WORDS CHECKED
4678 031100 031106         6$                  ;RETURN POINT FOR ERROR HEADER
4679 031102 031112         7$                  ;RETURN POINT FOR ERROR DATA
4680 031106 104004         6$:   ERROR   4            ;READ NEXT ERROR 5

```

E10

CZRJGB0.RP04/5/6 DSKLS\_CTRLR1  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 119  
T67 WRITE CHECK HEADER AND DATA

SEQ 0121

4681 031110 000207  
4682 031112 104005  
4683  
4684  
4685  
4686  
4687 031114 000207  
4688  
4689 031116 000240

7\$: RTS PC  
ERROR 5  
  
RTS PC  
10\$: NOP

: RETURN TO COMPARISON SUBROUTINE  
: DATA IN REINTO BUFFER GOT  
: CHANGED AFTER A WRITE  
: CHECK HEADER AND DATA COMMAND  
: WORD NO CONTAINS THE WORD  
: NUMBER THAT GOT CHANGED  
: RETURN TO COMPARISON SUBROUTINE  
  
: ONLY A BRANCH POINT

```

4690
4691
4692
4693
4694
4695
4696
4697
4698 031122 012706 001000
4699
4700
4701
4702 031134 012700 000001
4703 031140 012701 003154
4704 031144 010021 1$:
4705 031146 006100
4706 031150 103375
4707 031152 012700 177776
4708 031156 012701 003214
4709 031162 010021 2$:
4710 031164 000261
4711 031166 006100
4712 031170 103774
4713
4714 031172 004037 042532
4715 031176 003254
4716 031200 004152
4717 031202 000001
4718
4719
4720
4721 031204 012700 000001
4722 031210 012701 051416
4723 031214 010021 3$:
4724 031216 006100
4725 031220 103375
4726
4727 031222 012700 177776
4728 031226 012701 051456
4729 031232 010021 4$:
4730 031234 000261
4731 031236 006100
4732 031240 103774
4733
4734 031242 004037 042532
4735 031246 051516
4736 031250 052414
4737 031252 000000
4738
4739 031254 004737 043652
4740
4741
4742
4743
4744
4745

```

```

;*DATA TABLE
;*TOTAL OF 32 WORDS CONSISTING OF
;*16 WORDS OF FLOATING ONES (EG. 1, 2, 4, 10)
;*16 WORDS OF FLOATING ZEROS (EG. 177776, 177775)

MOV #STACK, SP ;RESET STACK
;SET UP "REINTO" FOR WHAT IS TO BE READ

MOV #1, RO ;GETTING READY TO FLOAT 1
MOV #REINTO, R1 ;STARTING ADDRESS WHERE 1 GOES
1$: MOV RO, (R1)+ ;MOVE FLOATING 1
ROL RO ;GET 1 ONE BIT LEFT
BCC 1$ ;BRANCH IF 16 NOT DONE
MOV #177776, RO ;GETTING READY TO FLOAT 0
MOV #REINTO+<16.*2>, R1 ;STARTING ADDRESS WHERE 177776 GOES
2$: MOV RO, (R1)+ ;MOVE IN FLOATING 0
SEC ;SET CARRY
ROL RO ;GET 0 ONE BIT LEFT
BCS 2$ ;BRANCH IF 16 NOT DONE

JSR RO, @#CLAREA ;FILL REST OF BUFFER WITH 1
.WORD REINTO+<32.*2> ;FROM
.WORD REINTO+776 ;TO
.WORD 1 ;WITH DATA

;SET UP SIMULATED DISK WITH WHAT IS TO BE READ

MOV #1, RO ;GETTING READY TO FLOAT 1
MOV #DISK, R1 ;STARTING ADDRESS WHERE 1 GOES
3$: MOV RO, (R1)+ ;MOVE FLOATING 1
ROL RO ;GET 1 ONE BIT LEFT
BCC 3$ ;BRANCH IF 16 NOT DONE

MOV #177776, RO ;GETTING READY TO FLOAT 0
MOV #DISK+<16.*2>, R1 ;STARTING ADDRESS WHERE 177776 GOES
4$: MOV RO, (R1)+ ;MOVE FLOATING 0
SEC ;SET CARRY
ROL RO ;GET 0 ONE BIT LEFT
BCS 4$ ;BRANCH IF 16 NOT DONE

JSR RO, @#CLAREA ;FILL REST OF BUFFER WITH 0
.WORD DISK+<32.*2> ;FROM
.WORD DISK+776 ;TO
.WORD 0 ;WITH DATA

JSR PC, @#WRCHDA ;WRITE CHECK DATA
;CYLINDER 0, TRACK 1, SECTOR 1
;KEYS 0, 32 WORDS.

;IF THE PROGRAM COMES BACK HERE THEN WRITE CHECK
;HAS BEEN COMPLETED NOW WRITE CHECK ERROR BIT IS TESTED

```

```

4746 031260 013746 001774      MOV      @#UNIT, -(SP)      ;GET UNIT NUMBER
4747 031264 052716 000100      BIS      #IR, (SP)         ;ONLY BIT 6 SHOULD BE SET
4748 031270 004737 042220      JSR      PC, @#PUTREG      ;SAVE REGISTERS
4749 031274 022637 001712      CMP      (SP)+, @#CS2      ;COMPARE RHCS2
4750 031300 001407              BEQ      6$                ;BRANCH IF GOOD
4751 031302 032737 040000 001712 BIT      #WCE, @#CS2        ;WRITE CHECK ERROR HIGH?
4752 031310 001402              BEQ      5$                ;BRANCH IF ERROR NOT DUE TO "WCE"
4753 031312 104017              ERROR   17                ;RHDB CONTAINS FAILING WORD
4754 031314 000401              BR      6$                ;RHBA CONTAINS ADDRESS+2
4755                                ;OF THE WORD IN MEMORY FROM
4756                                ;THE DISK THAT DID NOT COMPARE
4757                                ;TRE AND SC WILL BE SET DUE TO WCE
4758 031316 104017              5$:      ERROR   17        ;WCE WAS CORRECTLY NOT SET
4759                                ;BUT SOME BITS OTHER THAN
4760                                ;IR AND UNIT NO. WERE SET
4761                                ;
4762                                ;*NOW CHECK MEMORY TO SEE IF ANYTHING GOT DESTROYED
4763                                ;*FILL "WRFROM" WITH WHAT SHOULD BE IN REINTO THEN CHECK IT
4764                                ;
4765 031320 005037 002006      6$:      CLR      @#ERFLG$        ;CLEAR ERROR FLAG
4766 031324 012700 000001      MOV      #1, RO           ;GETTING READY TO FLOAT 1
4767 031330 012701 002110      MOV      #WRFROM, R1      ;START ADDRESS WHERE 1 GOES
4768 031334 010021              7$:      MOV      RO, (R1)+        ;MOVE FLOATING 1
4769 031336 006100              ROL      RO               ;GET 1 ONE BIT LEFT
4770 031340 103375              BCC     7$                ;BRANCH IF 16 NOT DONE
4771                                ;
4772 031342 012700 177776      MOV      #177776, RO      ;GETTING READY TO FLOAT 0
4773 031346 012701 002150      MOV      #WRFROM+<16.*2>, R1 ;STARTING ADDRESS WHERE 177776 GOES
4774 031352 010021              10$:     MOV      RO, (R1)+        ;MOVE IN FLOATING 0
4775 031354 000261              SEC                     ;SET CARRY
4776 031356 006100              ROL      RO               ;GET 0 ONE BIT LEFT
4777 031360 103774              BCS     10$              ;BRANCH IF CARRY SET
4778                                ;
4779 031362 004037 042532      JSR      RO, @#CLAREA      ;FILL REST OF BUFFER WITH 1
4780 031366 002210      .WORD   WRFROM+<32.*2>    ;FROM
4781 031370 003106      .WORD   WRFROM+776        ;TO
4782 031372 000001      .WORD   1                 ;WITH DATA
4783                                ;
4784                                ;*NOW THE READ BUFFER WILL BE CHECKED
4785                                ;
4786 031374 004037 043510      JSR      RO, @#COMPAR      ;CHECK
4787 031400 002110      WRFROM  ;GOOD BUFFER
4788 031402 003154      REINTO  ;TEST BUFFER
4789 031404 000400      256.   ;NUMBER OF WORDS CHECKED
4790 031406 031414      11$    ;RETURN POINT FOR ERROR HEADER
4791 031410 031420      12$    ;RETURN POINT FOR ERROR DATA
4792                                ;
4793                                ;
4794 031414 104004              11$:     ERROR   4          ;READ NEXT ERROR 5
4795 031416 000207              RTS      PC              ;RETURN TO COMPARISON SUBROUTINE
4796 031420 104005              12$:     ERROR   5          ;DATA IN REINTO BUFFER GOT
4797                                ;CHANGED AFTER A WRITE
4798                                ;CHECK DATA COMMAND
4799                                ;WORD NO CONTAINS THE WORD
4800                                ;NUMBER THAT GOT CHANGED
4801 031422 000207              RTS      PC              ;RETURN TO COMPARISON SUBROUTINE

```

H10

CZRJGBD.RP04/5/6 DSKLS\_CTRLR1  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 122  
T70 WRITE CHECK DATA

SEQ 0124

4802  
4803 031424 000240  
4804  
4805

13\$: NOP

:ONLY A BRANCH POINT

.SBTTL ERROR BIT FUNCTIONAL TESTS

```

4806
4807
4808
4809
4810
4811
4812
4813 031430 012706 001000      MOV      #STACK,SP      ;RESET STACK
4814 031442 004737 042614      JSR      PC,@#CLDISK   ;CLEAR DISK REGISTERS
4815
4816
4817
4818 031460 012700 003154      MOV      #REINTO,RO    ;BUFFER STARTING FOR 3 ERROR
4819                                ;REGISTERS
4820 031464 013720 001642      MOV      @#RHER1,(RO)+ ;RHER1 STORED IN REINTO
4821 031470 012720 000000      MOV      #0,(RO)+     ;BITS NOT TO BE CHECKED IN RHER1
4822 031474 013720 001646      MOV      @#RHER2,(RO)+ ;RHER2 STORED IN REINTO+4
4823 031500 012720 000000      MOV      #0,(RO)+     ;BITS NOT TO BE CHECKED IN RHER2
4824 031504 013720 001654      MOV      @#RHER3,(RO)+ ;RHER3 STORED IN REINTO+10
4825 031510 012720 000000      MOV      #0,(RO)+     ;BITS NOT TO BE CHECKED IN RHER3
4826
4827 031514 013704 001656      MOV      @#RHAS,R4     ;R4 HAS RHAS
4828 031520 013705 002016      MOV      @#ATTENT,R5  ;R5 HAS ATA BIT IN RHAS
4829 031524 012737 031552 001110  MOV      #2$,@#SLPERR ;THAT SHOULD SET WITH ERROR
4830                                ;RETURN POINT TO ERROR
4831 031532 012737 000003 001200  MOV      #3,@#STMP1   ;ERROR REGISTER COUNTER
4832 031540 012700 003154      MOV      #REINTO,RO   ;REGISTER BUFFER POINTER
4833
4834 031544 012002                1$:  MOV      (RO)+,R2     ;R2 HAS ADDRESS OF ERROR REG
4835 031546 012701 000001                MOV      #BIT0,R1    ;R1 WILL HAVE BIT UNDER TEST
4836 031552 052777 000040 150056 2$:  BIS      #CLR,@RHCS2  ;CLEAR RHCS2
4837 031560 013777 001774 150050  MOV      @#UNIT,@RHCS2 ;REINSTATE UNIT NO.
4838 031566 010112                MOV      R1,@R2     ;SET ERROR BIT
4839 031570 004737 042220      JSR      PC,@#PUTREG  ;READ AND SAVE REGISTERS
4840 031574 120537 001732      CMPB    R5,@#AS      ;ONLY THE BIT IN R5 SHOULD BE
4841                                ;SET IN RHAS
4842 031600 001401                BEQ     3$          ;LOOK @ RHDS1 IF GOOD
4843 031602 104020                ERROR  20          ;WITH THE SETTING OF ONE
4844                                ;ERROR BIT IN AN ERROR
4845                                ;REGISTER, THE CORRESPONDING
4846                                ;RHAS BIT DID NOT SET
4847
4848 031604 013746 001736                3$:  MOV      @#DS1 -(SP)  ;GET RHDS1
4849 031610 042716 001100      BIC     #VV!PROG,(SP); REMOVE VV AND PROG
4850 031614 022726 140600      CMP     #ATA!ERR!DPR!DRY,(SP)+; THESE BITS PLUS VV SHOULD BE IN RHDS1
4851 031620 001401                BEQ     4$          ;CHECK 'GO' NEXT, IF THIS WAS OK
4852 031622 104020                ERROR  20          ;WITH THE SETTING OF ONE
4853                                ;ERROR BIT, COMPOSITE ERROR
4854                                ;OR ATTENTION ACTIVE, OR
4855                                ;ONE OF THE OTHER
4856                                ;PERMANENT BITS DID NOT SET
4857
4858 031624 012777 000001 150006 4$:  MOV      #GO,@RHCS1  ;GIVE NO-OP
4859 031632 004737 042220      JSR      PC,@#PUTREG  ;SAVE REGISTERS
4860 031636 020112                CMP     R1,@R2     ;GO SHOULD NOT CLEAR ERROR
4861 031640 001410                BEQ     5$          ;FURTHER CHECK OF 'GO' FUNCTIONALITY

```





K10

CZRJGB0.RP04/5/6 DSKLS CTRLR1  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 125  
T71 ATTENTION WITH ERROR TEST

SEQ 0127

;THE DRIVE BIT POSITION

4918  
4919  
4920  
4921  
4922  
4923

```

4924
4925 032020 012706 001000      MOV      #STACK, SP          ;RESET STACK
4926
4927 032032 004037 042532      JSR      RD, @#CLAREA        ;CLEAR SIMULATED DISK
4928 032036 051416                .WORD   DISK                 ;FROM
4929 032040 052414                .WORD   DISK+776            ;TO
4930 032042 177400                .WORD   177400              ;DATA
4931
4932 032044 004037 042532      JSR      RD, @#CLAREA        ;CLEAR READ INTO BUFFER
4933 032050 003154                .WORD   REINTO              ;FROM
4934 032052 004152                .WORD   REINTO+776          ;TO
4935 032054 000000                .WORD   0                   ;DATA
4936
4937                               ;*THESE ARE TO SETUP FOR DISKLESS USE ONLY
4938
4939 032056 012737 010000 047500  MOV      #FMT22, @#CYL       ;CYLINDER 0 16 BITS PER WORD FORMAT
4940 032064 105037 047503      CLRB    @#SECOTR+1          ;TRACK 0
4941 032070 112737 000001 047502  MOV     #1, @#SECOTR        ;SECTOR 1
4942 032076 005037 047504      CLR     @#KEY1              ;KEY1=0
4943 032102 005037 047506      CLR     @#KEY2              ;KEY2=0
4944 032106 012737 000012 047560  MOV     #10., @#DAWORD      ;NO. OF DATA WORDS
4945 032114 005037 047510      CLR     @#X                 ;THIS IS A READ COMMAND
4946 032120 004537 044022      JSR     R5, @#CRC           ;GO TO CALCULATE CRC
4947 032124 047500      CYL
4948 032126 051400      WCRC
4949
4950                               ;*THESE ARE REGULAR SETUPS
4951
4952 032130 004737 042614      JSR     PC, @#CLDISK        ;SETUP GENERAL REGISTERS
4953 032134 013711 002066      MOV     @#READAT, @R1       ;READ DATA INTO RHCS1=70
4954 032140 012777 177766 147464  MOV     #-10., @RHWC        ;10 DATA WORDS
4955 032146 012777 003154 147460  MOV     @#REINTO, @RHBA     ;STARTING ADDRESS OF READ BUFFER
4956 032154 112746 000001      MOV     #1, -(SP)           ;IN LOWER BYTE GET SECTOR 1
4957 032160 112766 000000 000001  MOV     #0, 1(SP)           ;GET TRACK0 IN UPPER BYTE
4958 032166 012677 147452      MOV     (SP)+, @RH DST     ;TRACK/SECTOR IN RH DST
4959 032172 012777 014000 147450  MOV     #FMT22!ECI, @RHOF   ;16 BITS PER WORD
4960
4961                               ;ECC CORRECTION INHIBIT BECAUSE
4962 032200 005077 147446      CLR     @RHCA              ;ECC IS NOT CHECKED HERE
4963 032216 052777 000010 147412  BIS     #BAI, @RHCS2        ;CYLINDER 0
4964 032224 005037 002006      CLR     @#ERFLGS           ;SET BUS ADDRESS INHIBIT
4965 032230 004737 047340      CLR     @#ERFLGS           ;CLEAR ERROR FLAG
4966
4967                               ;READ DATA
4968
4969                               ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUT
4970                               ;*FROM "COMHD" ROUTINE IN MEANS DATA IS TO BE CHECKED
4971
4972                               ;*NOW THE DATA READ INTO "REINTO" BUFFER WILL
4973                               ;*BE CHECKED, ONLY ONE WORD SHOULD BE CHANGED
4974                               ;*ALL OTHER WORDS SHOULD REMAIN UNCHANGED
4975                               ;*THE "WRFROM" BUFFER IS FILLED WITH EXPECTED DATA AND CHECKED
4976
4975 032234 005037 002006      CLR     @#ERFLGS           ;CLEAR FLAG
4976 032240 004037 042532      JSR     RD, @#CLAREA        ;CLEAR BUFFER
4977 032244 002110                WRFROM                      ;FROM
4978 032246 003106                WRFROM+776                  ;TO
4979 032250 000000                0                           ;DATA

```

```

4980
4981
4982 032252 012737 177400 002110
4983
4984
4985
4986 032260 012700 002110
4987 032264 012701 003154
4988 032270 012702 000400
4989 032274 012737 000401 047620 1$:
4990 032302 022021
4991 032304 001424
4992 032306 014037 001124
4993 032312 014137 001126
4994 032316 160237 047620
4995 032322 005737 002006
4996 032326 001002
4997 032330 104004
4998 032332 000401
4999 032334 104005 3$:
5000
5001
5002
5003
5004
5005
5006
5007
5008
5009
5010 032336 022021 4$:
5011 032340 017746 146574
5012 032344 042716 177177
5013 032350 022726 000200
5014 032356 005302 2$:
5015 032360 001345
5016
5017
5018
5019
5020

```

```

; *EXPECTED DATA IS 177400 IN FIRST LOCATION ONLY
MOV #177400, @#WRFROM ; EXPECTED DATA
; *NOW READ DATA BUFFER IS CHECKED
MOV #WRFROM, R0 ; GOOD DATA
MOV #REINTO, R1 ; DATA READ
MOV #256, R2 ; COUNTER
MOV #257, @#ERWORD ; FOR ERROR WORD NO
CMP (R0)+, (R1)+ ; COMPARE GOOD WITH READ BUFFER
BEQ 2$ ; BRANCH IF GOOD
MOV -(R0), @#SGDDAT ; GOOD DATA
MOV -(R1), @#SBDDAT ; BAD DATA
SUB R2, @#ERWORD ; ERROR WORD NO
TST @#ERFLGS ; ANY ERRORS ALREADY THERE
BNE 3$ ; IF YES BRANCH DO NOT TYPE HEADER
ERROR 4 ; ERROR ON READ DATA
BR 4$ ; BRANCH TO AVOID PRINTING NEXT ERROR
ERROR 5 ; WORD NO 1-10 ARE DATA
; WORDS
; WORD NOS 11-256 HAVE NOT BEEN
; READ AND BUFFER SHOULD BE
; ZERO IF OTHER THAN ZERO
; WRONG NUMBER OF WORDS HAVE
; BEEN READ IN THE DISK NOW
; CONTAINS 177400 ALL 256
; WORDS BUT ONLY 10 WORDS
; SHOULD BE READ IN
; UNDO -(R0) AND -(R1) FOR ERROR
MOV @SWR, -(SP) ; GET SWITCH SETTING
BIC #177177, (SP) ; KEEP ONLY SWITCH 7 AND 8
CMP #SW07, (SP)+ ; IS 7 SET AND 8 RESET
DEC R2 ; COUNT
BNE 1$ ; BRANCH IF NOT COMPLETE

```

```

5021
5022 032364 012706 001000      MOV      #STACK,SP      ;RESET STACK
5023
5024 032410 004037 042532      JSR      R0, @#CLAREA   ;CLEAR SIMULATED DISK
5025 032414 051416                    .WORD   DISK            ;FROM
5026 032416 052414                    .WORD   DISK+776       ;TO
5027 032420 177400                    .WORD   177400         ;DATA
5028
5029
5030
5031      ;*THESE ARE TO SETUP FOR DISKLESS USE ONLY
5032 032422 012737 010000 047500    MOV      #FMT22, @#CYL  ;CYLINDER 0, 16 BITS PER WORD FORMAT
5033 032430 105037 047503                    CLR      @#SECTR+1     ;TRACK 0
5034 032434 112737 000001 047502    MOV      @#1, @#SECTR  ;SECTOR 1
5035 032442 005037 047504                    CLR      @#KEY1        ;KEY1=0
5036 032446 005037 047506                    CLR      @#KEY2        ;KEY2=0
5037 032452 012737 000001 047560    MOV      @#1, @#DAWORD ;NO. OF DATA WORDS
5038 032460 005037 047510                    CLR      @#X           ;THIS IS A READ COMMAND
5039 032464 004537 044022                    JSR      R5, @#CRC      ;GO TO CALCULATE CRC
5040 032470 047500                    CYL
5041 032472 051400                    WCRC
5042
5043      ;*THESE ARE REGULAR SETUPS
5044
5045 032474 004737 042614                    JSR      PC, @#CLDISK  ;SETUP GENERAL REGISTERS
5046 032500 013711 002066                    MOV      @#READAT, @R1 ;READ DATA INTO RHCS1=70
5047 032504 012777 177777 147120    MOV      @#-1, @RHWC   ;10 DATA WORDS
5048 032512 012777 160000 147114    MOV      @#160000, @RHBA ;STARTING ADDRESS OF READ BUFFER
5049 032520 052711 001400                    BIS      @#A16!A17, @R1 ;IS 760000
5050 032524 112746 000001                    MOV      @#1, -(SP)    ;IN LOWER BYTE GET SECTOR 1
5051 032530 112766 000000 000001    MOV      @#0, 1(SP)    ;GET TRACK0 IN UPPER BYTE
5052 032536 012677 147102                    MOV      (SP)+, @RHDST ;TRACK/SECTOR IN RHDST
5053 032542 012777 014000 147100    MOV      @#FMT22!ECI, @RHOF ;16 BITS PER WORD
5054
5055
5056 032550 005077 147076                    CLR      @RHCA         ;CYLINDER 0
5057 032566 052777 000010 147042    BIS      @#BAI, @RHCS2 ;SET BUS ADDRESS INHIBIT
5058 032574 005037 002006                    CLR      @#ERFLG$     ;CLEAR ERROR FLAG
5059 032600 004737 047340                    JSR      PC, @#COMHD   ;READ DATA
5060
5061
5062
5063 032604 011137 001126      1$:  MOV      @R1, @#$BDDAT ;TEST DATA
5064
5065 032610 022737 145670 001126    CMP      @#SC!TRE!DVA!A16!A17!RDY!70, @#$BDDAT ;COMPARE RHCS1
5066 032616 001406                    BEQ      2$           ;BRANCH IF GOOD
5067 032620 012737 144270 001124    MOV      @#SC!TRE!DVA!RDY!70, @#$GDDAT ;GOOD DATA
5068 032626 010137 042264                    MOV      R1, @#REGADR  ;REGISTER RHCS1
5069 032632 104001                    ERROR 1              ;REFERENCE NON EXISTANT
5070
5071
5072 032634 013746 001774      2$:  MOV      @#UNIT, -(SP) ;GET UNIT NUMBER
5073 032640 052716 004110                    BIS      @#NEM!IA!BAI, (SP) ;INCLUDE NEM BAI AND IA
5074 032644 012637 001124                    MOV      (SP)+, @#$GDDAT ;
5075 032650 011237 001126                    MOV      @R2, @#$BDDAT ;TEST DATA
5076 032654 023737 001124 001126    CMP      @#$GDDAT, @#$BDDAT ;COMPARE RHCS2

```



5097  
5098  
5099  
5100  
5101  
5102  
5103  
5104  
5105  
5106  
5107  
5108  
5109  
5110  
5111  
5112  
5113  
5114  
5115  
5116  
5117  
5118  
5119  
5120  
5121  
5122  
5123  
5124  
5125  
5126  
5127  
5128  
5129  
5130  
5131  
5132  
5133  
5134  
5135  
5136  
5137  
5138  
5139  
5140  
5141  
5142  
5143  
5144  
5145  
5146  
5147  
5148  
5149  
5150  
5151  
5152

```

; *DATA TABLE
; *TOTAL OF 32 WORDS CONSISTING OF
; *16 WORDS OF FLOATING ONES (EG. 1, 2, 4, 10)
; *16 WORDS OF FLOATING ZEROS (EG. 177776, 177775)
    
```

032730 012706 001000  
032742 004737 042614

```

MOV #STACK, SP ; RESET STACK
JSR PC, @#CLDISK ; INIT AND SET UP GENERAL REGISTERS
    
```

; \*SET UP "REINTO" FOR WHAT IS TO BE READ

032746 012700 000001  
032752 012701 003154

```

MOV #1, RO ; GETTING READY TO FLOAT 1
MOV #REINTO, R1 ; STARTING ADDRESS WHERE 1 GOES
    
```

1\$:

032756 010021  
032760 006100  
032762 103375

```

MOV RO, (R1)+ ; MOVE FLOATING 1
ROL RO ; GET 1 ONE BIT LEFT
BCC 1$ ; BRANCH IF 16 NOT DONE
    
```

032764 012700 177776  
032770 012701 003214

```

MOV #177776, RO ; GETTING READY TO FLOAT 0
MOV #REINTO+<16.*2>, R1 ; STARTING ADDRESS WHERE 177776 GOES
    
```

2\$:

032774 010021  
032776 000261  
033000 006100  
033002 103774

```

MOV RO, (R1)+ ; MOVE IN FLOATING 0
SEC ; SET CARRY
ROL RO ; GET 0 ONE BIT LEFT
BCS 2$ ; BRANCH IF 16 NOT DONE
    
```

033004 004037 042532  
033010 003254  
033012 004152  
033014 000001

```

JSR RO, @#CLAREA ; FILL REST OF BUFFER WITH 1
    
```

```

.WORD REINTO+<32.*2> ; FROM
.WORD REINTO+776 ; TO
.WORD 1 ; WITH DATA
    
```

; \*SET UP SIMULATED DISK WITH WHAT IS TO BE READ

033016 012700 000001  
033022 012701 051416

```

MOV #1, RO ; GETTING READY TO FLOAT 1
MOV #DISK, R1 ; STARTING ADDRESS WHERE 1 GOES
    
```

3\$:

033026 010021  
033030 006100  
033032 103375

```

MOV RO, (R1)+ ; MOVE FLOATING 1
ROL RO ; GET 1 ONE BIT LEFT
BCC 3$ ; BRANCH IF 16 NOT DONE
    
```

033034 012700 177776  
033040 012701 051456

```

MOV #177776, RO ; GETTING READY TO FLOAT 0
MOV #DISK+<16.*2>, R1 ; STARTING ADDRESS WHERE 177776 GOES
    
```

4\$:

033044 010021  
033046 000261  
033050 006100  
033052 103774

```

MOV RO, (R1)+ ; MOVE FLOATING 0
SEC ; SET CARRY
ROL RO ; GET 0 ONE BIT LEFT
BCS 4$ ; BRANCH IF 16 NOT DONE
    
```

033054 004037 042532  
033060 051516  
033062 052414  
033064 000000

```

JSR RO, @#CLAREA ; FILL REST OF BUFFER WITH 0
    
```

```

.WORD DISK+<32.*2> ; FROM
.WORD DISK+776 ; TO
.WORD 0 ; WITH DATA
    
```

; \*CHANGE FIFTH WORD TO 0 ON DISK

033066 005037 051426  
033072 005037 002006  
033076 004737 043652

```

CLR @#DISK+10 ; CLEAR FIFTH WORD ON DISK
CLR @#ERFLG$ ; CLEAR ERROR FLAG
    
```

```

JSR PC, @#WRCHDA ; WRITE CHECK DATA
; CYLINDER 0, TRACK 1, SECTOR 1
    
```

;KEYS 0, 32 WORDS.

;\*IF THE PROGRAM COMES BACK HERE THEN WRITE CHECK  
;\*HAS BEEN COMPLETED, NOW WRITE CHECK ERROR BIT IS TESTED  
;\*ALONG WITH RHWC FOR PROPER WORD COUNT AND RHBA FOR ADDRESS

```

5153
5154
5155
5156
5157
5158
5159 033102 013746 001774 MOV @#UNIT -(SP) ;GET UNIT NUMBER
5160 033106 052716 040300 BIS #IR!OR!WCE (SP) ;ONLY BIT 6 SHOULD BE SET
5161 033112 004737 042220 JSR PC,@#PUTREG ;SAVE REGISTERS
5162 033116 022637 001712 CMP (SP)+,@#CS2 ;COMPARE RHCS2
5163 033122 001407 BEQ 6$ ;BRANCH IF GOOD
5164 033124 032737 040000 001712 BIT #WCE,@#CS2 ;WRITE CHECK ERROR HIGH?
5165 033132 001002 BNE 5$ ;BRANCH IF ERROR NOT DUE TO "WCE"
5166 033134 104017 ERROR 17 ;RHDB CONTAINS FAILING WORD
5167 033136 000401 BR 6$ ;RHBA CONTAINS ADDRESS+2
5168 ;OF THE WORD IN MEMORY FROM
5169 ;THE DISK THAT DID NOT COMPARE
5170
5171 033140 104017 5$: ERROR 17 ;TRE AND SC WILL BE SET DUE TO WCE
5172 ;WCE WAS CORRECTLY NOT SET
5173 ;BUT SOME BITS OTHER THAN
5174 ;IR AND UNIT NO. WERE SET
5175 033142 005737 002040 6$: TST @#RH70 ;TEST FOR RH70 CONTROLLER
5176 033146 001414 BEQ 16$ ;SKIP RH70 CODE AND DO RH11 IF NOT
5177
5178 033150 022737 177750 001706 CMP #-24.,@#WC ;COMPARE RHWC AFTER A FORCED
5179 ;WRITE CHECK ERROR
5180 033156 001402 BEQ 17$ ;CHECK RHBA IF GOOD
5181 033160 104017 ERROR 17 ;WORD COUNT REGISTER IN ERROR AFTER A
5182 ;FORCED WRITE CHECK ERROR ON FIFTH WORD
5183 033162 000421 BR 15$ ;BRANCH TO CONTINUE TEST
5184
5185 033164 022737 003174 001710 17$: CMP #REINTO+(8.*2),@#BA ;COMPARE RHBA AFTER A FORCED
5186 ;WRITE CHECK ERROR IN FIFTH WORD
5187 033172 001415 BEQ 15$ ;CONTINUE IF GOOD
5188 033174 104017 ERROR 17 ;BUS ADDRESS REGISTER IN ERROR AFTER
5189 ;FORCED WRITE CHECK ERROR ON FIFTH WORD
5190 033176 000413 BR 15$ ;SKIP RH11 CODE AND CONTINUE WITH TEST
5191
5192 033200 022737 177745 001706 16$: CMP #-27.,@#WC ;COMPARE RHWC AFTER A FORCED
5193 ;WRITE CHECK ERROR
5194 033206 001402 BEQ 14$ ;CHECK RHBA IF GOOD
5195 033210 104017 ERROR 17 ;WORD COUNT REGISTER IN ERROR AFTER A
5196 ;FORCED WRITE CHECK ERROR ON FIFTH WORD
5197 033212 000405 BR 15$ ;BRANCH TO CONTINUE TEST
5198
5199 033214 022737 003166 001710 14$: CMP #REINTO+(5.*2),@#BA ;COMPARE RHBA AFTER FORCED
5200 ;WRITE CHECK ERROR IN FIFTH WORD
5201 033222 001401 BEQ 15$ ;CONTINUE IF GOOD
5202 033224 104017 ERROR 17 ;BUS ADDRESS REGISTER IN ERROR AFTER
5203 ;FORCED WRITE CHECK ERROR ON FIFTH WORD
5204
5205
5206 ;*NOW CHECK MEMORY TO SEE IF ANYTHING GOT DESTROYED
5207 ;*FILL "WRFROM" WITH WHAT SHOULD BE IN REINTO THEN CHECK
5208 033226 005037 002006 15$: CLR @#ERFLG$ ;CLEAR ERROR FLAG

```





5250  
5251  
5252  
5253  
5254  
5255  
5256  
5257  
5258  
5259  
5260  
5261  
5262  
5263  
5264  
5265  
5266  
5267  
5268  
5269  
5270

033336 012706 001000

MOV #STACK,SP ;RESET STACK

;\*NOW SEE THAT FORMAT ERROR BIT GOT SET

033720 004737 042220

BS:

JSR PC,@#PUTREG ;SAVE REGISTERS

033724 022737 100020 001716

CMP #FER!DCK,@#ER1

;FORMAT ERROR SHOULD BE SET  
;A 16 BIT PER WORD READ WAS ATTEMPTED  
;WHEN THE DISK HAD  
;THE FORMAT BIT=0= 18 BITS PER  
;WORD THE READ WAS  
;COMPLETED BUT ERROR REG  
;WAS NOT RIGHT  
;NOTE DCK WILL BE SET BECAUSE  
;ECC HAS NOT BEEN GENERATED

033734 104020

ERROR 20

```

5271 ;*NOW A WRITE DATA WILL BE ATTEMPTED WITH
5272 ;*WRONG FORMAT BIT
5273
5274
5275 033740 012706 001000 MOV #STACK,SP ;RESET STACK
5276
5277
5278 033752 012737 177777 047614 MOV #-1,@#NOSYNC ;SET FLAG SO THAT DATA SYNC
5279 ;AND DATA IS NOT READ
5280 033760 004037 042532 FRMAT1: JSR RO,@#CLAREA ;CLEAR SIMULATED DISK
5281 033764 051416 .WORD DISK ;FROM
5282 033766 052442 .WORD TOLGAP+16 ;TO
5283 033770 000000 .WORD 0 ;DATA
5284 ;*THESE ARE SETUP FOR DISKLESS USE ONLY
5285 033772 005037 047500 CLR @#CYL ;CYLINDER 0, FORMAT 18 BIT WORDS
5286 033776 105037 047503 CLR @#SECOTR+1 ;TRACK 0
5287 034002 105037 047502 CLR @#SECOTR ;SECTOR 0
5288 034006 005037 047504 CLR @#KEY1 ;KEY1 0
5289 034012 005037 047506 CLR @#KEY2 ;KEY2 0
5290 034016 012737 000004 047546 MOV #4,@#NOWORD ;NO OF DATA WORDS
5291 034024 012737 000001 047510 MOV #1,@#X ;WRITE DATA
5292 034032 004537 044022 JSR R5,@#CRC ;GO TO CALCULATE CRC
5293 034036 052636 WCYL
5294 034040 052646 GCRC
5295
5296 ;*THESE AER REGULAR SETUPS
5297
5298 034042 004037 042532 JSR RO,@#CLAREA ;FILL WRITE FROM BUFFER WITH 125252
5299 034046 002110 WRFROM ;FROM
5300 034050 002116 WRFROM+6 ;TO
5301 034052 125252 125252 ;DATA
5302 034054 004737 042614 JSR PC,@#CLDISK ;SETUP GENERAL REGISTERS
5303 034060 012777 177774 145544 MOV #-4,@#RHWC ;256 DATA WORDS
5304 034066 012777 002110 145540 MOV @#WRFROM,@#RHBA ;STARTING ADDRESS OF WRITE BUFFER
5305 034074 005077 145544 CLR @#RHDS ;TRACK=0 SECTOR=0
5306 034100 012777 010000 145542 MOV @#FMT22,@#RHOF ;16 BITS PER WORD FORMAT
5307 034106 005077 145540 CLR @#RHCA ;CYLINDER 0
5308 034124 013711 002062 MOV @#WRIDAT,@#R1 ;WRITE DATA=60
5309 034130 005037 002006 CLR @#ERFLG ;CLEAR ERROR FLAG
5310 034134 004737 047340 JSR PC,@#COMHD ;WRITE DATA
5311 ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
5312 ;*FROM THE "COMHD" ROUTINE IT MEANS SECTOR GAP, SYNC BYTE
5313 ;*HEADER, HEADER CRC, HEADER GAP AND SYNC BYTE HAVE GONE BY
5314 ;*AND SYNC'S WERE CORRECTLY DETECTED
5315 ;*DATA IS TO BE CHECKED
5316 034140 004737 042220 JSR PC,@#PUTREG ;SAVE REGISTERS
5317 034144 005737 002006 TST @#ERFLG ;HAS ANY ERRORS OCCURED?
5318 034150 001041 BNE 4$ ;BRANCH IF YES
5319 034152 012700 000000 MOV #0,R0 ;GOOD DATA
5320 034156 012701 051416 MOV @#DISK,R1 ;DATA WRITTEN INTO "DISK"
5321 034162 012702 000004 MOV #4,R2 ;COUNTER
5322 034166 012737 000005 047620 1$: MOV #5,@#ERWORD ;FOR ERROR WORD
5323 034174 020021 CMP RO,(R1)+ ;COMPARE GOOD DATA WITH DATA ON DISK
5324 034176 001424 BEQ 3$ ;BRANCH IF GOOD
5325 034200 010037 001124 MOV RO,@#$GDDAT ;GOOD DATA
5326 034204 014137 001126 MOV -(R1),@#$BDDAT ;BAD DATA

```

```

5327 034210 160237 047620
5328 034214 005737 002006
5329 034220 001002
5330 034222 104004
5331
5332
5333
5334
5335 034224 000401
5336 034226 104005
5337 034230 005721
5338 034232 017746 144702
5339 034236 042716 177177
5340 034242 022726 000200
5341 034246 001402
5342 034250 005302
5343 034252 001345
5344
5345
5346
5347
5348
5349
5350
5351
5352
5353
5354
5355
5356
5357
5358
5359
5360
5361
5362
5363
5364
5365
5366
5367
5368
5369
5370
5371
5372
5373
5374
5375
5376
5377
5378
5379
5380
5381
5382
5383
5384
5385
5386
5387
5388
5389
5390
5391
5392
5393
5394
5395
5396
5397
5398
5399
5400

```

```

2$:
5$:
3$:
4$:

```

```

SUB R2, @#ERWORD ;ERROR WORD NO
TST @#ERFLG$ ;ANY ERRORS ALREADY THERE?
BNE 2$ ;BRANCH IF YES
ERROR 4 ;ERROR ON WRITE DATA COMMAND
;ON A WRITE DATA WITH
;WRONG FORMAT NO DATA
;SHOULD BE WRITTEN
;WORD NO GIVES WORD IN ERROR
;BRANCH TO AVOID PRINTING NEXT ERROR

BR 5$
ERROR 5
TST (R1)+ ;UNDO -(R1) FOR BAD DATA
MOV @SWR, -(SP) ;GET SWITCH SETTING
BIC #177177, (SP) ;KEEP ONLY SWITCH 7 AND 8
CMP #SW07, (SP)+ ;IS 7 SET AND 8 RESET.
BEQ 4$ ;BRANCH IF YES
DEC R2 ;IF NOT COUNT 256 WORDS
BNE 1$ ;BRANCH IF 256 NOT DONE

; *NOW CHECK TO SEE THAT FORMAT ERROR BIT GOT SET

CMP #FER, @#ER1 ;FORMAT ERROR SHOULD BE SET
ERROR 20 ;A 16 BIT PER WORD WRITE DATA
;WAS ATTEMPTED WHEN THE DISK
;HAD THE FORMAT BIT =0=18
;BITS PER WORD THE WRITE
;WAS CORRECTLY ABORTED
;BUT ERROR REG. 1 WAS WRONG

```

```

5355
5356
5357 034270 012706 001000      MOV      #STACK, SP      ;RESET STACK
5358 034302 004737 042614      JSR      PC, @#CLDISK    ;CLEAR DISK
5359 034320 012700 001652      MOV      #RHCA, RD
5360 034324 012005              MOV      (RD)+, R5      ;R5 HAS ADDRESS OF REG. UNDER TEST
5361 034326 052777 000040 145302 1$:  BIS      #CLR, @RHCS2
5362 034334 013777 001774 145274 2$:  MOV      @#UNIT, @RHCS2 ;REINSTATE UNIT NO.
5363
5364 ;*SET UP FOR AN OPERATION (WRITE HEADER AND DATA)
5365
5366 034342 013777 002064 145270      MOV      @#WRIFOR, @RHCS1 ;WRITE HEADER AND DATA=62
5367 ;IN RHCS1
5368 034350 012777 177766 145254      MOV      #-10, @RHWC     ;10 WORDS
5369 034356 012777 002110 145250      MOV      #WRFROM, @RHBA  ;BUS ADDRESS = WRFROM
5370 034364 012777 000010 145252      MOV      #10, @RHDS1     ;DESIRED TRACK=0, SECTOR=10
5371 034372 052777 000010 145236      BIS      #BA1, @RHCS2    ;BUS ADDRESS INCREMENT INHIBIT
5372 034400 012777 010000 145242      MOV      #FMT22, @RHOF   ;FORMAT 16 BIT WORDS
5373 034406 005077 145240      CLR      @RHCA           ;CYLINDER =0
5374
5375 ;*SAVE REGISTERS
5376
5377 034412 004037 043306      JSR      RD, @#SAVER     ;SAVE
5378 034416 001640              RHCS1      ;FROM
5379 034420 003154              REINTO     ;TO
5380 034422 000016              14.       ;NUMBER OF REGISTERS SAVED
5381
5382 ;*NOW THE COMMAND IS GIVES TO
5383 ;*WRITE HEADER AND DATA FOR CYL=0, SECTOR=10
5384 ;*TRACK=0 IT COMES BACK AFTER ONE SECTOR
5385 ;*HAS PASSED
5386
5387 034424 012777 000001 145226      MOV      #DMD, @RHMR     ;SET DIAGNOSTIC MODE
5388 034432 005277 145202      INC      @RHCS1         ;GO TO RHCS1 WITH 62
5389 034436 012715 177672      MOV      #177672, @RS    ;TRY WRITING ALL BITS EXCEPT
5390 ;GO, RMR, IE
5391 034442 052737 000001 003174      BIS      #DMD, @#REINTO+20 ;SET DMD IN SAVED REGISTER RHMR
5392 034450 052737 000004 003156      BIS      #RMR, @#REINTO+2 ;SET RMR IN SAVED REG. RHER1
5393 034456 042737 000200 003176      BIC      #DRY, @#REINTO+22 ;CLEAR DRY IN RHDS1
5394 034464 052737 040000 003176      BIS      #ERR, @#REINTO+22 ;SET ERR IN RHDS1
5395 034472 052737 000001 003154      BIS      #GO, @#REINTO    ;SET GO IN SAVED REG. RHCS1
5396 034500 042737 000200 003154      BIC      #RDY, @#REINTO   ;CLEAR RDY BIT
5397
5398 ;*AFTER AN ATTEMPT TO WRITE INTO A REGISTER
5399 ;*SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE
5400
5401 034506 004037 043306      JSR      RD, @#SAVER     ;SAVE
5402 034512 001640              RHCS1      ;FROM
5403 034514 002110              WRFROM     ;TO
5404 034516 000016              14.       ;NUMBER
5405
5406 ;*AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
5407 ;*OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
5408 ;*SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
5409 034520 113737 003173 002127      MOVB     @#REINTO+17, @#WRFROM+17 ;SAVE UPPER RHAS
5410

```

5411  
5412  
5413  
5414  
5415  
5416  
5417  
5418  
5419  
5420  
5421  
5422  
5423  
5424  
5425  
5426  
5427  
5428  
5429  
5430  
5431  
5432

034526 004037 043510  
034532 003154  
034534 002110  
034536 000016  
034540 034546  
034542 034546  
034544 034566  
034546 013705 047620  
034552 060505  
034554 016537 001636 042264  
034562 104001  
034564 000207

4\$:

5\$:

; \*COMPARE REGISTERS BEFORE ATTEMPTED WRITE WITH AFTER

JSR RD, @#COMPAR ; COMPAR  
REINTO ; GO BUFFER  
WRFROM ; TEST BUFFER  
14. ; NUMBER  
4\$ ; RETURN FOR ERROR  
4\$ ; SAME  
5\$ ; RETURN FOR GOOD COMPARISON  
MOV @#ERWORD, R5 ; GETTING READY TO INDEX  
ADD R5, R5 ; DOUBLE ERROR WORD  
MOV RHCSI-2(R5), @#REGADR ; FAILING REG. ADDRESS  
ERROR 1 ; CONTENTS OF REGISTER  
RTS PC ; CHANGED WITH  
; AN ATTEMPT TO WRITE  
; DURING AN OPERATION

; \*THE FOLLOWING CLEAR MAY SET THE ATA BIT BECAUSE GO IS HIGH

\*  
JSR PC, @#CLDISK ; CLEAR DISK

K11

CZRJG80.RP04/5/6 DSKLS CTRLR1  
CZRJG8.P11 08-NOV-77 08:30

MACY11 30(1046)  
T77

08-NOV-77 08:54 PAGE 138  
RHER1 - BIT #2 - REG. MODIFICATION REFUSED

SEQ 0140

5433  
5434  
5435  
5436  
5437  
5438

L11

CZRJGBO RPO4/5/6 DSKLS CTRLR1  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 139  
T101 ERROR REG1 - BIT #7 - HEADER COMPARE ERROR

SEQ 0141

5  
5  
5  
5  
5  
5  
5  
5  
5  
5

M11

CZRJGBD.RP04/5/6 DSKLS CTRLRI  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 140  
T103 ERROR REG1 - BIT #7 - HEADER COMPARE ERROR

SEG 0142

TTTTT  
TTTTT  
TTTTT  
TTTTT  
TTTTT





CZRJGBD.RP04/5/6 DSKLS CTRLR1  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046)  
T107

08-NOV-77 08:54 PAGE 142  
ERROR REG.1 - BIT #7 - HEADER COMPARE ERROR

SEQ 0144

5454  
5555  
5656  
5757

C12

CZRJGB0.RP04/5/6 DSKLS CTRLRI  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 143  
T110 RHER1 - BIT #8 - CRC ERROR (READING)

SEQ 0145

5458  
5459  
5460  
5461

462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498

;\*SET UP FOR THE TWO LAST SECTOR TRANSFERRED TESTS FOLLOWING

035650 005737 002036  
035654 001401  
035656 000402  
035660 000137 036334  
035664

TST @#RPO6 ;MOVE RPO6 FLAG TO ITSELF TO TEST  
BEQ 2\$ ;IF = 0 TREAT DRIVE AS RPO4  
BR 3\$ ;TREAT AS RPO6 - DO NEXT "MAKECL" & TEST  
JMP @#DOG ;DO SECOND FOLLOWING "MAKECL" AND TEST  
2\$:  
3\$:

036252 013746 001736  
036256 042716 001000  
036262 022726 002700  
036266 001412  
036270 013737 001662 042264  
036276 012737 002700 001124  
036304 013737 001736 001126  
036312 104001

4\$: MOV @#DS1, -(SP) ;GET RHDS1  
BIC #PROG, (SP) ;CLEAR PROG  
CMP @#LST!DPR!DRY!VV, (SP)+ ;IS 'LST' HIGH ?  
BEQ 5\$ ;BRANCH IF GOOD  
MOV @#RHDS1, @#REGADR ;FAILING REG. ADDRESS  
MOV @#LST!DPR!DRY!VV, @#SGDDAT ;GOOD DATA  
MOV @#DS1, @#SBDDAT ;BAD DATA  
ERROR 1 ;'LST' DID NOT SET AFTER  
;LAST SECTOR ON LAST TRACK  
;ON LAST CYLINDER WAS  
;WRITTEN  
;VV BIT #6 MAY OR MAY NOT BE HIGH  
5\$: MOV @#RHCS1, @#6\$ ;SET UP "WAT" SUBROUTINE  
WAT  
6\$: 0 ;RHCS1 ADDRESS  
RDY ;WAIT FOR READY

036314 013737 001640 036324  
036322 104415  
036324 000000  
036326 000200  
036330 000137 037000

JMP @#CAT ;DON'T DO THE RPO4 'LST' TEST FOLLOWING

5499  
5500  
5501  
5502  
5503  
5504  
5505  
5506  
5507  
5508  
5509  
5510  
5511  
5512  
5513  
5514  
5515  
5516  
5517  
5518  
5519  
5520  
5521  
5522  
5523

036334

DOG:

036722 013746 001736  
036726 042716 001000  
036732 022726 002700  
036736 001412  
036740 013737 001662 042264  
036746 012737 002700 001124  
036754 013737 001736 001126  
036762 104001  
  
036764 013737 001640 036774  
036772 104415  
036774 000000  
036776 000200

4\$:

MOV @#DS1,-(SP) ;GET RHDS1  
BIC #PROG,(SP) ;CLEAR PROG BIT  
CMP #LST:DPR:DRY:VV,(SP)+ ;IS 'LST' HIGH ?  
BEQ 5\$ ;WAIT FOR 'RDY' IF GOOD  
MOV @#RHDS1,@#REGADR ;FAILING REG. ADDRESS  
MOV #LST:DPR:DRY:VV,@#SGDDAT ;GOOD DATA  
MOV @#DS1,@#SBDDAT ;BAD DATA  
ERROR 1 ;'LST' DID NOT SET AFTER  
;LAST SECTOR ON LAST TRACK ON LAST  
;CYLINDER WAS WRITTEN - 'VV' BIT #6  
;MAY OR MAY NOT BE HIGH

5\$:

MOV @#RHCS1,@#6\$ ;SET UP "WAT" SUBROUTINE

6\$:

WAT 0 ;RHCS1 ADDRESS  
RDY ;WAIT FOR 'RDY' BIT

```

5524
5525
5526 037000
5527 037002 012706 001000
5528 037014 004737 042614
5529 037020 004037 042532
5530 037024 051416
5531 037026 052442
5532 037030 000000
5533
5534
5535
5536
5537 037032 005737 002036
5538 037036 001404
5539
5540 037040 012737 011456 047500
5541 037046 000403
5542
5543 037050 012737 010632 047500 10$:
5544
5545
5546 037056 112737 000022 047503 11$:
5547 037064 112737 000025 047502
5548 037072 005037 047504
5549 037076 005037 047506
5550 037102 012737 000400 047546
5551 037110 012737 000001 047510
5552 037116 004537 044022
5553 037122 047500
5554 037124 051400
5555
5556
5557
5558 037126 004037 042532
5559 037132 002110
5560 037134 003110
5561 037136 000377
5562 037140 004737 042614
5563 037144 012777 177272 142460
5564 037152 012777 002110 142454
5565 037160 012746 000025
5566 037164 112766 000022 000001
5567 037172 012677 142446
5568 037176 012777 010000 142444
5569
5570
5571
5572
5573 037204 005737 002036
5574 037210 001404
5575 037212 012777 001456 142432
5576 037220 000403
5577 037222 012777 000632 142422 12$:
5578 037230 13$:
5579

```

CAT:

```

MOV #STACK, SP ; RESET STACK
JSR PC, @CLDISK ; INIT AND SET UP GENERAL REG. CORRES.
JSR RO, @CLAREA ; CLEAR SIMULATED DISK
.WORD DISK ; FROM
.WORD TOLGAP+16 ; TO
.WORD 0 ; DATA

```

```

; *THESE ARE TO SETUP FOR DISKLESS USE ONLY
; *AND WILL HANDLE RPO4 OR RPO6 DRIVES

```

```

TST @RPO6 ; MOVE RPO6 FLAG TO ITSELF TO TEST
BEQ 10$ ; TREAT DRIVE AS RPO4 IF = 0

MOV #814, !FMT22, @CYL ; CYLINDER 814, 16 BITS PER WORD
BR 11$ ; TREAT DRIVE AS RPO6

MOV #410, !FMT22, @CYL ; CYLINDER 410, 16 BITS PER WORD
; TREAT DRIVE AS RPO4

MOVB #18, @SECOTR+1 ; TRACK 18.
MOVB #21, @SECOTR ; SECTOR 21.
CLR @KEY1 ; KEY1 0
CLR @KEY2 ; KEY2 0
MOV #256, @NOWORD ; NO OF DATA WORDS
MOV #1, @X ; WRITE DATA
JSR R5, @CRC ; GO TO CALCULATE CRC
CYL
WCRC

```

```

; *THESE ARE REGULAR SETUPS

```

```

JSR RO, @CLAREA ; FILL WRITE BUFFER WITH 377
WRFROM ; FROM
WRFROM+<256.*2> ; TO
377 ; DATA
JSR PC, @CLDISK ; SETUP GENERAL REGISTERS
MOV #-326, @RHWC ; 326. DATA WORDS
MOV @WRFROM, @RHBA ; STARTING ADDRESS OF WRITE BUFFER
MOV #21, -(SP) ; SECTOR 21.
MOVB #18, 1(SP) ; TRACK 18.
MOV (SP)+, @RHDS1 ; SECTOR 21. TRACK 18.
MOV @FMT22, @RHOF ; 16 BITS PER WORD FORMAT

```

```

; *CHECK TO SEE WHAT TYPE OF DRIVE IS BEING TESTED
; *AND LOAD CYLINDER ADDRESS REGISTER WITH THE PROPER NUMBER

```

```

TST @RPO6 ; MOVE FLAG TO ITSELF TO TEST
BEQ 12$ ; TREAT AS RPO4 IF = 0
MOV #814, @RHCA ; CYLINDER 814.
BR 13$ ; TREAT AS RPO6
MOV #410, @RHCA ; CYLINDER 410.

```

```

5580 037242 013711 002062 MOV @#WRIDAT,@R1 ;WRITE DATA=60
5581 037246 005037 002006 CLR @#ERFLGS ;CLEAR ERROR FLAG
5582
5583 ;*THE REGISTERS WILL BE SAVED IN REINTO BUFFER
5584
5585 037252 004037 043306 JSR RO,@#SAVER ;SAVE
5586 037256 001632 RHW ;FROM
5587 037260 003154 REINTO ;TO
5588 037262 000023 19. ;NUMBER SAVED
5589
5590 ;*GIVE WRITE DATA COMMAND
5591
5592 037264 004737 047340 JSR PC,@#COMMD ;WRITE DATA COMMAND
5593
5594 ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
5595
5596 037270 005737 002040 TST @#RH70 ;CHECK FOR RH70 CONTROLLER
5597 037274 001407 BEQ 8$ ;SKIP RH70 CODE AND DO RH11 IF NOT
5598
5599 037276 012737 177702 003154 MOV #-76,@#REINTO ;SAVED RHW SHOULD BE = 76 (OCTAL)
5600 037304 012737 003130 003156 MOV #WRFROM+(2*256.)+<2*8.>,@#REINTO+2 ;SAVED RHBA SHOULD BE WRFROM+256+8
5601 ;SAVED RHBA SHOULD BE WRFROM+256+8
5602 037312 000406 BR 9$ ;SKIP NEXT RH11 CODE
5603
5604 037314 012737 177774 003154 8$: MOV #-4,@#REINTO ;SAVED RHW SHOULD BE = 4
5605 037322 012737 003314 003156 MOV #WRFROM+(2*256.)+<2*66.>,@#REINTO+2 ;SAVED RHBA SHOULD BE WRFROM+256+66
5606
5607
5608 037330 052737 000200 003160 9$: BIS #OR,@#REINTO+4 ;SAVED RHCS2
5609 037336 042737 000100 003160 BIC #IR,@#REINTO+4 ;SAVED RHCS2
5610 037344 052737 140000 003162 BIS #SC!TRE,@#REINTO+6 ;SAVED RHCS1 SHOULD HAVE 'SC' & 'TRE'
5611 037352 012737 001000 003164 MOV #AOE,@#REINTO+10 ;SAVED RHER1 SHOULD HAVE 'AOE'
5612 037360 017737 142260 003166 MOV @#RHDST,@#REINTO+12 ;SAVED RHDST SHOULD HAVE=
5613 ;RHDST IS UNDEFINED
5614
5615 ;*CHECK TO SEE WHAT TYPE OF DRIVE IS BEING TESTED
5616 ;*AND SET UP CYLINDER ADDRESS ACCORDINGLY
5617
5618 037366 005737 002036 TST @#RPO6 ;MOVE RPO6 FLAG TO ITSELF TO TEST
5619 037372 001404 BEQ 14$ ;TREAT AS RPO4 IF = 0
5620 037374 012737 001457 003174 MOV #815.,@#REINTO+20 ;SAVED DESIRED CYLINDER ADDRESS
5621 037402 000403 BR 15$ ;TREAT AS RPO6
5622 037404 012737 000633 003174 14$: MOV #411.,@#REINTO+20 ;SAVED DESIRED CYLINDER ADDRESS
5623
5624 037412 013737 002016 003200 15$: MOV @#ATTENT,@#REINTO+24 ;SAVED RHAS SHOULD HAVE APPRO. BIT
5625 037420 052737 000001 003202 BIS #DMD,@#REINTO+26 ;SAVED RHMR
5626 037426 052737 142000 003204 BIS #ATA!ERR!LST,@#REINTO+30 ;SAVED RHDS1
5627
5628 ;*AFTER A WRITE DATA COMMAND WITH 'AOE' ERROR
5629 ;*SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE
5630
5631 037434 004037 043306 JSR RO,@#SAVER ;SAVE
5632 037440 001632 RHW ;FROM
5633 037442 002110 WRFROM ;TO
5634 037444 000021 17. ;NUMBER OF REGISTERS SAVED
5635

```

```

5636 ;*AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
5637 ;*OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
5638 ;*SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
5639
5640 037446 113737 003201 002135 MOVB @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
5641
5642 ;*COMPARE REGISTERS BEFORE WRITE DATA COMMAND
5643 ;*WITH AFTER COMMAND
5644
5645 037454 004037 043510 JSR RD,@#COMPAR ;COMPARE
5646 037460 003154 REINTO ;GOOD BUFFER
5647 037462 002110 WRFROM ;TEST BUFFER
5648 037464 000021 17. ;NUMBER OF REGISTERS
5649 037466 037474 1$ ;RETURN FOR ERROR
5650 037470 037474 1$ ;SAME
5651 037472 037514 2$ ;RETURN FOR GOOD COMPARISON
5652
5653 037474 013705 047620 1$: MOV @#ERWORD,R5 ;GETTING READY TO INDEX
5654 037500 060505 ADD R5,R5 ;DOUBLE ERROR WORD
5655 037502 016537 001630 042264 MOV RHWC-2(R5),@#REGADR ;FAILING REG. ADDRESS
5656 037510 104001 ERROR 1 ;FORCED AOE ERROR CAUSED IMPROPER
5657 ;REGISTER CHANGE
5658 037512 000207 RTS PC ;RETURN FOR FURTHER COMPARISONS
5659 ;NO ERRORS
5660 037514 005037 002006 2$: CLR @#ERFLG$ ;CLEAR ERROR FLAG
5661
5662 ;*DATA IS TO BE CHECKED HERE
5663
5664
5665 037520 004737 042220 JSR PC,@#PUTREG ;SAVE REGISTERS
5666 037524 012700 000377 MOV #377,R0 ;GOOD DATA
5667 037530 012701 051416 MOV #DISK,R1 ;DATA WRITTEN INTO "DISK"
5668 037534 012702 000400 MOV #256,R2 ;COUNTER
5669 037540 012737 000400 047620 3$: MOV #256,@#ERWORD ;FOR ERROR WORD
5670 037546 020021 CMP R0,(R1)+ ;COMPARE GOOD DATA WITH DATA ON DISK
5671 037550 001424 BEQ 6$ ;BRANCH IF GOOD
5672 037552 010037 001124 MOV R0,@#SGDDAT ;GOOD DATA
5673 037556 014137 001126 MOV -(R1),@#SBDDAT ;BAD DATA
5674 037562 160237 047620 SUB R2,@#ERWORD ;ERROR WORD NO
5675 037566 005737 002006 TST @#ERFLG$ ;ANY ERRORS ALREADY THERE?
5676 037572 001002 BNE 4$ ;BRANCH IF YES
5677 037574 104004 ERROR 4 ;ERROR ON WRITE DATA COMMAND WITH FORCED 'AOE'
5678 037576 000401 BR 5$ ;BRANCH TO AVOID PRINTING NEXT ERROR
5679 037600 104005 4$: ERROR 5 ;WORD NO. GIVES WORD IN ERROR
5680 037602 005721 5$: TST (R1)+ ;UNDO -(R1) FOR BAD DATA
5681 037604 017746 141330 MOV @SWR, -(SP) ;GET SWITCH SETTING
5682 037610 042716 177177 BIC #177177,(SP) ;KEEP ONLY SWITCH 7 AND 8
5683 037614 022726 000200 CMP #SW07,(SP)+ ;IS 7 SET AND 8 RESET
5684 037620 001402 BEQ 7$ ;BRANCH OUT IF YES -----)
5685
5686 037622 005302 6$: DEC R2 ;IF NOT COUNT 256 WORDS
5687 037624 001345 BNE 3$ ;BRANCH IF 256. NOT DONE
5688
5689 037626 7$:

```



```

5690
5691
5692
5693 037664 012706 001000      MOV      #STACK,SP      ;RESET STACK
5694 037676 004737 042614      JSR      PC,@#CLDISK    ;CLEAR REGISTERS AND SET UNIT NO.
5695
5696      ;*GIVE INDEX PULSE
5697 037702 012777 000001 141750  MOV      #DMD,@RHMR     ;SET DIAGNOSTIC MODE
5698 037710 052777 000004 141742  BIS      #MINX,@RHMR    ;SET INDEX
5699 037716 042777 000004 141734  BIC      #MINX,@RHMR    ;CLEAR INDEX
5700
5701
5702      ;*THESE ARE REGULAR SETUPS
5703
5704 037724 012777 177400 141700  MOV      #-256,@RHWC    ;256 DATA WORDS 4 HEADER WORDS
5705 037732 012700 003154      MOV      #REINT0,RO     ;THESE TWO INSTRUCTIONS GETS
5706 037736 010077 141672      MOV      RO,@RHBA      ;ADDR. OF WRFROM INTO RO AND
5707      ;BUS ADDRESS REGISTER
5708 037742 012720 010000      MOV      #FMT22,(RO)+   ;FORMAT=16 BIT WORDS
5709      ;CYLINDER=0
5710 037746 012720 012000      MOV      #12000,(RO)+   ;TRACK=20 SECTOR=0 KEYS=0
5711 037752 005020      CLR      (RO)+          ;KEY1=0
5712 037754 005020      CLR      (RO)+          ;KEY2=0
5713 037756 012705 000400      MOV      #256,R5        ;COUNTER
5714 037762 012720 177777      MOV      #-1,(RO)+     ;MOVE ALL ONES FOR DATA
5715 037766 005305      DEC      R5
5716 037770 001374      BNE     1$             ;BRANCH IF DATA NOT COMPLETE
5717 037772 012777 012000 141644  MOV      #12000,@RHDST  ;TRACK=20 SECTOR=0
5718
5719
5720 040012 013711 002070      MOV      @#REFOR,@R1    ;GET READY FOR WRITE HEADER AND
5721      ;DATA WITH 62 IN RHCS1
5722 040016 005037 002006      CLR      @#ERFLGS      ;CLEAR ERROR FLAG
5723 040022 012777 010000 141620  MOV      #FMT22,@RHOF   ;FORMAT BIT=1 (16 BIT WORDS)
5724 040030 005077 141616      CLR      @RHCA         ;CYLINDER =0
5725
5726      ;*THE REGISTERS WILL BE SAVED IN REINT0 BUFFER
5727 040034 004037 043306      JSR      RO,@#SAVER     ;SAVE
5728 040040 001632      RHWC     ;FROM
5729 040042 003154      REINT0   ;TO
5730 040044 000023      19.     ;NUMBER SAVED
5731
5732      ;*GO TO WRITE HEADER AND DATA
5733
5734 040046 013700 001660      MOV      @#RHMR,RO     ;NOW RO WAS MAINTENANCE REG. ADDR.
5735 040052 012710 000001      MOV      #DMD,@RO      ;SET DIAGNOSTIC MODE
5736 040056 052777 000001 141554  BIS      #GO,@RHCS1    ;GO
5737
5738      ;*CHANGE SAVED REGISTERS TO EXPECTED VALUE
5739 040064 052737 140000 003162  BIS      #SC!TRE,@#REINT0+6 ;SAVED RHCS1
5740 040072 012737 002000 003164  MOV      #IAE,@#REINT0+10 ;SAVED RHER1
5741 040100 012737 012001 003166  MOV      #12001,@#REINT0+12 ;SAVED RHDST
5742 040106 013737 002016 003200  MOV      @#ATTENT,@#REINT0+24 ;SAVED RHAS
5743 040114 052737 000001 003202  BIS      #DMD,@#REINT0+26 ;SAVED RHMR
5744 040122 052737 140000 003204  BIS      #ATA!ERR,@#REINT0+30 ;SAVED RHDS1
5745

```

```

5746
5747 040130 004037 043306
5748 040134 001632
5749 040136 002110
5750 040140 000023
5751
5752
5753
5754
5755 040142 113737 003201 002135
5756
5757
5758
5759
5760 040150 004037 043510
5761 040154 003154
5762 040156 002110
5763 040160 000021
5764 040162 040170
5765 040164 040170
5766 040166 040210
5767
5768 040170 013705 047620 2$: MOV 2#ERWORD,R5 ;GETTING READY TO INDEX
5769 040174 060505 ADD R5,R5 ;DOUBLE ERROR WORD
5770 040176 016537 001630 042264 MOV RHW-2(R5),2#REGADR ;FAILING REG. ADDRESS
5771 040204 104001 ERROR 1 ;FORCED IAE CAUSED IMPROPER
5772
5773 040206 000207 RTS PC ;REGISTER CHANGE
5774
5775 ;NO ERRORS
5776
5777 040210 004737 042614 3$: JSR PC,2#CLDISK ;CLEAR GO BIT
5778

```

```

; *SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE
JSR RO,2#SAVER ;SAVE
RHW ;FROM
WRFROM ;TO
19. ;NUMBER OF REGISTERS SAVED

; *AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
; *OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
; *SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
MOVB 2#REINTO+25,2#WRFROM+25;SAVE UPPER RHAS

; *COMPARE REGISTERS BEFORE READ IN PRESET COMMAND
; *WITH AFTER COMMAND
JSR RO,2#COMPAR ;COMPARE
REINTO ;GOOD BUFFER
WRFROM ;TEST BUFFER
17. ;NUMBER OF REGISTERS
2$ ;RETURN FOR ERROR
2$ ;SAME
3$ ;RETURN FOR GOOD COMPARISON

;GETTING READY TO INDEX
;DOUBLE ERROR WORD
;FAILING REG. ADDRESS
;FORCED IAE CAUSED IMPROPER
;REGISTER CHANGE
;RETURN FOR FURTHER COMPARISONS
;NO ERRORS
;CLEAR GO BIT

```

```

5779
5780 040216 012706 001000      MOV      #STACK, SP      ;RESET STACK
5781 040230 004737 042614      JSR      PC, @#CLDISK   ;CLEAR REGISTERS AND SET UNIT NO.
5782
5783      ;*GIVE INDEX PULSE
5784 040234 012777 000001 141416  MOV      #DMD, @RHMR    ;SET DIAGNOSTIC MODE
5785 040242 052777 000004 141410  BIS      #MINX, @RHMR   ;SET INDEX
5786 040250 042777 000004 141402  BIC      #MINX, @RHMR   ;CLEAR INDEX
5787
5788      ;*THESE ARE REGULAR SETUPS
5789
5790 040256 012777 177400 141346  MOV      #-256., @RHWC  ;256 DATA WORDS 4 HEADER WORDS
5791 040264 012700 002110      MOV      #WRFROM, RO    ;THESE TWO INSTRUCTIONS GETS
5792 040270 010077 141340      MOV      RO, @RHBA      ;ADDR. OF WRFROM INTO RO AND
5793      ;BUS ADDRESS REGISTER
5794 040274 012720 010000      MOV      #FMT22, (RO)+  ;FORMAT=16 BIT WORDS
5795      ;CYLINDER=0
5796 040300 012720 000026      MOV      #22., (RO)+    ;TRACK=0, SECTOR=22, KEYS=0
5797 040304 005020      CLR      (RO)+          ;KEY1=0
5798 040306 005020      CLR      (RO)+          ;KEY2=0
5799 040310 012705 000400      MOV      #256., R5      ;COUNTER
5800 040314 012720 177777      MOV      #-1., (RO)+    ;MOVE ALL ONES FOR DATA
5801 040320 005305      DEC      R5
5802 040322 001374      BNE      1$            ;BRANCH IF DATA NOT COMPLETE
5803 040324 012777 000026 141312  MOV      #22., @RHDS1   ;TRACK=0 SECTOR=22
5804
5805
5806 040344 013711 002064      MOV      @#WRIFOR, @R1  ;GET READY FOR WRITE HEADER AND
5807      ;DATA WITH 62 IN RHCS1
5808 040350 005037 002006      CLR      @#ERFLGS      ;CLEAR ERROR FLAG
5809 040354 012777 010000 141266  MOV      #FMT22, @RHOF  ;FORMA BIT=1 (16 BIT WORDS)
5810 040362 005077 141264      CLR      @RHCA          ;CYLINDER =0
5811
5812      ;*AS EXCEPTION IS ASSERTED BEFORE RUN IS
5813      ;*LATCHED RHWC, RHBA, RHCS1, RHCS2 CANNOT BE CHECKED
5814      ;*BECAUSE RHWC WILL VARY DEPENDING UPON GATE DELAYS
5815      ;*ON DIFFERENT UNITS
5816
5817      ;*THE REGISTERS WILL BE SAVED IN REINTO BUFFER
5818 040366 004037 043306      JSR      RO, @#SAVER    ;SAVE
5819 040372 001642      RHER1      ;FROM
5820 040374 003154      REINTO     ;TO
5821 040376 000015      13.        ;NUMBER SAVED
5822
5823      ;*GO TO WRITE HEADER AND DATA
5824
5825 040400 013700 001660      MOV      @#RHMR, RO     ;NOW RO HAS MAINTENANCE REG. ADDR.
5826 040404 012710 000001      MOV      #DMD, @RO      ;SET DIAGNOSTIC MODE
5827 040410 052777 000001 141222  BIS      #GO, @RHCS1    ;GO
5828
5829      ;*CHANGE SAVED REGISTERS TO EXPECTED VALUE
5830 040416 012737 002000 003154  MOV      #IAE, @#REINTO ;SAVED RHER1
5831 040424 012737 000027 003156  MOV      #23., @#REINTO+2;SAVED RHDST
5832 040432 013737 002016 003170  MOV      @#ATTENT, @#REINTO+14 ;SAVED RHAS
5833 040440 052737 000001 003172  BIS      #DMD, @#REINTO+16 ;SAVED RHMR
5834 040446 052737 140000 003174  BIS      #ATA!ERR, @#REINTO+20 ;SAVED RHDS1

```

```

5835
5836
5837 040454 004037 043306
5838 040460 001642
5839 040462 002110
5840 040464 000015
5841
5842
5843
5844
5845 040466 113737 003171 002125
5846
5847
5848
5849
5850 040474 004037 043510
5851 040500 003154
5852 040502 002110
5853 040504 000015
5854 040506 040514
5855 040510 040514
5856 040512 040534
5857
5858 040514 013705 047620 2$: MOV @#ERWORD,RS ;GETTING READY TO INDEX
5859 040520 060505 ADD RS,RS ;DOUBLE ERROR WORD
5860 040522 016537 001640 042264 MOV RHER1-2(R5),@#REGADR ;FADING REG. ADDRESS
5861 040530 104001 ERROR 1 ;FORCED IAE CAUSED IMPROPER
5862
5863 040532 000207 RTS PC ;REGISTER CHANGE
5864
5865 ;RETURN FOR FURTHER COMPARISONS
5866 ;NO ERRORS
5867 040534 004737 042614 3$: JSR PC,@#CLDISK ;CLEAR GO BIT
5868
5869

```

```

; *SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE
JSR RD,@#SAVER ;SAVE
RHER1 ;FROM
WRFROM ;TO
13. ;NUMBER OF REGISTERS SAVED

```

```

; *AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
; *OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
; *SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
MOVB @#REINTO+15,@#WRFROM+15;SAVE UPPER RHAS

```

```

; *COMPARE REGISTERS BEFORE READ IN PRESET COMMAND
; *WITH AFTER COMMAND
JSR RD,@#COMPAR ;COMPARE
REINTO ;GOOD BUFFER
WRFROM ;TEST BUFFER
13. ;NUMBER OF REGISTERS
2$ ;RETURN FOR ERROR
2$ ;SAME
3$ ;RETURN FOR GOOD COMPARISON

```

```

;GETTING READY TO INDEX
;DOUBLE ERROR WORD
;FADING REG. ADDRESS
;FORCED IAE CAUSED IMPROPER
;REGISTER CHANGE
;RETURN FOR FURTHER COMPARISONS

```

```

;NO ERRORS
;CLEAR GO BIT

```

```

5870
5871 040542 012706 001000      MOV      #STACK,SP      ;RESET STACK
5872 040554 004737 042614      JSR      PC,@#CLDISK    ;CLEAR REGISTERS AND SET UNIT NO.
5873
5874      ;*GIVE INDEX PULSE
5875 040560 012777 000001 141072  MOV      #DMD,@RHMR     ;SET DIAGNOSTIC MODE
5876 040566 052777 000004 141064  BIS      #MINX,@RHMR     ;SET INDEX
5877 040574 042777 000004 141056  BIC      #MINX,@RHMR     ;CLEAR INDEX
5878
5879      ;*THESE ARE REGULAR SETUPS
5880
5881 040602 012777 177400 141022  MOV      #-256,@RHWC     ;256 DATA WORDS 4 HEADER WORDS
5882 040610 012700 002110      MOV      #WRFROM,RO     ;THESE TWO INSTRUCTIONS GETS
5883 040614 010077 141014      MOV      RO,@RHBA       ;ADDR. OF WRFROM INTO RO AND
5884      ;BUS ADDRESS REGISTER
5885 040620 012705 000400      MOV      #256,R5        ;COUNTER
5886 040624 012720 177777      1$:     MOV      #-1,(RO)+      ;MOVE ALL ONES FOR DATA
5887 040630 005305      DEC      R5
5888 040632 001374      BNE     1$              ;BRANCH IF DATA NOT COMPLETE
5889 040634 012777 000000 141002  MOV      #0,@RH DST     ;TRACK=0 SECTOR=0
5890
5891
5892 040654 013711 002062      MOV      @#WRIDAT,@R1   ;GET READY FOR WRITE
5893      ;DATA WITH 60 IN RHCS1
5894 040660 005037 002006      CLR      @#ERFLGS      ;CLEAR ERROR FLAG
5895 040664 012777 010000 140756  MOV      #FMT22,@RHOF   ;FORMA BIT=1 (16 BIT WORDS)
5896
5897 040672 005737 002036      TST     @#RP06 ;MOVE FLAG TO ITSELF TO TEST
5898 040676 001404      BEQ     4$              ;TREAT DRIVE AS RP04 IF FLAG = 0
5899
5900 040700 012777 001457 140744  MOV      #815,@RHCA     ;CYLINDER = 815 (ONE TOO MANY)
5901 040706 000403      BR      5$              ;TREAT DRIVE AS RP06
5902
5903 040710 012777 000633 140734  4$:     MOV      #411,@RHCA     ;CYLINDER = 411 (ONE TOO MANY)
5904      ;TREAT DRIVE AS RP04
5905
5906      ;*AS EXCEPTION IS ASSERTED BEFORE RUN IS
5907      ;*LATCHED RHWC,RHBA,RHCS1,RHCS2 CANNOT BE CHECKED
5908      ;*BECAUSE RHWC WILL VARY DEPENDING UPON GATE DELAYS
5909      ;*ON DIFFERENT UNITS
5910
5911      ;*THE REGISTERS WILL BE SAVED IN REINTO BUFFER
5912 040716 004037 043306      5$:     JSR      RO,@#SAVER     ;SAVE
5913 040722 001642      RHER1    ;FROM
5914 040724 003154      REINTO   ;TO
5915 040726 000015      13.     ;NUMBER SAVED
5916
5917      ;*GO TO WRITE HEADER AND DATA
5918
5919 040730 013700 001660      MOV      @#RHMR,RO     ;NOW RO HAS MAINTENANCE REG. ADDR.
5920 040734 012710 000001      MOV      #DMD,@RO      ;SET DIAGNOSTIC MODE
5921 040740 052777 000001 140672  BIS      #GO,@RHCS1     ;GO
5922
5923      ;*CHANGE SAVED REGISTERS TO EXPECTED VALUE
5924 040746 012737 002000 003154  MOV      #IAE,@#REINTO  ;SAVED RHER1
5925 040754 012737 000001 003156  MOV      #1,@#REINTO+2;SAVED RH DST

```

```

5926 040762 013737 002016 003170 MOV @#ATTENT,@#REINTO+14 ;SAVED RHAS
5927 040770 052737 000001 003172 BIS #DMD,@#REINTO+16 ;SAVED RHMR
5928 040776 052737 140000 003174 BIS #ATA!ERR,@#REINTO+20 ;SAVED RHDS1
5929
5930 ;*SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE
5931 041004 004037 043306 JSR RO,@#SAVER ;SAVE
5932 041010 001642 RHER1 ;FROM
5933 041012 002110 WRFROM ;TO
5934 041014 000015 13. ;NUMBER OF REGISTERS SAVED
5935
5936 ;*AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
5937 ;*OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
5938 ;*SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
5939 041016 113737 003171 002125 MOVB @#REINTO+15,@#WRFROM+15;SAVE UPPER RHAS
5940
5941 ;*COMPARE REGISTERS BEFORE READ IN PRESET COMMAND
5942 ;*WITH AFTER COMMAND
5943
5944 041024 004037 043510 JSR RO,@#COMPAR ;COMPARE
5945 041030 003154 REINTO ;GOOD BUFFER
5946 041032 002110 WRFROM ;TEST BUFFER
5947 041034 000015 13. ;NUMBER OF REGISTERS
5948 041036 041044 2$ ;RETURN FOR ERROR
5949 041040 041044 2$ ;SAME
5950 041042 041064 3$ ;RETURN FOR GOOD COMPARISON
5951
5952 041044 013705 047620 2$: MOV @#ERWORD,R5 ;GETTING READY TO INDEX
5953 041050 060505 ADD R5,R5 ;DOUBLE ERROR WORD
5954 041052 016537 001640 042264 MOV RHER1-2(R5),@#REGADR ;FAILING REG. ADDRESS
5955 041060 104001 ERROR 1 ;FORCED IAE CAUSED IMPROPER
5956 ;REGISTER CHANGE
5957 041062 000207 RTS PC ;RETURN FOR FURTHER COMPARISONS
5958
5959 ;NO ERRORS
5960
5961 041064 004737 042614 3$: JSR PC,@#CLDISK ;CLEAR GO BIT

```

```

5962 041100 012737 000000 177776 MOV #0,PS ;REINSTATE PS TO 0
5963 041166 013746 001774 MOV @#UNIT,-(SP) ;GET READY TO TYPE UNIT NUMBER
5964 041172 104405 TYPDS ;
5965 041206 013746 001112 MOV @#SERTTL,-(SP) ;GET READY TO TYPE NUMBER OF ERRORS
5966 041212 104405 TYPDS ;
5967 041214 005037 001112 CLR @#SERTTL ;CLEAR TOTAL NUMBER OF ERRORS
5968 041220 005037 001102 CLR @#STSTNM ;CLEAR TEST NUMBER
5969 041224 005737 002002 TST @#SELECT ;STARTING FROM 210 ?
5970
5971 041230 001413 BEQ 3$ ;TEST NEXT DRIVE IF NOT
5972 ;CONTINUE ON THIS ONE IF SO
5973 041232 005237 001100 INC @#SPASS ;INCREASE PASS COUNT
5974 041236 104401 041421 TYPE $ENDMG ;TYPE END PASS #
5975 041242 013746 001100 MOV @#SPASS,-(SP) ;
5976 041246 104405 TYPDS ;
5977 041250 104401 041416 TYPE $ENULL ;
5978 041254 000137 007540 JMP @#TST5 ;DO NEXT TESTS ----->
5979
5980 041260 005337 001776 3$: DEC @#NUNITS ;NO. OF UNITS PRESENT DECREMENTED
5981 041264 001413 BEQ $EOP ;BRANCH IF ALL DRIVES COMPLETE
5982 041266 013700 001774 MOV @#UNIT,R0 ;UNIT UNDER TEST
5983 041272 012701 001754 MOV @#UNITS,R1 ;TABLE
5984 041276 022100 1$: CMP (R1)+,R0 ;IS THIS UNIT JUST TESTED
5985 041300 001401 BEQ 2$ ;BRANCH IF YES
5986 041302 000775 BR 1$ ;BRANCH IF NO
5987 041304 011137 001774 2$: MOV (R1),@#UNIT ;THIS IS NEXT UNIT
5988 041310 000137 007540 JMP @#TST5 ;GO FOR NEXT TESTS ----->
5989
5990
5991
5992
5993
5994
5995
5996
5997
5998
5999
6000
6001
6002
6003
6004
6005
6006
6007
6008
6009
6010
6011
6012
6013
6014
6015
6016
6017

```

```

.SBTTL
.SBTTL ***SUBROUTINES***
.SBTTL

```

```

; **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.

```

```

6018 041436 000000
6019
6020 041440
6021 041440 005037 177776
6022 041514 013746 002032
6023 041520 104402
6024 041560 013746 001110
6025 041564 104402
6026 041566 104401 001223
6027 042020 104412
6028 042022 062716 000002
6029 042026 012637 001106
6030 042204 104412
6031 042206 012637 001110
6032 042212 013746 001106
6033 042216 000002
6034
6035
6036
6037
6038
6039
6040
6041
6042
6043
6044
6045
6046
6047
6048 042220
6049 042226 012700 001632
6050 042232 012701 001706
6051 042236 012702 000023
6052 042242 013021
6053 042244 005302
6054 042246 001375
6055 042256 000207
6056
6057
6058
6059
6060
6061
6062
6063
6064
6065
6066
6067 042260 000000
6068 042262 000000
6069 042264 000000
6070
6071 042266 012537 042260
6072 042272 012504
6073 042274 010437 042264

```

```

TESTAD: 0 ;FIRST ADDRESS OF TEST
OPERSEL:
CLR PS ;MAKE PROCESSOR STATUS ZERO
MOV @#TSTNM, -(SP) ;GET READY TO TYPE TEST
TYPOC ;NUMBER
MOV @#SLPERR, -(SP) ;GET READY TO TYPE LOOP BACK PC
TYPOC
TYPE , $CRLF
RDOCT
ADD #2 (SP) ;GET LPADR
MOV (SP)+, @#SLPADR
RDOCT
MOV (SP)+, @#SLPERR ;GET LPERR
MOV @#SLPADR, -(SP)
RTI

```

.SBTTL SAVE REGISTERS ROUTINE

```

; *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 #RHW, R0 ;STARTING ADDRESS OF REG
MOV #WC, R1 ;STARTING ADDRESS OF WERE SAVED
MOV #RHC-RHW+2/2, R2 ;NUMBER OF REG. INTO R2
10$: MOV @ (R0)+, (R1)+ ;SAVE HARDWARE REG.
DEC R2
BNE 10$
RTS PC

```

.SBTTL FLOAT 1 AND 0

```

; *FLOAT A ONE AND A ZERO THRU A DESIGNATED REGISTER
; *ABSOLUTE ADDRESS OF REG. UNDER TEST IS IN R4
MASK: 0 ;BITS UNDER TEST
LERR: 0 ;ERROR HLT ADDRESS
REGADR: 0
BITST: MOV (R5)+, MASK ;FETCH DATA MASK
MOV (R5)+, R4 ;GET ADDRESS OF REG. UNDER TEST
MOV R4, REGADR

```



```

6074 042300 010537 042262      MOV      R5,      LERR      ;GET ERROR RETURN ADDR.
6075 042304 062705 000004      ADD      #4,      R5        ;MODIFY RETURN ADDR. TO JUMP OVER RTS
6076 042310 012703 000001      MOV      #1,      R3        ;INITIALIZE DATA PATTERN
6077 042314 004737 042336      JSR      PC,      BLT2      ;OUTPUT FLOATING ZERO
6078 042320 004737 042336      JSR      PC,      BLT2      ;OUTPUT FLOATING ONE
6079 042324 000241      CLC
6080 042326 006103      ROL      R3              ;SHIFT PATTERN
6081 042330 005703      TST      R3
6082 042332 001370      BNE      BLT1            ;BRANCH IF NOT COMPLETE
6083 042334 000205      RTS
6084 042336 005103      COM      R3              ;COMPLEMENT PATTERN
6085 042340 012737 042346 042574      MOV      #BLT3, @#LAD      ;SET SCOPE LOOP
6086 042346 032777 001000 136564      BIT      #SW09, @SWR      ;LOOP ON ERROR
6087 042354 001411      BEQ      4$              ;BRANCH IF NO
6088 042356 105737 001103      TSTB    @#SERFLG        ;ANY ERRORS
6089 042362 001406      BEQ      4$              ;BRANCH IF NO
6090 042364 000005      RESET
6091 042366 013777 001774 137242      MOV      @#UNIT, @RHCS2   ;SET UNIT NUMBER UNDER TEST
6092 042374 004737 055172      JSR      PC, @#STKINT     ;INITILIZE TK
6093
6094 042400 010337 001124      4$:      MOV      R3, @#SGDDAT     ;INIT FOR SCOPING LOOPS
6095 042404 005137 042260      COM      @#MASK          ;STORE GOOD DATA
6096 042410 043737 042260 001124      BIC      @#MASK, @#SGDDAT ;AND MASK WITH PATTERN
6097 042416 005137 042260      COM      @#MASK          ;CLEAR THE REST
6098 042422 013714 001124      MOV      @#SGDDAT, (R4)  ;RESTORE MASK
6099 042426 011437 001126      MOV      (R4), @#SBDDAT  ;OUTPUT TO REGISTER
6100 042432 005137 042260      COM      @#MASK          ;INPUT FROM REGISTER
6101 042436 043737 042260 001126      BIC      @#MASK, @#SBDDAT ;AND MASK OUT RECEIVED DATA
6102 042444 005137 042260      COM      @#MASK          ;RESTORE MASK
6103 042450 023737 001124 001126      CMP      @#SGDDAT, @#SBDDAT ;IS DATA CORRECT
6104 042456 001424      BEQ      1$              ;BRANCH IF GOOD
6105 042460 011437 001126      MOV      (R4), @#SBDDAT
6106 042464 023704 001640      CMP      @#RHCS1, R4     ;REGISTER UNDER TEST RHCS1?
6107 042470 001004      BNE      2$              ;BRANCH IF NOT
6108 042472 052737 004200 001124      BIS      #RDY!DVA, @#SGDDAT ;SET RDY AND DVA
6109 042500 000410      BR      3$
6110 042502 023704 001636      2$:      CMP      @#RHCS2, R4     ;REGISTER UNDER TEST RHCS2?
6111 042506 001005      BNE      3$              ;BRANCH IF NOT
6112 042510 011446      MOV      @R4, -(SP)      ;GET RHCS2
6113 042512 042716 177477      BIC      #C<IR!OR>, (SP)  ;KEEP IR AND OR BIT
6114 042516 052637 001124      BIS      (SP)+, @#SGDDAT ;SET IR OR BITS IF NEEDED
6115 042522 004777 177534      3$:      JSR      PC,      @LERR   ;GO TO REPORT ERROR
6116 042526 000240      NOP
6117 042530 000207      1$:      RTS      PC              ;REPLACE BY 104420 FOR LOCAL SCOPE LOOP

```

.SBTTL CLEAR MEMORY ROUTINE

THIS CLEARS ANY BLOCK OF MEMORY  
FILLING IT WITH ANY DATA

```

CALL
JSR      RO, CLAREA
X
Y
Z
;STARTING ADDRESS OF BLOCK
;DATA TO BE FILLED

```

```

6118
6119
6120
6121
6122
6123
6124
6125
6126
6127
6128
6129

```

```

6130
6131
6132
6133
6134
6135
6136 042532
6137 042540 012001
6138 042542 012002
6139 042544 012003
6140 042546 160102
6141 042550 062702 000002
6142 042554 010321
6143 042556 005302
6144 042560 005302
6145 042562 001374
6146 042572 000200

```

```

;*R1 WILL HAVE STARTING ADDRESS OF BLOCK TO BE FILLED
;*R2 AFTER SUBTRACTION WILL HAVE TWICE NUMBER OF LOCATIONS
;*R3 WILL HAVE DATA TO BE FILLED
;*TO AVOID DIVIDE ROUTINE TWO DECREMENT R2 WILL BE USED

```

```

CLAREA:
MOV (R0)+,R1 ;FROM
MOV (R0)+,R2 ;TO
MOV (R0)+,R3 ;DATA
SUB R1,R2 ;NO. OF LOCATIONS MINUS TWO
ADD #2,R2 ;GET TWICE NO OF LOCATIONS
1$: MOV R3,(R1)+ ;MOVE IN DATA
DEC R2
DEC R2
BNE 1$ ;BRANCH IF NOT COMPLETE
RTS R0 ;RETURN

```

```

6147          .SBTTL  LOCAL TRAPS
6148 042574 000000          LAD:      0
6149
6150 042576 032777 001000 136334 T.SCOPI: BIT      #SW09, @SWR
6151 042604 001402          BEQ      1$
6152 042606 013716 042574          MOV      @#LAD, (SP)
6153 042612 000002          1$:      RTI
6154
6155          ;*EXAMPLE OF THE USE OF THE ABOVE
6156          ;*THIS WILL LOOP BETWEEN X: AND SCOP1 PROVIDED THERE IS NO "NEWTST"
6157          ;*MOV      #X,      @#LAD
6158          ;*X:      ----      ---
6159          ;*      ----      ---
6160          ;*      ----      ---
6161          ;*      SCOP1
6162
6163          .SBTTL  CLEAR DISK ROUTINE
6164
6165          CLDISK: MOV      @#RHCS1,      R1      ;R1 WILL BE CONTROL AND STATUS1
6166 042614 013701 001640          MOV      @#RHCS2,      R2      ;R2 WILL BE CONTROL AND STATUS2
6167 042620 013702 001636          MOV      @#RHDS1,      R3      ;R3 WILL BE DISK STATUS REGISTER1
6168 042624 013703 001662          MOV      @#RHER1,      R4      ;R4 WILL BE ERROR REGISTER #1
6169 042630 013704 001642
6170
6171          MOV      #CLR,@R2          ;CLEAR ALL REG.
6172 042634 012712 000040          MOV      @#UNIT,@R2      ;REINSTATE UNIT NO.
6173 042640 012712 001774          CLR      @R1          ;CLEAR FUNCTION BITS
6174 042644 0050.1          RTS      PC
6175 042646 000207
    
```

.SBTTL CHECK DISK STATUS ROUTINES

```

6175
6176
6177
6178
6179
6180
6181
6182
6183
6184 042650 011637 002014 CHECKT: MOV (SP), @#PCJSR ;SAVE PC OF JSR+4
6185 042654 162737 000004 002014 SUB #4, @#PCJSR ;GET PC OF JSR
6186 042662 004737 042220 JSR PC, @#PUTREG ;SAVE REGISTERS
6187 042666 022737 004200 001714 CMP #DVA!RDY, @#CS1 ;RHCS1 SHOULD HAVE DEVICE AVAILABLE
6188 ;AND BE READY
6189 042674 001423 BEQ 3$ ;BRANCH IF GOOD
6190
6191 042676 032737 004000 001714 BIT #DVA, @#CS1 ;BAD SO TEST DEVICE AVAILABLE
6192 042704 001004 BNE 1$ ;BRANCH IF DVA THERE
6193 042706 010137 001122 MOV R1, @#$BDADR ;ADDRESS OF BAD REGISTER (RHCS1)
6194 042712 104026 ERROR 26 ;RHCS1 DID NOT HAVE DEVICE
6195 ;AVAILABLE AT START OF TEST
6196 042714 000413 BR 3$ ;BRANCH TO NEXT COMPARE
6197 042716 032737 000200 001714 1$: BIT #RDY, @#CS1 ;TEST READY
6198 042724 001003 BNE 2$ ;IF RDY THERE BRANCH
6199 042726 010137 001122 MOV R1, @#$BDADR ;ADDRESS OF BAD REGISTER (RHCS1)
6200 042732 104026 ERROR 26 ;RHCS1 DID NOT HAVE READY
6201 ;RIGHT AT START OF TEST
6202 042734 000403 BR 3$ ;BRANCH TO NEXT COMPARE
6203 042736 010137 001122 MOV R1, @#$BDADR ;ADDRESS OF BAD REGISTER (RHCS1)
6204 042742 104026 ERROR 26 ;RHCS1 HAD SOME BITS OTHER
6205 ;THAN DVA AND RDY SET
6206 ;ALL OTHER BITS SHOULD BE 0
6207
6208 042744 013746 001736 3$: MOV @#DS1, -(SP) ;GET RHDS1
6209 042750 042716 001100 BIC #VV!PROG, (SP) ;CLEAR VV AND PROGRAMABLE BIT
6210 042754 022726 000600 CMP #DPR!DRY, (SP)+; RHDS1 SHOULD HAVE THESE SET
6211 042760 001424 BEQ 8$ ;BRANCH IF THEY ARE
6212
6213 042762 032737 000400 001736 4$: BIT #DPR, @#DS1 ;TEST DRIVE PRESENT
6214 042770 001004 BNE 5$ ;CONTINUE IF THERE
6215 042772 010337 001122 MOV R3, @#$BDADR ;ADDRESS OF BAD REGISTER (RHDS1)
6216 042776 104026 ERROR 26 ;RHDS1 DOES NOT HAVE DPR
6217 043000 000413 BR 7$ ;BRANCH OUT
6218 043002 032737 000200 001736 5$: BIT #DRY, @#DS1 ;TEST DRIVE READY
6219 043010 001004 BNE 6$ ;IF DPR WAS THERE, BRANCH IF GOOD
6220 043012 010337 001122 MOV R3, @#$BDADR ;ADDRESS OF BAD REGISTER (RHDS1)
6221 043016 104026 ERROR 26 ;RHDS1 DOES NOT HAVE DRY
6222 043020 000403 BR 7$ ;BRANCH OUT
6223 043022 010337 001122 6$: MOV R3, @#$BDADR ;ADDRESS OF BAD REGISTER (RHDS1)
6224 043026 104026 ERROR 26 ;RHDS1 HAS SOME BITS OTHER
6225 ;THAN MOL, DRY, DPR, SET
6226 ;ALL OTHER BITS SHOULD BE 0
6227 043030 000207 7$: RTS PC ;RETURN TO TEST AND HALT
6228
6229 043032 062716 000006 8$: ADD #6, (SP) ;ADJUST STACK TO GET OVER HALT IN TEST
6230 043036 000207 RTS PC ;RETURN TO TEST AND CONTINUE TESTING

```

H13

CZRJGBD.RP04/5/6 DSKLS\_CTRLR1  
CZRJGB.F11 09-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 161  
CHECK DISK STATUS ROUTINES

SEQ 0163

6231

```

6232
6233
6234
6235
6236 043040 011637 002014
6237 043044 162737 000004 002014
6238 043052 004737 042220
6239 043056 032737 000200 001714
6240
6241 043064 001004
6242 043066 010137 001122
6243 043072 104026
6244
6245 043074 000427
6246 043076 032737 004000 001714 1$:
6247
6248 043104 001004
6249 043106 010137 001122
6250 043112 104026
6251
6252 043114 000417
6253 043116 032737 000200 001736 2$:
6254 043124 001004
6255 043126 010337 001122
6256 043132 104026
6257 043134 000407
6258 043136 032737 000400 001736 3$:
6259 043144 001004
6260 043146 010337 001122
6261 043152 104026
6262
6263 043154 000207 4$:
6264
6265 043156 062716 000006 5$:
6266 043162 000207

```

```

; *THIS CHECKS THAT DEVICE AVAILABLE (DVA) AND READY (RDY) IN RHCS1 = 1
; *AND CHECKS THAT DEVICE PRESENT (DPR), DEVICE READY (DRY) IN RHDS1 = 1
CHECKE: MOV (SP), @#PCJSR ;SAVE PC OF JSR+4
SUB #4, @#PCJSR ;GET PC OF JSR
JSR PC, @#PUTREG ;READ & SAVE REGISTERS
BIT #RDY, @#CS1 ;RHCS1 SHOULD HAVE DEVICE AVAILABLE
;AND BE READY
BNE 1$ ;BRANCH IF GOOD
MOV R1, @#$BDADR ;FAILING REGISTER
ERROR 26 ;RHCS1 IS IN ERROR
;DOES NOT HAVE DVA, RDY
BR 4$ ;BRANCH OUT
BIT #DVA, @#CS1 ;RHCS1 SHOULD HAVE DEVICE AVAILABLE
;AND BE READY
BNE 2$ ;BRANCH IF GOOD
MOV R1, @#$BDADR ;FAILING REGISTER
ERROR 26 ;RHCS1 IS IN ERROR
;DOES NOT HAVE DVA, RDY
BR 4$ ;BRANCH OUT
BIT #DRY, @#DS1 ;RHDS1 SHOULD HAVE DPR, DRY
;AND BE READY
BNE 3$ ;BRANCH IF THERE
MOV R3, @#$BDADR ;FAILING REGISTER RHDS1
ERROR 26 ;RHDS1 DOES NOT HAVE DPR, DRY
BR 4$ ;BRANCH OUT
BIT #DPR, @#DS1 ;RHDS1 SHOULD HAVE DPR, DRY
;AND BE READY
BNE 5$ ;BRANCH OUT AND CONTINUE IF THERE
MOV R3, @#$BDADR ;FAILING REGISTER RHDS1
ERROR 26 ;RHDS1 DOES NOT HAVE DPR, DRY
RTS PC ;RETURN TO TEST AND HALT
ADD #6, (SP) ;ADJUST STACK TO GET OVER HALT IN TEST
RTS PC ;RETURN TO TEST AND CONTINUE TESTING

```

6267  
6268  
6269  
6270  
6271  
6272  
6273  
6274  
6275  
6276  
6277  
6278  
6279  
6280  
6281  
6282  
6283  
6284  
6285  
6286  
6287  
6288  
6289  
6290  
6291  
6292  
6293  
6294  
6295  
6296  
6297  
6298  
6299  
6300  
6301  
6302  
6303  
6304  
6305  
6306  
6307  
6308  
6309

```

043164 177777 TIMCNT: 177777 ;WAITING COUNT
043166 010046 WAIT.T: MOV RO, -(SP) ;SAVE RO
043170 016600 MOV 2(SP), RO ;GET ADDRESS OF REG. ADDRESS
043174 010037 001204 MOV RO, @#STMP3 ;WAT PC+2 IN STMP3
043200 162737 000002 001204 SUB #2, @#STMP3 ;WAT PC FOR TYPEOUT
043206 012037 001176 MOV (RO)+, @#STMP0 ;WAIT REGISTER ADDRESS
043212 012037 001200 MOV (RO)+, @#STMP1 ;WAIT ON BIT
043216 010066 000002 MOV RO, 2(SP) ;RESTORE RETURN ON STACK
043222 012600 MOV (SP)+, RO ;RESTORE RO
043224 013737 043164 001202 MOV @#TIMCNT, @#STMP2 ;TEMPORARY COUNT
043232 033777 001200 135736 1$: BIT @#STMP1, @#STMP0 ;IS REQUIRED BIT THERE?
043240 001021 BNE 2$ ;BRANCH IF YES
043242 005337 001202 DEC @#STMP2 ;COUNT
043246 001371 BNE 1$ ;BRANCH IF NOT TIME UP
043250 013737 043164 001202 MOV @#TIMCNT, @#STMP2 ;TEMPORARY COUNT
043256 033777 001200 135712 3$: BIT @#STMP1, @#STMP0 ;IS REQUIRED BIT THERE?
043264 001007 BNE 2$ ;BRANCH IF YES
043266 005337 001202 DEC @#STMP2 ;COUNT
043272 001371 BNE 3$ ;BRANCH IF NOT TIME UP
043274 017737 135676 C01126 MOV @#STMP0, @#SBDDAT ;REGISTER CONTENTS
043302 104016 ERROR 16 ;WAITED ON BIT FAILED TO SET
043304 000002 2$: RTI

```

```

; * CALL FOR THE ABOVE WAITLOOP IS
; *
; * MOV @A, @#XS ;A CONTAINS REGISTER ADDRESS
; * - - - ;HENCE XS WILL HAVE ABSOLUTE REG. ADR.
; * - - -
; * WAT
; *XS: 0 ;ABSOLUTE REG. ADDRESS UNDER WAIT
; * .WORD 0 ;BIT WAITED FOR
; * ;CONTINUE

```

6310  
 6311  
 6312  
 6313  
 6314  
 6315  
 6316  
 6317  
 6318  
 6319  
 6320  
 6321 043306  
 6322 043314 012001  
 6323 043316 012002  
 6324 043320 012003  
 6325 043322 013122  
 6326 043324 005303  
 6327 043326 001375  
 6328 043336 000200  
 6329  
 6330  
 6331  
 6332  
 6333  
 6334  
 6335  
 6336  
 6337  
 6338  
 6339 043340 012737 010000 047500  
 6340 043346 112737 000001 047503  
 6341 043354 112737 000001 047502  
 6342 043362 005037 047504  
 6343 043366 005037 047506  
 6344 043372 012737 000044 047560  
 6345 043400 005037 047510  
 6346 043404 004537 044022  
 6347 043410 047500  
 6348 043412 051400  
 6349  
 6350  
 6351  
 6352 043414 004737 042614  
 6353  
 6354 043420 012777 177730 136204  
 6355 043426 012777 003154 136200  
 6356 043434 112746 000001  
 6357 043440 112766 000001 000001  
 6358 043446 012677 136172  
 6359 043452 012777 014000 136170  
 6360  
 6361  
 6362 043460 005077 136166  
 6363 043476 013711 002060  
 6364  
 6365 043502 004737 047340

```

.SBTTL SAVE ROUTINE
; *THIS IS A SUBROUTINE TO SAVE REGISTERS
; *IN THE REGISTER TABLE TO ANY LOCATION
; *THE CALL IS
; *JSR RO, @SAVER
; *FROM
; *TO
; *NUMBER OF WORDS SAVED

SAVER:
MOV (RO)+, R1 ; FROM
MOV (RO)+, R2 ; TO
MOV (RO)+, R3 ; NUMBER
1$: MOV @ (R1)+, (R2)+ ; SAVE REGISTER CONTENTS
DEC R3 ; COUNT
BNE 1$ ; BRANCH IF NOT DONE
RTS RO

.SBTTL WRITE CHECK ROUTINE
; *THIS IS A SUBROUTINE TO DO WRITE CHECK HEADER AND DATA
; *CYLINDER 0, TRACK 1, SECTOR 1, KEYS 0
; *THESE ARE TO SET UP FOR DISKLESS USE ONLY
WRCHHD: MOV #FMT22, @#CYL ; CYLINDER 0 FORMAT 16 BIT WORDS
MOV #1, @#SECOTR+1 ; TRACK=1
MOV #1, @#SECOTR ; SECTOR=1
CLR @#KEY1 ; KEY1=0
CLR @#KEY2 ; KEY2=0
MOV #36., @#DAWORD ; NO OF DATA WORDS
CLR @#X ; THIS IS A READ OPERATION
JSR R5, @#CRC ; GO TO CALCULATE CRC
CYL
WCRC

; *THESE ARE REGULAR SETUPS
JSR PC, @#CLDISK ; SET UP GENERAL REGISTERS
; AND CLEAR DISK REGISTERS
MOV #-40., @#RHWC ; 36 DATA WORDS 4 HEADER WORDS
MOV #REINT0, @#RHBA ; STARTING ADDRESS OF READ BUFFER
MOV #1, -(SP) ; SECTOR=1
MOV #1, 1(SP) ; TRACK=1 IN UPPER BYTE
MOV (SP)+, @#RHDST ; TRACK=1, SECTOR=1 IN RHDST
MOV #FMT22!ECI, @#RHOF ; 16 BIT WORDS
; ECC CORRECTION INHIBIT BECAUSE
; ECC LOGIC IS NOT CHECKED YET
CLR @#RHCA ; CYLINDER=0
MOV @#WRCHDT, @#R1 ; WRITE CHECK HEADER AND DATA=52
; INTO RHCS1
JSR PC, @#COMHD ; WRITE CHECK HEADER AND DATA

```



CZRJGBD.RP04/5/6 DSKLS CTRLR1  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 165  
WRITE CHECK ROUTINE

L13

SEQ 0167

6366  
6367  
6368  
6369  
6370

043506 000207

RTS PC

;SAME AS READ HEADER AND DATA  
;RETURN TO WRITE CHECK TEST

6371  
6372  
6373  
6374  
6375  
6376  
6377  
6378  
6379  
6380  
6381  
6382  
6383  
6384  
6385  
6386  
6387  
6388  
6389  
6390  
6391  
6392  
6393  
6394  
6395  
6396  
6397  
6398  
6399  
6400  
6401  
6402  
6403  
6404  
6405  
6406  
6407  
6408  
6409  
6410  
6411  
6412  
6413  
6414  
6415  
6416  
6417  
6418  
6419  
6420  
6421  
6422  
6423  
6424  
6425  
6426

.SBTTL COMPARE ROUTINE

\*THIS IS A SUBROUTINE TO COMPARE TWO BLOCKS IN MEMORY  
\*R1 HAS GOOD DATA BUFFER ADDRESS  
\*R2 HAS TEST DATA BUFFER ADDRESS  
\*STMP0 HAS ADDRESS OF RETURN ON ERROR TO PRINT HEADER  
\*STMP1 HAS ADDRESS OF RETURN ON ERROR TO PRINT DATA  
\*R3 HAS NUMBER OF WORDS TO BE COMPARED  
\*R4 HAS ONE MORE THAN NUMBER OF WORDS TO BE COMPARED

COMPAR:

MOV (R0)+,R1 ; ADDRESS OF GOOD DATA BUFFER  
MOV (R0)+,R2 ; ADDRESS OF TEST DATA BUFFER  
MOV (R0)+,R3 ; NO OF WORDS TO BE COMPARED  
MOV (R0)+,STMP0 ; RETURN ON ERROR TO PRINT HEADER  
MOV (R0)+,STMP1 ; RETURN ON ERROR TO PRINT DATA  
MOV (R0),R0 ; RETURN ON NO ERROR  
MOV R3,R4 ; NO OF WORDS TO BE COMPARED  
INC R4  
1\$: MOV R4,2#ERWORD ; FOR ERROR WORD NO  
CMP (R1)+,(R2)+ ; COMPARE GOOD WITH TEST DATA  
BEQ 3\$ ; BRANCH IF GOOD  
MOV -(R1),2#SGDDAT ; GOOD DATA  
MOV -(R2),2#SBDDAT ; BAD DATA  
SUB R3,2#ERWORD ; ERROR WORD NO.  
TST 2#ERFLGS ; ANY ERRORS ALREAY THERE  
BNE 2\$ ; BRANCH IF YES  
JSR PC,2\$STMP0 ; RETURN TO PRINT HEADER  
BR 5\$ ; BRANCH TO AVOID PRINTING NEXT ERROR  
2\$: JSR PC,2\$STMP1 ; RETURN TO PRINT DATA  
5\$: CMP (R1)+,(R2)+ ; UNDO -(R1) AND -(R2) FOR ERRORS  
MOV 2\$SWR,-(SP) ; GET SWITCH SETTING  
BIC 2#C600,(SP) ; KEEP ONLY SWITCH 7 AND 8  
CMP 2\$SWD7,(SP)+ ; IS 7 SET AND 8 RESET  
BEQ 4\$ ; BRANCH OUT IF YES  
3\$: DEC R3 ; COUNT  
BNE 1\$ ; BRANCH IF ALL NOT DEVICE  
4\$: RTS R0 ; RETURN TO MAIN PROGRAM

\*THIS IS A SUBROUTINE TO DO WRITE CHECK DATA  
\*CYLINDER 0, TRACK 1, SECTOR 1, KEYS 0  
\*THESE ARE TO SET UP FOR DISKLESS USE ONLY

WRCHDA: MOV #FMT22,2#CYL ; CYLINDER 0 FORMAT 16 BIT WORDS  
MOV #1,2#SECOTR+1 ; TRACK=1  
MOV #1,2#SECOTR ; SECTOR=1  
CLR 2#KEY1 ; KEY1=0

6427	043700	005037	047506		CLR	@#KEY2	;KEY2=0
6428	043704	012737	000040	047560	MOV	#32.,@#DAWORD	;NO OF DATA WORDS
6429	043712	005037	047510		CLR	@#X	;THIS IS A READ OPERATION
6430							
6431	043716	004537	044022		JSR	R5,@#CRC	;GO TO CALCULATE CRC
6432	043722	047500			CYL		
6433	043724	051400			WCRC		
6434							
6435							
6436							
6437	043726	004737	042614		JSR	PC,@#CLDISK	;SET UP GENERAL REGISTERS ;AND CLEAR DISK REGISTERS
6438							
6439							
6440	043732	012777	177740	135672	MOV	#-32,@#RHWC	;36 DATA WORDS 4 HEADER WORDS
6441	043740	012777	003154	135666	MOV	#REINTO,@#RHBA	;STARTING ADDRESS OF READ BUFFER
6442	043746	112746	000001		MOVB	#1,-(SP)	;SECTOR=1
6443	043752	112766	000001	000001	MOVB	#1,1(SP)	;TRACK=1 IN UPPER BYTE
6444	043760	012677	135660		MOV	(SP)+,@#RHDST	;TRACK=1, SECTOR=1 IN RHDST
6445	043764	012777	014000	135656	MOV	#FMT22!ECI,@#RHOF	;16 BIT WORDS
6446							
6447							
6448	043772	005077	135654		CLR	@#RHCA	;ECC CORRECTION INHIBIT BECAUSE
6449	044010	013711	002056		MOV	@#WRCHEK,@#R1	;ECC LOGIC IS NOT CHECKED YET
6450	044014	004737	047340		JSR	PC,@#COMHD	;CYLINDER=0 ;WRITE CHECK DATA=50 INTO RHCS1
6451							
6452							
6453	044020	000207			RTS	PC	;WRITE CHECK HEADER AND DATA ;SAME AS READ HEADER AND DATA ;RETURN TO WRITE CHECK TEST

6454  
6455  
6456  
6457  
6458  
6459  
6460  
6461  
6462  
6463  
6464  
6465  
6466  
6467  
6468  
6469  
6470  
6471  
6472  
6473  
6474  
6475  
6476  
6477  
6478  
6479  
6480  
6481  
6482  
6483  
6484  
6485  
6486  
6487  
6488  
6489  
6490  
6491  
6492  
6493  
6494  
6495  
6496  
6497  
6498  
6499  
6500  
6501  
6502  
6503  
6504  
6505  
6506  
6507  
6508  
6509

.SBTTL CRC GENERATION ROUTINE

```

; *THIS IS A SUBROUTINE TO CALCULATE CRC FOR THE FOUR
; *HEADER WORDS AND STORE THEM IN "WCRC" AND "GCRC"
; *R1 - REGISTER FOR CRC, INCREMENTED CRC VALUE IS HERE
; *R2 - THIS HAS BIT POSITION 2 VALUE C
; *R3 - THIS HAS BIT POSITION 16 I.E. OUTPUT BIT VALUE B
; *R4 - THIS HAS BIT POSITION 15 VALUE E
; *STMP0 - NUMBER OF WORDS
; *STMP2 - NUMBER OF BITS PER WORD = 16
; *STMP3 - TEMPORARY REG.
; *STMP4 - TEMPORARY REG TO TRANSFER CARRY
; *STMP5 - THIS HAS DATA BIT VALUE D

; *FETCH DATA BIT D
; *B = D XOR 16
; *C = B XOR 2
; *E = B XOR 15
; *ROTATE RIGHT ONE POSITION
; *B GOES TO POSITION 1
; *C GOES TO POSITION 3
; *E GOES TO POSITION 16
; *REPET 64 TIMES
; *CALL JSR R5 @#CRC
; *X ;FIRST LOCATION AT
; *Y ;PUT CRC IN WCRC FOR READ GCRC FOR WRITE

```

```

044022      CRC:
044024      012500
044036      005001
044040      005037 001210
044044      012737 000004 001176
044052      012037 001204 16$:
044056      012737 000020 001202
044064      013737 001204 001206
044072      006037 001204 15$:
044076      006037 001210
044102      032701 000001
044106      001403
044110      012703 100000
044114      000401
044116      005003 1$:
044120      063703 001210 2$:
044124      032701 040000
044130      001403
044132      012702 100000
044136      000401
044140      005002 3$:
044142      060302 4$:
044144      032701 000002
044150      001403
044152      012704 100000

```

```

MOV (R5)+,R0 ;GET POINTER TO CYL NO.
CLR R1 ;CLEAR WORKING LOCATION
CLR @#STMP5
MOV #4,@#STMP0 ;WORD COUNT
MOV (R0)+,@#STMP3 ;TEMPORARY WORD STORAGE
MOV #16,@#STMP2 ;BIT COUNT
MOV @#STMP3,@#STMP4 ;TEMPORARY WORD STORAGE
ROR @#STMP3 ;GET LSB INTO "C"
ROR @#STMP5 ;GET ABOVE "C" INTO STMP5
BIT #BIT0,R1 ;IS POSITION 15 HIGH
BEQ 1$ ;BRANCH IF POSITION 16 LOW
MOV #BIT15,R3 ;GET POSITION 16
BR 2$
1$: CLR R3 ;GET POSITION 16
2$: ADD @#STMP5,R3 ;XOR POSITION 16 WITH D
;TO GIVE B
BIT #BIT14,R1 ;IS POSITION 2 HIGH
BEQ 3$ ;BRANCH IF POSITION 2 LOW
MOV #BIT15,R2 ;GET POSITION 2
BR 4$
3$: CLR R2 ;GET POSITION 2
4$: ADD R3,R2 ;XOR B WITH POSITION 2
;TO GIVE C
BIT #BIT1,R1 ;IS POSITION 15 HIGH
BEQ 5$ ;BRANCH IF POSITION 15 LOW
MOV #BIT15,R4 ;GET POSITION 15

```

6510	044156	000401		BR	6\$	
6511	044160	005004		5\$: CLR	R4	: GET POSITION 15
6512	044162	060304		6\$: ADD	R3,R4	: XOR POSITION 15 WITH B
6513						: TO GIVE E
6514	044164	006037	001206	ROR	2#STMP4	: GET LSB INTO "C"
6515	044170	006001		ROR	R1	: GET ABOVE C INTO R1
6516	044172	005703		TST	R3	: TEST B
6517	044174	100403		BMI	7\$	: BRANCH IF B=1
6518	044176	042701	100000	BIC	#BIT15,R1	: SET B IN POSITION 1
6519	044202	000402		BR	10\$	
6520	044204	052701	100000	7\$: BIS	#BIT15,R1	: SET B IN POSITION 1
6521	044210	005702		10\$: TST	R2	: TEST C
6522	044212	100403		BMI	11\$	: BRANCH IF C=1
6523	044214	042701	020000	BIC	#BIT13,R1	: GET C IN POSITION 3
6524	044220	000402		BR	12\$	
6525	044222	052701	020000	11\$: BIS	#BIT13,R1	: GET C IN POSITION 3
6526	044226	005704		12\$: TST	R4	: TEST E
6527	044230	100403		BMI	13\$	: BRANCH IF E=1
6528	044232	042701	000001	BIC	#BIT0,R1	: GET E IN POSITION 16
6529	044236	000402		BR	14\$	
6530	044240	052701	000001	13\$: BIS	#BIT0,R1	: GET E IN POSITION 16
6531	044244	005337	001202	14\$: DEC	2#STMP2	: BIT COUNTER
6532	044250	001310		BNE	15\$	: BRANCH IF 16 NOT DONE
6533	044252	005337	001176	DEC	2#STMP0	: WORD COUNTER
6534	044256	001275		BNE	16\$	: BRANCH IF 4 NOT DONE
6535	044260	010135		MOV	R1,2(R5)+	: PUT CRC WHERE DESIRED
6536	044274	000205		RTS	R5	

6537  
6538  
6539  
6540  
6541  
6542  
6543  
6544  
6545  
6546  
6547  
6548  
6549  
6550  
6551  
6552  
6553  
6554  
6555  
6556  
6557  
6558  
6559  
6560  
6561  
6562  
6563  
6564  
6565

```

; *THIS IS A SUBROUTINE TO SET UP THE SIMULATOR DISK FOR
; *CYLINDER 0 (16 BITS PER WORD)
; *TRACK 1, SECTOR 1
; *KEY1 1
; *KEY2 1
; *CRC THROUGH THE JSR R5,2#CRC
; *256 WORDS OF 177400

```

```

; *CALL JSR PC,2#SETDSK

```

SETDSK:	MOV	#177400,R0	: DATA IN THE DISK
	MOV	#256,R1	: COUNTER
	MOV	#DISK,R2	: START OF SIMULATOR DISK
1\$:	MOV	R0,(R2)+	: MOVE IN DATA
	DEC	R1	: COUNT FOR 256
	BNE	1\$	: BRANCH IF 256 NOT COMPLETE
	MOV	#17.,R1	: 2 ECC WORDS, 1 DATA GAP
			: 14 TOLERANCE GAP
2\$:	CLR	(R2)+	: CLEAR ECC, DATA GAP AND
			: TOLERANCE GAP
	DEC	R1	: COUNT
	BNE	2\$	: BRANCH IF NOT COMPLETE

```

; *NOW SET UP FOR DISKLESS USE

```

6566								
6567	044340	012737	010000	047500	MOV	#FMT22,@#CYL	;CYLINDER 0 (16 BIT WORDS)	
6568	044346	112737	000001	047503	MOVB	#1,@#SECOTR+1	;TRACK=1	
6569	044354	112737	000001	047502	MOVB	#1,@#SECOTR	;SECTOR=1	
6570	044362	012737	000001	047504	MOV	#1,@#KEY1	;KEY1=1	
6571	044370	012737	000001	047506	MOV	#1,@#KEY2	;KEY2=1	
6572	044376	013737	000400	047560	MOV	256,@#DAWORD	;NO. OF DATA WORDS	
6573	044404	004537	044022		JSR	R5,@#CRC	;GO TO CALCULATE CRC	
6574	044410	047500			CYL		;FIRST CRC WORD	
6575	044412	051400			WCRC		;PUT CALCULATED CRC	
6576	044422	000207			RTS	PC		

```

6577
6578
6579
6580
6581
6582
6583
6584
6585
6586
6587
6588
6589
6590
6591
6592
6593 044424 010037 002014
6594 044430 162737 000004 002014
6595 044436 004737 042614
6596 044454 011037 001210
6597 044460 012011
6598 044462 012077 135164
6599 044466 112046
6600 044470 105720
6601 044472 112066 000001
6602 044476 105720
6603 044500 012677 135140
6604 044504 012077 135122
6605
6606 044510 012077 135120
6607 044514 012037 047510
6608
6609 044520 012777 014000 135122
6610
6611 044526 005037 002006
6612 044532 004737 047340
6613
6614
6615
6616
6617
6618
6619
6620
6621 044536 004737 042220
6622 044542 005737 002006
6623 044546 001034
6624 044550 005737 047510
6625 044554 001015

; * THIS IS A SUBROUTINE TO CHECK HEADER COMPARE ERROR
; * (BIT #7) AND CRC ERROR (BIT #8)
; * CALL JSR RD, @#HCCRCE
; *
; * COM ; COMMAND-READ HEADER AND DATA
; * ; -WRITE DATA
; * C ; CYLINDER
; * S ; SECTOR
; * T ; TRACK
; * -N. ; WORD COUNT
; * B ; RHBA BUFFER START
; * X ; 1=WRITE DATA 0=READ
; * H ; H=1 HEADER CHECK, H=0 CRC CHECK

HCCRCE: MOV RD, @#PCJSR ; SAVE PC OF JSR+4
SUB #4, @#PCJSR ; GET PC OF JSR
JSR PC, @#CLDISK ; INIT AND SETUP GENERAL REG.
MOV (RD), @#STMP5 ; SAVE COMMAND
MOV (RD)+, @R1 ; COMMAND
MOV (RD)+, @RHCA ; CYLINDER
MOV (RD)+, -(SP) ; SECTOR
MOV (RD)+, 1(SP) ; UP DATE RD
TSTB (RD)+, 1(SP) ; TRACK
MOV (SP)+, @RHDS ; UPDATE RD
MOV (RD)+, @RHWC ; TRACK SECTOR
; NO. OF DATA WORDS +4 HEADER
; IF A READ HEADER AND DATA
; STARTING ADDRESS OF BUFFER
; X=0 READ HEADER AND DATA
; X=1 WRITE DATA
; 16 BITS PER WORD
; ECC CORRECTION INHIBIT
; CLEAR ERROR FLAG
; COMMAND
; * IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
; * FROM THE "COMHD" ROUTINE THAT MEANS SECTOR GAP,
; * FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
; * SYNC BYTE HAVE GONE BY AND SYNC'S WERE CORRECTLY
; * DETECTED
; * HEADER AND DATA ARE TO BE CHECKED.

JSR PC, @#PUTREG ; SAVE REGISTERS
TST @#ERFLGS ; ANY ERRORS ALREADY THERE
BNE 10$ ; BRANCH IF YES
TST @#X ; IS THIS A READ
BNE 3$ ; IF A WRITE DATA BRANCH

```

```

6626
6627
6628
6629
6630
6631
6632
6633 044556 004037 043510 JSR RD,@#COMPAR ;CHECK
6634 044562 002110 WRFROM ;GOOD DATA
6635 044564 003154 REINTO ;TEST BUFFER
6636 044566 000400 256. ;4 HEADER 252 DATA
6637 044570 044576 1$ ;RETURN POINT FOR ERROR HEADER
6638 044572 044602 2$ ;RETURN POINT FOR ERROR DATA
6639 044574 044640 10$ ;RETURN FOR GOOD COMPARISON
6640 044576 104004 1$: ERROR 4 ;READ NEXT ERROR 5
6641 044600 000207 RTS PC ;RETURN TO COMPARISON SUBROUTINE
6642 044602 104005 2$: ERROR 5 ;WORD NO 1 THRU 4 ARE
6643 ;HEADER WORDS AND HENCE
6644 ;SHOULD BE READ AS WRITTEN ON
6645 ;DISK, WORD NOS. 5 ONWARDS
6646 ;SHOULD NOT BE READ AND HENCE
6647 ;READ INTO BUFFER
6648 ;SHOULD BE UNCHANGED
6649 044604 000207 RTS PC ;RETURN TO COMPARISON
6650
6651 044606 000414 BR 10$ ;JUMP OUT
6652
6653
6654
6655
6656
6657
6658
6659 044610 004037 043510 3$: JSR RD,@#COMPAR ;CHECK
6660 044614 003154 REINTO ;GOOD DATA BUFFER
6661 044616 051416 DISK ;TEST BUFFER
6662 044620 000400 256.
6663 044622 044630 4$ ;RETURN POINT FOR ERROR HEADER
6664 044624 044634 5$ ;RETURN POINT FOR ERROR DATA
6665 044626 044640 10$ ;RETURN POINT FOR GOOD COMPARISON
6666 044630 104004 4$: ERROR 4 ;READ NEXT ERROR 5
6667 044632 000207 RTS PC ;RETURN TO COMPARISON SUBROUTINE
6668 044634 104005 5$: ERROR 5 ;WORD NO ARE ALL DATA
6669 ;WORDS THE SHOULD NOT
6670 ;HAVE BEEN CHANGED BY THE
6671 ;WRITE COMMAND
6672 044636 000207 RTS PC ;RETURN TO COMPARISON SUBROUTINE
6673 044640 005720 10$: TST (R0)+ ;IS THIS A HCRC ON HCE CHECK?
6674 044642 001442 BEQ 6$ ;BRANCH IF HCRC
6675 044644 022737 000072 001210 CMP #72,@#STMP5 ;IS THIS A READ COMMAND
6676 044652 001417 BEQ 11$ ;BRANCH IF YES
6677 044654 017737 134762 001126 MOV @RHER1,@#BDDAT ;TEST DATA
6678 044662 022737 000200 001126 CMP #HCE,@#BDDAT ;ONLY HEADER COMPARE BIT?
6679 ;SHOULD BE SET
6680 044670 001470 BEQ 7$ ;BRANCH IF GOOD
6681 044672 013737 001642 042264 MOV @RHER1,@#REGADR ;REGISTER ADDRESS RHER1

```



```

6682 044700 012737 000200 001124      MOV      #HCE, @#SGDDAT ;GOOD DATA
6683 044706 104027                      ERROR    27           ;AFTER AN ERROR ON THE
6684                                     ;HEADER ONLY HCE SHOULD
6685 044710 000460                      BR       7$           ;BE SET
6686 044712                                     11$:
6687 044712 017737 134724 001126      MOV      @#RHER1, @#SBDDAT ;TEST DATA
6688 044720 022737 100200 001126      CMP      #DCK!HCE, @#SBDDAT ;ONLY HEADER COMPARE BIT?
6689                                     ;SHOULD BE SET
6690                                     ;DCK IS SET BECAUSE ECC IS NOT READ
6691 044726 001451                      BEQ      7$           ;BRANCH IF GOOD
6692 044730 013737 001642 042264      MOV      @#RHER1, @#REGADR ;REGISTER ADDRESS RHER1
6693 044736 012737 100200 001124      MOV      #DCK!HCE, @#SGDDAT ;GOOD DATA
6694 044744 104027                      ERROR    27           ;AFTER AN ERROR ON THE
6695                                     ;HEADER ONLY HCE SHOULD
6696 044746 000441                      BR       7$           ;BE SET
6697 044750 022737 000072 001210 6$:    CMP      #72, @#STMP$    ;IS THIS A READ COMMAND?
6698 044756 001417                      BEQ      12$          ;BRANCH IF A READ
6699 044760 017737 134656 001126      MOV      @#RHER1, @#SBDDAT ;TEST DATA
6700 044766 022737 000400 001126      CMP      #HCRC, @#SBDDAT ;ONLY CRC ERROR SHOULD BE THERE
6701 044774 001426                      BEQ      7$
6702 044776 013737 001642 042264      MOV      @#RHER1, @#REGADR ;REG. ADDR = RHER1
6703 045004 012737 000400 001124      MOV      #HCRC, @#SGDDAT ;GOOD DATA
6704 045012 104027                      ERROR    27           ;AFTER A CRC ERROR ONLY CRC
6705                                     ;SHOULD BE SET
6706 045014 000416                      BR       7$           ;BRANCH OUT
6707 045016 017737 134620 001126 12$:  MOV      @#RHER1, @#SBDDAT ;TEST DATA
6708
6709 045024 022737 100400 001126      CMP      #DCK!HCRC, @#SBDDAT ;HCRC AND DCK SHOULD BE SET
6710                                     ;DCK IS SET BECAUSE ECC IS NOT READ
6711 045032 001407                      BEQ      7$           ;BRANCH IF GOOD
6712 045034 012737 100400 001124      MOV      #DCK!HCRC, @#SGDDAT ;GOOD DATA
6713 045042 013737 001642 042264      MOV      @#RHER1, @#REGADR ;FAILING REGISTER RHER1
6714 045050 104027                      ERROR    27           ;AFTER A CRC ERROR ON A READ
6715                                     ;DCK AND HCRC SHOULD BE SET
6716                                     ;DCK IS SET BECAUSE ECC IS NOT READ
6717 045052 000200                      7$:    RTS       R0         ;RETURN TO MAIN TEST
6718
6719
6720
6721                                     ;*THIS IS A SUBROUTINE TO LEAVE AT THE MIDDLE OF
6722                                     ;*A WRITE HEADER AND DATA COMMAND
6723                                     ;*IT TRIES TO GET SECTOR 10, TRACK 0, CYLINDER 0
6724                                     ;*BUT COMES OUT AFTER ONE SECTOR
6725                                     ;*THE COMMAND OS JSR PC, @#MIDDLE
6726                                     ;*BAI IS SET
6727
6728 045054                                     MIDDLE:
6729 045060 013777 002064 134552      MOV      @#WRIFOR, @#RHCSI ;WRITE HEADER AND DATA=62
6730                                     ;IN RHCSI
6731 045066 012777 177766 134536      MOV      #-10, @#RHWC    ;10 WORDS
6732 045074 012777 002110 134532      MOV      #WRFROM, @#RHBA ;BUS ADDRESS=WRFROM
6733 045102 012777 000010 134534      MOV      #10, @#RHDSST   ;DESIRED TRACK=0 SECTOR=10
6734 045110 052777 000010 134520      BIS      #BAI, @#RHCS2   ;BUS ADDRESS INCREMENT INHIBIT
6735 045116 012777 010000 134524      MOV      #FMT22, @#RHOF  ;FORMAT 16 BIT WORDS
6736 045124 005077 134522                      CLR      @#RHCA         ;CYLINDER=0
6737 045130 012737 000001 045156      MOV      #1, @#MID      ;SECTOR IS SET TO 1 SO THAT

```

: WE CAN GET OUT AT THE  
: MIDDLE OF AN OPERATION  
: LOOKING FOR SECTOR 10  
: SET DIAGNOSTIC MODE  
: ;GO TO RHCSI WITH 62

: SECTOR

6738  
6739  
6740  
6741 045136 012777 000001 134514  
6742 045144 052777 000001 134466  
6743 045152 004137 053546  
6744 045156 000000  
6745 045164 000207

MOV #DMD, @RHMR  
BIS #GO, @RHCSI  
JSR R1, @SEARCH  
MID: .WORD 0  
RTS PC

.SBTTL JAM CURRENT CYLINDER ROUTINE

: \*THIS SUBROUTINE WILL CHANGE THE CURRENT CYLINDER REGISTER  
: \*THIS IS DONE BY GIVING A SEEK COMMAND THEN AN INIT  
: \*WHICH WILL LOAD THE CURRENT CYLINDER WITH THE DESIRED CYLINDER VALUE  
: \*  
: \*CALL IS  
: \*JSR RO, @MAKECYL  
: \*XC ; DESIRED VALUE OF CURRENT CYLINDER

6760  
6761 045166  
6762 045170 010037 002014  
6763 045174 162737 000004 002014  
6764 045202 012005  
6765 045204 010577 134442  
6766 045210 005077 134430  
6767 045214 013777 002072 134416  
6768 045222 012777 000001 134430  
6769 045230 052777 000001 134402  
6770 045236 000240  
6771 045240 000240  
6772 045242 000240  
6773 045244 000240  
6774 045246 004737 042614  
6775 045252 017737 134420 001126  
6776 045260 020537 001126  
6777 045264 001406  
6778 045266 010537 001124  
6779 045272 013737 001676 042264  
6780 045300 104030

MAKECYL:  
MOV RO, @PCJSR ; PC OF JSR+4  
SUB #4, @PCJSR ; SAVE PC OF JSR  
MOV (RO)+, R5 ; GETTING READY TO FILL DESIRED CYLINDER  
MOV R5, @RHCA ; FILL DESIRED CYLINDER REGISTER  
CLR @RHST ; MAKE SURE DESIRED SECTOR TRACK IS NOT ILLEGAL  
MOV @SEECOM, @RHCSI ; FILL SEEK COMMAND  
MOV #DMD, @RHMR ; SET DIAGNOSTIC MODE  
BIS #GO, @RHCSI ; GO TO SEEK  
NOP ; ALLOW TIME FOR SEEK TO HANG UP  
NOP ; ALLOW TIME FOR SEEK TO HANG UP  
NOP ; ALLOW TIME FOR SEEK TO HANG UP  
NOP ; ALLOW TIME FOR SEEK TO HANG UP  
JSR PC, @CLDISK ; GIVE INIT  
MOV @RHCC, @SBDDAT ; TEST DATA  
CMP R5, @SBDDAT ; COMPARE CURRENT CYLINDER  
BEQ 1\$ ; BRANCH IF GOOD  
MOV R5, @SGDDAT ; GOOD VALUE OF RHCC  
MOV @RHCC, @REGADR ; FAILING REGISTER ADDRESS  
ERROR 30 ; CURRENT CYLINDER DOES NOT MATCH DESIRED CYLINDER  
; REGISTER AFTER A SEEK AND AN INIT

1\$: RTS RO

.SBTTL ECC GENERATION AND COMPARISON ROUTINE

6781  
6782 045302  
6783 045304 000200  
6784  
6785  
6786  
6787  
6788  
6789  
6790  
6791  
6792  
6793

: \*THIS SUBROUTINE GENERATES AND TESTS ECC  
: \*CALL JSR PC, ECTEST

6794			PIE1	=100000
6795			PIE2	=40000
6796			PIE3	=20000
6797			PIE4	=10000
6798	100000		PIE5	=4000
6799	040000		PIE6	=2000
6800	020000		PIE7	=1000
6801	010000		PIE8	=400
6802	004000		PIE9	=200
6803	002000		PIE10	=100
6804	001000		PIE11	=40
6805	000400		PIE12	=20
6806	000200		PIE13	=10
6807	000100		PIE14	=4
6808	000040		PIE15	=2
6809	000020		PIE16	=1
6810	000010		PIE17	=100000
6811	000004		PIE18	=40000
6812	000002		PIE19	=20000
6813	000001		PIE20	=10000
6814	100000		PIE21	=4000
6815	040000		PIE22	=2000
6816	020000		PIE23	=1000
6817	010000		PIE24	=400
6818	004000		PIE25	=200
6819	002000		PIE26	=100
6820	001000		PIE27	=40
6821	000400		PIE28	=20
6822	000200		PIE29	=10
6823	000100		PIE30	=4
6824	000040		PIE31	=2
6825	000020		PIE32	=1
6826	000010			
6827	000004			
6828	000002			
6829	000001			
6830	045306	000000	ECDATA:	0
6831				
6832				
6833				
6834	045310	000000	GECC1:	0
6835				
6836				
6837	045312	000000	GECC2:	0
6838				
6839				
6840	045314	000000	TSECCG:	0
6841				
6842				
6843	045316	113713	NCODE:	38859.
6844	045320	000000	NCOUNT:	0
6845	045322	000000	POSITI:	0
6846	045324	010041	HARDER:	4129.
6847				
6848				
6849				

```

: DATA BIT FOR ECC
: IF ALL ONES THEN CURRENT BIT IS A ONE
: IF ZERO THEN CURRENT BIT IS A ZERO

: LOW ORDER ECC WORD TO BE GENERATED HERE
:=R1

: HIGH ORDER ECC WORD TO BE GENERATED HERE
:=R2

: IF =177777 GENERATE AND TEST ECC FOR THIS BIT
: IF =0 DO NOT GENERATE AND TEST ECC FOR THIS BIT

: N-CODE WORD
: TEMPORARY N CODE
: POSITION REGISTER
: HARD ERROR COUNT
: TRUE COUNT IS 4128 BUT AS COMPARES ARE
: DONE ONE STAGE LATER SO 4129

```

```

6850 045326 000000 DATENV: 0 ;DATA ENVELOPE FOR TYPE OUT
6851 ;MAX FOR WRITE IS 4096
6852 ;MAX FOR READ IS 4128
6853 045330 000000 ZCODE: 0 ;LEADING ZEROS ENVELOPE FOR TYPE OUT
6854 ;THIS IS SHUT OFF WHEN POSITION COUNTER
6855 ;IN ENABLED
6856 ;MAX COUNT IS 38859
6857
6858
6859
6860 045332 000000 HADTMP: 0 ;TEMPORARY HARD ERROR COUNT
6861 045334 000000 P3: 00
6862 045336 000000 P12: 00
6863 045340 000000 P22: 00
6864 045342 000000 P24: 0
6865
6866
6867
6868
6869
6870 045344 ECTEST:
6871 045360 013701 045310 MOV @#GECC1,R1 ;ECC1 WORD
6872 045364 013702 045312 MOV @#GECC2,R2 ;ECC2 WORD
6873 045370 005737 045306 TST @#ECDATA ;IS CURRENT BIT A ONE
6874 045374 001406 BEQ 2$ ;BRANCH IF CURRENT DATA D=0
6875
6876 ;*IF CARRY IS NOT ZERO THEN D=1
6877 ;*INVERT X32 TO GIVE R0
6878
6879 045376 010103 1$: MOV R1,R3
6880 045400 052703 177776 BIS #1CPIE32,R3
6881 045404 005103 COM R3
6882 045406 010300 MOV R3,R0
6883 045410 000404 BR 3$
6884
6885 ;*IF CARRY IS ZERO THEN D=0
6886 ;*X32 BECOMES R0
6887 045412 010103 2$: MOV R1,R3
6888 045414 042703 177776 BIC #1CPIE32,R3
6889 045420 010300 MOV R3,R0
6890
6891 045422 000241 3$: CLC
6892 045424 006000 ROR R0
6893 045426 006000 ROR R0
6894 045430 005700 TST R0
6895 045432 001462 BEQ 10$ ;BRANCH IF R0=0
6896 ;*INVERT X2
6897
6898 045434 MOV R2,R3
6899 045436 052703 137777 BIS #1CPIE2,R3
6900 045442 005103 COM R3
6901 045444 010337 045334 MOV R3,@#P3
6902 045450 006237 045334 ASR @#P3
6903
6904 ;*INVERT X11
6905

```

6906					
6907	045454	010203		MOV	R2,R3
6908	045456	052703	177737	BIS	#PIE11,R3
6909	045462	005103		COM	R3
6910	045464	010337	045336	MOV	R3,@#P12
6911	045470	006237	045336	ASR	@#P12
6912					
6913					;*INVERT X21
6914					
6915	045474	010103		MOV	R1,R3
6916	045476	052703	173777	BIS	#PIE21,R3
6917	045502	005103		COM	R3
6918	045504	010337	045340	MOV	R3,@#P22
6919	045510	006237	045340	ASR	@#P22
6920					
6921					;*INVERT X23
6922					
6923	045514	010103		MOV	R1,R3
6924	045516	052703	176777	BIS	#PIE23,R3
6925	045522	005103		COM	R3
6926	045524	010337	045342	MOV	R3,@#P24
6927	045530	006237	045342	ASR	@#P24
6928					
6929					;*NOW THAT R0 FOR POSITION 1
6930					;* P3 FOR POSITION 3
6931					;* P12 FOR POSITION 12
6932					;* P22 FOR POSITION 22
6933					;* P24 FOR POSITION 24
6934					;*ARE KNOWN THE ROTATE WILL BE DONE AND
6935					;*THESE BITS JAMED IN
6936					
6937	045534	006002		ROR	R2
6938	045536	006001		ROR	R1
6939	045540	053700	045334	BIS	@#P3,R0
6940	045544	053700	045336	BIS	@#P12,R0
6941	045550	042702	120020	BIC	#PIE1!PIE3!PIE12,R2
6942	045554	050002		BIS	R0,R2
6943					
6944	045556	005000		CLR	R0
6945	045560	053700	045340	BIS	@#P22,R0
6946	045564	053700	045342	BIS	@#P24,R0
6947	045570	042701	002400	BIC	#PIE22!PIE24,R1
6948	045574	050001		BIS	R0,R1
6949	045576	000404		BR	12\$
6950					
6951					;*THE PROGRAM COMES HERE IF R0=0
6952					;*SO AFTER ROTATE R0 GETS PUT INTO POSITION 1
6953					
6954	045600	006002		10\$: ROR	R2
6955	045602	006001		ROR	R1
6956	045604	042702	100000	BIC	#PIE1,R2
6957	045610	010137	045310	12\$: MOV	R1,@#GECC1
6958	045614	010237	045312	MOV	R2,@#GECC2
6959	045620	005737	045314	TST	@#SECCG
6960					;SAVE ECC1
6961					;SAVE ECC2
					;IS HARDWARE TO BE CHECKED
					;IF =1777777 TEST HARDWARE
					;IF = 0 DO NOT TEST HARDWARE

6962	045624	001432			BEQ	14\$			; BRANCH IF HARDWARE NOT TO BE CHECKED
6963									
6964									
6965									
6966	045626	032777	000400	133304	; *CHECK HARDWARE				
6967	045634	001005			BIT	#SW8, @SWR			; IS SWITCH 8 SET
6968	045636	032777	000100	133274	BNE	15\$			; BRANCH IF SW8 IS SET
6969	045644	001401			BIT	#SW6, @SWR			; IS SWITCH 6 SET
6970	045646	000421			BEQ	15\$			; BRANCH IF SW6 IS NOT SET
6971					BR	14\$			; IF SWITCH 8 IS NOT SET AND
6972									; SWITCH 6 IS SET THEN
6973	045650	010146							; DO NOT DO COMPARES
6974	045652	042716	174000		15\$:	MOV	R1, -(SP)		; GOOD PATTERN REGISTER
6975	045656	022677	134010			BIC	#174000, (SP)		; GET ONLY PATTERN BITS
6976	045662	001404				CMP	(SP)+, @RHEC2		; COMPARE PATTERN REGISTER
6977						BEQ	13\$		; BRANCH IF GOOD
6978	045664	004737	042220			; *TO SAVE TIME			
6979	045670	104035				JSR	PC, @PUTREG		; SAVE REGISTERS
6980	045672	000407				ERROR	35		; PATTERN REGISTER IN 11 BITS IN ERROR
6981	045674	023777	045322	133766		BR	14\$		; BRANCH OUT
6982	045702	001403				CMP	@POSITI, @RHEC1		; COMPARE POSITION REGISTER
6983						BEQ	14\$		; BRANCH IF GOOD
6984	045704	004737	042220			; *TO SAVE TIME			
6985	045710	104035				JSR	PC, @PUTREG		; SAVE REGISTERS
6986						ERROR	35		; POSITION REGISTER IN ERROR
6987									; "DATA ENVLOP" GIVES NUMBER OF CLOCK
6988									; PULSES FROM BEGINING OF COMMAND
6989									; THAT IS THE CLOCKS IN THE R/W DATA FIELD ENVELOPE
6990									
6991									; IN A WRITE THERE ARE 10000 OCTAL CLOCKS
6992									; IN A READ THERE ARE 10040 OCTAL CLOCKS
6993									
6994									
6995									; "N-CODE ZEROS" GIVE THE NUMBER OF CLOCKS
6996									; GIVEN FOR THE LEADING ZEROS FIELD
6997									; MAX COUNT IS 113713 OCTAL
6998									
6999									; "GOOD POSITION" GIVES NUMBER OF CLOCKS
7000									; GIVEN AFTER LEADING ZEROS WHICH IS FOR THE DATA
7001									; FIELD
7002									; MAX COUNT IS 10040 OR 10041 OCTAL
7003									
7004	045712								
7005	045726	000207			14\$:	RTS	PC		

M14

CZRJGB0,RP04/5/6 DSKLS CTRLRI  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 179  
ECC GENERATION AND COMPARISON ROUTINE

SEQ 0181

```

7006
7007
7008
7009
7010
7011
7012
7013
7014
7015 045730 000000
7016
7017
7018
7019 045732 010037 002014
7020 045736 162737 000004 002014
7021 045744 012037 045730
7022 045752 013701 001660
7023 045756 012711 000001
7024 045762 005037 045306
7025
7026
7027
7028 045766 005737 045322
7029 045772 001007
7030 045774 005337 045320
7031 046000 001001
7032 046002 000403
7033 046004 005237 045330
7034 046010 000420
7035
7036 046012 005237 045322
7037 046016 023737 045730 045322
7038 046024 103012
7039 046026 023737 045332 045322
7040
7041 046034 001415
7042 046036 032711 000400
7043 046042 001016
7044
7045 046044 004737 042220
7046 046050 104034
7047
7048
7049
7050 046052 052711 000002
7051 046056 042711 000002
7052 046062 004737 045344
7053 046066 000737
7054
7055
7056
7057
7058 046070 052711 000002
7059 046074 042711 000002
7060
7061 046100

```

```

; *THIS SUBROUTINE WILL CONTROL THE ECC GENERATION ROUTINE
; *FOR ERROR CORRECTION PROCESS
; *CALL JSR, PC, @#ECORR
; * XP ; EXPECTED POSITION REGISTER WHEN CORRECTION IS COMPLETE

ERPOS: 0 ; POSITION REG. WHEN CORRECTION IS COMPLETE

ECORR: MOV RO, @#PCJSR ; SAVE PC OF JSR + 4
SUB #4, @#PCJSR ; SAVE PC OF JSR
MOV (RO)+, @#ERPOS ; GET POSITION REG. WHEN CORRECTION IS COMPLETE
MOV @#RHMR, R1 ; MAINTENANCE REGISTER
MOV @#DMD, @R1 ; SET DIAGNOSTIC MODE BIT
CLR @#ECCDATA ; ECC DATA IS ZERO

1$: TST @#POSITI ; IS SOFTWARE POSITION NON ZERO
BNE 2$ ; BRANCH IF N-CODE S COMPLETE
DEC @#NCOUNT ; DECREMENT N-CODE
BNE 6$ ; BRANCH IF N-CODE IS NOT COMPLETE
BR 2$ ; BRANCH AS N-CODE IS COMPLETE
6$: INC @#ZCODE ; INCREMENT CLOCKS GIVEN FOR LEADING ZEROS
BR 3$ ; BRANCH AS N-CODE IS NOT COMPLETE
2$: INC @#POSITI ; INCREMENT SOFTWARE POSITION
CMP @#ERPOS, @#POSITI ; HAVE ENOUGH CLOCKS BEEN GIVEN TO DETECT ERROR
BHS 3$ ; BRANCH IF MORE CLOCKS TO BE GIVEN
CMP @#HADTMP, @#POSITI ; HAVE ENOUGH CLOCKS BEEN GIVEN FOR HARD ERROR
; THAT IS HAVE 4128 MORE CLOCKS BEEN GIVEN
BEQ 5$ ; BRANCH IF YES
BIT #ZER, @R1 ; CHECK ZERO DETECT BIT IN RHMR
BNE 4$ ; BRANCH IS ZER SET
; *TO SAVE TIME
JSR PC, @#PUTREG ; SAVE REGISTERS
ERROR 34 ; ZERO DETECT BIT NOT HIGH
; WHEN 21 BITS IN ECC 32 BIT REGISTER IS 0

3$: BIS #MCLK, @R1 ; SET CLOCK
BIC #MCLK, @R1 ; CLEAR CLOCK
JSR PC, @#ECTEST ; GO TO GENERATE AND TEST ECC
BR 1$ ; CONTINUE

; *THIS EXTRA CLOCK IS TO BRING ECH HIGH
; *AFTER THIS CLOCK POSITION REGISTER MAY BE 10040 OR 10041 OCTAL

5$: BIS #MCLK, @R1 ; SET CLOCK
BIC #MCLK, @R1 ; CLEAR CLOCK

4$:

```

RTS RC

7062 046102 000200

7063  
7064  
7065  
7066  
7067  
7068  
7069  
7070  
7071  
7072  
7073

;\*THIS SUBROUTINE GENERATES THE ECC FOR WHAT IS ON DISK AND INSERTS THEM  
;\*ON LOCATIONS "DISK+1000" AND "DISK+1002"

7074 046104  
7075 046120 005037 045322  
7076 046124 005037 045310  
7077 046130 005037 045312  
7078 046134 012701 051416  
7079 046140 012702 000400  
7080 046144 012703 000020  
7081 046150 012104  
7082 046152 006004  
7083 046154 103004  
7084 046156 012737 177777 045306  
7085 046164 000402  
7086 046166 005037 045306  
7087 046172 004737 045344  
7088 046176 005303  
7089 046200 001364  
7090 046202 005302  
7091 046204 001357  
7092 046206 013737 045310 052416  
7093 046214 013737 045312 052420  
7094 046236 000207

FILLEC:  
CLR @#POSITI ;CLEAR POSITION  
CLR @#GECC1 ;CLEAR GECC1  
CLR @#GECC2 ;CLEAR  
MOV @#DISK,R1 ;POINTER TO DATA FOR ECC GENERATION  
MOV @#256,R2 ;COUNTER FOR NUMBER OF DATA WORDS  
9\$: MOV @#16,R3 ;COUNTER FOR NUMBER OF BITS PER WORD  
MOV (R1)+,R4 ;DATA IN R4  
10\$: ROR R4 ;GET ONE DATA BIT IN CARRY  
BCC 11\$ ;BRANCH IF DATA BIT IS ZERO  
MOV #-1,@#ECDATA ;ECC DATA BIT IS A ONE  
BR 12\$ ;BRANCH TO GENERATE ECC  
11\$: CLR @#ECDATA ;ECC DATA BIT IS A ZERO  
12\$: JSR PC,@#ECTEST ;GO TO GENERATE ECC  
DEC R3 ;DECREMENT BIT COUNT  
10\$: BNE 9\$ ;BRANCH IF 16 BITS NOT DONE  
DEC R2 ;DECREMENT WORD COUNT  
9\$: BNE 9\$ ;BRANCH IF 256 WORDS NOT DONE  
MOV @#GECC1,@#DISK+<256.\*2>; INSERT ECC1 ON DISK  
MOV @#GECC2,@#DISK+<257.\*2>; INSERT ECC2 ON DISK  
RTS PC

.SBTTL RH BASE ADDRESS CHANGE ROUTINE

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

7104  
7105 046240  
7106 046320 013746 001640  
7107 046324 104402  
7108 046406 004737 055172  
7109 046412 104412  
7110 046414 012700 001630  
7111 046420 012701 000026  
7112 046424 012737 047224 000004  
7113 046432 021637 001640  
7114 046436 001407  
7115 046440 005776 000000  
7116 046444 163716 001640  
7117 046450 061620

BASECH:  
MOV @#RHCS1,-(SP) ;GET READY TO TYPE OLD BASE  
TYPOC  
JSR PC,@#STKINT ;INITIALIZE THE TTY KEYBOARD  
RDOCT  
MOV @#RHDB,R0 ;GET STARTING ADDRESS OF REGISTERS  
MOV @#22,R1 ;NUMBER OF REGISTERS  
MOV @#ADTIMO,@#4 ;SET UP TO CHECK NEW ADDRESS  
CMP @#SP,@#RHCS1 ;NEW CSR?  
BEQ 1\$ ;NO-SKIP NEXT  
1\$: TST @#(SP) ;ACCESS THE NEW ADDRESS  
2\$: SUB @#RHCS1,@#SP ;GET THE ADDRESS OFFSET  
ADD @#SP,(R0)+ ;AND PLUG IT IN



```

7118 046453 005301          DEC      R1          ;DONE ALL OF THEM YET?
7119 046454 001375          BNE      2$          ;NOT YET, SO DO MORE
7120 046456                1$:          MOV      @#RPVEC, -(SP) ;GET READY TO TYPE OLD VECTOR ADDRESS
7121 046522 013746 001626        TYP0C
7122 046526 104402          RDOCT
7123 046634 104412          MOV      (SP)+, @#RPVEC ;SETUP VECTOR ADDRESS
7124 046636 012637 001626~  MOV      @#RHCS1, -(SP)
7125 046704 013746 001640        TYP0C
7126 046710 104402          MOV      @#RPVEC, -(SP)
7127 046754 013746 001626        TYP0C
7128 046760 104402          MOV      @#RA, -(SP)
7129 047200 012746 000200        TYP0C
7130 047204 104402          JMP      @#BEGIN      ;OK, NOW START OVER WITH NEW ADDRESS
7131 047220 000137 004240        ADTIMO:  CMP      (SP)+, (SP)+ ;RESTORE THE STACK
7132 047224                JMP      @#BASECH    ;AND DO IT AGAIN.
7133 047302 022626
7134 047304 000137 046240
7135
7136
7137
7138
7139
7140
7141
7142
7143
7144 047310 000000          ;*THIS IS A LITTLE ROUTINE THAT TESTS NED BIT 11 IN RHCS2
7145 047312 004737 042614          ;*THIS LOOPS HERE FOR EVER
7146 047316 013712 047310          ;*TO BE USED ONLY IF DRIVES PRESENT LOOKING AT NED DOES NOT AGREE
7147 047322 005714          ;*WITH WHAT IS REALY THERE
7148 047324 032712 010000          ERUNIT: 0          ;UNIT UNDER MANUAL TEST
7149 047330 001401          ERSTART: JSR     PC, @#CLDISK ;SET GENERAL REG.
7150 047332 000773          MOV      @#ERUNIT, @R2 ;SELECT UNIT
7151 047334 000772          1$:          TST      @R4          ;TEST RHER1
              BIT      @#NED, @R2 ;TEST NED
              BEQ     2$          ;BRANCH IF GOOD
              BR     1$          ;NED NOT SET
              BR     1$          ;NED SET

```

7152  
7153  
7154  
7155  
7156  
7157  
7158  
7159  
7160  
7161  
7162  
7163  
7164  
7165  
7166  
7167  
7168  
7169  
7170  
7171  
7172  
7173  
7174  
7175  
7176  
7177  
7178  
7179  
7180  
7181  
7182  
7183  
7184  
7185  
7186  
7187  
7188  
7189  
7190  
7191  
7192  
7193  
7194  
7195  
7196  
7197  
7198  
7199  
7200  
7201  
7202  
7203  
7204  
7205

```

.SBTTL DISK SIMULATION
*IN A WRITE HEADER AND DATA COMMAND FILL THE FOLLOWING
*WCLY=WITH CYLINDER TO BE ON DISK
*WSECTR=WITH SECTOR AND TRACK TO BE ON DISK
*WKEY1= WITH KEY1 TO BE ON DISK
*WKEY2= WITH KEY2 TO BE ON DISK
*FNWORD= NO OF DATA WORDS TO BE WRITTEN ON DISK
*THE COMMAND THEN IS JSR PC,COMWHD
*
*
*
*IN A WRITE DATA COMMAND FILL THE FOLLOWING
*CYL=WITH CYLINDER TO BE FOUND ON DISK
*SECTR= WITH SECTOR AND TRACK TO BE FOUND ON DISK
*KEY1= WITH KEY1 TO BE FOUND ON DISK
*KEY2= WITH KEY2 TO BE FOUND ON DISK
*X= 1 MUST BE ONE
*NOWORD= WITH NUMBER OF DATA WORDS TO BE WRITTEN
*THE COMMAND THEN IS JSR PC,COMHD
*
*
*
*IN A READ HEADER AND DATA COMMAND FILL THE FOLLOWING
*CYL= WITH CYLINDER TO BE FOUND ON DISK
*SECTR= WITH SECTOR AND TRACK TO BE FOUND ON DISK
*KEY1= WITH KEY1 TO BE FOUND ON DISK
*KEY2=WITH KEY2 TO BE FOUND ON DISK
*DAWORD= WITH NUMBER OF WORDS TO BE FOUND ON DISK
* X=0 MUST BE ZERO
*THE COMMAND THEN IS JSR PC,COMHD
*
*
*
*IN A READ DATA COMMAND FILL THE FOLLOWING
*CYL= WITH CYLINDER TO BE FOUND ON DISK
*SECTR= WITH SECTOR AND TRACK TO BE FOUND ON DISK
*KEY1= WITH KEY1 TO BE FOUND ON DISK
*KEY2=WITH KEY2 TO BE FOUND ON DISK
*DAWORD= WITH NUMBER OF WORDS TO BE FOUND ON DISK
* X=0 MUST BE ZERO
*THE COMMAND THEN IS JSR PC,COMHD
*

```

7206  
7207  
7208  
7209  
7210  
7211  
7212  
7213  
7214  
7215  
7216  
7217  
7218  
7219  
7220  
7221  
7222  
7223  
7224  
7225  
7226  
7227  
7228  
7229  
7230  
7231  
7232  
7233  
7234  
7235  
7236  
7237  
7238  
7239  
7240  
7241  
7242  
7243  
7244  
7245  
7246  
7247  
7248  
7249  
7250  
7251  
7252  
7253  
7254  
7255  
7256  
7257  
7258  
7259  
7260  
7261

;\*WRITE DATA COMMAND  
;\*OR READ COMMAND I.E DATA ONLY OR HEADER AND DATA

\*\*\*THIS SUBROUTINE IS THE FIRST IN A SERIES OF NESTED SUBROUTINES  
\*\*IT ISSURE DIAGNOSTIC MODE, AN EXTRA DIAGNOSTIC INDEX, AND THE  
\*\*'GO' BIT  
\*\*IT THEN CALL THREE OTHER SUBROUTINES, WHICH IN TURN CALL OTHER  
\*\*SUBROUTINES. THESE ARE:

\*\*\* SEARCH  
\*\*\* RDHEAD  
\*\*\* WRDATA  
\*\*\* REDATA

```

047336 000000          RUNCTR: .WORD 0
047340 011637 002014  COMMD: MOV (SP), @#PCJSR ;SAVE PC OF JSR + 4
047344 162737 000004 002014  SUB #4, @#PCJSR ;SAVE PC OF JSR
047366 012777 000001 132264  MOV #DMD, @RHMR ;SET DIAGNOSTIC MODE
047374 052777 000004 132256  BIS #MINX, @RHMR ;SET DIAGNOSTIC INDEX
047402 042777 000004 132250  BIC #MINX, @RHMR ;CLEAR DIAGNOSTIC INDEX
047410 052777 000001 132222  BIS #GO, @RHCSI ;ISSUE 'GO' BIT & STALL 'TILL 'RUN'
                                ;(FUNCTION CODE IS ISSUED BY THE TEST)
047416 012737 000113 047336  RUNWAT: MOV #75., @#RUNCTR ;LOAD STALL COUNT = APPROX. 450US
                                ;FOR 11/50 CPU WITH CORE MEMORY
047424 005337 047336  1$: DEC @#RUNCTR ;COUNT DOWN ONE
047430 001375 1$: BNE 1$ ;CONTINUE UNTIL = 0
047432 013746 047502  MOV SECOTR, -(SP) ;GET DESIRED SECTOR/TRACK
047436 042716 177740  BIC #177740, (SP) ;MAKE ONLY SECTOR
047442 012637 047452  MOV (SP)+, @#TRK ;SAVE SECTOR
047446 004137 053546  JSR R1, @#SEARCH ;ISSUE SECTOR CLOCKS (-----)
047452 000000          TRK: .WORD 0
047454 012701 000240  MOV #+NOP, R1 ;GOING TO MOVE NOPS
047460 010137 047512  MOV R1, @#SSYN ;NOP INTO SSYN
047464 010137 047514  MOV R1, @#HEDGAP ;NOP INTO HEDGAP
047470 010137 047516  MOV R1, @#HEDSYN ;NOP INTO HEDSYN
047474 004137 047622  JSR R1, @#RDHEAD

```

```

7262
7263
7264                                     ;*DUMMY ERROR CALL LOCATIONS FOR THE READ HEADER OPERATION
7265
7266 047500 000000          CYL:  .WORD  0          ;CYLINDER ADDRESS
7267 047502 000000          SECOTR: .WORD  0          ;SECTOR/TRACK ADDRESS
7268 047504 000000          KEY1:  .WORD  0          ;KEY1 WORD
7269 047506 000000          KEY2:  .WORD  0          ;KEY2 WORD
7270 047510 000000          X:      .WORD  0          ;X=1 WRITE COMMAND
7271                                     ;X=0 READ COMMAND
7272
7273 047512 000240          SSYN:  NOP          ; IF "ERROR 2" INSERTED BY RDHEAD
7274                                     ; SUBROUTINE THEN THE FIRST SYNC.
7275                                     ; IS NOT DETECTED. NO BAD DATA
7276                                     ; IS GIVEN BECAUSE SYNC=144000
7277                                     ; CANNOT BE READ. WORD NO
7278                                     ; IS "1" BECAUSE THIS IS THE FIRST
7279                                     ; WORD TESTED
7280
7281 047514 000240          HEDGAP: NOP          ; IF "ERROR 3" INSERTED BY
7282                                     ; RDHEAD SUBROUTINE THEN THE
7283                                     ; HEADER GAP 0'S WERE NOT
7284                                     ; WRITTEN RIGHT.
7285                                     ; IF "WORD NO" CONTAINS, SAY
7286                                     ; 3(8), THEN IT IS THE THIRD
7287                                     ; WORD OF A 5 WORD HEADER
7288                                     ; GAP THAT IS WRONG
7289                                     ; "BAD DATA" CONTAINS WHAT IS
7290                                     ; GOING ON THE DISK
7291
7292 047516 000240          HEDSYN: NOP          ; IF "ERROR 3" INSERTED BY RDHEAD
7293                                     ; SUBROUTINE THEN THE HEADER SYNC.
7294                                     ; GENERATED BY DCL IS WRONG
7295                                     ; OR THE LAST BYTE
7296                                     ; OF THE HEADER GAP 0'S IS WRONG
7297                                     ; IN EITHER CASE WORD NO=6
7298                                     ; RIGHT BYTE IS HEADER 0
7299                                     ; LEFT BYTE IS SYNC
7300                                     ; "BAD DATA" HAS WHAT IS GOING
7301                                     ; ON DISK
7302
7303 047520 005737 002006          TST      @#ERFLGS          ; ARE ANY ERRORS DETECTED
7304 047524 001017          BNE      OUT          ; IF YES, EXIT ----->
7305 047526 005737 047510          TST      @#X          ; IS IT A DATA WRITE ?
7306 047532 001410          BEQ      DAREAD          ; NO... THEN DO A DATA READ
7307 047534 005737 047614          TST      @#NOSYNC          ; IS THIS FORCED HEADER ERROR COMMAND
7308                                     ; IF YES NOSYNC=-1 THEN WRITE OR READ
7309                                     ; IS SHUT OFF SO BRANCH OUT
7310                                     ; IF NOSYNC=0 THEN CONTINUE
7311 047540 001011          BNE      OUT          ;EXIT IF SET ----->
7312
7313 047542 004137 051066          JSR      R1,@#WRDATA          ;WRITE DATA (<----->)
7314
7315 047546 000000          NOWORD: .WORD  0          ;NO OF WORDS TO BE WRITTEN
7316 047550 000000          Y:      .WORD  0          ;
7317 047552 000404          BF      OUT          ;EXIT ----->

```

CZRJGBD.RP04/5/6 DSKLS CTRLR1  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 185  
DISK SIMULATION

SEQ 0187

7318			
7319	047554	004137	054022
7320	047560	000000	
7321	047562	000000	
7322	047564		
7323	047600	000207	

DAREAD:	JSR	R1,2#REDATA	;READ DATA <----->
DAWORD:	.WORD	0	;NO OF WORDS TO BE READ
	.WORD	0	
OUT:	RTS	PC	;EXIT ROUTINE

7332  
73325  
73326  
73327  
73328  
73329  
73330  
73331  
73332  
73333  
73334  
73335  
73336  
73337  
73338  
73339  
73340  
73341  
73342  
73343  
73344  
73345  
73346  
73347  
73348  
73349  
73350  
73351  
73352  
73353  
73354  
73355  
73356  
73357  
73358  
73359  
73360  
73361  
73362  
73363  
73364  
73365  
73366  
73367  
73368  
73369  
73370  
73371  
73372  
73373  
73374  
73375  
73376  
73377  
73378  
73379  
73380  
73381  
73382  
73383  
73384  
73385  
73386  
73387  
73388  
73389  
73390  
73391  
73392  
73393  
73394  
73395  
73396  
73397  
73398  
73399  
73400

047602 014400  
047604 000000  
047606 000000  
047610 000000  
047612 000000

:\*THE DISK SECTOR IS DEVIDED AS FOLLOWS  
:\*19 WORDS OF 0, ONE WORD 144000  
:\*THESE MAKE 39 BYTES FOR SECTOR GAP AND ONE SYNC. BYTE

RSYNC: 14400  
RCYL: 0  
RSETR: 0  
RKEY1: 0  
RKEY2: 0

:\*5 WORDS OF 0 ONE WORD 144000  
:\*THESE MAKE 11 BYTES FOR HEADER GAP AND ONE SYNC. BYTE  
:\*THESE ARE DCL GENERATED

:\*THERE ARE 256 WORDS OF DATA  
:\*THERE ARE 2 WORDS FOR ECC GENERATED BY DCL  
:\*15 WORDS OF 0 FOR DATA GAP AND TOLERANCE GAP

7334  
7335  
7336  
7337  
7338  
7339  
7340  
7341  
7342  
7343  
7344  
7345  
7346  
7347  
7348  
7349  
7350  
7351  
7352  
7353  
7354  
7355  
7356  
7357  
7358  
7359  
7360  
7361  
7362  
7363  
7364  
7365  
7366  
7367  
7368  
7369  
7370  
7371  
7372  
7373  
7374  
7375  
7376  
7377  
7378  
7379  
7380  
7381  
7382  
7383  
7384  
7385  
7386  
7387  
7388  
7389  
7390  
7391  
7392  
7393  
7394  
7395  
7396  
7397  
7398  
7399  
7400  
7401  
7402  
7403

;\*READ DISK HEADER

047614 000000  
047616 000000  
047620 000000

NOSYNC: 0  
TY: 0  
ERWORD: 0

;FORCED HEADER ERROR = -1  
;NORMAL = 0  
;ERROR TYPE NO.  
;ERROR WORD NO.

RDHEAD: MOV (R1)+, @#RCYL ;STORE CYLINDER ADDRESS  
MOV (R1)+, @#RSETR ;STORE SECTOR AND TRACK ADDRESS  
MOV (R1)+, @#RKEY1 ;STORE KEY1  
MOV (R1)+, @#RKEY2 ;STORE KEY2  
MOV (R1)+, @#COMPA ;STORE COMPARE OR NOT  
MOV @#RHMR, R0 ;R0 CONTAINS MAINTANENCE REG.  
MOV #2, R5 ;R5 IS A COUNTER FOR WORDS  
MOV @#DMD, @RO ;DIAG. MODE  
BIS #MSTCK, @RO ;SET SECTOR FOR FIRST WORD  
BIS #MCLK, @RO ;SET CLOCK FOR FIRST WORD  
BIC #MSTCK!MCLK, @RO ;RESET SECTOR AND CLOCK  
BR 2\$ ;BRANCH OVER GIVING SECTOR FOR FIRST TIME  
1\$: MOV #MSTCK!MCLK!DMD, @RO ;SET SECTOR, CLOCK, DIAG. MODE, RESET INDEX  
BIC #MSTCK!MCLK, @RO ;RESET SECTOR, CLOCK  
2\$: MOV #7, R2 ;R2 IS A COUNTER FOR BYTES  
3\$: BIS #MCLK, @RO ;SET CLOCK  
BIC #MCLK, @RO ;RESET CLOCK  
DEC R2 ;BYTE COUNTER  
3\$: BNE ;BRANCH IF BYTE NOT COMPLETE  
DEC R5 ;WORD COUNTER  
1\$: BNE ;BRANCH IF WORD NOT COMPLETE  
4\$: MOV #18, R2 ;NO OF WORDS OF ZEROS  
CLR @#WORD ;READ 0  
JSR PC, @#READ ;GO TO READ  
DEC R2 ;COUNT  
4\$: BNE ;COUNT  
5\$: MOV @#RSYNC, @#WORD ;SYNC. WORD  
JSR PC, @#READ  
BIT @#DSY, @RO ;SYNC. BYTE DETECTED?  
BNE 5\$ ;BRANCH IF SYNC DETECTED  
MOV #1, @#ERWORD ;ERROR WORD NO  
MOV @#RSYNC, @#SGDDAT ;SYNC WORD  
MOV #104002, @#SSYN ;INSERT "ERROR 2" IN SSYN  
BR 13\$ ;BRANCH OUT  
5\$: MOV @#RCYL, @#WORD ;SETUP CYLINDER  
JSR PC, @#READ ;READ  
MOV @#RSETR, @#WORD ;SETUP SECTOR/TRACK  
JSR PC, @#READ ;READ

7404	050046	013737	047610	050410	MOV	Q#RKEY1,Q#WORD	; SETUP KEY1
7405	050054	004737	050414		JSR	PC,Q#READ	; READ
7406	050060	013737	047612	050410	MOV	Q#RKEY2,Q#WORD	; SETUP KEY2
7407	050066	004737	050414		JSR	PC,Q#READ	; READ
7408	050072	013737	051400	050410	MOV	Q#WCRC,Q#WORD	; SETUP CRC
7409	050100	004737	050414		JSR	PC,Q#READ	; READ
7410	050104	005737	002026		TST	Q#TESDTE	; IS THIS A DRIVE TIMING ERROR
7411	050110	001135			BNE	13\$	; BRANCH OUT IF YES
7412	050112	005737	050412		TST	Q#COMPA	; IS THIS A READ OR WRITE COMMAND
7413	050116	001472			BEQ	11\$	
7414	050120	012705	051402		MOV	#HEGAP,R5	; POINTER FOR HEADER GAP
7415	050124	012702	000005		MOV	#5,R2	; NO OF WORDS OF ZEROS
7416	050130	012737	000006	047620	MOV	#6,Q#ERWORD	; ERROR WORD NO SET
7417	050136	004737	050646		JSR	PC,Q#WRITE	; FOR HEADER GAP
7418	050142	005737	050644		TST	Q#WORD	; TEST WRITTEN WORD
7419	050146	001413			BEQ	7\$	; BRANCH IF GOOD THAT IS 0
7420	050150	160237	047620		SUB	R2,Q#ERWORD	; WORD NO IN ERROR
7421	050154	005037	001124		CLR	Q#\$GDDAT	; GOOD WORD SHOULD BE 0
7422	050160	013737	050644	001126	MOV	Q#WORD,Q#\$BDDAT	; BAD DATA
7423	050166	012737	104003	047514	MOV	#104003,Q#HEDGAP	; "ERROR 2" GOES IN HEDGAP
7424	050174	000503			BR	13\$	; BRANCH OUT
7425	050176	013725	050644		MOV	Q#WORD,(R5)+	; SAVE HEADER GAP
7426	050202	005302			DEC	R2	
7427	050204	001351			BNE	6\$	
7428	050206	004737	050646		JSR	PC,Q#WRITE	; WRITE HEADER (DATA) GAP SYNC
7429	050212	023737	047602	050644	CMP	Q#RSYNC,Q#WORD	
7430	050220	001426			BEQ	10\$	
7431	050222	005737	047614		TST	Q#NOSYNC	; IS THIS FORCED HEADER ERROR COMMAND
7432							; IF YES NOSYNC=-1 THEN WRITE OR READ
7433							; IS SHUT OFF SO BRANCH OUT
7434							; IF NO NOSYNC=0 THEN CONTINUE
7435	050226	001406			BEQ	14\$	; BRANCH IF TRUE ERROR
7436	050230	005737	050644		TST	Q#WORD	
7437	050234	001420			BEQ	10\$	; BRANCH IF GOOD
7438	050236	005037	001124		CLR	Q#\$GDDAT	; IT SHOULD BE ZERO
7439	050242	000403			BR	15\$	; BRANCH TO TYPE ERROR
7440	050244	013737	047602	001124	MOV	Q#RSYNC,Q#\$GDDAT	; GOOD DATA
7441	050252	013737	050644	001126	MOV	Q#WORD,Q#\$BDDAT	; BAD DATA
7442	050260	012737	000006	047620	MOV	#6,Q#ERWORD	
7443	050266	012737	104003	047516	MOV	#104003,Q#HEDSYN	
7444	050274	000443			BR	13\$	; BRANCH OUT
7445	050276	013725	050644		MOV	Q#WORD,(R5)+	; SAVE DATA SYNC.
7446	050302	000440			BR	13\$	
7447							; *READ COMMAND START FROM HERE
7448	050304	012702	000005		MOV	#5,R2	
7449	050310	005037	050410		CLR	WORD	
7450	050314	004737	050414		JSR	PC,READ	; READ HEADER GAP
7451	050320	005302			DEC	R2	; IS 5 HEADER GAP ZEROS COMPLETE
7452	050322	001372			BNE	12\$	; IF NOT BRANCH
7453	050324	013737	047602	050410	MOV	Q#RSYNC,Q#WORD	; SYNC WORD
7454	050332	004737	050414		JSR	PC,READ	; READ HEADER (DATA) SYNC
7455	050336	005737	047614		TST	Q#NOSYNC	
7456	050342	001404			BEQ	16\$	; IF NOT ERROR COMMAND BRANCH
7457	050344	032710	001000		BIT	#DTSY,Q#D	; SYNC. DETECTED
7458	050350	001415			BEQ	13\$	; IF ZERO BRANCH OUT
7459	050352	000403			BR	17\$	; IF NOT ZERO BRANCH TO ERROR



CZRJGBO,RP04/5/6 DSKLS CTRLR1  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 189  
DISK SIMULATION

SEQ 0191

7460	050354	032710	001000		16\$:	BIT	#DTSY, @RO	; SYNC. DETECTED?
7461	050360	001011				BNE	13\$	; BRANCH IF YES
7462	050362	012737	000006	047620	17\$:	MOV	#6, @#ERWORD	; ERROR WORD NO.
7463	050370	013737	047602	001124		MOV	@#RSYNC, @#\$GDDAT	; SYNC WORD
7464	050376	012737	104002	047516		MOV	#104002, @#HEDSYN	
7465	050404				13\$:			
7466	050406	000201				RTS	R1	
7467								
7468								

7469  
7470  
7471  
7472  
7473  
7474  
7475  
7476  
7477  
7478  
7479  
7480  
7481  
7482  
7483  
7484  
7485  
7486  
7487  
7488  
7489  
7490  
7491  
7492  
7493  
7494  
7495  
7496  
7497  
7498  
7499  
7500  
7501  
7502  
7503  
7504  
7505  
7506  
7507  
7508  
7509  
7510  
7511  
7512  
7513  
7514  
7515  
7516  
7517  
7518  
7519  
7520  
7521  
7522  
7523  
7524

050410 000000  
050412 000000  
  
050414  
050416 012705 000002  
050422 012710 000001  
050426 006037 050410  
050432 103002  
050434 052710 000020  
050440 012702 000007  
050444 052710 000012  
050450 005737 045314  
050454 001411  
050456 032710 000020  
050462 001404  
050464 012737 177777 045306  
050472 000402  
050474 005037 045306  
050500 012746 000001  
050504 006037 050410  
050510 103002  
050512 012716 000021  
050516 012610  
050520 005737 045314  
050524 001404  
050526 005237 045326  
050532 004737 045344  
050536 052710 000002  
050542 005737 045314  
050546 001411  
050550 032710 000020  
050554 001404  
050556 012737 177777 045306  
050564 000402  
050566 005037 045306  
050572 012746 000001  
050576 006037 050410  
050602 103002  
050604 012716 000021  
050610 012610  
050612 005737 045314  
050616 001404

; \*READ ONE WORD IN "WORD"

WORD: 0  
COMPA: 0

READ:

```
MOV #2, R5 ; WORD COUNTER
ROR @#WORD ; SET DIAG. MODE
BCC 1$ ; CHECKING IF THERE IS A ONE
; IF NO ONE BRANCH
BIS #MRD, @RO ; SET BIT 4 IF DATA HAS ONE
MOV #7, R2 ; BYTE COUNTER
BIS @#STCK!MCLK, @RO ; SET CLOCK DATA IF ANY, SECTOR
TST @#TSECCG ; IS THIS BIT TO GENERATE AND TEST ECC
BEQ 6$ ; BRANCH IF NO
BIT #MRD, @RO ; IS DATA BIT A ONE
BEQ 5$ ; BRANCH IF DATA BIT IS 0
MOV #-1, @#ECDATA ; ECC DATA BIT IS A ONE
BR 6$ ; BRANCH
CLR @#ECDATA ; ECC DATA BIT IS A 0
MOV #DMD, -(SP) ; KEEP ONLY DIAG. MODE
ROR @#WORD ; CHECKING IF THERE IS A ONE
BCC 2$ ; IF NO ONE BRANCH
MOV #MRD!DMD, (SP) ; KEEP DATA AND DIAG. MODE
MOV (SP)+, @RO ; PUT IN DATA, RESET CLOCK, SECTOR
TST @#TSECCG ; IS ECC TO BE GENERATED FOR THIS BIT
BEQ 3$ ; BRANCH IF NO
INC @#DATENV ; NUMBER OF CLOCKS GIVEN FOR DATA ENVELOPE
JSR PC, @#ECTEST ; GO TO GENERATE AND TEST ECC
BIS #MCLK, @RO ; SET CLOCK
TST @#TSECCG ; IS THIS BIT TO GENERATE ECC
BEQ 8$ ; BRANCH IF NO
BIT #MRD, @RO ; IS DATA BIT A ONE
BEQ 7$ ; BRANCH IF DATA BIT IS = 0
MOV #-1, @#ECDATA ; ECC DATA BIT IS A ONE
BR 8$ ; BRANCH
CLR @#ECDATA ; ECC DATA BIT IS = 0
MOV #DMD, -(SP) ; KEEP DIAG. MODE
ROR @#WORD ; CHECKING IF THERE IS A ONE
BCC 4$ ; BRANCH IF NO ONE
MOV #MRD!DMD, (SP) ; KEEP DIAG. MODE AND DATA
MOV (SP)+, @RO ; SET DATA, DIAG. MODE, CLEAR CLOCK
TST @#TSECCG ; IS THIS BIT TO GENERATE ECC
BEQ 9$ ; BRANCH IF NO
```

7525	050620	005237	045326
7526	050624	004737	045344
7527	050630	005302	
7528	050632	001341	
7529	050634	005305	
7530	050636	001300	
7531	050642	000207	
7532			
7533			

9\$:

INC	Q#DATENV
JSR	PC,Q#ECTEST
DEC	R2
BNE	3\$
DEC	R5
BNE	1\$
RTS	PC

```

;NUMBER OF CLOCKS GIVEN FOR DATA ENVELOPE
;GO TO GENERATE AND TEST ECC
;BYTE COUNTER
;BRANCH IF ONE BYTE NOT COMPLETE
;WORD COUNTER
;BRANCH IF ONE WORD NOT COMPLETE

```

7534  
7535  
7536  
7537  
7538  
7539  
7540  
7541  
7542  
7543  
7544  
7545  
7546  
7547  
7548  
7549  
7550  
7551  
7552  
7553  
7554  
7555  
7556  
7557  
7558  
7559  
7560  
7561  
7562  
7563  
7564  
7565  
7566  
7567  
7568  
7569  
7570  
7571  
7572  
7573  
7574  
7575  
7576  
7577  
7578  
7579  
7580  
7581  
7582  
7583  
7584  
7585  
7586  
7587  
7588  
7589

; \*WRITE ONE WORD WHICH COMES BACK IN "WORD"

050644 000000

WORD: 0

050646

WRITE:

050656 012705 000002  
050662 012710 000001

MOV #2, R5  
MOV #1, @RO

; WORD COUNTER  
; SET DIAG. MODE  
; RO HAS RHMR ADDRESS IN IT

050666 012702 000007  
050672 012710 000013  
050676 032710 000040

1\$:

MOV #7, R2  
MOV #MSTCK!MCLK!DMD, @RO  
BIT #MWR, @RO

; BYTE COUNTER  
; @RO; SET SECTOR AND CLOCK  
; CHECK WRITE BIT IN MAINT. REG.

050702 001406  
050704 012737 177777 045306

BEQ #2\$  
MOV #-1, @#ECDATA

; BRANCH IF ZERO  
; ECC DATA BIT IS A ONE

050712 000261  
050714 006003

SEC  
ROR R3

; SET CARRY  
; MOVE 1 FORWARD

050716 000404  
050720 005037 045306

2\$:

BR 3\$  
CLR @#ECDATA

; ECC DATA BIT IS = 0

050724 000241  
050726 006003

CLC  
ROR R3

; CLEAR CARRY  
; MOVE 0 FOR WORD

050730 012710 000001  
050734 005737 045314

3\$:

MOV #DMD, @RO  
TST @#TSECCG

; CLEAR SECTOR AND CLOCK  
; IS THIS BIT TO GENERATE ECC

050740 001404  
050742 005237 045326

BEQ 4\$  
INC @#DATENV

; BRANCH IF NO  
; NUMBER OF CLOCKS GIVEN FOR DATA ENVELOPE

050746 004737 045344  
050752 052710 000002

4\$:

JSR PC, @#ECTEST  
BIS #MCLK, @RO

; GO TO GENERATE AND TEST ECC  
; SET CLOCK IN RHMR

050756 032710 000040  
050762 001406  
050764 012737 177777 045306

BIT #MWR, @RO  
BEQ 5\$  
MOV #-1, @#ECDATA

; CHECK WRITE BIT IN RHMR  
; BRANCH IF ZERO  
; ECC DATA BIT IS A ONE

050772 000261  
050774 006003

SEC  
ROR R3

; SET CARRY  
; MOVE 1 FOR WORD

050776 000404  
051000 005037 045306

5\$:

BR 6\$  
CLR @#ECDATA

; ECC DATA BIT IS ZERO

051004 000241  
051006 006003

CLC  
ROR R3

; CLEAR CARRY  
; MOVE 0 FOR WORD

051010 012710 000001  
051014 005737 045314

6\$:

MOV #DMD, @RO  
TST @#TSECCG

; CLEAR CLOCK  
; IS THIS BIT TO GENERATE ECC

051020 001404  
051022 005237 045326

BEQ 7\$  
INC @#DATENV

; BRANCH IF NO  
; NUMBER OF CLOCKS GIVEN FOR DATA ENVELOPE

051026 004737 045344  
051032 005302

7\$:

JSR PC, @#ECTEST  
DEC R2

; GO TO GENERATE AND TEST ECC  
; COUNT FOR BYTE END

051034 001346  
051036 005305

BNE 4\$  
DEC R5

; IF NOT BYTE END BRANCH  
; COUNT FOR WORD END

051040 001312

BNE 1\$

; IF NOT WORD END BRANCH

N15

CZRJGB0.RP04/5/6 DSKLS CTRLR1  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 193  
DISK SIMULATION

SEQ 0195

7590 051042 010337 050644  
7591  
7592 051056 000207  
7593  
7594

MOV R3,2#WORD ;STORE THE WORD  
RTS PC

7595  
7596  
7597  
7598  
7599  
7600  
7601  
7602  
7603  
7604  
7605  
7606  
7607  
7608  
7609  
7610  
7611  
7612  
7613  
7614  
7615  
7616  
7617  
7618  
7619  
7620  
7621  
7622  
7623  
7624  
7625  
7626  
7627  
7628  
7629  
7630  
7631  
7632  
7633  
7634  
7635  
7636  
7637  
7638  
7639  
7640  
7641  
7642  
7643  
7644  
7645  
7646  
7647  
7648  
7649  
7650

051060 000000  
051062 000400  
051064 000000  
051066 011137 051060  
051072 012102  
051074 012137 050412  
051112 012701 000016  
051116 012703 052424  
051122 012723 177777  
051126 005301  
051130 001374  
051132 013700 001660  
051136 013746 051062  
051142 163716 051060  
051146 011637 051064  
051152 012604  
051154 005737 002024  
051160 001403  
051162 012737 177777 045314  
051170 012703 051416  
051174 004737 050646  
051200 013723 050644  
051204 005302  
051206 001372  
051210 005704  
051212 001406  
051214 004737 050646  
051220 013723 050644  
051224 005304  
051226 001372  
051230 005037 045314  
051234 012701 000002  
051240 004737 050646  
051244 013723 050644  
051250 005301  
051252 001372  
051254 004737 050646  
051260 013723 050644  
051264 012701 000016  
051270 004737 050646

;\*WRITE DATA - PUT DATA INTO "DISK" AREA FROM "WORD"  
;\*ONE WORD AT A TIME

COUNTD: 0  
FORMAT: 256.  
ZWORDS: 0  
WRDATA:  
MOV (R1), @#COUNTD ; STORE NO. OF WORDS TO BE WRITTEN  
MOV (R1)+, R2 ; SAME IN R2  
MOV (R1)+, @#COMPA ; COMPARE OR NOT  
MOV #14, R1 ; NO. OF TOLERANCE GAP WORDS  
MOV @#TOLGAP, R3 ; START OF TOLERANCE GAP TABLE  
1\$: MOV #-1, (R3)+ ; MAKE IT 177777  
DEC R1 ; IS 14 COMPLETED  
BNE 1\$ ; IF NO BRANCH  
MOV @#RHMR, R0 ; R0 CONTAINS MAINTANENCE REG.  
MOV @#FORMAT, -(SP)  
SUB @#COUNTD, (SP)  
MOV (SP), @#ZWORDS ; NO. OF ZERO WORDS TO BE WRITTEN  
MOV (SP)+, R4  
TST @#TSECC ; IS THIS AN ECC TEST ?  
BEQ 7\$ ; BRANCH IF NO  
MOV #-1, @#TSECCG ; THESE BITS ARE TO GENERATE ECC  
7\$: MOV @#DISK, R3 ; ADDRESS THE "DISK" AREA  
2\$: JSR PC, @#WRITE ; WRITE INTO "WORD"  
MOV @#WORD, (R3)+ ; STORE ON SIMULATED DISK  
DEC R2 ; COUNT DOWN  
BNE 2\$ ; CONTINUE IF ALL WORDS NOT WRITTEN  
TST R4 ; ANY ZEROS TO BE WRITTEN ?  
BEQ 4\$ ; BRANCH IF NONE TO BE WRITTEN  
3\$: JSR PC, @#WRITE ; WRITE ZEROS INTO "WORD"  
MOV @#WORD, (R3)+ ; STORE INTO "DISK"  
DEC R4  
BNE 3\$  
4\$: CLR @#TSECCG ; NO MORE ECC TO BE GENERATED  
MOV #2, R1  
5\$: JSR PC, WRITE ; WRITE ECC1 AND ECC2 ON SIMULATED DISK  
MOV @#WORD, (R3)+ ; STORE ON WEEC1 AND WEEC2  
DEC R1  
BNE 5\$  
6\$: JSR PC, WRITE ; WRITE DATA GAP INTO "WORD"  
MOV @#WORD, (R3)+ ; STORE INTO "DISK"  
MOV #14, R1  
7\$: JSR PC, @#WRITE ; WRITE TOLERANCE GAP ZEROS

7651	051274	013723	050644
7652	051300	005301	
7653	051302	001372	
7654			
7655			
7656	051316	000201	
7657			
7658			

MOV	2#WORD, (R3)+	; STORE INTO "DISK"
DEC	R1	
BNE	6\$	
RTS	R1	

7659  
7660  
7661  
7662  
7663  
7664  
7665  
7666  
7667  
7668  
7669  
7670  
7671  
7672  
7673  
7674  
7675  
7676  
7677  
7678  
7679  
7680  
7681  
7682  
7683  
7684  
7685  
7686  
7687  
7688  
7689  
7690

;\*WRITE HEADER AND DATA  
\*  
\*  
\*THIS IS THE SIMULATED DISK  
\*ONLY ONE SECTOR OF SPACE IS ALLOWED

051320 000023  
051366 000001  
051370 000004  
051400 000001  
051402 000005  
051414 000001  
051416  
051416 000400  
052416 000001  
052420 000001  
052422 000001  
052424 000016

SECGAP: .BLKW 19.  
WSSYNC: .BLKW 1  
HEADER: .BLKW 4  
WCRC: .BLKW 1  
HEGAP: .BLKW 5  
HDWSYN: .BLKW 1  
SILOTB:  
DISK: .BLKW 256.  
WECC1: .BLKW 1  
WECC2: .BLKW 1  
DTAGAP: .BLKW 1  
TOLGAP: .BLKW 14.

; SECTOR GAP 38 BYTES OF 0  
; SECTOR GAP 1 BYTE OF 0 ONE SYNC BYTE  
; HEADER = CYL, SECTOR/TRACK, KEY1, KEY2  
; CRC  
; HEADER GAP 10 BYTES OF 0  
; HEADER GAP 1 BYTE OF 0 ONE SYNC. BYTE  
; USED IN SILO TEST AS SILO TABLE  
; DATA SPACE  
; ECC1  
; ECC2  
; DATA GAP 2 BYTES OF 0  
; TOLERANCE GAP 28 BYTES OF 0



7691  
7692  
7693  
7694  
7695  
7696  
7697  
7698  
7699  
7700  
7701  
7702  
7703  
7704  
7705  
7706  
7707  
7708  
7709  
7710  
7711  
7712  
7713  
7714  
7715  
7716  
7717  
7718  
7719  
7720  
7721  
7722  
7723  
7724  
7725  
7726  
7727  
7728  
7729  
7730  
7731  
7732  
7733  
7734  
7735  
7736  
7737  
7738  
7739  
7740  
7741  
7742  
7743  
7744  
7745  
7746

;\*WRITE HEADER AND DATA

;\*\*THIS SUBROUTINE IS THE FIRST IN A SERIES OF NESTED SUBROUTINES

;\*\*IT ISSUES DIAGNOSTIC MODE, AN EXTRA DIAGNOSTIC INDEX, AND THE  
;\*\*'GO' BIT

;\*\*IT THEN CALL THREE OTHER SUBROUTINES, WHICH IN TURN CALL OTHER  
;\*\*SUBROUTINES. THESE ARE:

;\*\* SEARCH  
;\*\* WRHEAD  
;\*\* WRDATA

052460	000000			RNCTR1: .WORD	0	;'RUN' LINE STALL COUNTER
052462	011637	002014		COMWHD: MOV	(SP), @#PCJSR	;SAVE PC OF JSR + 4
052466	162737	000004	002014	SUB	#4, @#PCJSR	;SAVE PC OF JSR
052510	012777	000001	127142	MOV	#DMD, @RHMR	;SET DIAGNOSTIC MODE
052516	052777	000004	127134	BIS	#MINX, @RHMR	;SET DIAGNOSTIC INDEX
052524	042777	000004	127126	BIC	#MINX, @RHMR	;CLEAR IT
052532	052777	000001	127100	BIS	#GO, @RHCS1	;SET 'GO' BIT & STALL 'TILL 'RUN'
052540	012737	000113	052460	RNWAT1: MOV	#75., @#RNCTR1	;LOAD STALL COUNTER = APPROX. 450US
052546	005337	052460		1\$: DEC	@#RNCTR1	;FOR 11/50 CPU WITH CORE MEMORY
052552	001375			BNE	1\$	;COUNT DOWN 1 TIME
052554	013746	052640		MOV	@#WSECTR, -(SP)	;GET DESIRED SECTOR/TRACK
052560	042716	177740		BIC	#177740, (SP)	;MAKE ONLY SECTOR
052564	012637	052574		MOV	(SP)+, @#WTRK	;SAVE SECTOR
052570	004137	053546		JSR	R1, @#SEARCH	;ISSUE SECTOR CLOCKS <----->
052574	000000			WTRK: .WORD	0	;SECTOR NO.
052576	012701	000240		MOV	#+NOP, R1	;GOING TO MOVE NOPS
052602	010137	052650		MOV	R1, @#SEGPER	;NOP INTO SEGAP
052606	010137	052652		MOV	R1, @#FSYNER	;NOP INTO FSYNER
052612	010137	052654		MOV	R1, @#ERHEAD	;NOP INTO ERHEAD
052616	010137	052656		MOV	R1, @#ERCRC	;NOP INTO ERCRC
052622	010137	052660		MOV	R1, @#ERHDGP	;NOP INTO ERHDGAP
052626	010137	052662		MOV	R1, @#HDESYN	;NOP INTO HDESYN



```

7803
7804
7805
7806
7807
7808
7809
7810
7811
7812
7813 052662 000240
7814
7815
7816
7817
7818
7819
7820
7821
7822
7823
7824
7825
7826 052664 005737 002006
7827 052670 001004
7828
7829 052672 004137 051066
7830 052676 000000
7831 052700 000000
7832
7833 052702
7834
7835 052716 000207

```

HDESYN: NOP

FNWORD:

FOUT:

```

TST  @#ERFLGS
BNE  FOUT
JSR  R1,@#WRDATA
      .WORD 0
      .WORD 0
RTS  PC

```

```

;GAP GOING ON DISK IS WRONG.
;WORD NO. GIVES WHICH OF
;THE HEADER GAP WORDS
;ARE WRONG. FOR EXAMPLE:
;WORD NO 1 = FIRST HEADER
;                GAP WORD
;BAD WORD IS WHAT IS GOING ON DISK
;IF "ERROR 6" INSERTED BY
;WRHEAD SUBROUTINE THEN LAST
;HEADER GAP BYTE OR HEADER
;SYNC BYTE GOING ON DISK IS WRONG.
;WORD NO = 5
;BAD DATA IS WHAT IS GOING
;ON DISK, RIGHT BYTE IS HEADER
;GAP 0 BYTE, LEFT BYTE IS HEADER
;GAP SYNC.
;ARE ANY ERRORS DETECTED
;IF YES EXIT -----)
;WRITE THE DATA (<-----)
;FORMAT COMMAND NO. OF DATA

```

7836  
7837  
7838  
7839  
7840  
7841  
7842  
7843  
7844  
7845  
7846  
7847  
7848  
7849  
7850  
7851  
7852  
7853  
7854  
7855  
7856  
7857  
7858  
7859  
7860  
7861  
7862  
7863  
7864  
7865  
7866  
7867  
7868  
7869  
7870  
7871  
7872  
7873  
7874  
7875  
7876  
7877  
7878  
7879  
7880  
7881  
7882  
7883  
7884  
7885  
7886  
7887  
7888  
7889  
7890  
7891

;\*WRITE HEADER

;\*R0 = MAINT.REG.  
;\*R1 = SIMULATED DISK  
;\*R2 = BYTE COUNT  
;\*R3 = WRITE WORD  
;\*R5 = WORD COUNT

052720 000000  
052722 000000  
052724 000000  
052726 000000  
052730 000000

SCYL: 0  
SSECTR: 0  
SKEY1: 0  
SKEY2: 0  
SCRC: 0

052732 012137 052720  
052736 012137 052722  
052742 012137 052724  
052746 012137 052726  
052752 012137 052730  
052760 012701 051320  
052764 013700 001660  
052770 012710 000001  
052774 012705 000002  
053000 052710 000010  
053004 012710 000013  
053010 032710 000040  
053014 001403  
053016 000261  
053020 006003  
053022 000402  
053024 000241  
053026 006003  
053030 012710 000001  
053034 012702 000007  
053040 052710 000002  
053044 032710 000040  
053050 001403  
053052 000261  
053054 006003  
053056 000402  
053060 000241  
053062 006003  
053064 012710 000001  
053070 005302  
053072 001362  
053074 005305  
053076 001342

WRHEAD: MOV (R1)+, @#SCYL  
MOV (R1)+, @#SSECTR  
MOV (R1)+, @#SKEY1  
MOV (R1)+, @#SKEY2  
MOV (R1)+, @#SCRC  
MOV #SECGAP, R1  
MOV @#RHMR, R0  
MOV #DMD, @R0  
MOV #2, R5  
BIS #MSTCK, @R0  
1\$: MOV #MSTCK!MCLK!DMD, @R0  
BIT #MWR, @R0  
BEQ 2\$  
SEC  
ROR R3  
BR 3\$  
2\$: CLC  
ROR R3  
3\$: MOV #DMD, @R0  
MOV #7, R2  
4\$: BIS #MCLK, @R0  
BIT #MWR, @R0  
BEQ 5\$  
SEC  
ROR R3  
BR 6\$  
5\$: CLC  
ROR R3  
6\$: MOV #DMD, @R0  
DEC R2  
BNE 4\$  
DEC R5  
BNE 1\$

;  
;  
;  
;  
;  
; SIMULATED DISK INDICATOR  
; R0 NOW HAS MAINT. REG. ADDR.  
; SET DIAG. MODE IN RHMR  
; WORD COUNTER  
; SET SECTOR FOR FIRST BYTE  
; SET SECTOR, CLOCK, DIAG. MODE, RESET INDEX  
; CHECK WRITE BIT IN MAINT. REG.  
;  
; SET CARRY  
; MOVE ONE FORWARD  
;  
; CLEAR CARRY  
; MOVE ZERO FORWARD  
; CLEAR CLOCK, SECTOR  
; BYTE COUNTER  
; SET CLOCK  
; CHECK WRITE BIT IN MAINT.REG.  
; BRANCH IF ZERO  
; SET CARRY  
; MOVE ONE FORWARD  
;  
; SET DIAG. MODE AGAIN IN RHMR  
;  
;  
; CONTINUE

```

7892 053100 010321      MOV      R3,(R1)+
7893 053102 005703      TST      R3
7894 053104 001414      BEQ      7$
7895 053106 012737 000001 047620      MOV      #1,@#ERWORD
7896 053114 005037 001124      CLR      @#$GDDAT
7897 053120 010337 001126      MOV      R3,@#$BDDAT
7898 053124 012737 104006 052650      MOV      #104006,@#SEGP
7899 053132 000137 053540      JMP      @#17$ ; BRANCH OUT -----)
7900
7901 053136 012702 000022      MOV      #18.,R2 ; COUNT NO. OF SECTOR GAP
7902 053142 012737 000024 047620 10$: MOV      #20.,@#ERWORD ; COUNT TO GIVE ERROR WORD
7903 053150 004737 050646      JSR      PC,@#WRITE ; WRITE SECTOR GAP
7904 053154 013721 050644      MOV      @#WWORD,(R1)+ ; STORE SECTOR GAP WORD
7905 053160 001413      BEQ      11$
7906 053162 160237 047620      SUB      R2,@#ERWORD ; IF NOT GET ERROR WORD NO.
7907 053166 005037 001124      CLR      @#$GDDAT ; GOOD WORD
7908 053172 013737 050644 001126      MOV      @#WWORD,@#$BDDAT ; BAD WORD
7909 053200 012737 104006 052650      MOV      #104006,@#SEGP ; STORE "ERROR 6" IN SEGP
7910 053206 000554      BR       17$ ; BRANCH OUT -----)
7911
7912 053210 005302      11$: DEC      R2 ; HAVE 18 WORDS OF ZEROS BEEN WRITTEN ?
7913 053212 001353      BNE      10$ ; IF NOT DO SO
7914
7915 ;*AT THIS POINT THE SECTOR FOUND FLOP SHOULD
7916 ;*BE HIGH. SO THAT THE HEADER SYNC BYTE CAN BE GIVEN
7917
7918 ;*HOWEVER IN THE DRIVE TIMING ERROR TEST THE REST OF THE ROUTINE
7919 ;*IS ABORTED - HEADER SYNC BYTE IS NOT GIVEN
7920
7921 053214 005737 002026      TST      @#TESDTE ; IS THIS A DRIVE TIMING ERROR
7922 053220 001147      BNE      17$ ; BRANCH OUT IF YES
7923 053222 004737 050646      JSR      PC,@#WRITE ; WRITE ONE SECTOR GAP 0 BYTE
7924 ; AND ONE SYNC. BYTE = 230
7925 053226 013711 050644      MOV      @#WWORD,(R1) ; SAVE 0 BYTE AND SYNC BYTE
7926 053232 023721 047602      CMP      @#RSYNC,(R1)+ ; IF SYNC. BYTE RIGHT
7927 053236 001414      BEQ      12$ ; IF YES BRANCH
7928 053240 012737 000024 047620      MOV      #20.,@#ERWORD ; IF NOT GET READY FOR ERROR
7929 053246 013737 047602 001124      MOV      @#RSYNC,@#$GDDAT ; GOOD WORD
7930 053254 014137 001126      MOV      -(R1),@#$BDDAT ; BAD WORD
7931 053260 012737 104006 052652      MOV      #104006,@#FSYNER ; INSERT "ERROR 6" IN FSYNER
7932 053266 000524      BR       17$ ; BRANCH OUT -----)
7933
7934 053270 012702 000004      12$: MOV      #4,R2 ; FOUR HEADER WORDS
7935 053274 012703 052720      MOV      #SCYL,R3 ; POINTER FOR HEADER TABLE
7936 053300 012737 000005 047620 13$: MOV      #5,@#ERWORD ; ERROR WORD NO SET
7937 053306 004737 050646      JSR      PC,@#WRITE ; WRITE 4 HEADER WORDS
7938 053312 013711 050644      MOV      @#WWORD,(R1) ; STORE WRITTEN WORD
7939 053316 022321      CMP      (R3)+,(R1)+ ; IS IT RIGHT?
7940 053320 001412      BEQ      14$ ; IF GOOD CONTINUE
7941 ; IF NOT GET READY FOR PRINT
7942 053322 160237 047620      SUB      R2,@#ERWORD ; WORD NO
7943 053326 014337 001124      MOV      -(R3),@#$GDDAT ; GOOD DATA
7944 053332 014137 001126      MOV      -(R1),@#$BDDAT ; BAD DATA
7945 053336 012737 104006 052654      MOV      #104006,@#ERHEAD ; INSERT "ERROR 6"
7946 053344 000475      BR       17$ ; BRANCH OUT -----)
7947

```

```

7948 053346 005302          14$: DEC      R2          ;ARE 4 HEADER WORDS DONE?
7949 053350 001353          BNE      13$          ;IF NOT DO THEM
7950 053352 004737 050646    JSR      PC, @WRITE   ;WRITE CRC
7951 053356 013711 050644    MOV      @WORD, (R1)  ;STORE CRC
7952 053362 022137 052646    CMP      (R1)+, @GCRC ;COMPARE GOOD CRC
7953 053366 001414          BEQ      20$          ;BRANCH IF GOOD
7954 053370 014137 001126    MOV      -(R1), @SBDATA ;BAD CRC WRITTEN
7955 053374 013737 052646 001124    MOV      @GCRC, @SGDDAT ;GOOD CRC
7956 053402 012737 000005 047620    MOV      #5, @ERWORD   ;ERROR WORD NO
7957 053410 012737 104006 052656    MOV      #104006, @ERCRC ;INSERT ERROR 6
7958 053416 000450          BR       17$          ;EXIT ----->
7959
7960 053420 012702 000005    20$: MOV      #5, R2      ;NO OF HEADER GAP
7961 053424 012737 000006 047620 15$: MOV      #6, @ERWORD  ;ERROR WORD NO SET
7962 053432 004737 050646    JSR      PC, @WRITE   ;WRITE HEADER GAP
7963 053436 013721 050644    MOV      @WORD, (R1)+ ;STORE
7964 053442 001412          BEQ      16$          ;IF GOOD BRANCH
7965 053444 160237 047620    SUB      R2, @ERWORD  ;ERROR WORD NO
7966 053450 005037 001124    CLR      @SGDDAT     ;GOOD DATA
7967 053454 014137 001126    MOV      -(R1), @SBDAT ;BAD DATA
7968 053460 012737 104006 052660    MOV      #104006, @ERHDP ;STORE "ERROR 6"
7969 053466 000424          BR       17$          ;BRANCH OUT ----->
7970
7971 053470 005302          16$: DEC      R2          ;ARE 5 HEADER GAP ZEROS DONE
7972 053472 001354          BNE      15$          ;IF NOT BRANCH
7973 053474 004737 050646    JSR      PC, @WRITE   ;WRITE HEADER GAP
7974 053500 013711 050644    MOV      @WORD, (R1)  ;STORE
7975 053504 023721 047602    CMP      @RSYNC, (R1)+ ;COMPARE GOOD CRC
7976 053510 001413          BEQ      17$          ;BRANCH IF GOOD
7977 053512 012737 000005 047620    MOV      #5, @ERWORD  ;ERROR WORD NO
7978 053520 014137 001126    MOV      -(R1), @SBDAT ;BAD DATA
7979 053524 013737 047602 001124    MOV      @RSYNC, @SGDDAT ;STORE "ERROR 6"
7980 053532 012737 104006 052662    MOV      #104006, @HDESYN ;STORE "ERROR 6"
7981
7982 053540          17$:
7983
7984 053542 000201          RTS      R1
7985
7986

```

7987  
7988  
7989  
7990  
7991  
7992  
7993  
7994  
7995  
7996  
7997  
7998  
7999  
8000  
8001  
8002  
8003  
8004  
8005  
8006  
8007  
8008  
8009  
8010  
8011  
8012  
8013  
8014  
8015  
8016  
8017  
8018  
8019  
8020  
8021  
8022  
8023  
8024  
8025  
8026  
8027  
8028  
8029  
8030  
8031  
8032  
8033  
8034  
8035  
8036  
8037  
8038  
8039  
8040  
8041  
8042

;\*SEARCH SECTOR

::\*  
::\* RO=RHMR ADDRESS  
::\* R1=PASSED ARGUMENT (SECTOR SEARCHED FOR)  
::\* R2=CLOCK COUNT (PER BYTE)  
::\* R3=SECTOR COUNTER FROM R1  
::\* R5=BYTES PER WORD COUNT  
::\*BEFORE INDEX IS GIVEN TWO SECTOR CLOCKS ARE GIVEN TO RESET  
::\*SECTOR PULSE IN CASE IT IS SET  
::\*AT BEGINNING OF EACH SECTOR ONE SECTOR CLOCK HAS TO RISE  
::\*BEFORE CLOCK THEN EVERY EIGHT CLOCKS ONE SECTOR CLOCK IS  
::\*IDENTICAL WITH CLOCK  
::\*NUMBERING THE SECTOR CLOCKS AS FOLLOWS  
::\*THE SECTOR CLOCK UNDER INDEX - 0  
::\*THE NEXT - 1  
::\*THE NEXT - 2  
::\*ETC.  
::\*THEN THE LAST SECTOR CLOCK IN ONE SECTOR HAS NUMBER - 608  
::\*THE NEXT SECTOR THEN HAS 608 SECTOR CLOCKS  
::\*THE NEXT SECTOR THEN HAS ANOTHER 608 SECTOR CLOCKS  
::\*AND SO ON

SECTR: 0 ;SECTOR SEARCHED FOR  
SEARCH: MOV (R1)+, @#SECTR ;SAVE SECTOR SEARCHED FOR  
MOV @#RHMR, RO ;NOW RO HAS MAINTENANCE REG. ADR.  
MOV @#SECTR, R3 ;SECTOR COUNTER  
MOV @#DMD, @RO ;SET DIAGNOSTIC MODE  
BIS @#MSTCK, @RO ;SET SECTOR CLOCK  
BIC @#MSTCK, @RO ;CLEAR SECTOR CLOCK  
BIS @#MSTCK, @RO ;SET SECTOR CLOCK  
BIC @#MSTCK, @RO ;CLEAR SECTOR CLOCK  
;THE ABOVE TWO SECTOR CLOCKS ARE GIVEN FOR  
;RESETTING SECTOR PULSE  
;IN CASE IT STARTS SET  
BIS @#MINX!MSTCK, @RO ;SET INDEX AND SECTOR CLOCK  
MOV @#DMD, @RO ;RESET INDEX AND SECTOR CLOCK  
TST R3 ;IF SECTOR REQUIRED JUMP OUT  
BEQ 7\$ ;BRANCH OF SECTOR ZERO REQUIRED  
;\*NOW THE 304 WORDS WILL START

8043  
 8044  
 8045  
 8046  
 8047  
 8048  
 8049  
 8050  
 8051 053636 012702 000010  
 8052 053642 012705 000002  
 8053 053646 052710 000010  
 8054 053652 052710 000002  
 8055 053656 000402  
 8056 053660 052710 000012  
 8057 053664 042710 000012  
 8058 053670 052710 000002  
 8059 053674 042710 000002  
 8060 053700 005302  
 8061 053702 001372  
 8062 053704 012702 000007  
 8063 053710 005305  
 8064 053712 001362  
 8065  
 8066  
 8067  
 8068  
 8069  
 8070 053714 012701 000457  
 8071 053720 012705 000002  
 8072 053724 012702 000007  
 8073 053730 052710 000012  
 8074 053734 042710 000012  
 8075 053740 052710 000002  
 8076 053744 042710 000002  
 8077 053750 005302  
 8078 053752 001372  
 8079 053754 005305  
 8080 053756 001362  
 8081 053760 005301  
 8082 053762 001356  
 8083 053764 052710 000010  
 8084 053770 042710 000010  
 8085 053774 005303  
 8086 053776 001317  
 8087  
 8088 054000  
 8089 054014 000201  
 8090  
 8091  
 8092  
 8093  
 8094 054016 000000  
 8095 054020 000000  
 8096  
 8097  
 8098

;\*FOR FIRST BYTE SECTOR CLOCK WILL GO HIGH THEN CLOCK WILL GO HIGH  
 ;\*BOTH WILL COME DOWN TOGETHER THEN SEVEN CLOCKS WILL BE GIVEN  
 ;\*FOR SECOND BYTE AND ALL OTHER BYTES TILL NEXT SECTOR SECTOR CLOCK  
 ;\*WILL BE IDENTICAL WITH ONE CLOCK

;\*ONE WORD ONLY

```

1$:  MOV    #8.,   R2    ;BYTE COUNTER
     MOV    #2.,   R5    ;BYTES PER WORD
     BIS    #MSTCK, @R0  ;SET SECTOR CLOCK
     BIS    #MCLK, @R0  ;SET CLOCK
     BR     3$          ;BRANCH TO CLEAR SECTOR AND CLOCK
2$:  BIS    #MSTCK!MCLK, @R0 ;SET SECTOR AND CLOCK
3$:  BIC    #MSTCK!MCLK, @R0 ;CLEAR SECTOR AND CLOCK
8$:  BIS    #MCLK,  @R0  ;SET CLOCK
     BIC    #MCLK,  @R0  ;CLEAR CLOCK
     DEC    R7          ;BYTE COUNTER
     BNE    8$          ;BRANCH IF BYTE NOT COMPLETE
     MOV    #7.,   R2    ;SETUP FOR SECOND BYTE
     DEC    R5          ;IS WORD COMPLETE?
     BNE    2$          ;BRANCH IF NOT COMPLETE
                          ;TO GIVE SECTOR CLOCK AND CLOCK
  
```

;\*NOW 303 WORDS ARE LEFT AND ALL ARE IDENTICAL

```

4$:  MOV    #303., R1    ;WORDS PER SECTOR COUNTER
5$:  MOV    #2.,   R5    ;BYTES PER WORD COUNTER
     MOV    #7.,   R2    ;BYTE COUNTER (CLOCK COUNTER)
     BIS    #MSTCK!MCLK, @R0 ;SET SECTOR CLOCK AND CLOCK
     BIC    #MSTCK!MCLK, @R0 ;CLEAR SECTOR CLOCK AND CLOCK
6$:  BIS    #MCLK,  @R0  ;SET CLOCK
     BIC    #MCLK,  @R0  ;RESET CLOCK
     DEC    R2          ;IS BYTE COMPLETE?
     BNE    6$          ;BRANCH IF NOT COMPLETE
     DEC    R5          ;IS WORD COMPLETE?
     BNE    5$          ;BRANCH IF NOT
     DEC    R1          ;IS SECTOR COMPLETE
     BNE    4$          ;BRANCH IF NOT
     BIS    #MSTCK, @R0  ;SET SECTOR
     BIC    #MSTCK, @R0  ;CLEAR SECTOR
     DEC    R3          ;IS REQUIRED NO OF SECTORS COMPLETE
     BNE    1$          ;BRANCH IF NOT
7$:  RTS     R1
  
```

;\*READ ONE SECTOR OF DATA

```

RNO:  0    ;NO. OF WORDS READ
RCOM: 0    ;EXTRA STORAGE
  
```



8099	054022	012137	054016		REDATA: MOV	(R1)+, @#RNO	:: SAVE NO. OF WORDS ONLY FOR INFORMATION
8100	054026	012137	054020		MOV	(R1)+, @#RCOM	:: EXTRA WORD ONLY FOR INFORMATION
8101	054034	005737	002024		TST	@#TSECC	:: IS THIS AN ECC TEST
8102	054040	001403			BEG	1\$	:: BRANCH IF NO
8103	054042	012737	177777	045314	MOV	#-1 @#TSECCG	:: THESE BITS ARE TO GENERATE ECC
8104	054050	012702	000402		1\$: MOV	#256., R2	:: 256 WORDS PER SECTOR
8105							:: PLUS 2 ECC WORDS
8106	054054	012703	051416		MOV	#DISK, R3	:: POINTE TO DISK SIMULATION
8107	054060	012337	050410		2\$: MOV	(R3)+, @#WORD	:: READY TO READ CONTENTS
8108	054064	004737	050414		JSR	PC, @#READ	:: READ
8109	054070	005302			DEC	R2	:: IS 256 WORDS DONE?
8110	054072	001372			BNE	2\$	:: IF NOT BRANCH
8111	054074	005737	002024		TST	@#TSECC	:: IS THIS AN ECC TEST
8112	054100	001012			BNE	4\$	:: BRANCH OUT IF YES
8113	054102	005037	045314		CLR	@#TSECCG	:: NO MORE ECC BITS ARE TO BE GENERATED
8114	054106	012702	000017		MOV	#15., R2	:: ONE DATA GAP, 14 TOLERANCE GAP
8115	054112	012337	050410		3\$: MOV	(R3)+, @#WORD	:: READY TO READ CONTENTS OF WORD
8116	054116	004737	050414		JSR	PC, @#READ	:: READ
8117	054122	005302			DEC	R2	:: COUNT
8118	054124	001372			BNE	3\$	:: BRANCH IF 14 NOT DONE
8119	054126				4\$:		
8120	054130	000201			RTS	R1	:: RETURN

B01

CZRJGB0.RP04/5/6 DSKLS CTRLR1  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 206  
DISK SIMULATION

SEQ 0208

8123					
8124					
8125	054132				
8126	054202	104402			
8127	054204	012777	054132	125414	
8128	054212	000000			

RPVECT: TYPOC  
MOV #RPVECT, @RPVEC  
HALT

:TYPE FROM PC  
:RESTORE TRAP RPO4 VECTOR  
:CHANGE TO CONTINUE

CO1

CZRJGB0.RP04/5/6 DSKLS CTRLRI  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 207  
SYSMAC LIBRARY ROUTINES

SEQ 0209

8129  
8130

.SBTTL SYSMAC LIBRARY ROUTINES

DO1

CZRJGBD.RP04/5/6 DSKLS CTRLR1  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 208  
TTY INPUT ROUTINE

SEG 0210

8131

;FROM THE TTY

E01

CZRJGBO, RPO4/5/6 DSKLS CTRLR1  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 209  
POWER DOWN AND UP ROUTINES

SEQ 0211

8132  
8133  
8134  
8135  
8136  
8137  
8138  
8139  
8140  
8141  
8142  
8143  
8144  
8145  
8146  
8147  
8148  
8149  
8150  
8151  
8152  
8153  
8154  
8155  
8156  
8157  
8158  
8159  
8160  
8161  
8162  
8163  
8164  
8165  
8166  
8167  
8168  
8169  
8170  
8171  
8172  
8173  
8174  
8175  
8176  
8177  
8178  
8179  
8180  
8181  
8182  
8183  
8184  
8185  
8186  
8187

057454 051127 047117 020107  
057462 040504 040524 044440  
057470 020116 042522 042101  
057476 047111 020107 051117  
057504 053440 044522 044524  
057512 043516 044040 051101  
057520 053504 051101 020105  
057526 042522 044507 052123  
057534 051105 000  
057537 105 051122 051117  
057544 047440 020116 042040  
057552 052101 020101 047503  
057560 046515 047101 000104  
057566 051105 047522 020122  
057574 047117 053440 044522  
057602 042524 044040 040505  
057610 042504 020122 047101  
057616 020104 040504 040524  
057624 000  
057625 103 047117 051124  
057632 046117 042514 020122  
057640 051117 042040 044522  
057646 042526 051440 040524  
057654 052524 000123  
057660 042522 044507 052123  
057666 051105 043040 044501  
057674 042514 000104  
057700 047516 020116 054105  
057706 051511 042524 052116  
057714 051040 043505 051511  
057722 042524 026122 020040  
057730 051120 043517 040522  
057736 020115 041101 051117  
057744 042524 027104 000  
057751 127 044501 020124  
057756 047514 050117 043040  
057764 044501 042514 000104  
057772 051127 052111 020105  
060000 044103 041505 020113  
060006 040506 046111 047111  
060014 000107  
060016 042522 044507 052123  
060024 051105 043040 044501  
060032 044514 043516 000  
060037 111 052116 051105  
060044 052522 052120 043040  
060052 044501 044514 043516

\*\*\*\*\*  
\*  
\*ERROR AND MESSAGE TABLE CONDIMITS  
\*  
\*\*\*\*\*

EM1: .ASCIZ /WRONG DATA IN READING OR WRITING HARDWARE REGISTER/  
EM2: .ASCIZ /ERROR ON DATA COMMAND/  
EM6: .ASCIZ /ERROR ON WRITE HEADER AND DATA/  
EM11: .ASCIZ /CONTROLLER OR DRIVE STATUS/  
EM14: .ASCIZ /REGISTER FAILED/  
EM15: .ASCIZ /NON EXISTENT REGISTER, PROGRAM ABORTED./  
EM16: .ASCIZ /WAIT LOOP FAILED/  
EM17: .ASCIZ /WRITE CHECK FAILING/  
EM20: .ASCIZ /REGISTER FAILING/  
EM21: .ASCIZ /INTERRUPT FAILING/

# F01

CZRJGB0,RP04/5/6 DSKLS CTRLR1  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 210  
POWER DOWN AND UP ROUTINES

SEQ 0212

8188	060060	000			
8189	060061	105	051122	051117	EM22: .ASCII /ERROR ON DRIVES PRESENT -/<15><12>
8190	060066	047440	020116	051104	
8191	060074	053111	051505	050040	
8192	060102	042522	042523	052116	
8193	060110	026440	005015		
8194	060114	044124	020105	047125	.ASCII /THE UNIT NO'S FOUND BY SETTING RHAS USING RHER1/<15><12>
8195	060122	052111	047040	023517	
8196	060130	020123	047506	047125	
8197	060136	020104	054502	051440	
8198	060144	052105	044524	043516	
8199	060152	051040	040510	020123	
8200	060160	051525	047111	020107	
8201	060166	044122	051105	006461	
8202	060174	012			
8203	060175	050	032124	020051	.ASCII /(T4) DO NOT AGREE WITH THE UNIT NO'S FOUND/<15><12>
8204	060202	047504	047040	052117	
8205	060210	040440	051107	042505	
8206	060216	053440	052111	020110	
8207	060224	044124	020105	047125	
8208	060232	052111	047040	023517	
8209	060240	020123	047506	047125	
8210	060246	006504	012		
8211	060251	102	020131	047514	.ASCII /BY LOOKING FOR 'NED' = 0 IN RHCS2 (BIT #12)/<15><12><15><12>
8212	060256	045517	047111	020107	
8213	060264	047506	020122	047047	
8214	060272	042105	020047	020075	
8215	060300	020060	047111	051040	
8216	060306	041510	031123	024040	
8217	060314	044502	020124	030443	
8218	060322	024462	005015	005015	
8219	060330	047516	042524	020072	.ASCII /NOTE: ON DUAL PORT SYSTEM, A DRIVE ON OTHER PORT WILL /<15><12>
8220	060336	047117	042040	040525	
8221	060344	020114	047520	052122	
8222	060352	051440	051531	042524	
8223	060360	026115	040440	042040	
8224	060366	044522	042526	047440	
8225	060374	020116	052117	042510	
8226	060402	020122	047520	052122	
8227	060410	053440	046111	020114	
8228	060416	005015			
8229	060420	047516	020124	044507	.ASCII /NOT GIVE 'NED', BUT WILL GIVE RHAS RESPONSES/<15><12>
8230	060426	042526	023440	042516	
8231	060434	023504	020054	052502	
8232	060442	020124	044527	046114	
8233	060450	043440	053111	020105	
8234	060456	044122	051501	051040	
8235	060464	051505	047520	051516	
8236	060472	051505	005015		
8237	060476	042510	041516	020105	.ASCIZ /HENCE THERE WILL BE AN EXTRA DRIVE/
8238	060504	044124	051105	020105	
8239	060512	044527	046114	041040	
8240	060520	020105	047101	042440	
8241	060526	052130	040522	042040	
8242	060534	044522	042526	000	
8243	060541	114	047517	020113	EM24: .ASCIZ /LOOK AHEAD REGISTER AT THE BEGINNING OF SECTOR IS IN ERROR/

8244	060546	044101	040505	020104	
8245	060554	042522	044507	052123	
8246	060562	051105	040440	020124	
8247	060570	044124	020105	042502	
8248	060576	044507	047116	047111	
8249	060604	020107	043117	051440	
8250	060612	041505	047524	020122	
8251	060620	051511	044440	020116	
8252	060626	051105	047522	000122	
8253	060634	047514	045517	040440	EM25: .ASCIZ /LOOK AHEAD REGISTER IS IN ERROR/
8254	060642	042510	042101	051040	
8255	060650	043505	051511	042524	
8256	060656	020122	051511	044440	
8257	060664	020116	051105	047522	
8258	060672	000122			
8259	060674	052503	051122	047105	EM30: .ASCII /CURRENT CYLINDER DOES NOT MATCH DESIRED CYLINDER REGISTER/<15><12>
8260	060702	020124	054503	044514	
8261	060710	042116	051105	042040	
8262	060716	042517	020123	047516	
8263	060724	020124	040515	041524	
8264	060732	020110	042504	044523	
8265	060740	042522	020104	054503	
8266	060746	044514	042116	051105	
8267	060754	051040	043505	051511	
8268	060762	042524	006522	012	
8269	060767	101	052106	051105	.ASCIZ /AFTER A SEEK AND INIT/
8270	060774	040440	051440	042505	
8271	061002	020113	047101	020104	
8272	061010	047111	052111	000	
8273	061015	105	041503	043440	EM31: .ASCII /ECC GENERATED IS INCORRECT/<15><12>
8274	061022	047105	051105	052101	
8275	061030	042105	044440	020123	
8276	061036	047111	047503	051122	
8277	061044	041505	006524	012	
8278	061051	105	042526	054522	.ASCIZ /EVERY WORD ON THIS SECTOR IS THAT GIVEN IN "DATA USED"/
8279	061056	053440	051117	020104	
8280	061064	047117	052040	044510	
8281	061072	020123	042523	052103	
8282	061100	051117	044440	020123	
8283	061106	044124	052101	043440	
8284	061114	053111	047105	044440	
8285	061122	020116	042042	052101	
8286	061130	020101	051525	042105	
8287	061136	000042			
8288	061140	047117	051040	040505	EM32: .ASCII /ON READ COMMAND, AFTER DATA AND ECC HAVE BEEN READ,/<15><12>
8289	061146	020104	047503	046515	
8290	061154	047101	026104	040440	
8291	061162	052106	051105	042040	
8292	061170	052101	020101	047101	
8293	061176	020104	041505	020103	
8294	061204	040510	042526	041040	
8295	061212	042505	020116	042522	
8296	061220	042101	006454	012	
8297	061225	105	041503	051040	.ASCII /ECC REGISTERS OR RHER1 ARE IN ERROR/<15><12>
8298	061232	043505	051511	042524	
8299	061240	051522	047440	020122	

8300	061246	044122	051105	020061
8301	061254	051101	020105	047111
8302	061262	042440	051122	051117
8303	061270	005015		
8304	061272	047117	054514	046040
8305	061300	053517	051105	030440
8306	061306	020061	044502	051524
8307	061314	047440	020106	040520
8308	061322	052124	051105	020116
8309	061330	042522	027107	041440
8310	061336	047101	041040	020105
8311	061344	042522	042101	005015
8312	061352	044124	051511	051440
8313	061360	047510	046125	020104
8314	061366	040515	041524	020110
8315	061374	047514	042527	020122
8316	061402	030461	041040	052111
8317	061410	020123	043117	043440
8318	061416	047517	020104	041505
8319	061424	030503	000	
8320	061427	110	043511	020110
8321	061434	047503	047125	020124
8322	061442	044502	020124	047516
8323	061450	020124	042523	020124
8324	061456	043101	042524	020122
8325	061464	034063	032470	020071
8326	061472	046103	041517	051513
8327	061500	000		
8328	061501	132	051105	020117
8329	061506	042504	042524	052103
8330	061514	041040	052111	047040
8331	061522	052117	044040	043511
8332	061530	020110	044127	047105
8333	061536	031440	020062	044502
8334	061544	020124	041505	020103
8335	061552	042522	027107	044040
8336	061560	051501	031040	020061
8337	061566	042532	047522	000123
8338	061574	047520	044523	044524
8339	061602	047117	051040	043505
8340	061610	051511	042524	020122
8341	061616	051117	030440	020061
8342	061624	044502	051524	047440
8343	061632	020106	040520	052124
8344	061640	051105	020116	042522
8345	061646	044507	052123	051105
8346	061654	044440	041516	051117
8347	061662	042522	052103	005015
8348	061670	047514	042527	020122
8349	061676	030461	041040	052111
8350	061704	020123	043117	050040
8351	061712	052101	042524	047122
8352	061720	051040	043505	051511
8353	061726	042524	020122	044123
8354	061734	052517	042114	046440
8355	061742	052101	044103	046040

.ASCII /ONLY LOWER 11 BITS OF PATTERN REG. CAN BE READ/<<15><12>

.ASCIZ /THIS SHOULD MATCH LOWER 11 BITS OF GOOD ECC1/

EM33: .ASCIZ /HIGH COUNT BIT NOT SET AFTER 38859 CLOCKS/

EM34: .ASCIZ /ZERO DETECT BIT NOT HIGH WHEN 32 BIT ECC REG. HAS 21 ZEROS/

EM35: .ASCII /POSITION REGISTER OR 11 BITS OF PATTERN REGISTER INCORRECT/<<15><12>

.ASCII /LOWER 11 BITS OF PATTERN REGISTER SHOULD MATCH LOWER/<<15><12>



8356	061750	053517	051105	005015
8357	061756	030461	041040	052111
8358	061764	020123	043117	043440
8359	061772	047517	020104	041505
8360	062000	030503	005015	
8361	062004	040504	020124	047105
8362	062012	046126	050117	043440
8363	062020	047517	020104	047520
8364	062026	044523	044524	047117
8365	062034	040440	042116	047040
8366	062042	041455	042117	020105
8367	062050	042532	047522	020123
8368	062056	051101	020105	047111
8369	062064	047440	052103	046101
8370	062072	000		
8371	062073	117	020116	042522
8372	062100	042101	041440	046517
8373	062106	040515	042116	053440
8374	062114	052111	020110	047516
8375	062122	026516	047503	051122
8376	062130	041505	040524	046102
8377	062136	020105	051105	047522
8378	062144	020122	042047	045503
8379	062152	020047	047101	020104
8380	062160	042447	044103	020047
8381	062166	044123	052517	042114
8382	062174	041040	020105	042523
8383	062202	000124		
8384	062204	051120	043517	040522
8385	062212	020115	051105	047522
8386	062220	020122	044502	020124
8387	062226	030443	020060	047111
8388	062234	051040	041510	031123
8389	062242	042040	042111	047040
8390	062250	052117	051440	052105
8391	062256	005015		
8392	062260	043111	050040	051517
8393	062266	052111	047511	020116
8394	062274	042522	044507	052123
8395	062302	051105	036440	030061
8396	062310	032060	020060	051117
8397	062316	030440	030060	030464
8398	062324	020054	052111	044440
8399	062332	020123	047507	042117
8400	062340	000		
8401				
8402	062341	122	053510	020103
8403	062346	044504	020104	047516
8404	062354	020124	020075	020060
8405	062362	050125	047117	041440
8406	062370	046517	046120	052105
8407	062376	047511	020116	043117
8408	062404	051040	040505	006504
8409	062412	012		
8410	062413	117	020122	051127
8411	062420	052111	020105	042510

.ASCII /11 BITS OF GOOD ECC1/<15><12>

.ASCIZ /DAT ENVLOP GOOD POSITION AND N-CODE ZEROS ARE IN OCTAL/

EM36: .ASCIZ /ON READ COMMAND WITH NON-CORRECTABLE ERROR 'DCK' AND 'ECH' SHOULD BE SE

EM37: .ASCII /PROGRAM ERROR BIT #10 IN RHCS2 DID NOT SET/<15><12>

.ASCIZ /IF POSITION REGISTER =10040 OR 10041, IT IS GOOD/

EM40: .ASCII /RHWC DID NOT = 0 UPON COMPLETION OF READ/<15><12>

.ASCIZ /OR WRITE HEADER AND DATA/

J01

CZRJGB0, RPO4/5/6 DSKLS CTRLRI  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 214  
POWER DOWN AND UP ROUTINES

SEQ 0216

8412	062426	042101	051105	040440
8413	062434	042116	042040	052101
8414	062442	000101		

K01

CZRJGB0.RP04/5/6 DSKLS CTRLRI  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 215  
POWER DOWN AND UP ROUTINES

SEQ 0217

8415										
8416	062444	040506	040524	020114	CPHALT:	.ASCII	/FATAL ERROR - SEE DOCUMENT LISTING/	<15>	<12>	
8417	062452	051105	047522	020122						
8418	062460	020055	042523	020105						
8419	062466	047504	052503	042515						
8420	062474	052116	046040	051511						
8421	062502	044524	043516	005015						
8422	062510	006440	103412	177777	.ASCII	/	<15>	<12>	<207>	<377>
8423	062516	177607	103777	177777						
8424	062524	044124	020105	047503	.ASCII	/	THE CONTROLLER OR DEVICE HAS GONE OFFLINE, LOST/	<15>	<12>	
8425	062532	052116	047522	046114						
8426	062540	051105	047440	020122						
8427	062546	042504	044526	042503						
8428	062554	044040	051501	043440						
8429	062562	047117	020105	043117						
8430	062570	046106	047111	026105						
8431	062576	046040	051517	006524						
8432	062604	012								
8433	062605	047	042522	042101	.ASCII	/	'READY', BECOME UNAVAILABLE, OR HAS STATUS BITS/	<15>	<12>	
8434	062612	023531	020054	042502						
8435	062620	047503	042515	052440						
8436	062626	040516	040526	046111						
8437	062634	041101	042514	020054						
8438	062642	051117	044040	051501						
8439	062650	051440	040524	052524						
8440	062656	020123	044502	051524						
8441	062664	005015								
8442	062666	044127	041511	020110	.ASCIZ	/	WHICH CANNOT BE CLEARED/			
8443	062674	040503	047116	052117						
8444	062702	041040	020105	046103						
8445	062710	040505	042522	000104						
8446										
8447	062716	020040	020040	020040	SPACE8:	.ASCII	/			
8448	062724	020040	000		SPACE2:	.ASCIZ	/			
8449										
8450										
8451	062727	120	020103	020040	DH1:	.ASCII	/PC	TEST	REG.	GOOD
8452	062734	020040	052040	051505						RECEIVED/
8453	062742	020124	020040	051040						<15>
8454	062750	043505	020056	020040						<12>
8455	062756	043440	047517	020104						
8456	062764	020040	051040	041505						
8457	062772	044505	042526	006504						
8458	063000	012								
8459	063001	040	020040	020040	.ASCIZ	/	NO	ADDR.	DATA	DATA
8460	063006	020040	047040	020117						/
8461	063014	020040	020040	040440						
8462	063022	042104	027122	020040						
8463	063030	042040	052101	020101						
8464	063036	020040	042040	052101						
8465	063044	020101	020040	000040						
8466	063052	041520	020040	020040	DH2:	.ASCII	/PC	TEST	WORD	GOOD
8467	063060	020040	042524	052123						BAD
8468	063066	020040	020040	047527						/
8469	063074	042122	020040	020040						<15>
8470	063102	047507	042117	020040						<12>

















8863											
8864	067350	001116	002032	001122	DT11:	.WORD	\$ERRPC,TSTNM,\$BDADR,CS1,CS2,DS1,ER1,0				
8865	067356	001714	001712	001736							
8866	067364	001716	000000								
8867	067370	001116	002032	001122	DT14:	.WORD	\$ERRPC,TSTNM,\$BDADR,\$BDDAT,CS1,CS2,DS1,ER1,0				
8868	067376	001126	001714	001712							
8869	067404	001736	001716	000000							
8870	067412	001116	002032	001200	DT15:	.WORD	\$ERRPC,TSTNM,\$TMP1,0				
8871	067420	000000									
8872	067422	001116	002032	001204	DT16:	.WORD	\$ERRPC,TSTNM,\$TMP3,\$TMP1,\$TMP0,\$BDDAT,0				
8873	067430	001200	001176	001126							
8874	067436	000000									
8875	067440	001116	002032	001710	DT17:	.WORD	\$ERRPC,TSTNM,BA,DB,WC,CS1,CS2,0				
8876	067446	001704	001706	001714							
8877	067454	001712	000000								
8878											
8879	067460	001116	002032	001716	DT20:	.WORD	\$ERRPC,TSTNM,ER1,ER2,ER3,AS,DS1,0				
8880	067466	001722	001730	001732							
8881	067474	001736	000000								
8882	067500	001116	002032	001714	DT21:	.WORD	\$ERRPC,TSTNM,CS1,AS,DS1,0				
8883	067506	001732	001736	000000							
8884	067514	001116	002032	000000	DT22:	.WORD	\$ERRPC,TSTNM,0				
8885	067522	001116	002032	001720	DT24:	.WORD	\$ERRPC,TSTNM,DST,\$BDDAT,\$TMP1,\$TMP2,\$TMP3,0				
8886	067530	001126	001200	001202							
8887	067536	001204	000000								
8888	067542	001116	002032	002014	DT26:	.WORD	\$ERRPC,TSTNM,PCJSR,\$BDADR,CS1,CS2,DS1,ER1,0				
8889	067550	001122	001714	001712							
8890	067556	001736	001716	000000							
8891	067564	001116	002032	002014	DT27:	.WORD	\$ERRPC,TSTNM,PCJSR,REGADR,\$GDDAT,\$BDDAT,0				
8892	067572	042264	001124	001126							
8893	067600	000000									
8894											
8895	067602	001116	002032	002014	DT30:	.WORD	\$ERRPC,TSTNM,PCJSR,REGADR,\$GDDAT,\$BDDAT,0				
8896	067610	042264	001124	001126							
8897	067616	000000									
8898	067620	001116	002032	047620	DT31:	.WORD	\$ERRPC,TSTNM,ERWORD,\$GDDAT,\$BDDAT,CS1,DS1,ER1,0				
8899	067626	001124	001126	001714							
8900	067634	001736	001716	000000							
8901	067642	001116	002032	002014	DT32:	.WORD	\$ERRPC,TSTNM,PCJSR,ERWORD,\$GDDAT,\$BDDAT,CS1,DS1,ER1,0				
8902	067650	047620	001124	001126							
8903	067656	001714	001736	001716							
8904	067664	000000									
8905	067666	001116	002032	002014	DT33:	.WORD	\$ERRPC,TSTNM,PCJSR,ERWORD,\$GDDAT,CS1,DS1,ER1,0				
8906	067674	047620	001124	001714							
8907	067702	001736	001716	000000							
8908	067710	001116	002032	045310	DT34:	.WORD	\$ERRPC,TSTNM,GECC1,GECC2,WECC1,WECC2,DISK,0				
8909	067716	045312	052416	052420							
8910	067724	051416	000000								
8911	067730	001116	002032	045310	DT35:	.WORD	\$ERRPC,TSTNM,GECC1,GECC2,EC2,EC1,POSITI,ER1,0				
8912	067736	045312	001746	001744							
8913	067744	045322	001716	000000							
8914	067752	001116	002032	002014	DT36:	.WORD	\$ERRPC,TSTNM,PCJSR,MR,EC1,EC2,0				
8915	067760	001734	001744	001746							
8916	067766	000000									
8917	067770	001116	002032	001744	DT37:	.WORD	\$ERRPC,TSTNM,EC1,POSITI,GECC1,GECC2,EC2,DATENV,ZCODE,0				
8918	067776	045322	045310	045312							

8919	070004	001746	045326	045330					
8920	070012	000000							
8921									
8922	070014	001116	002032	001126	DT40:	.WORD	\$ERRPC, TSTNM, \$BDDAT, 0		
8923	070022	000000							
8924									
8925	070024	000	000	000	DF1:	.BYTE	0,0,0,0,0		
8926	070027	000	000						
8927	070031	000	000	001	DF2:	.BYTE	0,0,1,0		
8928	070034	000							
8929	070035	000	000	001	DF3:	.BYTE	0,0,1,0,0		
8930	070040	000	000						
8931									
8932	070042	000	000	000	DF11:	.BYTE	0,0,0,0,0,0,0		
8933	070045	000	000	000					
8934	070050	000							
8935	070051	000	000	000	DF14:	.BYTE	0,0,0,0,0,0,0,0		
8936	070054	000	000	000					
8937	070057	000	000						
8938	070061	000	000	000	DF15:	.BYTE	0,0,0		
8939	070064	000	000	000	DF16:	.BYTE	0,0,0,0,0		
8940	070067	000	000						
8941	070071	000	000	000	DF17:	.BYTE	0,0,0,0,0,0,0		
8942	070074	000	000	000					
8943	070077	000							
8944									
8945	070100	000	000	000	DF20:	.BYTE	0,0,0,0,0,0,0		
8946	070103	000	000	000					
8947	070106	000							
8948	070107	000	000	000	DF21:	.BYTE	0,0,0,0,0		
8949	070112	000	000						
8950	070114	000	000	000	DF22:	.BYTE	0,0,0,0		
8951	070117	000							
8952	070120	000	000	000	DF24:	.BYTE	0,0,0,0,0,0,0		
8953	070123	000	000	000					
8954	070126	000							
8955	070127	000	000	000	DF26:	.BYTE	0,0,0,0,0,0,0,0		
8956	070132	000	000	000					
8957	070135	000	000						
8958	070137	000	000	000	DF27:	.BYTE	0,0,0,0,0,0		
8959	070142	000	000	000					
8960									
8961	070145	000	000	000	DF30:	.BYTE	0,0,0,0,0,0		
8962	070150	000	000	000					
8963	070153	000	000	001	DF31:	.BYTE	0,0,1,0,0,0,0,0		
8964	070156	000	000	000					
8965	070161	000	000						
8966	070163	000	000	000	DF32:	.BYTE	0,0,0,1,0,0,0,0,0		
8967	070166	001	000	000					
8968	070171	000	000	000					
8969	070174	000	000	000	DF33:	.BYTE	0,0,0,1,0,0,0,0		
8970	070177	001	000	000					
8971	070202	000	000						
8972	070204	000	000	000	DF34:	.BYTE	0,0,0,0,0,0,0		
8973	070207	000	000	000					
8974	070212	000							

CZRJGB0, RPO4/5/6 DSKLS CTRLRI  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 225  
POWER DOWN AND UP ROUTINES

SEQ 0227

8975	070213	000	000	000	DF35:	.BYTE	0,0,0,0,0,0,0,0
8976	070216	000	000	000			
8977	070221	000	000				
8978	070223	000	000	000	DF36:	.BYTE	0,0,0,0,0,0
8979	070226	000	000	000			
8980	070231	000	000	000	DF37:	.BYTE	0,0,0,0,0,0,0,0,0
8981	070234	000	000	000			
8982	070237	000	000	000			
8983							
8984	070242	000	000	000	DF40:	.BYTE	0,0,0
8985		070246				.EVEN	
8986							
8987	000001				.END		





DF15	070061	682	8938#															
DF16	070064	693	8939#															
DF17	070071	706	8941#															
DF2	070031	8927#																
DF20	070100	719	8945#															
DF21	070107	730	8948#															
DF22	070114	745	8950#															
DF24	070120	766	779	8952#														
DF26	070127	792	8955#															
DF27	070137	807	8958#															
DF3	070035	622	629	8929#														
DF30	070145	820	8961#															
DF31	070153	585	593	8963#														
DF32	070163	569	610	8966#														
DF33	070174	552	8969#															
DF34	070204	838	8972#															
DF35	070213	857	929	8975#														
DF36	070223	872	889	8978#														
DF37	070231	912	8980#															
DF40	070242	950	8984#															
DF5	= 000001	1009#																
DH1	062727	529	648	8451#														
DH11	063175	634	933	8480#														
DH14	063355	665	8500#															
DH15	063555	679	8523#															
DH16	063606	687	8528#															
DH17	063751	698	8546#															
DH2	063052	616	8466#															
DH20	064134	711	8566#															
DH21	064317	724	8586#															
DH22	064437	740	8601#															
DH24	064542	759	772	8613#														
DH26	064725	783	8633#															
DH27	065130	799	8656#															
DH30	065273	813	8674#															
DH31	065436	575	8692#															
DH32	065641	558	599	8714#														
DH33	066064	543	8740#															
DH34	066267	828	8762#															
DH35	066452	848	919	8782#														
DH36	066655	863	880	8804#														
DH37	067014	900	8821#															
DH40	067232	946	8846#															
DIGB	= 000004	1011#																
DISK	051416	1839	1847	1900	1914	4154	4248	4351	4451	4458	4464	4470	4477	4478				
		4606	4607	4611	4618	4619	4722	4728	4735	4736	4928	4929	5025	5026				
		5130	5136	5143	5144	5149*	5255	5281	5320	5479	5507	5530	5667	6554				
		6661	7078	7092*	7093*	7630	7684#	8106	8908									
DISPLA	001142	524#	1387*	8131*	8132*													
DISPRE	000174	504#	1387															
DLT	= 100000	983#	2245	2396	2465	2467												
DL64	= 000020	1013#																
DMD	= 000001	1046#	1746	1749	1770	1986	2101	2171	2644	2661	2742	2798	2855	2936				
		2957	3075	3163	3216	3276	3326	3393	3446	3463	3464	3470	3529	3578				
		3601	3602	3648	3698	3722	3723	3769	3817	3831	3835	3874	3904	3913				
		4035	4072	4413	5387	5391	5435	5440	5445	5451	5473	5503	5625	5691				

















F03

CZRJGBO, RPD4/5/6 DSKLS CTRLR1  
 CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 237  
 CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0238

RH70CK	005364	1405#																		
RKEY1	047610	7333#	7368*	7404																
RKEY2	047612	7334#	7369*	7406																
RMR	= 000004	1029#	5392																	
RNCTR1	052460	7718#	7727*	7729*																
RNO	054016	8094#	8099*																	
RNWAT1	052540	7727#																		
RPVEC	001626	1212#	1391*	7121	7124*	7127	8127*													
RPVECT	054132	1391#	8125#	8127																
RPD6	002036	1327#	1615*	1621*	3086	5465	5537	5573	5618	5897										
RSETR	047606	7332#	7367*	7402																
RSYNC	047602	7330#	7392	7397	7429	7440	7453	7463	7926	7929	7975	7979								
RTN	006020	1447#	1449#																	
RUNCTR	047336	7237#	7246*	7248*																
RUNWAT	047416	7246#																		
SAVDI	002010	1298#	1659*	2063	3036															
SAVER	043306	2105#	2120	2175	2208	2745	2763	2820	2888	3108	3127	3179	3224	3242						
		3292#	3334	3360	3406	3478	3495	3543	3590	3615	3660	3711	3736	3781						
		3824#	3841	3887	4043	5377	5401	5585	5631	5727	5747	5818	5837	5912						
		5931#	6321#																	
SAVSN	002012	1301#	1658*	2067	3040															
SC	= 100000	1005#	1695	1698	2255	4032	5065	5067	5610	5739										
SCOP1	= 104413	8132#																		
SCRC	052730	7855#	7862*																	
SCYL	052720	7851#	7858*	7935																
SC1	= 000100	1093#																		
SC10	= 001000	1096#																		
SC2	= 000200	1094#																		
SC20	= 002000	1097#																		
SC4	= 000400	1095#																		
SEARCH	053546	6743#	7254	7736	8024#															
SECGAP	051320	4080#	4170	4270	4374	7677#	7863													
SECOTR	047502	4451*	4458*	4464*	4470*	4489*	4490*	4940*	4941*	5033*	5034*	5255*	5286*	5287*						
		5479*	5507*	5546*	5547*	6340*	6341*	6424*	6425*	6568*	6569*	7251	7267#							
SECTR	053544	3903#	3925	3995	8022#	8024*	8026													
SEECOM	002072	1350#	3079	3220	3458	6767														
SEGPER	052650	7740#	7757#	7898*	7909*															
SELECT	002002	1293#	1380*	1382*	1409	1579	1595	5969												
SELTST	007524	1509#	1579#																	
SERCH	002054	1343#	3884																	
SETDSK	044276	5438#	5443	5448	5454	5457	5461	6551#												
SILOTB	051416	2329#	2349	2354	7683#															
SKEY1	052724	7853#	7860*																	
SKEY2	052726	7854#	7861*																	
SKI	= 040000	1186#																		
SN	001742	1279#																		
SND1	005326	1397#	1402#																	
SPACE2	062724	1926#	8448#																	
SPACE8	062716	1924#	1925	1929	1930	1931	8447#													
SRO	= 177572	522#																		
SR1	= 177574	522#																		
SR2	= 177576	522#																		
SR3	= 172516	522#																		
SSECTR	052722	7852#	7859*																	
SSYN	047512	7258#	7273#	7398*																
STACK	= 001000	503#	1387	1418	1456	1672	1678	1682	1727	1732	1737	1792	1799	1805						





CZRJGBO, RPO4/5/6 DSKLS CTRLRI  
CZRJGB.P11 08-NOV-77 08:30

MACY11 30(1046) 08-NOV-77 08:54 PAGE 239  
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0240

TRP 005762  
TRTVEC= 000014  
TSECC 002024  
TSECCG 045314  
TSTNM 002032

1444  
503#  
1312#  
6841#  
1323#  
1737#  
2386#  
3444#  
4463#  
5435#  
5507#  
8872  
8911  
1410  
1670

1448#  
7627  
6959  
1419#  
1742#  
2410#  
3576#  
4469#  
5438#  
5528#  
8875  
8914  
1418#  
1678#

8101  
7494  
1457#  
1792#  
2478#  
3696#  
4470#  
5440#  
5691#  
8879  
8917  
5996

8111  
7506  
1472#  
1799#  
2507#  
3815#  
4476#  
5443#  
5694#  
8882  
8922

8131\*  
7511  
1485#  
1805#  
2536#  
3874#  
4579#  
5445#  
5781#  
8884

7523  
1585\*  
1811#  
2572#  
3877#  
4699#  
5448#  
5872#  
8885

7566  
1663\*  
1813#  
2597#  
4072#  
4814#  
5451#  
6022  
8888

7581  
1670#  
1827#  
2643#  
4079#  
4927#  
5454#  
8857  
8891

7629\*  
1672#  
2098#  
2729#  
4169#  
5023#  
5457#  
8859  
8895

7641\*  
1678#  
2156#  
2795#  
4269#  
5106#  
5461#  
8861  
8898

8103\*  
1683#  
2241#  
2921#  
4373#  
5255#  
5473#  
8864  
8901

8113\*  
1727#  
2282#  
3073#  
4450#  
5277#  
5479#  
8867  
8905

8131\*  
1732#  
2328#  
3324#  
4457#  
5358#  
5503#  
8870  
8908

TST1 005500  
TST10 010746  
TST100 034572  
TST101 034626  
TST102 034750  
TST103 035004  
TST104 035126  
TST105 035162  
TST106 035262  
TST107 035316  
TST11 011012  
TST110 035416  
TST111 035544  
TST112 035664  
TST113 035720  
TST114 036334  
TST115 036370  
TST116 037000  
TST117 037626  
TST12 011162  
TST120 037662  
TST121 040214  
TST122 040540  
TST123 041070  
TST13 011226  
TST14 011272  
TST15 011336  
TST16 011524  
TST17 011570  
TST2 006034  
TST20 011634  
TST21 011700  
TST22 011744  
TST23 012010  
TST24 012416  
TST25 013442  
TST26 013604  
TST27 014154  
TST3 006120  
TST30 014332  
TST31 014560  
TST32 015020  
TST33 015144

5435#  
5437#  
5440#  
5442#  
5445#  
5447#  
5451#  
5453#  
1681#  
5456#  
5460#  
5473#  
5478#  
5503#  
5506#  
5527#  
5691#  
1727#  
5693#  
5780#  
5871#  
5962#  
1732#  
1737#  
1741#  
1791#  
1798#  
1456#  
1805#  
1811#  
1813#  
1826#  
1969#  
2087#  
2155#  
2240#  
1457#  
2241#  
2282#  
2328#  
2386#

2097#  
1471#  
2277#  
2319#  
2386#  
2403#

2281#  
2328#  
2409#

TST34	015444	2410	2466	2477#										
TST35	015602	2478	2506#											
TST36	015722	2497	2503	2527	2535#									
TST37	016100	2536	2564	2571#										
TST4	006172	1477	1483#											
TST40	016224	2589	2595#											
TST41	016350	2613	2641#											
TST42	016644	2728#												
TST43	017070	2795#												
TST44	017460	2920#												
TST45	020612	3059	3072#											
TST46	021530	3323#												
TST47	022204	3443#												
TST5	007540	1580	1584#	5978	5988									
TST50	022612	3575#												
TST51	023156	3439	3695#											
TST52	023522	3814#												
TST53	023700	3874#												
TST54	023734	3876#												
TST55	024622	4072#												
TST56	024656	4078#												
TST57	025166	4139	4158	4168#										
TST6	010312	1663#												
TST60	025524	4228	4252	4268#										
TST61	026064	4332	4355	4372#										
TST62	026330	4449#												
TST63	026714	4451	4456#											
TST64	027300	4458	4462#											
TST65	027664	4464	4468#											
TST66	030224	4470	4474#											
TST67	030570	4524	4564	4572#										
TST7	010656	1665	1670#											
TST70	031120	4680	4691#											
TST71	031426	4793	4812#											
TST72	032016	4916	4925#											
TST73	032362	5014	5022#											
TST74	032726	5023	5098#											
TST75	033334	5236	5251#											
TST76	033736	5255	5261	5272#										
TST77	034266	5348	5356#											
TTNC =	000122	1419#	1457#	1472#	1485#	1585#	1663#	1670#	1672#	1678#	1683#	1727#	1732#	1737#
		1742#	1792#	1799#	1805#	1811#	1813#	1827#	2098#	2156#	2241#	2282#	2328#	2386#
		2410#	2478#	2507#	2536#	2572#	2597#	2643#	2729#	2795#	2921#	3073#	3324#	3444#
		3576#	3696#	3815#	3874#	3877#	4072#	4079#	4169#	4269#	4373#	4450#	4457#	4463#
		4469#	4470#	4476#	4579#	4699#	4814#	4927#	5023#	5106#	5255#	5277#	5358#	5435#
		5438#	5440#	5443#	5445#	5448#	5451#	5454#	5457#	5461#	5473#	5479#	5503#	5507#
		5528#	5691#	5694#	5781#	5872#								
		1112#	1130#											
TUF =	000100	1559	1567	1628	3354	5964	5966	5976	5996	8132#				
TY =	047616	1399	1400	1401	1410	1436	1447	1449	1489	1513	1557	1560	1562	1565
TYPDS =	104405	1627	1629	1651	1653	1655	1665	1667	1924	1925	1926	1929	1930	1931
TYPE =	104401	1936	3068	3352	3893	4114	4139	4206	4228	4308	4332	4407	4451	4458
		4464	4470	4512	4815	4963	5057	5255	5308	5359	5479	5507	5580	5719
		5805	5891	5963	5965	5974	5977	5996	6022	6024	6026	6027	6030	6363
		6449	6596	7106	7108	7121	7123	7125	7127	7129	7131	7133	8126	8131







SRDLIN 056112  
SRDOCT 056272  
SRDSZ = 000011  
SREGAD 001160  
SREG0 001162  
SREG1 001164  
SREG2 001166  
SREG3 001170  
SREG4 001172  
SREG5 001174  
SRTNAD 041414  
SR2A = \*\*\*\*\*  
SSAVRE = \*\*\*\*\*  
SSAVR6 057442  
SSCOPE 054214  
SSETUP= 000117  
SSS1 = 000000  
SSTUP = 177777  
SSVLAD 054442  
SSVPC = 000200  
SSWR = 167770

U  
U

8131#	8132													
8132#														
8131#														
524#														
524#														
524#														
524#														
524#														
524#														
524#														
5596#														
8132#														
8132#														
8132#*														
1387	8131#													
1386#	1387	1404	5996	8131	8132									
1388#														
1386#														
8131#														
505#														
487#	503	524	1387	1418	1456	1471	1483	1584	1663	1670	1678	1681		
1727	1732	1737	1741	1791	1798	1805	1811	1813	1826	1969	2097	2155		
2240	2281	2328	2386	2409	2477	2506	2535	2571	2595	2641	2728	2795		
2920	3072	3323	3443	3575	3695	3814	3874	3876	4072	4078	4168	4268		
4372	4449	4456	4462	4468	4474	4572	4691	4812	4925	5022	5098	5251		
5272	5356	5435	5437	5440	5442	5445	5447	5451	5453	5456	5460	5473		
5478	5503	5506	5527	5691	5693	5780	5871	5962	5996	8131	8132			
503	8131													
524#	1387*	1418*	1456*	1471*	1483*	1584*	5962*	5996*	8131*					
524#	8131													
8131#*														
1392	1485	2182	6092	7108	8131#									
8131#														
8131#*														
8131#*														
8131#														
524#	8131*													
8131#														
524#	1530*	1531*	6279*	6284	6289	6293	6388*	6402	6487*	6533*	8872			
524#	1431*	3897*	3922	3989*	3993	4007*	4037	4831*	4886*	6280*	6284	6289		
6389#	6404	8870	8872	8885										
524#	3925*	3926*	3995*	3996*	6283*	6286*	6288*	6291*	6489*	6531*	8885			
524#	3929*	3999*	6277*	6278*	6488*	6490	6491*	8872	8885					
524#	3919*	3928	3950*	3979*	3998	6490*	6514*							
524#	3092*	3094*	3095	3101*	3103*	3104	3173	6486*	6492*	6498	6596*	6675		
6697														
487#	1410	1418#	1419	1456#	1457	1471#	1472	1477	1483#	1485	1580	1584#		
1585	1663#	1665	1670#	1672	1678#	1681#	1683	1727#	1732#	1737#	1741#	1742		
1791#	1792	1798#	1799	1805#	1811#	1813#	1826#	1827	1969#	2086	2097#	2098		
2155#	2156	2240#	2241	2277	2281#	2282	2319	2328#	2386#	2403	2409#	2410		
2466	2477#	2478	2497	2503	2506#	2507	2527	2535#	2536	2564	2571#	2572		
2589	2595#	2597	2613	2641#	2643	2728#	2729	2795#	2920#	2921	3058	3072#		
3073	3323#	3324	3439	3443#	3444	3575#	3576	3695#	3696	3814#	3815	3874#		
3876#	3877	4072#	4078#	4079	4139	4158	4168#	4169	4228	4252	4268#	4269		
4332	4355	4372#	4373	4449#	4450	4451	4456#	4457	4458	4462#	4463	4464		
4468#	4469	4470	4474#	4476	4524	4564	4572#	4579	4680	4691#	4699	4793		

	4812#	4814	4916	4925#	4927	5014	5022#	5023	5098#	5106	5236	5251#	5255	
	5261	5272#	5277	5348	5356#	5358	5435#	5437#	5438	5440#	5442#	5443	5445#	
	5447#	5448	5451#	5453#	5454	5456#	5457	5460#	5461	5473#	5478#	5479	5503#	
	5506#	5507	5527#	5528	5691#	5693#	5694	5780#	5781	5871#	5872	5962#		
	524#	8131*												
	524#	8131												
	524#	8131												
STPB	001152													
STPFLG	001157													
STPS	001150													
STRAP	057202	1387												
STRAP2	057224	8132#												
STRP =	000016	8132#												
STRPAD	057236	8132#												
STSTNM	001102	524#	5968*	5996*	8131*	8132								
STTYIN	056220	8131#												
STYPBN=	***** U	8132												
STYPDS	054506	8131#	8132											
STYPE	054732	8131#	8132											
STYPEC	055102	8131#												
STYPEX	055150	8131#												
STYPOC	057000	8132#												
STYPON	057014	8132#												
STYPOS	056754	8132#												
SXTSTR	054246	8131#												
SSGET4=	000000	5996#												
SOFILL	057177	8132#*												
S4OCAT=	***** U	8131	8132											
=	070246	504#	505#	508#	522#	524#	1287#	1360#	1361#	1362#	1387	1399#	1400#	1401#
		1447#	1449#	1489#	1513#	1557#	1560#	1562#	1627#	1651#	1653#	1655#	1665#	1667#
		3352#	5965#	5996	6027#	6030#	7121#	7129#	7677#	7678#	7679#	7680#	7681#	7682#
		7684#	7685#	7686#	7687#	7688#	8131#	8132#	8985#					



CHECKA	502#	4139	4228	4332	4451	4458	4464	5255							
CHECKB	502#	3068	3893	4114	4206	4308	4407	4451	4458	4464	4470	4512	4815	4963	5057
	5255	5308	5359	5479	5507	5580	5719	5805	5891	6363	6449	6596			
COMMEN	488	503#	1946												
ENDCOM	491	503#	1965												
ERROR	503#	1432	1463	1479	1672	1678	1700	1711	1720	1727	1732	1737	1755	1777	1792
	1799	1805	1811	1813	1911	2000	2090	2146	2230	2250	2259	2271	2278	2290	2305
	2316	2322	2347	2364	2366	2399	2404	2418	2438	2453	2461	2471	2490	2499	2520
	2529	2551	2565	2584	2591	2608	2615	2668	2679	2694	2718	2787	2845	2913	3061
	3153	3206	3267	3317	3385	3430	3520	3566	3639	3685	3760	3805	3865	3930	4000
	4067	4135	4158	4160	4224	4252	4254	4326	4355	4357	4422	4451	4458	4464	4470
	4547	4549	4635	4641	4680	4682	4753	4758	4794	4796	4843	4852	4865	4872	4900
	4910	4916	4997	4999	5069	5079	5087	5166	5171	5181	5188	5195	5202	5237	5239
	5255	5261	5330	5336	5348	5424	5479	5487	5507	5515	5656	5677	5679	5771	5861
	5955	6194	6200	6204	6216	6221	6224	6243	6250	6256	6261	6294	6640	6642	6666
	6668	6683	6694	6704	6714	6780	6979	6985	7046						
ESCAPE	503#														
GETPRI	503#														
GETSWR	487#	503#	1404												
HCOMPR	502#	5438	5443	5457											
HCOMPW	502#	5448	5454	5461											
MAKECL	502#	3874	4072	5435	5440	5445	5451	5473	5502	5691					
MSG	1417#	1418	1455#	1456	1583#	1584	1663#	1670#	1677#	1678	1680#	1681	1726#	1727	1731#
	1732	1736#	1737	1740#	1741	1790#	1791	1797#	1798	1804#	1805	1810#	1811	1812#	1813
	1825#	1826	1967#	1969	2096#	2097	2154#	2155	2239#	2240	2280#	2281	2327#	2328	2385#
	2386	2408#	2409	2476#	2477	2505#	2506	2534#	2535	2570#	2571	2594#	2595	2640#	2641
	2727#	2728	2794#	2795	2919#	2920	3071#	3072	3322#	3323	3442#	3443	3574#	3575	3694#
	3695	3813#	3814	3875#	3876	4077#	4078	4167#	4168	4267#	4268	4371#	4372	4448#	4449
	4455#	4456	4461#	4462	4467#	4468	4473#	4474	4569#	4572	4690#	4691	4810#	4812	4924#
	4925	5021#	5022	5097#	5098	5250#	5251	5271#	5272	5355#	5356	5477#	5478	5505#	5506
	5525#	5527	5692#	5693	5779#	5780	5870#	5871	5962#						
MSGA	5437#	5442	5456												
MSGB	5446#	5447	5453	5460											
MULT	503#														
NEWTST	503#	1418	1456	1471	1483	1584	1663	1670	1678	1681	1727	1732	1737	1741	1791
	1798	1805	1811	1813	1826	1969	2097	2155	2240	2281	2328	2386	2409	2477	2506
	2535	2571	2595	2641	2728	2795	2920	3072	3323	3443	3575	3695	3814	3874	3876
	4072	4078	4168	4268	4372	4449	4456	4462	4468	4474	4572	4691	4812	4925	5022
	5098	5251	5272	5356	5435	5437	5440	5442	5445	5447	5451	5453	5456	5460	5473
	5478	5503	5506	5526	5691	5693	5780	5871	5962						
POP	503#	6055	6146	6328	6412	6536	6576	6745	6782	7004	7061	7094	7322	7465	7531
	7592	7655	7833	7982	8088	8119	8131	8132							
PUSH	503#	6048	6136	6321	6384	6483	6485	6551	6728	6761	6870	7022	7074	7240	7371
	7486	7550	7616	7721	7863	8025	8101	8131	8132						
REPORT	503#														
RFORGC	502#	4158	4252	4355	4451	4458	4464	4680	4793	5236					
RH70CK	502#	1457	1670	2241	2282	2328	2386	2410	2478	2536	5023				
SAVE	502#	8132													
SAVTST	502#	1419	1457	1472	1485	1585	1663	1670	1672	1678	1683	1727	1732	1737	1742
	1792	1799	1805	1811	1813	1827	2098	2156	2241	2282	2328	2386	2410	2478	2507
	2536	2572	2597	2643	2729	2795	2921	3073	3324	3444	3576	3696	3815	3874	3877
	4072	4079	4169	4269	4373	4450	4457	4463	4469	4470	4476	4579	4699	4814	4927
	5023	5106	5255	5277	5358	5435	5438	5440	5443	5445	5448	5451	5454	5457	5461
	5473	5479	5503	5507	5528	5691	5694	5781	5872						
SCOPE	503#	1418	1456	1471	1483	1584	1663	1670	1678	1681	1727	1732	1737	1741	1791
	1798	1805	1811	1813	1826	1969	2097	2155	2240	2281	2328	2386	2409	2477	2506

	2535	2571	2595	2641	2728	2795	2920	3072	3323	3443	3575	3695	3814	3874	3876
	4072	4078	4168	4268	4372	4449	4456	4462	4468	4474	4572	4691	4812	4925	5022
	5098	5251	5272	5356	5435	5437	5440	5442	5445	5447	5451	5453	5456	5460	5473
	5478	5503	5506	5527	5691	5693	5780	5871	5962	5996					
SETPRI	503#	8131													
SETTRA	8132#														
SETUP	503#	1387													
SKIP	502#	503#	1410	1477	1580	1665	2086	2277	2319	2403	2466	2497	2527	2564	2589
	2613	3058	4139	4228	4332	4451	4458	4464	4470	4524	4564	4916	5014	5255	5261
	5348														
SLASH	503#														
SMORE	502#	8131													
SPACE	503#														
STARS	495	501	503#	505	522	524	952	953	990	991	1418	1456	1471	1483	1542
	1551	1584	1614	1624	1630	1655	1663	1670	1678	1681	1727	1732	1737	1741	1791
	1798	1805	1811	1813	1818	1823	1826	1874	1883	1954	1956	1969	2097	2155	2240
	2281	2328	2386	2409	2477	2506	2535	2571	2595	2620	2629	2635	2638	2641	2728
	2795	2920	3072	3323	3443	3575	3695	3814	3874	3876	4072	4078	4168	4268	4372
	4426	4445	4449	4456	4462	4468	4474	4572	4691	4812	4925	5022	5098	5251	5272
	5356	5435	5437	5440	5442	5445	5447	5451	5453	5456	5460	5465	5470	5473	5478
	5503	5506	5527	5691	5693	5780	5871	5962	5996	5998	6016	7154	7212	7214	7325
	7346	7352	7353	7475	7476	7538	7539	7599	7601	7665	7670	7695	7696	7837	7838
	7993	7994	8092	8093	8131	8132									
SWRSU	503#	1387#													
TJUMP	502#	1457	1670	2241	2282	2328	2386	2410	2478	2503	2536	3439	5023		
TRMTRP	8132#														
TSCLR	502#	2010	2013	2019	2022	2025	2028	2031	2034	2037	2049	2070	2074		
TSCLR1	502#	2063	2067												
TSCLR2	502#	2965	2969	2973	2996	2999	3002	3005	3008	3011	3023	3043	3046		
TSCLR3	502#	3036	3040												
TSCLR4	502#	2016													
TSCLR5	502#	2978													
TYPBIN	503#														
TYPDEC	503#	5996	8132												
TYPNAM	503#														
TYPNUM	503#														
TYPOCS	503#	1926	1935												
TYPOCT	503#	1629	8131	8132											
TYPTXT	503#	1398	1400	1401	1410	1436	1447	1449	1489	1512	1513	1557	1560	1562	1627
	1629	1650	1652	1654	1665	1667	3352	5963	5965	6022	6024	6027	6030	7105	7108
	7120	7123	7125	7127	7129	7131	7132	8125							
WDATAR	502#	4470	5479	5507											
XREAD	502#	4451	4458	4464	5255										
\$\$CMRE	524#														
\$\$CMTM	524#														
\$\$ESCA	503#														
\$\$NEWT	503#	1418	1456	1471	1483	1584	1663	1670	1678	1681	1727	1732	1737	1741	1791
	1798	1805	1811	1813	1826	1969	2097	2155	2240	2281	2328	2386	2409	2477	2506
	2535	2571	2595	2641	2728	2795	2920	3072	3323	3443	3575	3695	3814	3874	3876
	4072	4078	4168	4268	4372	4449	4456	4462	4468	4474	4572	4691	4812	4925	5022
	5098	5251	5272	5356	5435	5437	5440	5442	5445	5447	5451	5453	5456	5460	5473
	5478	5503	5506	5527	5691	5693	5780	5871	5962						
\$\$SET	8132#														
\$\$SKIP	503#														
.EQUAT	487#	503													
.FLOAT	502#	1672	1678	1727	1732	1737	1792	1799	1805	1811	1813				

.HEADE	487#	
.KT11	487#	522
.SETUP	487#	1385
.SWRHI	487#	503
.SWRLO	487#	503#
.SACT1	487#	505
.SCATC	487#	504
.SCHTA	487#	524
.SEOP	487#	5996
.SERRC	487#	8132
.SERRT	487#	8132
.SPOWE	487#	8132
.SRDOC	487#	8132
.SREAD	487#	8131
.SSCOP	487#	8131
.STRAP	487#	8132
.STYPD	487#	8131
.STYPE	487#	8131
.STYPO	487#	8132

. ABS. 070246 000

ERRORS DETECTED: 0

RM03:CZRJGB,CZRJGB,SEQ/CRF/SOL/NL:MC:ME:CND=CZRJGB.P11  
RUN-TIME: 35 30 2 SECONDS  
RUN-TIME RATIO: 1342/68=19.5  
CORE USED: 29K (57 PAGES)

EO4