

RP11E

FUNCT LOGIC&READ/WRITE
MD-11-DZRPY-B

EP-DZRPY-B-DL-A
COPYRIGHT © 1976
FICHE 1 OF 1

NOV 1976
digital
MADE IN U.S.A

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

1. ABSTRACT

THE RP11E FUNCTIONAL LOGIC & READ/WRITE TEST PERFORMS A CURSORY CHECK OF THE RP11E CONTROLLER LOGIC AND PERFORMS DATA STORAGE/RETRIEVAL TESTING OF THE SUBSYSTEM. INCLUDED IN THE PROGRAM ARE UTILITY TESTS WHICH ASSIST IN PERFORMING HEAD ALIGNMENT, WHICH CHECK SUBSYSTEM POWER FAIL OPERATION, AND WHICH CHECK THE RP11E AND DRIVE CONTROL SWITCHES.

THE PROGRAM CONTAINS A 'CONVERSATION MODE' WHICH ALLOWS THE OPERATOR TO MODIFY THE OPERATION PARAMETERS FOR THE DATA STORAGE/RETRIEVAL TEST. THE PROGRAM WILL USE MEMORY MANAGEMENT IN THE OPERATION OF THE DATA TEST IF THE PROGRAM IS BEING RUN ON A MEMORY MANAGEMENT SYSTEM.

THIS PROGRAM COMPLETES THE TESTING STARTED IN THE DISKLESS CONTROLLER TEST (MD-11-DZRPW). TESTING IN THE FUNCTIONAL LOGIC PROGRAM IS PERFORMED WITH 1 - 8 RPO2, RPRO2, OR RPO3 DISK DRIVES.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP-11 COMPUTER WITH 12K OF MEMORY; CONSOLE TELETYPE; RP11E DISK CONTROLLER; 1 TO 8 RPO2, RPRO2, OR RPO3 DISK DRIVES WITH FORMATTED PACKS.

2.2 STORAGE

THIS PROGRAM WILL LOAD AND RUN IN 12K.

2.3 PRELIMINARY PROGRAMS

MAINDC-11-DZRPW - RP11E DISKLESS LOGIC TEST

3. STARTING PROCEDURES

3.1 LOADING PROCEDURES

THE PROGRAM MAY BE LOADED FROM EITHER PAPER TAPE OR AN 'XXDP' DEVICE. REFER TO EITHER THE STANDARD PROCEDURES FOR LOADING 'ABS' FORMAT PAPER TAPES OR TO THE 'XXDP' SYSTEM REFERENCE DOCUMENT.

THIS PROGRAM MAY BE INCLUDED IN AN 'XXDP' CHAIN. IF THE PROGRAM IS PART OF A CHAIN AND IS LOADED FROM AN RPO2, RPRO2, OR AN RPO3, THE PROGRAM WILL NOT USE DRIVE 0 FOR TESTING.

155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
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

3.2 STARTING ADDRESSES

200 NORMAL STARTING ADDRESS
204 SELECT OPERATING PARAMETERS
210 SELECT RP11 ADDRESS
214 COMBINATION OF 204 AND 210

3.2.1 START FROM LOCATION 200

WHEN THE PROGRAM IS STARTED FROM LOCATION 200, ALL ONLINE DRIVES WILL BE USED (SEE SECTION 3.1 FOR DRIVE 0 EXCEPTION) USING TESTS 0 - 14. ALL PARAMETERS FOR TEST 14 WILL BE RETURNED TO THEIR DEFAULT VALUES.

3.2.2 START FROM LOCATION 204

WHEN THE PROGRAM IS STARTED FROM LOCATION 204, PARAMETERS FOR TEST 14 ARE RESET TO THEIR DEFAULT VALUES, AND THE PROGRAM ENTERS CONVERSATIONAL MODE.

3.2.3 START FROM LOCATION 210

WHEN THE PROGRAM IS STARTED FROM LOCATION 210, OPERATION IS THE SAME AS THE START FROM 200, EXCEPT THAT THE PROGRAM ASKS FOR AN RP11E ADDRESS. THE OPERATOR MAY ENTER A NEW ADDRESS VALUE OR HE MAY RETAIN THE CURRENT VALUE. THE PARAMETERS FOR TEST 14 ARE RESET TO THEIR DEFAULT VALUES.

3.2.4 START FROM LOCATION 214

PROGRAM START FROM LOCATION 214 IS THE SAME AS THE START FROM LOCATION 204 EXCEPT THAT THE PROGRAM ASKS FOR A NEW RP11E ADDRESS. THE PARAMETERS FOR TEST 14 ARE RESET TO THEIR DEFAULT VALUES.

3.3 OPERATOR ACTION

BEFOR STARTING REFER TO SECTION 4.1

1. LOAD PROGRAM INTO MEMORY (SEE SECTION 3.1)
2. CHANGE TELETYPE KEYBOARD AND PRINTER ADDRESSES AS REQUIRED (SEE SECTION 4.2)
3. PLACE A FORMATTED PACK ON DRIVE(S) TO BE USED
4. BRING DRIVE(S) TO ONLINE STATE.
5. PLACE 'FORMAT' AND 'MAINTENANCE' SWITCHES ON THE RP11E CONTROL PANEL TO 'NORMAL'
6. SET CONTROLLER AND DRIVE WRITE LOCKOUT SWITCHES AS REQUIRED
7. LOAD ADDRESS 200, 204, 210, OR 214.
8. SET CPU SWITCH REGISTER SWITCHES (SEE SECTION 4.1)
9. PRESS START ON THE CPU CONTROL PANEL

3.4 'CONTROL C' OPTION

THE OPERATOR MAY TERMINATE THE CURRENT TEST AT ANY TIME BY TYPING A 'CONTROL C' ON THE TELETYPE KEYBOARD. TYPING A 'CONTROL C' CAUSES THE PROGRAM TO ENTER THE 'CONVERSATION MODE' ROUTINE. ALL THE PARAMETERS FOR TEST 14 WILL BE RETURNED TO THEIR DEFAULT VALUES.

210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
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

3.5 DEFAULT RP11E UNIBUS ADDRESS

THE DEFAULT RP11E UNIBUS ADDRESS IS 176710. WHEN THE PROGRAM IS STARTED FROM EITHER LOCATION 200 OR LOCATION 204, ADDRESS 176710 WILL BE CHECKED AND IF NO RESPONSE IS RECEIVED FROM THAT ADDRESS THE PROGRAM WILL ENTER THE RP11E ADDRESS CHANGE ROUTINE. ADDRESS CHANGE OPERATION WILL BE IDENTICAL TO THE OPERATION WHEN THE PROGRAM IS STARTED FROM EITHER LOCATION 210 OR 214. (REFER TO SECTION 4.4)

4. OPERATING PROCEDURES

IF THE DIAGNOSTIC IS RUN ON A CPU WITHOUT A SWITCH REGISTER THEN A SOFTWARE SWITCH REGISTER IS USED WHICH ALLOWS THE USER THE SAME SWITCH OPTIONS AS THE HARDWARE SWITCH REGISTER. IF THE HARDWARE SWITCH REGISTER DOES NOT EXIST OR IF ONE DOES AND IT CONTAINS ALL ONES (177777) THEN THE SOFTWARE SWITCH REGISTER (LOC. 176) IS USED.

CONTROL:

THIS PROGRAM ALSO SUPPORTS THE DYNAMIC LOADING OF THE SOFTWARE SWITCH REGISTER (LOC. 176) FROM THE TTY. THIS CAN BE ACCOMPLISHED BY DOING THE FOLLOWING:

- 1) TYPE CONTROL G (<↑G>): THIS WILL ALLOW THE TTY TO ENTER DATA INTO LOC. 176 AT SELECTED POINTS WITHIN THE PROGRAM.
- 2) THE MACHINE WILL THEN TYPE: SWR=XXXXXXNEW= (XXXXXX IS THE OCTAL CONTENTS OF THE SOFTWARE SWITCH REGISTER.)
- 3) AFTER THE "NEW=" HAS BEEN TYPED THEN THE OPERATOR CAN DO ONE OF THE FOLLOWING AT THE TTY:
 - A) TYPE A NUMBER TO BE LOADED INTO LOC. 176 FOLLOWED BY A <CR>. (ONLY NUMBERS BETWEEN 0-7 WILL BE ACCEPTED AND ONLY 6 NUMBERS WILL BE ALLOWED) IF A <CR> IS THE FIRST KEY DEPRESSED THE SOFTWARE SWITCH REGISTER CONTENTS WILL NOT BE CHANGED.
 - B) IF A CONTROL U (<↑U>) IS DEPRESSED THEN THE PROGRAM WILL SEND YOU BACK TO STEP 2.

4.1 OPERATIONAL SWITCH SETTINGS

WITH SW<15:0>=1, THE PROGRAM WILL TYPEOUT ERRORS AND CONTINUE WITH THE TEST.

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

THE SWITCH SETTINGS ARE:

SW<15>=1...HALT ON ERROR
 SW<14>=1...LOOP ON TEST
 SW<13>=1...INHIBIT ERROR TIMEOUTS
 SW<11>=1...INHIBIT ITERATIONS
 SW<10>=1...RING BELL ON ERROR
 SW<09>=1...LOOP ON ERROR
 SW<07>=1...DON'T TYPE DRIVE STATUS WHEN PROGRAM STARTED OR WHEN
 ENTERING CONVERSATION MODE.
 CHANGE HEAD SELECTION IN TEST 16.
 SW<06>=1...DON'T USE MEMORY MANAGEMENT IN TEST 14
 SW<05>=1...DISPLAY ALL READ RETRY ATTEMPTS IN TEST 14
 SW<04>=1...TYPE ALL DATA COMPARE ERRORS IN TESTS 13 & 14
 SW<03>=1...INHIBIT DATA COMPARE IN TESTS 13 & 14
 SW<02>=1...BYPASS READ & DATA COMPARE IN TEST 14
 SW<01>=1...BYPASS WRITE CHECK IN TEST 14
 SW<00>=1...BYPASS WRITE IN TEST 14

4.2 TELETYPE KEYBOARD AND PRINTER UNIBUS ADDRESSES

THE PROGRAM ASSUMES THE FOLLOWING UNIBUS ADDRESSES. THESE ADDRESSES
 MAY BE CHANGED AT THE INDICATED LOCATION BEFORE STARTING THE
 PROGRAM.

MEMORY LOCATION	CONTENTS	FUNCTION
1136	177560	TTY KEYBOARD STATUS REG
1140	177562	TTY KEYBOARD BUFFER REG
1142	177564	TTY PRINTER STATUS REGISTER
1144	177566	TTY PRINTER BUFFER REG

(THE RP11E UNIBUS ADDRESS IS CHANGED IN RESPONSE TO THE PROGRAM'S
 TYPED REQUEST IF THE PROGRAM IS STARTED FROM EITHER LOCATION 210 OR
 214 OR IF THERE IS NO RESPONSE WHEN LOCATION 176710 IS ADDRESSED.)

4.3 TTY KEYBOARD ENTRY CONTROL CODES AND CONVENTIONS

THE FOLLOWING CONTROL CODES AND CONVENTIONS ARE USED BY THE PROGRAM:

CARRIAGE RETURN (CR) - ALL ENTRIES MUST BE TERMINATED BY A CARRIAGE
 RETURN.

CONTROL U (↑U) - IF THIS CONTROL CODE IS TYPED BEFORE AN ENTRY IS
 TERMINATED, THE ENTRY IS CLEARED AND ANOTHER VALUE MAY BE ENTERED.
 A ↑U IS TYPED BY PRESSING THE 'CONTROL' AND 'U' KEYS TOGETHER.

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
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363

RUBOUT (RO) - IF THIS CONTROL CODE IS TYPED, THE CHARACTER TYPED BEFORE THE 'RO' WILL BE DELETED FROM THE ENTRY. EACH TIME A 'RO' IS TYPED, THE LAST CHARACTER IN THE INPUT ENTRY IS DELETED. WHEN A 'RO' IS TYPED, THE ENTRY ROUTINE WILL TYPE A BACKSLASH ('\') CHARACTER AND THE CHARACTER BEING DELETED. WHEN A CHARACTER OTHER THAN A 'RO' IS TYPED, THE ENTRY ROUTINE WILL TYPE ANOTHER BACKSLASH CHARACTER AND TYPE THE NEW CHARACTER ENTERED. THE 'RO' CANNOT BE USED TO DELETE A 'CR'.

CONTROL C (IC) - THE OPERATION OF THE 'IC' CHARACTER IS DESCRIBED IN SECTION 3.4. 'IC' MAY ALSO BE USED TO ABORT THE CONVERSATION MODE AND TO RETURN TO THE BEGINNING OF CONVERSATION MODE. 'IC' IS GENERATED BY PRESSING THE 'CONTROL' KEY AND THE 'C' KEY.

IF ONLY A 'CR' IS ENTERED IN RESPONSE TO AN ENTRY REQUEST FROM THE PROGRAM, THE PROGRAM WILL USE THE DEFAULT VALUE FOR THE REQUESTED ENTRY.

LEADING ZEROS ARE NOT REQUIRED ON ANY ENTRY.

4.4 RP11E UNIBUS ADDRESS CHANGE ROUTINE

WHEN THE PROGRAM IS STARTED FROM LOCATION 200 OR LOCATION 204, THE PROGRAM WILL ASSUME THAT THE RP11E IS AT UNIBUS ADDRESS 176710. WHEN THE PROGRAM IS STARTED FROM EITHER LOCATION 210 OR LOCATION 214, THE PROGRAM ENTERS A ROUTINE WHICH ALLOWS THE OPERATOR TO CHANGE THE RP11E UNIBUS ADDRESS TO BE USED BY THE PROGRAM.

WHEN THE ADDRESS CHANGE ROUTINE IS ENTERED, THE PROGRAM TYPES THE PRESENT RP11E UNIBUS ADDRESS:

RPADR = AAAAAA/

NOTE: 'AAAAAA' REPRESENTS THE PRESENT ADDRESS

THE ADDRESS TYPED OUT MAY BE USED BY TYPING A 'CR' ONLY. A NEW ADDRESS IS ENTERED BY TYPING THE NEW ADDRESS AND TERMINATING THE ENTRY WITH A 'CR'. (THE NEW ADDRESS WILL BE USED UNTIL ANOTHER ADDRESS IS ENTERED.)

AFTER THE OPERATOR HAS RESPONDED TO THE ADDRESS ENTRY REQUEST, THE PROGRAM WILL VERIFY THAT THE RP11E ADDRESS RESPONDS. IF NO RESPONSE IS RECEIVED FROM THE RP11E ADDRESS, THE FOLLOWING ERROR MESSAGE IS TYPED:

RP11E DIDN'T RESPOND TO ADDRESSING
RPADR
AAAAAA

THE PROGRAM RETURNS TO THE ADDRESS ENTRY ROUTINE. THE PROGRAM WILL NOT CONTINUE UNTIL THE OPERATOR ENTERS AN ADDRESS FOR WHICH THERE IS A UNIBUS RESPONSE.

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
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414

4.5 CONVERSATION MODE

THE PROGRAM CONTAINS A 'CONVERSATION MODE' ROUTINE WHICH ALLOWS THE OPERATOR TO SELECT AND EXECUTE INDIVIDUAL TESTS AND TO SPECIFY THE DRIVE TO BE USED. THE CONVERSATION MODE ALSO ALLOWS THE OPERATOR TO CHANGE THE OPERATING PARAMETERS OF THE 'DATA STORAGE/RETRIEVAL TEST' (TEST 14).

WHEN THE PROGRAM ENTERS CONVERSATION MODE, THE FOLLOWING MESSAGE IS TYPED:

ENTER TEST #:

A 'CR' MAY BE ENTERED OR A VALID TEST NUMBER (E.G., 0 - 17) MAY BE ENTERED. THE DEFAULT ENTRY ('CR') WILL CAUSE THE PROGRAM TO SELECT TESTS 0 - 14. IF A SINGLE TEST NUMBER IS ENTERED, ONLY THAT TEST WILL BE EXECUTED.

THE PROGRAM WILL THEN ASK FOR A DRIVE ASSIGNMENT.

ENTER DRIVE #:

A 'CR' ENTRY WILL CAUSE THE PROGRAM TO SELECT ALL ONLINE DRIVES FOR TESTING. IF A VALID DRIVE NUMBER IS ENTERED (0 - 7), THE PROGRAM WILL CHECK THE STATE OF THE DRIVE: IF THE DRIVE IS NOT ONLINE, THE DRIVE'S STATUS WILL BE TYPED AND THE PROGRAM WILL ASK FOR ANOTHER DRIVE ENTRY.

IF TEST 15, 16, OR 17 IS SELECTED, A 'CR' RESPONSE TO THE DRIVE NUMBER REQUEST WILL NOT BE ACCEPTED; A SPECIFIC DRIVE NUMBER MUST BE ENTERED FOR TEST 15, 16, OR 17.

IF TESTS 0 - 13, 15, 16 OR 17 HAVE BEEN SELECTED, OR IF THE ENTIRE TEST SEQUENCE WERE SPECIFIED, THE PROGRAM WILL BEGIN EXECUTING THE TEST OR SEQUENCE IMMEDIATELY. IF TEST 14 IS SPECIFIED, THE PROGRAM WILL ASK IF THE OPERATOR WANTS TO CHANGE PARAMETERS FOR TEST 14. THE FOLLOWING MESSAGE IS TYPED.

CHANGE PARAMETERS FOR TEST 14 ?

EITHER A 'CR' OR A 'Y' MAY BE ENTERED. IF A 'CR' IS ENTERED, THE PROGRAM WILL BYPASS THE TEST 14 PARAMETER ENTRY ROUTINE AND START TEST 14. IF A 'Y' IS ENTERED, THE PROGRAM STARTS THE TEST 14 PARAMETER CHANGE DIALOG.

THE PROGRAM WILL THEN TYPE THE WORD LENGTH FOR ALL TEST 14 TRANSFERS.

MAXIMUM WORD COUNT FOR TEST 14 (IN OCTAL): SSSSSS

NOTE: 'SSSSSS' = THE OCTAL VALUE OF THE WORD COUNT

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
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467

THE PROGRAM WILL THEN ASK FOR A NEW WORD COUNT.

ENTER NEW WORD COUNT:

EITHER A VALUE (LESS THAN THE MAXIMUM TYPED ABOVE) OR A 'CR' MAY BE ENTERED. IF A VALUE IS ENTERED, THE NEW VALUE WILL BE USED FOR ALL DATA TRANSFER OPERATIONS IN TEST 14. WHEN THE PROGRAM IS RESTARTED OR A 'CONTROL C' IS TYPED, THE WORD COUNT VALUE WILL BE RETURNED TO THE SYSTEM DEFAULT VALUE. IF A 'CR' IS TYPED, THE PROGRAM WILL USE THE DEFAULT WORD COUNT VALUE.

THE PROGRAM WILL THEN ASK FOR A PATTERN SELECTION WORD ENTRY.

ENTER PATTERN SELECTION CODE FOR TEST 14:

ENTER THE OCTAL VALUE OF THE PATTERN SELECTION WORD OR A 'CR' FOR THE DEFAULT VALUE. REFER TO SECTION 7.9 FOR A DESCRIPTION OF THE PATTERN SELECTION WORD AND ITS USE IN TEST 14. IF TESTING IS TO BE AT A SPECIFIC DISK LOCATION, ENTER THE CODE FOR A SINGLE PATTERN. (IF TESTING IS AT A SPECIFIC ADDRESS, THE PROGRAM WILL NOT ROTATE THE DATA PATTERNS BUT WILL USE THE LOWEST NUMBERED PATTERN SPECIFIED BY THE PATTERN SELECT WORD.)

THE PROGRAM WILL ASK IF THE OPERATOR WANTS TO TEST AT A SPECIFIC DISK ADDRESS.

USE A SPECIFIC DISK ADDRESS FOR TEST 14 ?

ENTER EITHER A 'CR' OR A 'Y'. IF A 'Y' IS ENTERED, THE PROGRAM WILL TYPE THE FOLLOWING REMINDER MESSAGE ABOUT DATA PATTERN USAGE.

(A SINGLE DATA PATTERN MUST HAVE BEEN SPECIFIED
IF NOT, PATTERN 0 (ALL ZEROS) WILL BE USED)

THE PROGRAM WILL ASK FOR CYLINDER, TRACK, AND SECTOR ADDRESSES. AN ADDRESS MUST BE ENTERED. THE PROGRAM WILL NOT ACCEPT DEFAULT ENTRIES FOR THESE ADDRESSES. THE ENTRY REQUESTS FOR THE ADDRESSES ARE GIVEN BELOW.

ENTER CYLINDER ADDRESS:

ENTER TRACK ADDRESS:

ENTER SECTOR ADDRESS:

IF A SPECIFIC ADDRESS IS ENTERED FOR TESTING, TESTING WILL TAKE PLACE ONLY AT THAT ADDRESS WITH THE WORD COUNT AND PATTERN ENTERED ABOVE. THE TEST WILL MAKE ONE PASS AND GO TO END OF TEST. SW(14) SHOULD BE SET TO INHIBIT THE END OF TEST MESSAGE TIMEOUT.

TO TERMINATE THE TEST OR SEQUENCE, EITHER TYPE A 'CONTROL C' OR HALT THE PROCESSOR AND RESTART AT THE APPROPRIATE STARTING ADDRESS.

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
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514

4.6 OPERATION OF TESTS 15, 16, & 17

TEST 15 AND TEST 17 CONTAIN IMBEDDED OPERATING INSTRUCTIONS. SETUP AND OPERATING INSTRUCTIONS ARE TYPED OUT AS THE TEST PROCEEDS.

TO OPERATE TEST 16, 'HEAD ALIGNMENT ROUTINE', REFER TO THE APPLICABLE MAINTENANCE MANUALS FOR THE DRIVE SETUP PROCEDURES. THE PROGRAM WILL TYPEOUT THE CONTROLLER SETUP PROCEDURES. TO CHANGE HEAD SELECTION, SW<07> MUST BE TOGGLED (TURNED ON, THEN OFF).

TO SETUP THE RP11E FOR HEAD ALIGNMENT, CONNECT A JUMPER BETWEEN PINS J13M2 AND J13C2 ON THE BACKPLANE.

5. ERROR HANDLING

5.1 ERROR REPORTING

WHEN THE PROGRAM DETECTS AN ERROR, THE ERROR ROUTINE IS CALLED; IF SW<13> IS NOT SET, THE ERROR MESSAGE FOR THAT ERROR WILL BE TYPED. EACH ERROR TYPEOUT CONTAINS THE FOLLOWING:

- A. AN ERROR MESSAGE
- B. A DATA HEADER LINE
- C. A DATA LINE CONTAINING
 1. THE TEST NUMBER
 2. THE ADDRESS (PROGRAM COUNTER CONTENTS) WHERE THE ERROR CALL WAS MADE
 3. CONTENTS OF THE APPROPRIATE REGISTERS
 4. OTHER SPECIFIC DATA

5.2 ERROR TYPES

THE ERRORS DETECTED ARE DIVIDED INTO 2 CATEGORIES: FATAL AND NON-FATAL.

FATAL ERRORS ARE DRIVE UNSAFE ERRORS WHICH OCCUR ON THE DRIVE BEING USED FOR TESTING OR DRIVES WHICH GO OFFLINE DURING TESTING. IF EITHER OF THESE TWO CONDITIONS OCCUR, THE PROGRAM WILL TERMINATE TESTING USING THE DRIVE WHICH GAVE THE FATAL ERROR.

ALL OTHER ERROR TYPES ARE NON-FATAL, E.G., TESTING CONTINUES AFTER THE ERROR IS REPORTED.

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

6. RESTRICTIONS

6.1 SUBSYSTEMS WITH MIXED DRIVE TYPES

THE PROGRAM ASSUMES THAT ONLY ONE KIND OF DRIVE IS ATTACHED TO A SYSTEM (E.G. RPO2'S & RPRO2'S OR RPO3'S). IF BOTH RPO2'S (AND RPRO2'S) OR RPO3'S ARE ON THE SAME SYSTEM, THE PROGRAM WILL ASSUME THAT ALL OF THE DRIVES ARE RPO3'S. TO TEST DRIVES ON A MIXED SYSTEM, TEST ALL OF THE SAME KIND OF DRIVE AT ONE TIME. THE OTHER DRIVES ON THE CONTROLLER MUST BE POWERED DOWN.

6.2 FORMATTED PACKS

EACH OF THE DRIVES BEING USED MUST HAVE A FORMATTED DISK PACK.

6.3 'CONTROL C' TERMINATION DURING TESTS 5 & 13

IF TESTS 5 OR 13 ARE TERMINATED WITH A 'CONTROL C', CYLINDER 0, TRACK 0, SECTOR 0 MAY BE LEFT WITH EITHER AN INVALID HEADER OR AN INVALID DATA FIELD. IF TEST 5 IS STOPPED, THE HEADER FOR SECTOR 0 MAY STILL BE MISFORMATTED; IF TEST 13 IS STOPPED, SECTOR 0 MAY BE STILL RECORDED IN 10/15 MODE.

6.4 CONTROLLER INTERRUPT

THE PROGRAM DOES NOT USE CONTROLLER INTERRUPT. THE PROGRAM WAITS FOR THE CONTROLLER 'READY' OR DRIVE 'READY' BITS TO SET TO INDICATE ORDER TERMINATION.

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
580
581
582
583
584
585
586
587
588
5897. MISCELLANEOUS

7.1 RUN TIME

TO MAKE ONE PASS OF THE PROGRAM (TESTS 0 - 14) TAKES APPROXIMATELY 20 MINUTES FOR EACH RPO2 OR RPRO2 USED AND APPROXIMATELY 40 MINUTES FOR EACH RPO3 USED.

7.2 STACK POINTER

STACK IS INITIALLY SET TO 1100 AND EXTENDS DOWNWARD IN MEMORY.

7.3 END OF PASS/END OF TEST

THE PROGRAM WILL PERFORM ONE PASS FOR EACH DRIVE BEING USED FOR TESTING. END OF TEST WILL OCCUR WHEN EACH DRIVE BEING USED FOR TESTING HAS HAD ONE PASS.

7.4 SUBROUTINE CALLS

THE SUBROUTINE CALLS USED BY THE PROGRAM ARE:

- A. 'SCOPE' (IOT INSTRUCTION). THIS CALL IS PLACED BETWEEN EACH TEST IN THE PROGRAM. THIS ROUTINE ESTABLISHES THE TEST ITERATION COUNT AND LOOP ON TEST AND LOOP ON ERROR ADDRESSES.
- B. 'ERROR' (EMT INSTRUCTION). THIS CALL IS USED TO REPORT ALL ERRORS. A CALL TO THIS ROUTINE IS FOLLOWED BY A NUMBER WHICH IDENTIFIES THE ERROR MESSAGE WHICH WILL BE TYPED.

THE TRAP INSTRUCTION IS USED FOR THE FOLLOWING SUBROUTINE CALLS:

TYPE - TTY TYPEOUT ROUTINE
 TYPOC - TYPE OCTAL NUMBER (WITH LEADING ZEROS)
 TYPOS - TYPE OCTAL NUMBER (NO LEADING ZEROS)
 TYPOS - TYPE DECIMAL NUMBER
 RDCHR - READ A CHARACTER FROM THE TTY KEYBOARD
 RDLIN - READ A LINE FROM THE TTY KEYBOARD
 SAVREG - ROUTINE TO SAVE R0 - R5
 RESREG - ROUTINE TO RESTORE R0 - R5

590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643

7.5 DISK SURFACE USAGE

THE PROGRAM USES THE FOLLOWING DISK LOCATIONS FOR TESTING.

TEST	DISK ADDRESS
0	CYL 0, TRK 0, SEC 0
1	CYL 0, TRK 0, SEC 0
2	CYL 0, TRK 0, SEC 0
3	CYL 0, TRK 0, SEC 0 MAXIMUM CYL, TRK 19, SEC 9
4	CYL 0, TRK 0, SEC 0
5	CYL 0, TRK 0, SEC 0
6	CYL 192, TRK 0, SEC 0
7	CYL 0, TRK 0, SEC 0
10	CYL 0, TRK 0, SEC 0 - 9
11	CYL 0, TRK 0 - 19, SEC 0
12	CYL 0, TRK 0, SEC 0
13	CYL 0, TRK 0, SEC 0
14	ENTIRE PACK
15	MAXIMUM CYL, TRK 0, SEC 0
16	N/A
17	NONE

7.6 TEST EXECUTION SEQUENCE

THE TEST SEQUENCE IS TESTS 0 - 14. TESTS 15, 16, & 17 ARE ONLY PERFORMED IF DIRECTED BY THE OPERATOR FROM CONVERSATION MODE. TESTS 15, 16, OR 17 CANNOT BE EXECUTED IN SEQUENCE.

7.7 DRIVE STATUS TYPEOUT

WHEN THE PROGRAM IS STARTED FROM LOCATION 200, 204, 210, OR 214 OR ENTERS THE CONVERSATION ROUTINE FROM A 'CONTROL C' THE PROGRAM WILL TYPE THE STATUS OF DRIVES 0 - 7. THIS DRIVE STATUS TYPEOUT CAN BE INHIBITED BY SETTING SW(07) BEFORE THE PROGRAM IS STARTED OR A 'CONTROL C' IS TYPED.

644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697

7.8 DATA COMPARISON USING MEMORY MANAGEMENT

IF THE SYSTEM HAS MORE THAN 28K, TEST 14 WILL ROTATE THE READ BUFFERS THROUGH ALL OF MEMORY. IF A DATA COMPARISON ERROR IS DETECTED, THE PROGRAM WILL TYPEOUT THE 'GOOD' AND THE 'BAD' DATA AND THE V I R T U A L ADDRESS OF THE COMPARISON ERROR LOCATION. THE CONTENTS OF THE APR'S IN USE WILL ALSO BE TYPED. IF MEMORY MANAGEMENT IS ENABLED, THE PROGRAM PERFORMS ALL DATA COMPARISONS FROM VIRTUAL LOCATIONS 60000 - 137776 USING APR'S 3, 4, & 5. THE PHYSICAL ADDRESS OF THE COMPARISON ERROR CAN BE FORMED BY ADDING THE APPLICABLE 'KIPAR' REGISTER CONTENTS (THE REGISTERS ARE DISPLAYED) TO THE VIRTUAL ADDRESS OF THE ERROR.

7.9 PATTERN SELECTION WORD

THE DATA PATTERNS USED BY TEST 14 ARE SELECTED FROM A SELECTION WORD WHICH IS LOADED BY THE PROGRAM OR LOADED BY THE OPERATOR IN CONVERSATION MODE. THE PATTERN SELECTION WORD CONTAINS A BIT FOR EACH PATTERN TO BE USED: BIT00 REPRESENTS PATTERN 0, BIT01 REPRESENTS PATTERN 1, ETC. REFER TO SECTION 8.13 FOR THE CONTENTS OF EACH PATTERN.

THE OCTAL VALUES FOR EACH DATA PATTERN ARE AS FOLLOWS.

PATTERN #	OCTAL SELECTION CODE
0	1
1	2
2	4
3	10
4	20
5	40
6	100
7	200
8	400
9	1000
10	2000
11	4000
12	10000
13	20000
14	40000
15	100000

THE INDIVIDUAL OCTAL CODES ARE 'OR ED' TOGETHER TO FORM THE PATTERN SELECTION WORD. TO SELECT PATTERNS 0, 6, & 13, THE PATTERN SELECTION WORD COULD BE THE FOLLOWING OCTAL VALUE: 20101.

7.10 CONTROLLER/DRIVE OPERATION TIME LIMIT

ALL CONTROLLER AND DRIVE OPERATIONS ARE TIMED BY THE PROGRAM. IF AN OPERATION EXCEEDS THE TIME LIMIT, THE PROGRAM WILL REPORT THE TIMEOUT, RESET THE CONTROLLER, TERMINATE THE CURRENT TEST, AND GO TO THE NEXT TEST. THE TIME LIMIT IS 150 MS TO 1.5 SEC (DEPENDING ON THE PROCESSOR).

698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750

- 8. TEST DESCRIPTIONS
 - 8.1 TEST 0 - 'CLEAR' COMMAND TERMINATION TEST
VERIFY THAT 'CLEAR' TERMINATES AN OPERATION AND CAUSES CONTROLLER READY TO SET.
 - 8.2 TEST 1 - WRITE CHECK COMMAND TEST
THIS TEST VERIFIES THE WRITE CHECK LOGIC AND TO VERIFY THAT A WRITE CHECK ERROR CAN BE DETECTED. 'FLOATING ONES' AND 'FLOATING ZEROS' PATTERNS ARE USED TO TEST THE WRITE CHECK COMPARE LOGIC.
 - 8.3 TEST 2 - PARTIAL SECTOR WRITE TEST
CHECK THE ABILITY OF THE RP11E TO CLEAR THE REMAINDER OF A SECTOR ON A PARTIAL WRITE OPERATION. A SECTOR OF ALL ONES IS WRITTEN AND THEN A TWO WORD WRITE OPERATION IS PERFORMED. THE SECTOR IS THEN READ BACK AND VERIFIED. THE FIRST TWO WORDS SHOULD BE ONES AND THE REST SHOULD BE ZEROS.
 - 8.4 TEST 3 - 'EOP' TEST
TEST THAT 'EOP' SETS WHEN THE CONTROLLER TRIES TO WRITE BEYOND THE LIMITS OF THE DRIVE. THE FIRST SECTOR OF THE PACK IS WRITTEN WITH ZEROS; THEN A TWO SECTOR WRITE OF ALL ONE'S IS ISSUED FOR THE MAXIMUM BE SET. THE FIRST SECTOR OF THE PACK IS CHECKED TO MAKE SURE THAT IS STILL CONTAINS ZEROS.
 - 8.5 TEST 4 - 'PROG' ERROR TEST
VERIFY THAT THE RP11E GENERATES A 'PROG' ERROR IF A COMMAND IS ISSUED WHILE THE CONTROLLER IS BUSY.
 - 8.6 TEST 5 - 'HEADER NOT FOUND' TEST
MISFORMAT THE FIRST SECTOR ON THE PACK AND THEN READ IT BACK. VERIFY THAT READ AND WRITE HEADER OPERATIONS WILL TRANSFER DATA CORRECTLY. ISSUE A WRITE COMMAND TO SECTOR ZERO; THIS SHOULD CAUSE 'HEADER NOT FOUND' TO SET. THE TEST SECTOR IS REFORMATTED.
 - 8.7 TEST 6 - COMMAND BUFFERING TEST
ISSUE A SEEK COMMAND AND WAIT FOR DONE TO SET. THEN ISSUE A READ COMMAND WHILE THE HEADS ARE STILL MOVING. THE RP11E SHOULD HOLD THE READ COMMAND UNTIL THE SEEK IS COMPLETE.
 - 8.8 TEST 7 - 'NOME' ERROR TEST
THE NON-EXISTENT MEMORY ERROR BIT IS TESTED BY ATTEMPTING TO READ 1 WORD INTO LOCATION 760000. 'NOME', 'HE', & 'ERR' SHOULD ALL BE SET.

751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802

8.9 TEST 10 - SECTOR ADDRESSING TEST

THE SECTOR ADDRESSING TEST TESTS THE CAPABILITY OF THE RP11E TO LOCATE A SECTOR BY USING THE 'SOT' COUNTER (FOR 'HDR' OPERATIONS) AND TO LOCATE A SECTOR BY COMPARING THE HEADER CONTENTS AGAINST THE CYLINDER AND TRACK/SECTOR ADDRESS REGISTERS (NORMAL MODE). THE TEST IS PERFORMED IN 3 PARTS.

1. WRITE ALL HEADERS ON TRACK 0, CYLINDER 0 IN ASCENDING SEQUENCE FROM INDEX (0, 1, 2 ... 9).

2. READ EACH HEADER, BEGINNING WITH SECTOR 0, AND VERIFY THAT THE SECTOR FIELD IN THE HEADER IS CORRECT FOR THE HEADER EXPECTED.

VERIFY THAT THE 'SOT' COUNTER AND THE SECTOR ADDRESS IN 'RPDA' ARE EQUAL AND ARE THE EXPECTED VALUE. THE EXPECTED VALUE IS 1 GREATER THAN THE SECTOR READ. THE SECTOR ADDRESS REGISTER AND THE 'SOT' WILL BE INCREMENTED BY THE TIME THE PROGRAM IS ABLE TO READ THE REGISTER.

WRITE THE SECTOR'S ADDRESS IN THE DATA FIELD. (FOR SECTOR 0, AN OCTAL 12 IS WRITTEN.)

CONTINUE THIS SEQUENCE FOR THE REMAINING SECTORS ON THE TRACK.

3. READ THE DATA FROM EACH SECTOR USING A NORMAL (I.E. NON-HEADER) READ AND VERIFY THAT THE DATA IS CORRECT FOR THE EXPECTED SECTOR.

8.10 TEST 11 - TRACK ADDRESSING TEST

TRACK ADDRESSING IS TESTED BY WRITING THE TRACK'S ADDRESS AS DATA IN SECTOR 0 OF EACH TRACK IN CYLINDER 0. SECTOR 0 IS THEN READ BACK AND THE DATA IS CHECKED TO VERIFY THAT THE PROPER HEAD WAS SELECTED.

8.11 TEST 12 - EXTENDED MEMORY ADDRESS BIT TEST

THIS TEST TESTS THE OPERATION OF THE EXTENDED MEMORY ADDRESS BITS. IF THE SYSTEM DOES NOT HAVE MEMORY MANAGEMENT OR HAS MEMORY MANAGEMENT AND ONLY 32K, THE TEST WILL NOT BE PERFORMED.

1. THE PROGRAM WRITES A 2 WORD TEST SECTOR OF ALL ONES.

2. EXTENDED ADDRESS BIT 'MEX00' IS TESTED BY CLEARING LOCATION 200000 AND READING THE TEST SECTOR INTO LOCATION 177776. LOCATION 200000 IS CHECKED TO VERIFY THAT THE DATA IS CORRECT (ONES). THE PROGRAM VERIFIES THAT 'MEX00' HAS SET AND THAT 'MEX01' IS NOT SET.

3. IF THE SYSTEM HAS AT LEAST 64K, 'MEX01' IS TESTED. LOCATION 400000 IS CLEARED AND THE TEST SECTOR IS READ INTO LOCATION 377776 ('MEX00' IS SET FOR THE READ). LOCATION 400000 IS CHECKED FOR THE PROPER CONTENTS (ONES). 'MEX00' SHOULD HAVE RESET AND 'MEX01' SHOULD BE SET.

803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853

8.12 TEST 13 - RP11E DATA BUFFER REGISTER BIT TEST

THE DATA BUFFER REGISTER TEST VERIFIES THAT ALL 36 BITS OF THE DATA BUFFER REGISTER CAN BE SET AND CLEARED.

1. THE TEST WRITES THE FOLLOWING TEST PATTERN IN CYLINDER 0, TRACK 0, SECTOR 0 USING 10/15 MODE ('MODE' BIT SET).

052525
052525
000005
125252
125252
000012
052525
052525
000005
125252

.
.
ETC

2. THE TEST SECTOR IS THEN READ AND THE DATA IS COMPARED. AT THE COMPLETION OF THE TEST, THE PROGRAM RESTORES THE TEST SECTOR TO PDP-11 MODE.

8.13 TEST 14 - DATA STORAGE/RETRIVAL TEST

THE DATA TEST PERFORMS DATA STORAGE AND RETREIVAL TESTING USING THE CONTROLLER AND THE CURRENTLY SELECTED DRIVE. UNLESS ALTERED BY THE OPERATOR, THE TEST WILL WRITE AND WRITE CHECK THE ENTIRE DISK PACK USING ALL 16 DATA PATTERNS IN A ROTATING MANNER. THE BUFFER SIZE USED FOR THE WRITE AND WRITE CHECK ORDERS IS DETERMINED BY THE AVAILABLE MEMORY ON THE SYSTEM (MINUS THE SPACE REQUIRED BY THE PROGRAM LOADER). THE PROGRAM WILL USE THE AVAILABLE MEMORY SIZE OR 8K, WHICHEVER IS LESS, AS THE BUFFER SIZE.

AFTER THE PACK HAS BEEN WRITTEN, THE TEST WILL READ AND COMPARE THE DATA. THE BUFFER SIZE USED FOR THE READ IS THE SAME SIZE AS THAT USED FOR THE WRITE/WRITE CHECK SEGEMENT OF THE TEST. FOR THE READ SEGEMENT OF THE TEST, THE PROGRAM ROTATES THE READ BUFFER THROUGH MEMORY SO THAT ALL OF AVAILABLE MEMORY IS USED BY THE READ AND COMPARE OPERATIONS. THE TEST USES MEMORY MANAGEMENT ON SYSTEMS WITH MORE THAN 28K. (MEMORY MANAGEMENT WILL NOT BE USED IF THE TEST IS RUN ON MEMORY MANAGEMENT SYSTEMS WITH 28K OR LESS.)

IF AN ERROR OCCURS DURING A READ OPERATION (EXCEPT DATA COMPARISON ERRORS), THE PROGRAM WILL RETRY THE READ ORDER UP TO 3 TIMES. WRITE AND WRITE CHECK ORDERS ARE NOT RETRIED.

THE DATA PATTERNS USED BY THE DATA TEST ARE GIVEN BELOW:

PAT 0	PAT 1	PAT 2	PAT 3	PAT 4	PAT 5	PAT 6	PAT 7
000000	000001	177776	000000	000000	052525	007417	026455
000000	000003	177774	000000	010421	052525	007417	026455
000000	000007	177770	000000	021042	052525	007417	026455
000000	000017	177760	177777	031463	125252	170360	151322
000000	000037	177740	177777	042104	125252	170360	151322
000000	000077	177700	177777	052525	125252	170360	151322
000000	000177	177600	000000	063146	052525	007417	026455
000000	000377	177400	000000	073567	052525	007417	026455
000000	000777	177000	177777	104210	125252	170360	151322
000000	001777	176000	177777	114631	125252	170360	151322
000000	003777	174000	000000	125252	052525	007417	026455
000000	007777	170000	177777	135673	125252	170360	151322
000000	017777	160000	000000	146314	052525	007417	026455
000000	037777	140000	177777	156735	125252	170360	151322
000000	077777	100000	177777	167356	125252	170360	151322
000000	177777	000000	000000	177777	052525	007417	026455
PAT 8	PAT 9	PAT 10	PAT 11	PAT 12	PAT 13	PAT 14	PAT 15
077577	000001	177776	172666	077777	177777	000000	177777
077577	000002	177775	155555	137777	177777	000000	177777
077577	000004	177773	172666	157777	177777	177777	000000
077577	000010	177767	155555	167777	177777	177777	000000
077577	000020	177757	172666	173777	177777	177777	000000
077577	000040	177737	155555	175777	177777	177777	000000
077577	000100	177677	172666	176777	177777	177777	000000
077577	000200	177577	155555	177377	177777	177777	000000
077577	000400	177377	172666	177577	177777	177777	000000
077577	001000	176777	155555	177677	177777	177777	000000
077577	002000	175777	172666	177737	177777	177777	000000
077577	004000	173777	155555	177757	177777	177777	000000
077577	010000	167777	172666	177767	177777	177777	000000
077577	020000	157777	155555	177773	177777	177777	000000
077577	040000	137777	172666	177775	177777	177777	000000
077577	100000	077777	155555	177776	177777	177777	000000

8.14 TEST 15 - POWER FAIL TEST

TEST THE ABILITY OF THE RP11E TO SENSE POWER FAILURE AND FOR THE DRIVES TO RETRACT THEIR HEADS. WHEN POWER IS TURNED ON AGAIN THE CYLINDER ADDRESS IS TESTED FOR ZERO. AFTER TYPING THE MESSAGE REQUESTING POWER TO BE TURNED OFF, THE PROGRAM GOES INTO A LOOP READING FROM THE SELECTED DISK DRIVE. AFTER POWER IS RESTORED, MEMORY IS CHECKED TO SEE THAT THE DISK DID NOT TRANSFER ANY DATA TO MEMORY WHILE POWER WAS GOING DOWN.

8.15 TEST 16 - HEAD ALIGNMENT ROUTINE

THIS ROUTINE PERFORMS HEAD SELECTION AS DIRECTED FROM THE KEYBOARD; ALIGNMENT OF THE SELECTED HEAD MAY BE CHECKED OR ADJUSTED.

854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908

909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

8.16 TEST 17 - RP11E/DRIVE CONTROL PANEL SWITCH TEST

THIS TEST TESTS THE 'ENABLE/DISABLE' AND 'READ ONLY' SWITCHES ON THE DRIVE CONTROL PANEL AND TESTS THE 'WRITE LOCKOUT' AND 'LOA' SWITCHES ON THE RP11E CONTROL PANEL.

9. PROGRAM LISTING

```

%
.TITLE MD-11-DZRPY-B, RP11-E FUNCTIONAL LOGIC & R/W TEST
*COPYRIGHT (C) 1975, 1976
*DIGITAL EQUIPMENT CORP.
*MAYNARD, MASS. 01754
*
*PROGRAM BY C. HESS/F. ROEMER
*
*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
*PACKAGE (MAINDEC-11-DZQAC-C1), MAR 24, 1976.
*
.SBTTL OPERATIONAL SWITCH SETTINGS
*
*      SWITCH      USE
*-----
*      15          HALT ON ERROR
*      14          LOOP ON TEST
*      13          INHIBIT ERROR TYPEOUTS
*      11          INHIBIT ITERATIONS
*      10          BELL ON ERROR
*      9           LOOP ON ERROR
*      7           DON'T TYPE DRIVE STATUS WHEN PROGRAM STARTED OR WHEN
*                  ENTERING CONVERSATION MODE
*                  CHANGE HEAD SELECTION IN TEST 16
*                  DON'T USE MEMORY MANAGEMENT IN TEST 14
*                  DISPLAY ALL READ RETRY ATTEMPTS IN TEST 14
*                  PRINT ALL COMPARE ERRORS IN TESTS 13 & 14
*                  INHIBIT DATA COMPARE IN TESTS 13 & 14
*                  BYPASS READ AND DATA COMPARE IN TEST 14
*                  BYPASS WRITE CHECK IN TEST 14
*                  BYPASS WRITE IN TEST 14
*

```

.SBTTL TRAP CATCHER

000000

```

.=0
*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A "+2,HALT"
*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS

```

000174 000174
000174 000000
000176 000000

```

.=174
DISPREG: .WORD 0      ;; SOFTWARE DISPLAY REGISTER
SWREG:   .WORD 0      ;; SOFTWARE SWITCH REGISTER

```

.SBTTL ACT11 HOOKS

;;*****

```

965 ;HOOKS REQUIRED BY ACT11
966 000200 $SVPC= ;SAVE PC
967 000046 =46
968 000046 SENDAD ;;1)SET LOC.46 TO ADDRESS OF SENDAD IN .SEOP
969 000052 =52
970 000052 .WORD 0 ;;2)SET LOC.52 TO ZERO
971 000200 .=$SVPC ;; RESTORE PC
972
973 .SBTTL STARTING ADDRESSES
974
975 000200
976 ;#200 = NORMAL START
977 000200 000137 003654 JMP START1
978 ;#204 = SELECT OPERATING PARAMETERS
979 000204 000137 003676 JMP START2
980 ;#210 = SELECT RP11E ADDRESS
981 000210 000137 003644 JMP START3
982 ;#214 = COMBINATION OF 204 AND 210
983 000214 000137 003666 JMP START4
984
985 .SBTTL BASIC DEFINITIONS
986
987 ;#INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
988 001100 STACK= 1100
989 .EQUIV EMT,ERROR ;;BASIC DEFINITION OF ERROR CALL
990 .EQUIV IOT,SCOPE ;;BASIC DEFINITION OF SCOPE CALL
991
992 ;#MISCELLANEOUS DEFINITIONS
993 000011 HT= 11 ;;CODE FOR HORIZONTAL TAB
994 000012 LF= 12 ;;CODE FOR LINE FEED
995 000015 CR= 15 ;;CODE FOR CARRIAGE RETURN
996 000200 CRLF= 200 ;;CODE FOR CARRIAGE RETURN-LINE FEED
997 177776 PS= 177776 ;;PROCESSOR STATUS WORD
998 .EQUIV PS,PSW
999 177774 STKLM= 177774 ;;STACK LIMIT REGISTER
1000 177772 PIRQ= 177772 ;;PROGRAM INTERRUPT REQUEST REGISTER
1001 177570 DSWR= 177570 ;;HARDWARE SWITCH REGISTER
1002 177570 DDISP= 177570 ;;HARDWARE DISPLAY REGISTER
1003
1004 ;#GENERAL PURPOSE REGISTER DEFINITIONS
1005 000000 R0= %0 ;;GENERAL REGISTER
1006 000001 R1= %1 ;;GENERAL REGISTER
1007 000002 R2= %2 ;;GENERAL REGISTER
1008 000003 R3= %3 ;;GENERAL REGISTER
1009 000004 R4= %4 ;;GENERAL REGISTER
1010 000005 R5= %5 ;;GENERAL REGISTER
1011 000006 R6= %6 ;;GENERAL REGISTER
1012 000007 R7= %7 ;;GENERAL REGISTER
1013 .EQUIV R6,SP ;;STACK POINTER
1014 .EQUIV R7,PC ;;PROGRAM COUNTER
1015
1016 ;#PRIORITY LEVEL DEFINITIONS
1017 000000 PRO= 0 ;;PRIORITY LEVEL 0
1018 000040 PRI= 40 ;;PRIORITY LEVEL 1
1019 000100 PR2= 100 ;;PRIORITY LEVEL 2
1020 000140 PR3= 140 ;;PRIORITY LEVEL 3

```

1021	000200	PR4=	200	::: PRIORITY LEVEL 4
1022	000240	PR5=	240	::: PRIORITY LEVEL 5
1023	000300	PR6=	300	::: PRIORITY LEVEL 6
1024	000340	PR7=	340	::: PRIORITY LEVEL 7

:*"SWITCH REGISTER" SWITCH DEFINITIONS

1025		SW15=	100000
1026		SW14=	40000
1027	100000	SW13=	20000
1028	040000	SW12=	10000
1029	020000	SW11=	4000
1030	010000	SW10=	2000
1031	004000	SW09=	1000
1032	002000	SW08=	400
1033	001000	SW07=	200
1034	000400	SW06=	100
1035	000200	SW05=	40
1036	000100	SW04=	20
1037	000040	SW03=	10
1038	000020	SW02=	4
1039	000010	SW01=	2
1040	000004	SW00=	1
1041	000002	.EQUIV	SW09, SW9
1042	000001	.EQUIV	SW08, SW8
1043		.EQUIV	SW07, SW7
1044		.EQUIV	SW06, SW6
1045		.EQUIV	SW05, SW5
1046		.EQUIV	SW04, SW4
1047		.EQUIV	SW03, SW3
1048		.EQUIV	SW02, SW2
1049		.EQUIV	SW01, SW1
1050		.EQUIV	SW00, SW0

:*DATA BIT DEFINITIONS (BIT00 TO BIT15)

1051		BIT15=	100000
1052		BIT14=	40000
1053		BIT13=	20000
1054		BIT12=	10000
1055	100000	BIT11=	4000
1056	040000	BIT10=	2000
1057	020000	BIT09=	1000
1058	010000	BIT08=	400
1059	004000	BIT07=	200
1060	002000	BIT06=	100
1061	001000	BIT05=	40
1062	000400	BIT04=	20
1063	000200	BIT03=	10
1064	000100	BIT02=	4
1065	000040	BIT01=	2
1066	000020	BIT00=	1
1067	000010	.EQUIV	BIT09, BIT9
1068	000004	.EQUIV	BIT08, BIT8
1069	000002	.EQUIV	BIT07, BIT7
1070	000001	.EQUIV	BIT06, BIT6
1071		.EQUIV	BIT05, BIT5
1072		.EQUIV	BIT04, BIT4
1073			
1074			
1075			
1076			


```

1077 .EQUIV BIT03,BIT3
1078 .EQUIV BIT02,BIT2
1079 .EQUIV BIT01,BIT1
1080 .EQUIV BIT00,BIT0
1081
1082 .#BASIC "CPU" TRAP VECTOR ADDRESSES
1083 000004 ERVVEC= 4 ;: TIME OUT AND OTHER ERRORS
1084 000010 RESVEC= 10 ;: RESERVED AND ILLEGAL INSTRUCTIONS
1085 000014 TBITVEC=14 ;: "T" BIT
1086 000014 TRIVEC= 14 ;: TRACE TRAP
1087 000014 BPTVEC= 14 ;: BREAKPOINT TRAP (BPT)
1088 000020 IOTVEC= 20 ;: INPUT/OUTPUT TRAP (IOT) **SCOPE**
1089 000024 PWRVEC= 24 ;: POWER FAIL
1090 000030 EMTVEC= 30 ;: EMULATOR TRAP (EMT) **ERROR**
1091 000034 TRAPVEC=34 ;: "TRAP" TRAP
1092 000060 TKVEC= 60 ;: TTY KEYBOARD VECTOR
1093 000064 TPVEC= 64 ;: TTY PRINTER VECTOR
1094 000240 PIRQVEC=240 ;: PROGRAM INTERRUPT REQUEST VECTOR
1095 .SBTTL MEMORY MANAGEMENT DEFINITIONS
1096
1097 ;#KT11 VECTOR ADDRESS
1098
1099 000250 MAVEC= 250
1100
1101 ;#KT11 STATUS REGISTER ADDRESSES
1102
1103 177572 SR0= 177572
1104 177574 SR1= 177574
1105 177576 SR2= 177576
1106 172516 SR3= 172516
1107
1108 ;#KERNEL "I" PAGE DESCRIPTOR REGISTERS
1109
1110 172300 KIPDR0= 172300
1111 172302 KIPDR1= 172302
1112 172304 KIPDR2= 172304
1113 172306 KIPDR3= 172306
1114 172310 KIPDR4= 172310
1115 172312 KIPDR5= 172312
1116 172314 KIPDR6= 172314
1117 172316 KIPDR7= 172316
1118
1119 ;#KERNEL "I" PAGE ADDRESS REGISTERS
1120
1121 172340 KIPAR0= 172340
1122 172342 KIPAR1= 172342
1123 172344 KIPAR2= 172344
1124 172346 KIPAR3= 172346
1125 172350 KIPAR4= 172350
1126 172352 KIPAR5= 172352
1127 172354 KIPAR6= 172354
1128 172356 KIPAR7= 172356
1129
1130
1131 ;RP11E REGISTER INDEX EQUATES
1132

```

1133	000000	RPDS=00	
1134	000002	RPER=02	
1135	000004	RPCS=04	
1136	000006	RPWC=06	
1137	000010	RPBA=10	
1138	000012	RPCA=12	
1139	000014	RPOA=14	
1140	000016	RPM1=16	
1141	000020	RPM2=20	
1142	000022	RPM3=22	
1143	000024	SUCA=24	
1144	000026	SIL0=26	
1145			
1146			
1147			
1148	000001	CLEAR=1	; CLEAR THE CONTROLLER
1149	000003	WRTSEK=3	; WRITE WITH IMPLIED SEEK AND HEAD SELECTION
1150	000005	ROSEK=5	; READ WITH IMPLIED SEEK AND HEAD SELECT
1151	000007	WRTCHK=7	; WRITE CHECK WITH IMPLIED SEEK AND HEAD SELECT
1152	000011	SEEK=11	; SEEK
1153	000013	WRITE=13	; WRITE (NO IMPLIED SEEK OR HEAD SELECT)
1154	000015	HOMSEK=15	; HOME SEEK
1155	000017	READ=17	; READ (NO IMPLIED SEEK OR HEAD SELECT)
1156			
1157			
1158			
1159			
1160	000400	SUMP=400	; SELECTED UNIT WRITE PROTECTED
1161	001000	SUFU=1000	; SELECTED UNIT FILE UNSAFE
1162	004000	SUSI=4000	; SELECTED UNIT SEEK INCOMPLETE
1163	020000	SURP03=20000	; SELECTED UNIT IS AN RPO3
1164	040000	SUOL=40000	; SELECTED UNIT IS ONLINE
1165	100000	SURDY=100000	; SELECTED UNIT IS READY
1166			
1167			
1168			
1169	000020	MEX0=20	; EXTENDED MEMORY BIT #1
1170	000040	MEX1=40	; EXTENDED MEMORY BIT #2
1171	000100	IDE=100	; INTERRUPT ON DONE ENABLE BIT
1172	000200	RDY=200	; RP11 CONTROLLER READY BIT
1173	004000	HDR=4000	; HEADER OPERATION BIT
1174	010000	MODE=10000	; MODE BIT
1175	020000	AIE=20000	; ATTENTION INTERRUPT ENABLE BIT
1176	040000	HE=40000	; HARD ERROR BIT
1177	100000	ERR=100000	; ERROR BIT
1178			
1179			

.SBTTL COMMON TAGS

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

1180
1181
1182
1183
1184
1185
1186 001100
1187 001100
1188 001100 000000
1189 001102 000
1190 001103 000
1191 001104 000000
1192 001106 000000
1193 001110 000000
1194 001112 000000
1195 001114 000
1196 001115 001
1197 001116 000000
1198 001120 000000
1199 001122 000000
1200 001124 000000
1201 001126 000000
1202 001130 000000
1203 001132 000000
1204 001134 000
1205 001135 000
1206 001136 000000
1207 001140 177570
1208 001142 177570
1209 001144 177560
1210 001146 177562
1211 001150 177564
1212 001152 177566
1213 001154 000
1214 001155 002
1215 001156 012
1216 001157 000
1217 001160 000000
1218 001162 000000
1219 001164 000000
1220 001166 177607 000377
1221 001172 077
1222 001173 015
1223 001174 000012
1224
1225 001176 000000
1226 001200 000000
1227 001202 000000
1228 001204 000000
1229 001206 000000
1230 001210 000000
1231 001212 001750
1232 001214 000000
1233 001216 000000
1234 001220 000000
1235 001222 000000

.=1100
\$CHTAG: .WORD 0
\$PASS: .WORD 0
\$STNM: .BYTE 0
\$ERFLG: .BYTE 0
\$ICNT: .WORD 0
\$LPADR: .WORD 0
\$LPERR: .WORD 0
\$ERTTL: .WORD 0
\$ITEMB: .BYTE 0
\$ERMAX: .BYTE 1
\$ERRPC: .WORD 0
\$GCADR: .WORD 0
\$BODR: .WORD 0
\$GDAT: .WORD 0
\$BDAT: .WORD 0
\$AUTOB: .BYTE 0
\$INTAG: .BYTE 0
\$SWR: .WORD DSWR
DISPLAY: .WORD DDISP
\$TKS: 177560
\$TKB: 177562
\$TPS: 177564
\$TPB: 177566
\$NULL: .BYTE 0
\$FILLS: .BYTE 2
\$FILLC: .BYTE 12
\$TPFLG: .BYTE 0
\$TMPO: .WORD 0
\$TIMES: 0
\$ESCAPE: 0
\$BELL: .ASCIZ <207><377><377>
\$QUES: .ASCII /?/
\$CRLF: .ASCII <15>
\$LF: .ASCIZ <12>

\$CHKDRV: .WORD 0
BYPASS: .WORD 0
FLAG: .WORD 0
PATNUM: .WORD 0
CNTRLC: .WORD 0
MAXCYL: .WORD 0
TIMEOUT: .WORD 1000.
ICNT: .WORD 0
BUSADR: .WORD 0
STALL: .WORD 0
MMACTV: .WORD 0

START OF COMMON TAGS
CONTAINS PASS COUNT
CONTAINS THE TEST NUMBER
CONTAINS ERROR FLAG
CONTAINS SUBTEST ITERATION COUNT
CONTAINS SCOPE LOOP ADDRESS
CONTAINS SCOPE RETURN FOR ERRORS
CONTAINS TOTAL ERRORS DETECTED
CONTAINS ITEM CONTROL BYTE
CONTAINS MAX. ERRORS PER TEST
CONTAINS PC OF LAST ERROR INSTRUCTION
CONTAINS ADDRESS OF 'GOOD' DATA
CONTAINS ADDRESS OF 'BAD' DATA
CONTAINS 'GOOD' DATA
CONTAINS 'BAD' DATA
RESERVED--NOT TO BE USED
AUTOMATIC MODE INDICATOR
INTERRUPT MODE INDICATOR
ADDRESS OF SWITCH REGISTER
ADDRESS OF DISPLAY REGISTER
TTY KBD STATUS
TTY KBD BUFFER
TTY PRINTER STATUS REG. ADDRESS
TTY PRINTER BUFFER REG. ADDRESS
CONTAINS NULL CHARACTER FOR FILLS
CONTAINS # OF FILLER CHARACTERS REQUIRED
INSERT FILL CHARS. AFTER A "LINE FEED"
"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
USER DEFINED
MAX. NUMBER OF ITERATIONS
ESCAPE ON ERROR ADDRESS
CODE FOR BELL
QUESTION MARK
CARRIAGE RETURN
LINE FEED

ADDRESS OF DRIVE BEING USED
TEST EXIT ADDRESS
INTERNAL PROGRAM FLAG WORD
CURRENTLY SELECTED PATTERN
CONVERSATION MODE FLAG
MAXIMUM CYLINDER ADDRESS FOR THE DRIVE IN USE
TIME ALLOWED FOR AN OPERATION
TEST ITERATION COUNT
CHANGE RP11 ADDRESS FLAG
STALL VALUE
MEMORY MANAGEMENT IS ACTIVE FLAG

1236	001224	000000	000000	ACTMEM: .WORD	0,0	; PHYSICAL ADDRESS OF NEXT READ BUFFER
1237	001230	000000	000000	MAXMEM: .WORD	0,0	; ADDRESS OF LAST MEMORY BANK (FILLED BY \$SIZE)
1238	001234	000000	000000	HIMEM: .WORD	0,0	; PHYSICAL ADDRESS OF LAST MEMORY BANK (FOR TEST 14)
1239	001240	000000	000000	LOMEM: .WORD	0,0	; PHYSICAL ADDRESS OF FIRST BUFFER LOCATION
1240	001244	000000	000000	PAKSIZ: .WORD	0,0	; SECTORS ON THE PACK ON THE DRIVE BEING TESTED
1241	001250	000000		INCSEC: .WORD	0	; SECTOR INCREMENT VALUE
1242	001252	000000		INCRK: .WORD	0	; TRACK INCREMENT VALUE
1243	001254	000000		BLKS14: .WORD	0	; NUMBER SECTORS IN TRANSFER IN TEST 14
1244	001256	000000		WC14: .WORD	0	; WORD COUNT FOR TEST 14
1245	001260	000000		OPRSEL: .WORD	0	; OPERATOR SELECTED ADDRESS FLAG
1246	001262	020000		LMTBK: .WORD	8192.	; LARGEST TRANSFER PERMITTED
1247	001264	000000		MAXWC: .WORD	0	; MAXIMUM WORD COUNT ALLOWED ON THIS SYSTEM
1248	001266	000000		SCYL: .WORD	0	; OPERATOR SELECTED CYLINDER
1249	001270	000000		STRK: .WORD	0	; OPERATOR SELECTED TRACK
1250	001272	000000		SSEC: .WORD	0	; OPERATOR SELECTED SECTOR
1251	001274	000000		RDERR: .WORD	0	; READ RETRY COUNTER
1252	001276	000000		TSTNMS: .WORD	0	; TEST SELECTION WORD. BIT 0 = TEST 0, ETC.
1253	001300	000000		PATRN: .WORD	0	; PATTERN SELECTION WORD FOR TEST 14
1254	001302	000000		MEMSIZ: .WORD	0	; ADDRESS OF HIGHEST NON-MEMORY MANAGEMENT
1255						; MEMORY LOCATION
1256	001304	000000		DRVTYP: .WORD	0	; CONTAINS A BIT FOR EACH RPO3 DRIVE
1257						; ON THE SYSTEM. BIT00 = DRIVE 0, ETC.
1258	001306	000000		DRVMSK: .WORD	0	; DRIVE TEST SELECTION MASK
1259	001310	000000		DRVBAD: .WORD	0	; CONTAINS A BIT FOR THE DRIVE IF THE DRIVE
1260						; GOES OFFLINE OR BECOMES UNSAFE DURING TESTING.
1261						; BIT00 = DRIVE 0, ETC.
1262	001312	000000		DRVSEL: .WORD	0	; CONTAINS A BIT FOR EACH DRIVE TO BE TESTED
1263	001314	000000		RETRY: .WORD	0	; RETRY COUNTER FOR TEST 14
1264	001316	000003		LRETRY: .WORD	3	; RETRY LIMIT FOR TEST 14
1265	001320	000003		CMP.CT: .WORD	3	; NUMBER OF COMPARE ERROR TIMEOUTS
1266	001322	000000		LDRFLG: .WORD	0	; WHEN =0 LOADERS ARE AT TOP OF MEMORY
1267						
1268						;RP11 ADDRESS VALUES
1269						
1270	001324	176710		RPADR: .WORD	176710	; RP11 BUS ADDRESS
1271	001326	000254	000256	RPVEC: .WORD	254,256	; RP11 VECTOR ADDRESS
1272	001332	000240		RPPRIO: .WORD	<5*32.>	; RP11 PRIORITY
1273						
1274						;SAVE THE RP11E REGISTERS HERE
1275						
1276	001334	000000		\$RPDS: .WORD	0	; DRIVE STATUS REGISTER
1277	001336	000000		\$RPER: .WORD	0	; ERROR REGISTER
1278	001340	000000		\$RPCS: .WORD	0	; COMMAND & STATUS REGISTER
1279	001342	000000		\$RPWC: .WORD	0	; WC.0 COUNT REGISTER
1280	001344	000000		\$RPBA: .WORD	0	; BUFFER ADDRESS REGISTER
1281	001346	000000		\$RPBA: .WORD	0	; CURRENT CYLINDER ADDRESS REGISTER
1282	001350	000000		\$RPDA: .WORD	0	; TRACK-SECTOR ADDRESS REGISTER
1283	001352	000000		\$RPM1: .WORD	0	; MAINTENANCE REGISTER #1
1284	001354	000000		\$SUCA: .WORD	0	; SELECTED UNIT CYLINDER ADDRESS REGISTER
1285	001356	000000		\$SILO: .WORD	0	; SILO REGISTER
1286						
1287						;ATTENTION BITS
1288						
1289	001360	001		ATABIT: .BYTE	1	; DRIVE 0
1290	001361	002		.BYTE	2	; DRIVE 1
1291	001362	004		.BYTE	4	; DRIVE 2

1292	001363	010
1293	001364	020
1294	001365	040
1295	001366	100
1296	001367	200

.BYTE	10	:DRIVE 3
.BYTE	20	:DRIVE 4
.BYTE	40	:DRIVE 5
.BYTE	100	:DRIVE 6
.BYTE	200	:DRIVE 7

```

;DRIVE STATUS INDICATORS (DRVSTA=8 BYTES)
;DRVSTA=0 DRIVE NONEXISTENT OR OFFLINE
;DRVSTA>0 DRIVE IS ONLINE
;DRVSTA<0 DRIVE IS UNSAFE

```

1304	001370	000
1305	001371	000
1306	001372	000
1307	001373	000
1308	001374	000
1309	001375	000
1310	001376	000
1311	001377	000

DRVSTA: .BYTE	0	:DRIVE 0
.BYTE	0	:DRIVE 1
.BYTE	0	:DRIVE 2
.BYTE	0	:DRIVE 3
.BYTE	0	:DRIVE 4
.BYTE	0	:DRIVE 5
.BYTE	0	:DRIVE 6
.BYTE	0	:DRIVE 7

;BIT TABLE

1315	001400	000001
1316	001402	000002
1317	001404	000004
1318	001406	000010
1319	001410	000020
1320	001412	000040
1321	001414	000100
1322	001416	000200
1323	001420	000400
1324	001422	001000
1325	001424	002000
1326	001426	004000
1327	001430	010000
1328	001432	020000
1329	001434	040000
1330	001436	100000

BITS: .WORD	BIT00
.WORD	BIT01
.WORD	BIT02
.WORD	BIT03
.WORD	BIT04
.WORD	BIT05
.WORD	BIT06
.WORD	BIT07
.WORD	BIT08
.WORD	BIT09
.WORD	BIT10
.WORD	BIT11
.WORD	BIT12
.WORD	BIT13
.WORD	BIT14
.WORD	BIT15

1332	001440	001500
1333	001442	001540
1334	001444	001600
1335	001446	001640
1336	001450	001700
1337	001452	001740
1338	001454	002000
1339	001456	002040
1340	001460	002100
1341	001462	002140
1342	001464	002200
1343	001466	002240
1344	001470	002300
1345	001472	002340
1346	001474	002400
1347	001476	002440

PAT.PT: .WORD	PAT0
.WORD	PAT1
.WORD	PAT2
.WORD	PAT3
.WORD	PAT4
.WORD	PAT5
.WORD	PAT6
.WORD	PAT7
.WORD	PAT8
.WORD	PAT9
.WORD	PAT10
.WORD	PAT11
.WORD	PAT12
.WORD	PAT13
.WORD	PAT14
.WORD	PAT15

```

;TABLE OF POINTERS WHICH POINT TO THE
;PATTERNS USED BY THE DATA TEST

```

1378
1379
1380
1381 001500 000000
1382 001502 000000
1383 001504 000000
1384 001506 000000
1385 001510 000000
1386 001512 000000
1387 001514 000000
1388 001516 000000
1389 001520 000000
1390 001522 000000
1391 001524 000000
1392 001526 000000
1393
1394 001540 000001
1395 001542 000003
1396 001544 000007
1397 001546 000017
1398 001550 000037
1399 001552 000077
1400 001554 000177
1401 001556 000377
1402 001560 000777
1403 001562 001777
1404 001564 003777
1405 001566 007777
1406 001570 017777
1407 001572 037777
1408 001574 077777
1409 001576 177777
1410
1411 001600 177776
1412 001602 177774
1413 001604 177770
1414 001606 177760
1415 001610 177740
1416 001612 177700
1417 001614 177600
1418 001616 177400
1419 001620 177000
1420 001622 176000
1421 001624 174000
1422 001626 170000
1423 001630 160000
1424 001632 140000
1425 001634 100000
1426 001636 000000
1427
1428 001640 000000
1429 001642 000000

; PATTERNS 0 THRU 15

PAT0: .WORD 000000 ; PATTERN 0
 .WORD 000000
 .WORD 000000
 .WORD 000000
 .WORD 000000
 .WORD 000000
 .WORD 000000
 .WORD 000000
 .WORD 000000
 .WORD 000000
 .WORD 000000
 .WORD 000000
 .WORD 000000
 .WORD 000000
 .WORD 000000
 .WORD 000000
 .WORD 000000
 .WORD 000000

PAT1: .WORD 000001 ; PATTERN 1
 .WORD 000003
 .WORD 000007
 .WORD 000017
 .WORD 000037
 .WORD 000077
 .WORD 000177
 .WORD 000377
 .WORD 000777
 .WORD 001777
 .WORD 003777
 .WORD 007777
 .WORD 017777
 .WORD 037777
 .WORD 077777
 .WORD 177777

PAT2: .WORD 177776 ; PATTERN 2
 .WORD 177774
 .WORD 177770
 .WORD 177760
 .WORD 177740
 .WORD 177700
 .WORD 177600
 .WORD 177400
 .WORD 177000
 .WORD 176000
 .WORD 174000
 .WORD 170000
 .WORD 160000
 .WORD 140000
 .WORD 100000
 .WORD 000000

PAT3: .WORD 000000 ; PATTERN 3
 .WORD 000000

1404	001644	000000	.WORD	000000
1405	001646	177777	.WORD	177777
1406	001650	177777	.WORD	177777
1407	001652	177777	.WORD	177777
1408	001654	000000	.WORD	000000
1409	001656	000000	.WORD	000000
1410	001660	177777	.WORD	177777
1411	001662	177777	.WORD	177777
1412	001664	000000	.WORD	000000
1413	001666	177777	.WORD	177777
1414	001670	000000	.WORD	000000
1415	001672	177777	.WORD	177777
1416	001674	177777	.WORD	177777
1417	001676	000000	.WORD	000000

1418				
1419	001700	000000		
1420	001702	010421	PAT4: .WORD	000000 ;PATTERN 4
1421	001704	021042	.WORD	010421
1422	001706	031463	.WORD	021042
1423	001710	042104	.WORD	031463
1424	001712	052525	.WORD	042104
1425	001714	063146	.WORD	052525
1426	001716	073567	.WORD	063146
1427	001720	104210	.WORD	073567
1428	001722	114631	.WORD	104210
1429	001724	125252	.WORD	114631
1430	001726	135673	.WORD	125252
1431	001730	146314	.WORD	135673
1432	001732	156735	.WORD	146314
1433	001734	167356	.WORD	156735
1434	001736	177777	.WORD	167356

1435				
1436	001740	052525	PAT5: .WORD	052525 ;PATTERN 5
1437	001742	052525	.WORD	052525
1438	001744	052525	.WORD	052525
1439	001746	125252	.WORD	052525
1440	001750	125252	.WORD	125252
1441	001752	125252	.WORD	125252
1442	001754	125252	.WORD	125252
1443	001756	125252	.WORD	125252
1444	001760	125252	.WORD	125252
1445	001762	125252	.WORD	125252
1446	001764	125252	.WORD	125252
1447	001766	125252	.WORD	125252
1448	001770	125252	.WORD	125252
1449	001772	125252	.WORD	125252
1450	001774	125252	.WORD	125252
1451	001776	125252	.WORD	125252

1452	002000	007417	PAT6: .WORD	007417 ;PATTERN 6
1453	002002	007417	.WORD	007417
1454	002004	007417	.WORD	007417
1455	002006	170360	.WORD	007417
1456	002010	170360	.WORD	170360
1457	002012	170360	.WORD	170360
1458	002014	007417	.WORD	170360
1459			.WORD	007417

1460	002016	007417	.WORD	007417
1461	002020	170360	.WORD	170360
1462	002020	170360	.WORD	170360
1463	002020	007417	.WORD	007417
1464	002020	170360	.WORD	170360
1465	002020	007417	.WORD	007417
1466	002032	170360	.WORD	170360
1467	002034	170360	.WORD	170360
1468	002036	007417	.WORD	007417
1469				
1470	002040	026455	.WORD	026455
1471	002042	026455	.WORD	026455
1472	002044	026455	.WORD	026455
1473	002046	151322	.WORD	151322
1474	002050	151322	.WORD	151322
1475	002052	151322	.WORD	151322
1476	002054	026455	.WORD	026455
1477	002056	026455	.WORD	026455
1478	002060	151322	.WORD	151322
1479	002062	151322	.WORD	151322
1480	002064	026455	.WORD	026455
1481	002066	151322	.WORD	151322
1482	002070	026455	.WORD	026455
1483	002072	151322	.WORD	151322
1484	002074	151322	.WORD	151322
1485	002076	026455	.WORD	026455
1486				
1487	002100	077577	.WORD	077577
1488	002102	077577	.WORD	077577
1489	002104	077577	.WORD	077577
1490	002106	077577	.WORD	077577
1491	002110	077577	.WORD	077577
1492	002112	077577	.WORD	077577
1493	002114	077577	.WORD	077577
1494	002116	077577	.WORD	077577
1495	002120	077577	.WORD	077577
1496	002122	077577	.WORD	077577
1497	002124	077577	.WORD	077577
1498	002126	077577	.WORD	077577
1499	002130	077577	.WORD	077577
1500	002132	077577	.WORD	077577
1501	002134	077577	.WORD	077577
1502	002136	077577	.WORD	077577
1503				
1504	002140	000001	.WORD	000001
1505	002142	000002	.WORD	000002
1506	002144	000004	.WORD	000004
1507	002146	000010	.WORD	000010
1508	002150	000020	.WORD	000020
1509	002152	000040	.WORD	000040
1510	002154	000100	.WORD	000100
1511	002156	000200	.WORD	000200
1512	002160	000400	.WORD	000400
1513	002162	001000	.WORD	001000
1514	002164	002000	.WORD	002000
1515	002166	004000	.WORD	004000

PAT7: ;PATTERN 7

PAT8: ;PATTERN 8 (WORST CASE PATTERN)

PAT9: ;PATTERN 9

1516	002170	010000	.WORD	010000
1517	002172	020000	.WORD	020000
1518	002174	040000	.WORD	040000
1519	002176	100000	.WORD	100000
1520				
1521	002200	177776	.WORD	177776
1522	002202	177775	.WORD	177775
1523	002204	177773	.WORD	177773
1524	002206	177767	.WORD	177767
1525	002210	177757	.WORD	177757
1526	002212	177737	.WORD	177737
1527	002214	177677	.WORD	177677
1528	002216	177577	.WORD	177577
1529	002220	177377	.WORD	177377
1530	002222	176777	.WORD	176777
1531	002224	175777	.WORD	175777
1532	002226	173777	.WORD	173777
1533	002230	167777	.WORD	167777
1534	002232	157777	.WORD	157777
1535	002234	137777	.WORD	137777
1536	002236	077777	.WORD	077777
1537				
1538	002240	172666	.WORD	172666
1539	002242	155555	.WORD	155555
1540	002244	172666	.WORD	172666
1541	002246	155555	.WORD	155555
1542	002250	172666	.WORD	172666
1543	002252	155555	.WORD	155555
1544	002254	172666	.WORD	172666
1545	002256	155555	.WORD	155555
1546	002260	172666	.WORD	172666
1547	002262	155555	.WORD	155555
1548	002264	172666	.WORD	172666
1549	002266	155555	.WORD	155555
1550	002270	172666	.WORD	172666
1551	002272	155555	.WORD	155555
1552	002274	172666	.WORD	172666
1553	002276	155555	.WORD	155555
1554				
1555	002300	077777	.WORD	077777
1556	002302	137777	.WORD	137777
1557	002304	157777	.WORD	157777
1558	002306	167777	.WORD	167777
1559	002310	173777	.WORD	173777
1560	002312	175777	.WORD	175777
1561	002314	176777	.WORD	176777
1562	002316	177377	.WORD	177377
1563	002320	177577	.WORD	177577
1564	002322	177677	.WORD	177677
1565	002324	177737	.WORD	177737
1566	002326	177757	.WORD	177757
1567	002330	177767	.WORD	177767
1568	002332	177773	.WORD	177773
1569	002334	177775	.WORD	177775
1570	002336	177776	.WORD	177776
1571				

PAT10: ;PATTERN 10

PAT11: ;PATTERN 11

PAT12: ;PATTERN 12

1572 002340 177777
1573 002342 177777
1574 002344 177777
1575 002346 177777
1576 002350 177777
1577 002352 177777
1578 002354 177777
1579 002356 177777
1580 002360 177777
1581 002362 177777
1582 002364 177777
1583 002366 177777
1584 002370 177777
1585 002372 177777
1586 002374 177777
1587 002376 177777

PAT13: .WORD 177777 ;PATTERN 13
.WORD 177777
.WORD 177777
.WORD 177777
.WORD 177777
.WORD 177777
.WORD 177777
.WORD 177777
.WORD 177777
.WORD 177777
.WORD 177777
.WORD 177777
.WORD 177777
.WORD 177777
.WORD 177777
.WORD 177777

1588
1589 002400 000000
1590 002402 000000
1591 002404 177777
1592 002406 177777
1593 002410 177777
1594 002412 177777
1595 002414 177777
1596 002416 177777
1597 002420 177777
1598 002422 177777
1599 002424 177777
1600 002426 177777
1601 002430 177777
1602 002432 177777
1603 002434 177777
1604 002436 177777

PAT14: .WORD 000000 ;PATTERN 14
.WORD 000000
.WORD 177777
.WORD 177777
.WORD 177777
.WORD 177777
.WORD 177777
.WORD 177777
.WORD 177777
.WORD 177777
.WORD 177777
.WORD 177777
.WORD 177777
.WORD 177777
.WORD 177777
.WORD 177777
.WORD 177777

1605
1606 002440 177777
1607 002442 177777
1608 002444 000000
1609 002446 000000
1610 002450 000000
1611 002452 000000
1612 002454 000000
1613 002456 000000
1614 002460 000000
1615 002462 000000
1616 002464 000000
1617 002466 000000
1618 002470 000000
1619 002472 000000
1620 002474 000000
1621 002476 000000

PAT15: .WORD 177777 ;PATTERN 15
.WORD 177777
.WORD 000000
.WORD 000000
.WORD 000000
.WORD 000000
.WORD 000000
.WORD 000000
.WORD 000000
.WORD 000000
.WORD 000000
.WORD 000000
.WORD 000000
.WORD 000000
.WORD 000000
.WORD 000000
.WORD 000000
.WORD 000000

1622
1623
1624
1625 002500 000
1626 002501 000
1627 002502 000

;DRIVE OPERATION CONTROL BLOCK
DPB: .BYTE 0 ;DRIVE NUMBER
.BYTE 0 ;'MEX' BITS
.BYTE 0 ;OPERATION CODE

1628 002503 000
1629 002504 000000
1630 002506 000000
1631 002510 000000
1632 002512 000
1633 002513 000
1634

SWC: .BYTE 0
SBUF: .WORD 0
SCYL: .WORD 0
SSEC: .BYTE 0
STRK: .BYTE 0

:MODE AND HDR BITS
:WORD COUNT
:BUFFER ADDRESS
:CYLINDER ADDRESS
:SECTOR ADDRESS
:TRACK ADDRESS

1635			.SBTTL	ERROR POINTER TABLE	
1636					
1637					
1638					
1639					
1640					
1641					
1642					
1643					
1644					
1645					
1646					
1647					
1648					
1649	002514		SERRTB:		
1650			;ERROR 1		
1651					
1652	002514	032117	EM1		;RP11 DIDN'T RESPOND TO ADDRESSING
1653	002516	040241	DH1		
1654	002520	042762	DT1		
1655	002522	043442	DF1		
1656					
1657					
1658					
1659	002524	032161	EM2		;CAN'T START READ COMMAND
1660	002526	040247	DH2		
1661	002530	042764	DT2		
1662	002532	043446	DF2		
1663					
1664					
1665					
1666	002534	032212	EM3		; 'CLEAR' COMMAND DIDN'T TERMINATE DATA TRANSFER OPERATION
1667	002536	040247	DH2		
1668	002540	042764	DT2		
1669	002542	043446	DF2		
1670					
1671					
1672					
1673	002544	032303	EM4		;ERROR WRITING TEST SECTOR(S)
1674	002546	040247	DH2		
1675	002550	042764	DT2		
1676	002552	043446	DF2		
1677					
1678					
1679					
1680	002554	032340	EM5		;WRITE CHECK ERROR CHECKING TEST SECTOR(S)
1681	002556	040411	DH5		
1682	002560	043016	DT5		
1683	002562	043456	DF5		
1684					
1685					
1686					
1687	002564	032412	EM6		;EXPECTED 'WCE' DIDN'T OCCUR
1688	002566	040411	DH5		
1689	002570	043016	DT5		
1690	002572	043456	DF5		

```

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

```

```

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

```

```

;SERRTB:
;ERROR 1
;RP11 DIDN'T RESPOND TO ADDRESSING
EM1
DH1
DT1
DF1

;ERROR 2
;CAN'T START READ COMMAND
EM2
DH2
DT2
DF2

;ERROR 3
;'CLEAR' COMMAND DIDN'T TERMINATE DATA TRANSFER OPERATION
EM3
DH2
DT2
DF2

;ERROR 4
;ERROR WRITING TEST SECTOR(S)
EM4
DH2
DT2
DF2

;ERROR 5
;WRITE CHECK ERROR CHECKING TEST SECTOR(S)
EM5
DH5
DT5
DF5

;ERROR 6
;EXPECTED 'WCE' DIDN'T OCCUR
EM6
DH5
DT5
DF5

```

1691				
1692				
1693				
1694	002574	032446	EM7	
1695	002576	040247	DH2	
1696	002600	042764	DT2	
1697	002602	043446	DF2	
1698				
1699				
1700				
1701	002604	000000	0	
1702	002606	000000	0	
1703	002610	000000	0	
1704	002612	000000	0	
1705				
1706				
1707				
1708	002614	032514	EM11	
1709	002616	040563	DH11	
1710	002620	043052	DT11	
1711	002622	043466	DF11	
1712				
1713				
1714				
1715	002624	032607	EM12	
1716				
1717	002626	040644	DH12	
1718	002630	043066	DT12	
1719	002632	043466	DF11	
1720				
1721				
1722				
1723	002634	032740	EM13	
1724				
1725	002636	040247	DH2	
1726	002640	042764	DT2	
1727	002642	043446	DF2	
1728				
1729				
1730				
1731	002644	033063	EM14	
1732	002646	040721	DH14	
1733	002650	043102	DT14	
1734	002652	043466	DF11	
1735				
1736				
1737				
1738	002654	033117	EM15	
1739	002656	040247	DH2	
1740	002660	042764	DT2	
1741	002662	043446	DF2	
1742				
1743				
1744				
1745	002664	033147	EM16	
1746	002666	040776	DH16	

;ERROR 7

;ERROR WRITING LESS THAN A FULL SECTOR

;ERROR 10

;UNUSED

;ERROR 11

;ONE OF 2 WORDS WRITTEN IN THE PARTIAL WRITE DIDN'T COMPARE

;ERROR 12

;NON-ZERO DATA IN PART OF SECTOR WHICH SHOULD HAVE
;BEEN ZERO FILLED DURING PARTIAL WRITE

;ERROR 13

;'EOP' DIDN'T SET DURING A 2 SECTOR WRITE BEGINNING
;AT THE LAST SECTOR OF THE PACK

;ERROR 14

;'ERR' DIDN'T SET WITH 'EOP'

;ERROR 15

;'EOP' DIDN'T CLEAR RPCA

;ERROR 16

;SUCA NOT CORRECT AFTER 'EOP'

1747	002670	043116	DT16	
1748	002672	043542	DF53	
1749				
1750				;ERROR 17
1751				
1752	002674	033204	EM17	;ERROR READING TEST SECTOR
1753	002676	040247	DH2	
1754	002700	042764	DT2	
1755	002702	043446	DF2	
1756				
1757				;ERROR 20
1758				
1759	002704	033236	EM20	;CONTENTS OF FIRST SECTOR OF PACK CHANGED AFTER
1760				;FORCING 'EOP' DURING WRITE
1761	002706	040563	DH11	
1762	002710	043052	DT11	
1763	002712	043466	DF11	
1764				
1765				;ERROR 21
1766				
1767	002714	033351	EM21	; 'PROG' ERROR DIDN'T SET WHEN READ COMMAND ISSUED
1768				; WHILE CONTROLLER BUSY
1769	002716	040721	DH14	
1770	002720	043102	DT14	
1771	002722	043466	DF11	
1772				
1773				;ERROR 22
1774				
1775	002724	033461	EM22	; 'ERR' DIDN'T SET WITH 'PROG'
1776	002726	040721	DH14	
1777	002730	043102	DT14	
1778	002732	043466	DF11	
1779				
1780				;ERROR 23
1781				
1782	002734	033516	EM23	;ERROR ATTEMPTING TO MISFORMAT SECTOR 0
1783	002736	040247	DH2	
1784	002740	042764	DT2	
1785	002742	043446	DF2	
1786				
1787				;ERROR 24
1788				
1789	002744	033565	EM24	;ERROR VERIFYING THE MISFORMATTED TEST HEADER
1790	002746	040721	DH14	
1791	002750	043102	DT14	
1792	002752	043466	DF11	
1793				
1794				;ERROR 25
1795				
1796	002754	033642	EM25	;MISFORMATTED TEST HEADER IS NOT CORRECT
1797	002756	041113	DH25	
1798	002760	043130	DT25	
1799	002762	043472	DF25	
1800				
1801				;ERROR 26
1802				

K03

MD-11-DZRPY-B, RP11-E FUNCTIONAL LOGIC & R/W TEST
DZRPYB.CMB ERROR POINTER TABLE

MACY11 27(732) 02-NOV-76 14:50 PAGE 37

1803	002764	033712	EM26	; 'HNF' DIDN'T SET WHEN SEARCHING FOR MISFORMATTED SECTOR
1804	002766	040247	DH2	
1805	002770	042764	DT2	
1806	002772	043446	DF2	
1807				
1808				; ERROR 27
1809				
1810	002774	034002	EM27	; 'DSK' NOT SET WITH 'HNF'
1811	002776	040721	DH14	
1812	003000	043102	DT14	
1813	003002	043466	DF11	
1814				
1815				; ERROR 30
1816				
1817	003004	034033	EM30	; 'HE' DIDN'T SET WITH 'HNF'
1818	003006	040721	DH14	
1819	003010	043102	DT14	
1820	003012	043466	DF11	
1821				
1822				; ERROR 31
1823				
1824	003014	034066	EM31	; 'ERR' NOT SET WITH 'HNF'
1825	003016	040721	DH14	
1826	003020	043102	DT14	
1827	003022	043466	DF11	
1828				
1829				; ERROR 32
1830				
1831	003024	034117	EM32	; ERROR WHILE RESTORING MISFORMATTED TEST HEADER
1832	003026	040247	DH2	
1833	003030	042764	DT2	
1834	003032	043446	DF2	
1835				
1836				; ERROR 33
1837				
1838	003034	034176	EM33	; CONTROLLER DIDN'T BECOME BUSY WHEN READ COMMAND ; ISSUED TO SEEKING DRIVE
1839				
1840	003036	040247	DH2	
1841	003040	042764	DT2	
1842	003042	043446	DF2	
1843				
1844				; ERROR 34
1845				
1846	003044	034307	EM34	; ERROR DURING READ COMMAND WHICH WAS ISSUED TO SEEKING DRIVE
1847	003046	040247	DH2	
1848	003050	042764	DT2	
1849	003052	043446	DF2	
1850				
1851				; ERROR 35
1852				
1853	003054	034403	EM35	; DATA INCORRECT FROM HEADER READ COMMAND ISSUED ; TO SEEKING DRIVE
1854				
1855	003056	041113	DH25	
1856	003060	043130	DT25	
1857	003062	043472	DF25	
1858				

1859				;ERROR 36	
1860					
1861	003064	034505	EM36		; 'NXME' DIDN'T SET WHEN LOCATION 760000 ADDRESSED
1862	003066	040247	DH2		
1863	003070	042764	DT2		
1864	003072	043446	DF2		
1865					
1866				;ERROR 37	
1867					
1868	003074	034566	EM37		; 'HE' DIDN'T SET WITH 'NXME'
1869	003076	040721	DH14		
1870	003100	043102	DT14		
1871	003102	043466	DF11		
1872					
1873				;ERROR 40	
1874					
1875	003104	034622	EM40		; 'ERR' DIDN'T SET WITH 'NXME'
1876	003106	040721	DH14		
1877	003110	043102	DT14		
1878	003112	043466	DF11		
1879					
1880				;ERROR 41	
1881					
1882	003114	034657	EM41		;ERROR FORMATTING TRACK 0
1883	003116	040247	DH2		
1884	003120	042764	DT2		
1885	003122	043446	DF2		
1886					
1887				;ERROR 42	
1888					
1889	003124	034710	EM42		;ERROR READING THE HEADER FROM THE TEST SECTOR
1890	003126	040247	DH2		
1891	003130	042764	DT2		
1892	003132	043446	DF2		
1893					
1894				;ERROR 43	
1895					
1896	003134	034766	EM43		;SECTOR FIELD FROM HEADER IS NOT CORRECT
1897	003136	041202	DH43		
1898	003140	043150	DT43		
1899	003142	043502	DF43		
1900					
1901				;ERROR 44	
1902					
1903	003144	035036	EM44		;ERROR WRITING SECTOR ADDRESS IN DATA FIELD
1904	003146	040247	DH2		
1905	003150	042764	DT2		
1906	003152	043446	DF2		
1907					
1908				;ERROR 45	
1909					
1910	003154	035111	EM45		; 'SOT' OR SECTOR ADDRESS REGISTER IS NOT CORRECT
1911	003156	041426	DH45		
1912	003160	043206	DT45		
1913	003162	043516	DF45		
1914					

1915				;ERROR 46	
1916					
1917	003164	035171	EM46		;DATA NOT CORRECT FOR SECTOR READ
1918	003166	041620	DM46		
1919	003170	043232	DT46		
1920	003172	043466	DF11		
1921					
1922				;ERROR 47	
1923					
1924	003174	035242	EM47		;SECTOR CONTENTS WRONG, DATA SHOULD BE THE
1925					;TRACK NUMBER OF THE CURRENTLY SELECTED HEAD
1926	003176	041755	DM47		
1927	003200	043246	DT47		
1928	003202	043526	DF47		
1929					
1930				;ERROR 50	
1931					
1932	003204	035371	EM50		;ERROR AFTER 2 WORD READ STARTING AT LOC 177776
1933	003206	040247	DM2		
1934	003210	042764	DT2		
1935	003212	043446	DF2		
1936					
1937				;ERROR 51	
1938					
1939	003214	035450	EM51		; 'MEX0' DIDN'T SET AFTER 2 WORD READ STARTING
1940					; AT LOC 177776
1941	003216	042071	DM51		
1942	003220	043272	DT51		
1943	003222	043536	DF51		
1944					
1945				;ERROR 52	
1946					
1947	003224	035544	EM52		; 'MEX1' SET AFTER 2 WORD READ STARTING AT LOC 177776
1948	003226	042071	DM51		
1949	003230	043272	DT51		
1950	003232	043536	DF51		
1951					
1952				;ERROR 53	
1953					
1954	003234	035630	EM53		; CONTENTS OF LOC 200000 WRONG AFTER 2 WORD READ
1955					; STARTING AT LOC 177776
1956	003236	041113	DM25		
1957	003240	043130	DT25		
1958	003242	043542	DF53		
1959					
1960				;ERROR 54	
1961					
1962	003244	035737	EM54		;ERROR AFTER 2 WORD READ STARTING AT LOC 377776
1963	003246	040247	DM2		
1964	003250	042764	DT2		
1965	003252	043446	DF2		
1966					
1967				;ERROR 55	
1968					
1969					
1970	003254	036016	EM55		; 'MEX0' DIDN'T CLEAR AFTER 2 WORD STARTING AT LOC 377776

1971	003256	042071	DH51	
1972	003260	043272	DT51	
1973	003262	043536	DF51	
1974				
1975				;ERROR 56
1976				
1977	003264	036106	EM56	; 'MEX1' DIDN'T SET AFTER 2 WORD READ STARTING AT LOC 377776
1978	003266	042071	DH51	
1979	003270	043272	DT51	
1980	003272	043536	DF51	
1981				
1982				;ERROR 57
1983				
1984	003274	036201	EM57	;CONTENTS OF LOC 400000 WRONG AFTER 2 WORD READ
1985				;STARTING AT LOCATION 377776
1986	003276	041113	DH25	
1987	003300	043130	DT25	
1988	003302	043542	DF53	
1989				
1990				;ERROR 60
1991				
1992	003304	036315	EM60	;ERROR RETURNING SECTOR TO PDP-11 MODE USING A
1993				;2 WORD WRITE
1994	003306	040247	DH2	
1995	003310	042764	DT2	
1996	003312	043446	DF2	
1997				
1998				;ERROR 61
1999				
2000	003314	036411	EM61	;ERROR WRITING TEST SECTOR(S)
2001	003316	040411	DH5	
2002	003320	043016	DT5	
2003	003322	043456	DF5	
2004				
2005				;ERROR 62
2006				
2007	003324	036446	EM62	;ERROR READING TEST SECTOR(S)
2008	003326	040411	DH5	
2009	003330	043016	DT5	
2010	003332	043456	DF5	
2011				
2012				;ERROR 63
2013				
2014	003334	036503	EM63	;DRIVE DIDN'T RETURN TO CYL 0 FROM THE MAXIMUM
2015				;CYL ON CONTROLLER POWER FAIL
2016	003336	042166	DH63	
2017	003340	043312	DT63	
2018	003342	043546	DF63	
2019				
2020				;ERROR 64
2021				
2022	003344	036617	EM64	;CONTENTS OF MEMORY CHANGED DURING POWER
2023				;FAIL WHILE READING THE DISK
2024	003346	040644	DH12	
2025	003350	043066	DT12	
2026	003352	043466	DF11	

2027				
2028				;ERROR 65
2029				
2030	003354	036724	EM65	; 'SUOL' SET WITH DRIVE DISABLED
2031	003356	040721	DH14	
2032	003360	043102	DT14	
2033	003362	043466	DF11	
2034				
2035				;ERROR 66
2036				
2037	003364	036763	EM66	; 'SUOL' NOT SET WITH DRIVE ENABLED
2038	003366	040721	DH14	
2039	003370	043102	DT14	
2040	003372	043466	DF11	
2041				
2042				;ERROR 67
2043				
2044	003374	037025	EM67	; 'SUMP' NOT SET WITH DRIVE SET TO 'READ ONLY'
2045	003376	040721	DH14	
2046	003400	043102	DT14	
2047	003402	043466	DF11	
2048				
2049				;ERROR 70
2050				
2051	003404	037102	EM70	; 'SUMP' SET WITH DRIVE SET TO 'READ/WRITE'
2052	003406	040721	DH14	
2053	003410	043102	DT14	
2054	003412	043466	DF11	
2055				
2056				;ERROR 71
2057				
2058	003414	037154	EM71	; 'SUMP' SET WITH RP11 'WRITE LOCKOUT' SET, CYLINDER
2059				; LOA'S = 0, AND RPCA = 0
2060	003416	042253	DH71	
2061	003420	043330	DT71	
2062	003422	043546	DF63	
2063				
2064				;ERROR 72
2065				
2066	003424	037270	EM72	; 'SUMP' SET WITH RPCA = 2 & CYL LOA'S = 0
2067	003426	042253	DH71	
2068	003430	043330	DT71	
2069	003432	043546	DF63	
2070				
2071				;ERROR 73
2072				
2073	003434	037341	EM73	; 'SUMP' NOT SET; RPCA = VALUE IN CYL LOA'S
2074	003436	042253	DH71	
2075	003440	043330	DT71	
2076	003442	043546	DF63	
2077				
2078				;ERROR 74
2079				
2080	003444	037413	EM74	; 'SUMP' SET WITH RPCA ONE GREATER THAN CYL LOA VALUE
2081	003446	042253	DH71	
2082	003450	043330	DT71	

2093	003452	043546	DF63	
2094				; ERROR 75
2095	003454	037477	EM75	; 'SUMP' SET WITH DRIVE LOA SWITCHES EQUAL SELECTED DRIVE
2096	003456	040721	DH14	
2097	003460	043102	DT14	
2098	003462	043466	DF11	
2099				; ERROR 76
2100	003464	037572	EM76	; DRIVE HAS GONE OFFLINE
2101	003466	040247	DH2	
2102	003470	042764	DT2	
2103	003472	043446	DF2	
2104				; ERROR 77
2105	003474	037621	EM77	; DRIVE IS UNSAFE
2106	003476	040247	DH2	
2107	003500	042764	DT2	
2108	003502	043446	DF2	
2109				; ERROR 100
2110	003504	037641	EM100	; DRIVE TIMED OUT
2111	003506	040247	DH2	
2112	003510	042764	DT2	
2113	003512	043446	DF2	
2114				; ERROR 101
2115	003514	037661	EM101	; CONTROLLER TIMED OUT
2116	003516	040247	DH2	
2117	003520	042764	DT2	
2118	003522	043446	DF2	
2119				; ERROR 102
2120	003524	037706	EM102	; DATA COMPARE ERROR
2121	003526	042340	DH102	
2122	003530	043346	DT102	
2123	003532	043552	DF102	
2124				; ERROR 103
2125	003534	037731	EM103	; DATA COMPARE ERROR (MEMORY MANAGEMENT ENABLED)
2126	003536	042340	DH102	
2127	003540	043374	DT103	
2128	003542	043562	DF103	
2129				; ERROR 104
2130	003544	000000	0	; DATA COMPARE ERROR DETAIL LINE
2131	003546	000000	0	
2132	003550	043430	DT104	

2139	003552	043576	DF104	
2140				
2141				;ERROR 105
2142				
2143	003554	000000	0	;DATA COMPARE ERROR SUMMARY LINE
2144	003556	042643	OH105	
2145	003550	043436	DT105	
2146	003562	043442	DF1	
2147				
2148				;ERROR 106
2149				
2150	003564	040010	EM106	; 'ERR' DIDN'T SET WITH 'WCE'
2151	003566	040721	OH14	
2152	003570	043112	DT14	
2153	003572	043466	DF11	
2154				
2155				;ERROR 107
2156				

2157	003574	040044	EM107	'HE' DIDN'T SET WITH 'PROG'
2158	003576	040721	DH14	
2159	003600	043102	DT14	
2160	003602	043466	DF11	
2161				
2162				
2163				
2164	003604	000000	0	
2165	003606	042671	DH110	
2166	003610	000000	0	
2167	003612	000000	0	
2168				
2169				
2170				
2171	003614	000000	0	
2172	003616	042724	DH111	
2173	003620	043440	DT111	
2174	003622	043442	DF1	
2175				
2176				
2177				
2178	003624	040100	EM112	
2179	003626	040247	DH2	
2180	003630	042764	DT2	
2181	003632	043446	DF2	
2182				
2183				
2184				
2185	003634	040136	EM113	
2186	003636	040721	DH14	
2187	003640	043102	DT14	
2188	003642	043466	DF11	
2189				

;ERROR 110

;UNSUCCESSFUL AFTER 3 RETRIES

;ERROR 111

;SUCCESSFUL AFTER 'N' RETRIES

;ERROR 112

;DRIVE UNSAFE AFTER POWER FAIL

;ERROR 113

;'SUMP' NOT SET WITH DRIVE LOA SWITCHES GREATER THAN SELECTED DR

2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245

```

;*****
.SBTTL START OF PROGRAM
;*****

003644 012737 177777 001216 START3: MOV    #-1,BUSADR ;SET BUSADR FLAG
003652 000402                BR      STRT1A
003654 005037 001216          START1: CLR    #BUSADR ;CLR BUSADR FLAG
003660 005037 001206          STRT1A: CLR    #CNTRLC ;NO CONVERSATION MODE
003664 000411                BR      START
003666 012737 177777 001216 START4: MOV    #-1,BUSADR ;SET BUSADR FLAG
003674 000402                BR      STRT2A
003676 005037 001216          START2: CLR    #BUSADR ;CLR BUSADR FLAG
003702 012737 177777 001206 STRT2A: MOV    #-1,CNTRLC ;SET CONVERSATION MODE FLAG
003710 000005          START:  RESET ;CLEAR THE BUS
.SBTTL INITIALIZE THE COMMON TAGS
;CLEAR THE COMMON TAGS ($CMTAG) AREA
MOV    #SCMTAG,R6 ;FIRST LOCATION TO BE CLEARED
CLR    (R6)+ ;CLEAR MEMORY LOCATION
CMP    #SWR,R6 ;;DONE?
BNE    -6 ;;LOOP BACK IF NO
MOV    #STACK,SP ;;SETUP THE STACK POINTER
;INITIALIZE A FEW VECTORS
MOV    #SCOPE,@IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
MOV    #340,@IOTVEC+2 ;;LEVEL 7
MOV    #ERROR,@EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
MOV    #340,@EMTVEC+2 ;;LEVEL 7
MOV    #TRAP,@TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
MOV    #340,@TRAPVEC+2 ;;LEVEL 7
MOV    #ENDCT,SEOPCT ;;SETUP END-OF-PROGRAM COUNTER
CLR    $TIMES ;;INITIALIZE NUMBER OF ITERATIONS
CLR    $ESCAPE ;;CLEAR THE ESCAPE ON ERROR ADDRESS
MOV    #1,$SERMAX ;;ALLOW ONE ERROR PER TEST
MOV    #,$SLPROR ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
MOV    #,$SLPERR ;;SETUP THE ERROR LOOP ADDRESS
;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
MOV    @ERRVEC,-(SP) ;;SAVE ERROR VECTOR
MOV    #64,$ERRVEC ;;SET UP ERROR VECTOR
MOV    #DSWR,$SWR ;;SETUP FOR A HARDWARE SWICH REGISTER
MOV    #DISP,$DISPLAY ;;AND A HARDWARE DISPLAY REGISTER
CMP    #-1,$SWR ;;TRY TO REFERENCE HARDWARE SWR
BNE    66$ ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
;AND THE HARDWARE SWR IS NOT = -1
BR     65$ ;;BRANCH IF NO TIMEOUT
MOV    #65$,(SP) ;;SET UP FOR TRAP RETURN
RTI
MOV    #SWREG,$SWR ;;POINT TO SOFTWARE SWR
MOV    #DISPREG,$DISPLAY ;;RESTORE ERROR VECTOR
MOV    (SP)+,@ERRVEC
INC    #-1 ;FIRST START ?

```

2246	004130	001002				BNE	STARTS		:BR IF NOT
2247	004132	104401	043602			TYPE	TITLE		:TYPE THE PROGRAM'S MAINDEC NUMBER AND NAME
2248	004136	004737	020374			STARTS: JSR	PC,STKINT		:SETUP THE TTY KEYBOARD
2249						.SBTTL	GET VALUE FOR SOFTWARE SWITCH REGISTER		
2250	004142	005737	000042			TST	#42		:ARE WE RUNNING UNDER XXDP/ACT?
2251	004146	001006				BNE	645		:BRANCH IF YES
2252	004150	023727	001140	000176		CMP	SWR,#SWREG		:SOFTWARE SWITCH REG SELECTED?
2253	004156	001005				BNE	655		:BRANCH IF NO
2254	004160	104406				GTSWR			:GET SOFT-SWR SETTINGS
2255	004162	000403				BR	655		
2256	004164	112737	000001	001134	645:	MOV	#1,SAUTOB		:;SET AUTO-MODE INDICATOR
2257	004172				655:				
2258	004176	004737	023404			JSR	PC,RESLDR		:RECALL THE LOADER
2259	004176	004737	025624			JSR	PC,SIZMEM		:SET MEMORY SIZE VARIABLES
2260	004202	005037	001112			CLR	\$ERTTL		:CLEAR TOTAL ERROR COUNT
2261	004206	005037	001260			CLR	OPRSEL		:CLEAR OPERATOR SELECTED ADDRESS FLAG
2262	004212	005037	001266			CLR	SCYL		:OPERATOR SPECIFIED CYLINDER
2263	004216	005037	001270			CLR	STRK		:OPERATOR SPECIFIED TRACK
2264	004222	005037	001272			CLR	SSEC		:OPERATOR SPECIFIED SECTOR
2265	004226	013737	001264	001256		MOV	MAXWC,WC14		:ASSUME THAT MAXIMUM WORD COUNT LESS THAN BK
2266	004234	023737	001264	001262		CMP	MAXWC,LMTBK		:MAXIMUM WORD COUNT GREATER THAN LIMIT
2267	004242	101403				BLOS	IS		:BR IF NOT
2268	004244	013737	001262	001256		MOV	LMTBK,WC14		:USE AN BK MAXIMUM
2269	004252	012737	177777	001300	15:	MOV	#177777,PATTRN		:ENABLE ALL PATTERNS
2270	004260	012737	017777	001276		MOV	#17777,†STNMS		:SELECT TESTS 0 - 14
2271	004266	005037	177776			CLR	PSW		:ENSURE THE PRIORITY = 0
2272	004272	005037	001310			CLR	DRVBD		:CLEAR OFFLINE/UNSAFE DRIVE BITS
2273	004276	005037	001162			CLR	\$TIMES		:INITIALIZE NUMBER OF ITERATIONS
2274	004302	005037	001164			CLR	\$ESCAPE		:CLEAR THE ESCAPE ON ERROR ADDRESS
2275	004306	012737	016606	001200		MOV	#SEOP,BYPASS		:INITIAL BYPASS ADDRESS
2276	004314	012737	000001	001104		MOV	#1,\$ICNT		:PRESET ITERATION COUNT TO 1
2277	004322	112737	000001	001115		MOVB	#1,\$ERMAX		:ALLOW ONE ERROR PER TEST
2278	004330	012737	004330	001106		MOV	#.,\$LPADR		:INITIALIZE THE LOOP ADDRESS FOR SCOPE
2279	004336	012737	004336	001110		MOV	#.,\$LPERR		:SETUP THE ERROR LOOP ADDRESS
2280	004344	005037	001222			CLR	MMACTV		:CLEAR MEMORY MANAGEMENT FLAG
2281	004350	005037	177776			START6: CLR	PS		:CLEAR THE PROCESSOR PRIORITY
2282	004354	004737	026074			JSR	PC,GETADR		:CHECK THE RP11 ADDRESS
2283	004360	004737	022536			JSR	PC,RPINIT		:FIND OUT WHICH DRIVES ARE ON SYSTEM
2284	004364	012737	000312	001210		MOV	#202.,MAXCYL		:ASSUME RPO2'S
2285	004372	005737	001304			TST	DRVTYP		:WHICH DRIVES ?
2286	004376	001403				BEQ	25		:BR IF THEY REALLY WERE RPO2'S
2287	004400	012737	000625	001210		MOV	#405.,MAXCYL		:SET MAX CYLINDER FOR RPO3'S
2288	004406	005227	177777		25:	INC	#-1		:FIRST START ?
2289	004412	001404				BEQ	95		:BR IF IT IS
2290	004414	032777	000200	174516		BIT	#SW07,\$SWR		:BYPASS DRIVE STATUS TYPEOUT ?
2291	004422	001044				BNE	START7		:BR IF YES
2292	004424	104401	027162		95:	TYPE	DRSTAT		: 'DRIVE STATUS'
2293	004430	005001				CLR	R1		:CLEAR TABLE POINTER
2294	004432	012702	000010			MOV	#8.,R2		:COUNTER
2295	004436				35:				
2296	004436	010146				MOV	R1,-(SP)		:SAVE R1 FOR TYPEOUT
2297									:TYPE DRIVE NUMBER
2298	004440	104403				TYPOS			:GO TYPE--OCTAL ASCII
2299	004442	001				.BYTE	1		:TYPE 1 DIGIT(S)
2300	004443	000				.BYTE	0		:SUPPRESS LEADING ZEROS
2301	004444	105761	001370			TSTB	DRVSTA(R1)		:CHECK DRIVE'S STATUS

2302	004450	001420			BEQ	7\$		DRIVE IS OFFLINE OR NON-EXISTENT
2303	004452	100403			BMI	4\$		DRIVE IS UNSAFE
2304	004454	104401	027205		TYPE	ONLINE		'ONLINE'
2305	004460	000402			BR	5\$		
2306	004462	104401	027222	4\$:	TYPE	UNSAFE		'UNSAFE'
2307	004466	136137	001360	001304	5\$:	BITB	ATABIT(R1),DRVTYP	;SEE WHICH DRIVE TYPE
2308	004474	001003			BNE	6\$		BR IF RPO3
2309	004476	104401	027253		TYPE	RPO2		'RPO2'
2310	004502	000405			BR	8\$		
2311	004504	104401	027262	6\$:	TYPE	RPO3		'RPO3'
2312	004510	000402			BR	8\$		
2313	004512	104401	027236	7\$:	TYPE	,OFFLIN		'OFFLINE'
2314	004516	104401	001173	8\$:	TYPE	\$CRLF		CR-LF
2315	004522	005201			INC	R1		INCREMENT TABLE POINTER
2316	004524	005302			DEC	R2		DECREMENT THE COUNTER
2317	004526	001343			BNE	3\$		CONTINUE
2318	004530	104401	001173		TYPE	\$CRLF		CR-LF
2319	004534	005737	001206	START7:	TST	CNTRLC		CONVERSATION MODE ?
2320	004540	001403			BEQ	1\$		NO--BRANCH
2321	004542	004737	025240		JSR	PC,ENTPRM		YES--GET PARAMETERS
2322	004546	000416			BR	5\$		GO TYPE DRIVES TO BE TESTED
2323	004550	005037	001312	1\$:	CLR	DRVSEL		NO DRIVES SELECTED
2324	004554	005000			CLR	R0		DETERMINE THE DRIVES THAT
2325	004556	012701	000001		MOV	#1,R1		ARE AVAILABLE FOR TESTING
2326	004562	105760	001370	2\$:	TSTB	DRVSTA(R0)		DRIVE ONLINE ?
2327	004566	003403			BLE	3\$		BR IF NOT
2328	004570	156037	001360	001312	BISB	ATABIT(R0),DRVSEL		;SET SELECTION BIT FOR DRIVE
2329	004576	005200			INC	R0		INCREMENT DRIVE ADDRESS
2330	004600	106301			R1			COUNT
2331	004602	001367			BNE	2\$		BR IF NOT ALL DRIVES CHECKED
2332	004604	104401	027271	5\$:	TYPE	DRVTST		'DRIVES TO BE TESTED'
2333	004610	005037	016774		CLR	\$ENDCT		DETERMINE PASSES TO MAKE AND
2334	004614	005000			CLR	R0		THE DRIVES TO BE TESTED
2335	004616	013701	001312		MOV	DRVSEL,R1		ANY DRIVES SELECTED?
2336	004622	001004			BNE	6\$		YES--BRANCH
2337	004624	104401	027316		TYPE	NONE		'NONE'
2338	004630	000137	016606		JMP	#SEOP		GO TO END OF PROGRAM
2339	004634	005737	001206	6\$:	TST	CNTRLC		CONVERSATION MODE START ?
2340	004640	001011			BNE	7\$		BR IF NOT
2341	004642	005737	000042		TST	42		UNDER MONITOR CONTROL ?
2342	004646	001406			BEQ	7\$		BR IF NOT
2343	004650	122737	000007	000041	CMPS	#7,41		LOADED BY RPO2/RPO3 ?
2344	004656	001002			BNE	7\$		BR IF NOT
2345	004660	006201			ASR	R1		EXCLUDE DRIVE 0 FROM TESTING
2346	004662	000413			BR	8\$		CHECK ON THE OTHER DRIVES
2347	004664	006201		7\$:	ASR	R1		REPORT THE DRIVES TO BE TESTED
2348	004666	103011			BCC	8\$		BR IF DRIVE IS NOT SELECTED
2349	004670	005237	016774		INC	\$ENDCT		GIVE THIS DRIVE A PASS
2350	004674	010046			MOV	R0,-(SP)		SAVE R0 FOR TYPEOUT
2351								TYPE THE DRIVE NUMBER
2352	004676	104403			TYPOS			GO TYPE--OCTAL ASCII
2353	004700	001			.BYTE	1		TYPE 1 DIGIT(S)
2354	004701	000			.BYTE	0		SUPPRESS LEADING ZEROS
2355	004702	005701			TST	R1		MORE DRIVES?
2356	004704	001404			BEQ	9\$		NO--BRANCH
2357	004706	104401	027323		TYPE	,COMMA		' , '


```

2358 004712 005200          BS:   INC   R0      ; INCREMENT DRIVE NUMBER
2359 004714 000763          BR     7$      ; CONTINUE
2360 004716 013737 016774 016766 9$:   MOV   SENDCT,SEOPCT ; SETUP TEST COUNT
2361 004724 005037 001206          CLR   CNTRLC   ; CLEAR CONVERSATION MODE FLAG
2362 004730 005037 001176          RSTRT1: CLR  CHKDRV ; INITIALIZE THE CHECK DRIVE KEY
2363 004734 012737 000001 001306  MOV   #1,DRVMSK ; START TO CHECK DESIRED DRIVES
2364 004742 033737 001306 001312 RSTRT2: BIT  DRVMSK,DRVSEL ; IS THIS DRIVE SELECTED?
2365 004750 001011          RSTRT3: BNE  DRVOK  ; YES--GO CHECK IF DRIVE IS READY FOR TESTING
2366 004752 005237 001176          RESTART: INC  CHKDRV ; MOVE TO NEXT DRIVE NUMBER
2367 004756 106337 001306          ASLB  DRVMSK   ; POSITION THE MASK
2368 004762 103367          BCC   RSTRT2  ; BR IF MORE DRIVES
2369 004764 043737 001310 001312  BIC   DRVBA0,DRVSEL ; CLEAR SELECTION BITS FOR ANY OFFLINE/UNSAFE
2370                                DRIVES
2371 004772 000756          BR     RSTRT1 ; CONTINUE WITH CYCLE
2372 004774 113737 001176 002500 DRVOK: MOV#  CHKDRV,DPB ; PICKUP THE DRIVE NUMBER
2373 005002 104401 027325          TYPE  TSTING  ; 'TESTING WITH DRIVE '
2374 005006 013746 001176          MOV   CHKDRV,-(SP) ; SAVE CHKDRV FOR TYPEOUT
2375                                ; TYPE THE DRIVE NUMBER
2376 005012 104403          TYPOS ; GO TYPE--OCTAL ASCII
2377 005014 001          .BYTE 1 ; TYPE 1 DIGIT(S)
2378 005015 000          .BYTE 0 ; SUPPRESS LEADING ZEROS
2379 005016 104401 001173          TYPE  $CR-LF ; CR-LF
2380 005022 013704 001324          MOV   RPA0R,R4 ; RP11 ADDRESS
2381 005026 004737 025606          JSR   PC,CLAP ; CLEAR THE RP11
2382 005032 112737 000015 002502  MOV#  #HOMSEK,DPB+2 ; HOME SEEK COMMAND
2383 005040 004737 022740          JSR   PC,RP11 ; START THE COMMAND
2384 005044 004737 023500          JSR   PC,DRVROY ; WAIT FOR THE DRIVE TO FINISH
2385 005050 000137 005054          JMP   TST0 ; START THE PROGRAM
2386
2387          .SBTTL  ##### TESTS #####
2388
2389
2390          ; *****
2391          ; *TEST 0 TEST 'CLEAR' TERMINATION
2392
2393          ; *VERIFY THAT 'CLEAR' TERMINATES AN OPERATION AND CAUSES CONTROLLER
2394          ; *READY TO SET.
2395
2396          ; *****
2397          TST0:
2398          BIT   BITS+(<STN*2>),TSTNMS ; IS THIS TEST SELECTED
2399          BNE  .+6 ; BR IF YES
2400          JMP  TST1 ; GO TO THE NEXT TEST
2401          MOV  #TEST0,$LPERR ; SETUP THE ERROR LOOP ADDRESS
2402          MOV  #TST0,$LPADR ; SETUP THE SCOPE LOOP ADDRESS
2403          MOV  #0,$STNM ; SETUP TEST NUMBER AND
2404                                ; CLEAR THE ERROR FLAG (SERFLG)
2405          MOV  $STNM,$DISPLAY ; LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
2406          MOV  #EXIT0,BYPASS ; SETUP BYPASS ADDRESS
2407          MOV  RPA0R,R4 ; RP11E BUS ADDRESS
2408          CLRB DPB+1 ; CLEAR ANY EXTENDED MEMORY BITS
2409          CLRB DPB+3 ; CLEAR 'MODE' & 'HOR' BITS
2410          MOV  #100,$TIMES ; DO 100 ITERATIONS
2411          MOV  #STACK,SP ; SETUP THE STACK POINTER
2412          JSR  PC,CLAP ; CLEAR THE RP11
2413          MOV  #BUFFER,$BUF ; LOAD BUFFER ADDRESS

```

```

2414 005166 012737 000001 002504 MOV #1,SWC ;SETUP WORD COUNT FOR 1 WORD
2415 005174 005037 002510 CLR $CYL ;CYLINDER 0
2416 005200 005037 002512 CLR $SEC ;TRACK & SECTOR 0
2417 005204 112737 000005 002502 MOVB #RDSEK,DPB+2 ;READ COMMAND
2418 005212 004737 022740 JSR PC,RP11 ;ISSUE THE COMMAND AND RETURN
2419 005216 032764 000200 000004 BIT #RDY,RPCS(R4) ;IS THE CONTROLLER READY ?
2420 005224 001404 BEQ #1 ;BR IF NOT
2421 005226 004737 023172 JSR PC,SAVRP ;STORE THE REGISTERS
2422 005232 104002 ERROR 2 ;CAN'T START READ COMMAND
2423 005234 000420 BR ;EXIT FROM TEST
2424 005236 012764 000001 000004 1$: MOV #CLEAR,RPCS(R4) ;ISSUE THE CONTROLLER CLEAR
2425 005244 012764 000001 000004 MOV #CLEAR,RPCS(R4) ;CLEAR THE CONTROLLER AGAIN
2426 005252 004737 023172 JSR PC,SAVRP ;STORE THE REGISTERS
2427 005256 032737 000200 001340 BIT #RDY,RPCS ;IS THE CONTROLLER READY ?
2428 005264 001004 BNE EXIT0 ;BR IF IT IS
2429 005266 104003 ERROR 3 ;'READY' DIDN'T SET AFTER ISSUING 'CLEAR'
2430 ;DURING READ COMMAND
2431 005270 000005 RESET ;FORCE THE CONTROLLER READY
2432 005272 004737 020374 JSR PC,STKINT ;SETUP THE TTY KEYBOARD
2433 005276 000004 EXIT0: SCOPE ;LOOP ?

```

```

*****
;*TEST 1 WRITE CHECK TEST

```

```

;*THIS TEST VERIFIES THE WRITE CHECK LOGIC AND TO VERIFY THAT A WRITE
;*CHECK ERROR CAN BE DETECTED. 'FLOATING ONES' AND 'FLOATING ZEROS'
;*PATTERNS ARE USED TO TEST THE WRITE CHECK COMPARE LOGIC.

```

```

*****
;*TST1:

```

```

2445 005300 033737 001402 001276 BIT BITS+(<STN*2>),TSTNMS ;IS THIS TEST SELECTED
2446 005306 001002 BNE .+6 ;BR IF YES
2447 005310 000137 005714 JMP TST2 ;GO TO THE NEXT TEST
2448 005314 012737 005374 001110 MOV #TEST1,SLPERR ;SETUP THE ERROR LOOP ADDRESS
2449 005322 012737 005300 001106 MOV #TST1,SLPAOR ;SETUP THE SCOPE LOOP ADDRESS
2450 005330 012737 000001 001102 MOV #1,$TSTNM ;SETUP TEST NUMBER AND
2451 ;CLEAR THE ERROR FLAG (SERFLG)
2452 005336 013777 001102 173576 MOV $TSTNM,$DISPLAY ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
2453 005344 012737 005712 001200 MOV #EXIT1,BYPASS ;SETUP BYPASS ADDRESS
2454 005352 013704 001324 MOV RPAOR,R4 ;RP11E BUS ADDRESS
2455 005356 105037 002501 CLRE DPB+1 ;CLEAR ANY EXTENDED MEMORY BITS
2456 005362 105037 002503 CLRB DPB+3 ;CLEAR 'MODE' & 'HDR' BITS
2457 005366 012737 000144 001162 MOV #100,$TIMES ;DO 100. ITERATIONS
2458 005374 012706 001100 TEST1: MOV #STACK,SP ;SETUP THE STACK POINTER
2459 005400 005037 002512 CLR $SEC ;CLEAR TRACK & SECTOR ADDRESS
2460 005404 005037 002510 CLR $CYL ;CLEAR CYLINDER ADDRESS
2461 005410 012737 000400 002504 MOV #256,$WC ;SET WORD COUNT TO 1 SECTOR
2462 005416 012737 000011 001204 MOV #9,$PATNUM ;STARTING PATTERN NUMBER
2463 005424 004737 024364 JSR PC,SETBUF ;FILL THE BUFFER WITH THE PATTERN
2464 005432 012737 043602 002506 MOV #BUFFER,$BUF ;SETUP OUTPUT BUFFER
2465 005436 012737 005436 001110 1$: MOV #1,$SLPERR ;SETUP LOOP ON ERROR ADDRESS
2466 005444 004737 025606 JSR PC,CLRP ;CLEAR THE RP11
2467 005450 012737 000003 002502 MOV #WATSEK,DPB+2 ;LOAD COMMAND
2468 005456 004737 022740 JSR PC,RP11 ;WRITE THE SECTOR WITH THE TEST PATTERN
2469 005462 004737 023616 JSR PC,CONRDY ;WAIT FOR THE OPERATION TO COMPLETE

```

2470	005465	004737	023172			JSR	PC,SAVRP	:SAVE THE REGISTERS
2471	005472	032737	100000	001340		BIT	#ERR,\$RPCS	:DID AN ERROR OCCUR ?
2472	005500	001401				BEQ	3\$:BR ? NOT
2473	005502	104004				ERROR	4	:ERROR OCCURED WRITING WRITE CHECK TEST PATTERN
2474	005504	012737	005504	001110	3\$:	MOV	#3\$,SLPERR	:CHANGE LOOP ON ERROR ADDRESS
2475	005512	004737	025606			JSR	PC,CLRP	:CLEAR THE RP11
2476	005516	012737	000207	002502		MOV	#WRTCHK,DPB+2	:LOAD WRITE CHECK COMMAND
2477	005524	004737	022740			JSR	PC,RP11	:CHECK THE SECTOR JUST WRITTEN
2478	005530	004737	023616			JSR	PC,CONRDY	:WAIT FOR WRITE CHECK TO COMPLETE
2479	005534	004737	023172			JSR	PC,SAVRP	:STORE THE REGISTERS
2480	005540	032737	100000	001340		BIT	#ERR,\$RPCS	:CHECK OK ?
2481	005546	001403				BEQ	4\$:BR IF OK
2482	005550	104005				ERROR	5	:ERROR ATTEMPTING TO CHECK TEST PATTERN
2483	005552	004737	025606			JSR	PC,CLRP	:CLEAR THE RP11
2484	005556	005001			4\$:	CLR	R1	:SETUP TO CLEAR BUFFER
2485	005560	005002				CLR	R2	:BUFFER INDEX
2486	005562	023727	001204	000011		CMP	PATNUM,#9.	: 'FLOATING 1' PATTERN ?
2487	005570	001002				BNE	5\$:BRANCH IF NOT
2488	005572	012701	177777			MOV	#-1,R1	:FILL BUFFER WITH ONES
2489	005576	010162	043602		5\$:	MOV	R1,BUFFER(R2)	:FILL BUFFER WITH PATTERN IN R1
2490	005602	022702	000400			CMP	#256.,R2	:SEE IF FINISHED
2491	005606	001403				BEQ	6\$:BR IF FINISHED
2492	005610	062702	000002			ADD	#2,R2	:INCREMENT THE INDEX
2493	005614	000770				BR	5\$:CONTINUE
2494	005616	012737	005616	001110	6\$:	MOV	#6\$,SLPERR	:CHANGE THE LOOP ON ERROR ADDRESS
2495	005624	004737	025606			JSR	PC,CLRP	:CLEAR THE RP11
2496	005630	004737	022740			JSR	PC,RP11	:CHECK THE SECTOR AGAIN, ERROR SHOULD OCCUR
2497	005634	004737	023616			JSR	PC,CONRDY	:WAIT FOR THE WRITE CHECK TO COMPLETE
2498	005640	004737	023172			JSR	PC,SAVRP	:STORE THE REGISTERS
2499	005644	032737	000010	001336		BIT	#BIT03,\$RPER	:DID WRITE CHECK ERROR SET?
2500	005652	001002				BNE	7\$:BRANCH IF YES
2501	005654	104006				ERROR	6	:WRITE CHECK ERROR DID NOT SET
2502	005656	000405				BR	8\$:BYPASS REST OF TEST
2503	005660	032737	100000	001340	7\$:	BIT	#ERR,\$RPCS	:DID 'ERR' SET ?
2504	005666	001001				BNE	8\$:BR IF IT SET
2505	005670	104106				ERROR	106	: 'ERR' DIDN'T SET WITH 'WCE'
2506	005672	023727	001204	000012	8\$:	CMP	PATNUM,#10.	:FLOATING A ZERO ?
2507	005700	001404				BEQ	EXIT1	:BR IF ZERO
2508	005702	012737	000012	001204		MOV	#10.,PATNUM	: 'FLOATING ZERO' PATTERN
2509	005710	000652				BR	1\$:DO THE TEST AGAIN
2510	005712	000004			EXIT1:	SCOPE		:LOOP ?

;TEST 2 PARTIAL SECTOR WRITE TEST

;CHECK THE ABILITY OF THE RP11E TO CLEAR THE REMAINDER OF A SECTOR
;ON A PARTIAL WRITE OPERATION. A SECTOR OF ALL ONES IS WRITTEN AND
;THEN A TWO WORD WRITE OPERATION IS PERFORMED. THE SECTOR IS THEN
;READ BACK AND VERIFIED. THE FIRST TWO WORDS SHOULD BE ONES AND
;THE REST SHOULD BE ZEROS.

2523	005714				†ST2:	BIT	BITS+(STN*2),TSTNMS	:IS THIS TEST SELECTED
2524	005714	033737	001404	001276		BNE	+.6	:BR IF YES
2525	005722	001002						

2526	005724	000137	006372		JMP	TST3	:GO TO THE NEXT TEST	
2527	005730	012737	006010	001110	MOV	#TEST2,\$LPERR	:SETUP THE ERROR LOOP ADDRESS	
2528	005736	012737	005714	001106	MOV	#TST2,\$LPADR	:SETUP THE SCOPE LOOP ADDRESS	
2529	005744	012737	000002	001102	MOV	#2,\$STSTNM	:SETUP TEST NUMBER AND	
2530							:CLEAR THE ERROR FLAG (SERFLG)	
2531	005752	013777	001102	173162	MOV	\$STSTNM,\$DISPLAY	:LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER	
2532	005760	012737	006370	001200	MOV	#EXIT2,BYPASS	:SETUP BYPASS ADDRESS	
2533	005766	013704	001324		MOV	RPADR,\$R4	:RP11E BUS ADDRESS	
2534	005772	105037	002501		CLRB	DPB+1	:CLEAR ANY EXTENDED MEMORY BITS	
2535	005776	105037	002503		CLRB	DPB+3	:CLEAR 'MODE' & 'HDR' BITS	
2536	006002	012737	000144	001162	MOV	#100,\$TIMES	:DO 100 ITERATIONS	
2537	006010	012706	001100		MOV	#STACK,\$SP	:SETUP THE STACK POINTER	
2538	006014	012737	006010	001110	MOV	#TEST2,\$LPERR	:SETUP INITIAL LOOP ON ERROR ADDRESS	
2539	006022	004737	025606		JSR	PC,\$CLRP	:CLEAR THE RP11	
2540	006026	012737	043602	002506	MOV	#BUFFER,\$BUF	:LOAD BUFFER ADDRESS	
2541	006034	012737	000400	002504	MOV	#256,\$WC	:WORD COUNT - 1 SECTOR	
2542	006042	005037	002512		CLR	\$SEC	:CLEAR SECTOR/TRACK ADDRESS	
2543	006046	005037	002510		CLR	\$CYL	:CLEAR CYLINDER ADDRESS	
2544	006052	012737	000015	001204	MOV	#13,\$PATNUM	:ONE'S PATTERN NUMBER	
2545	006060	004737	024364		JSR	PC,\$SETBUF	:LOAD THE PATTERN INTO THE BUFFER	
2546	006064	012737	000003	002502	MOV	#WRTSEK,\$DPB+2	:WRITE COMMAND	
2547	006072	004737	022740		JSR	PC,\$RP11	:WRITE THE SECTOR	
2548	006076	004737	023616		JSR	PC,\$CONRDY	:WAIT FOR THE WRITE TO COMPLETE	
2549	006102	004737	023172		JSR	PC,\$SAVRP	:STORE THE REGISTERS	
2550	006106	032737	100000	001340	BIT	#ERR,\$RPCS	:DID AN ERROR OCCUR ?	
2551	006114	001402			BEQ	1\$:BR IF NOT	
2552	006116	104004			ERROR	4	:ERROR OCCURED DURING WRITE	
2553	006120	060523			BR	EXIT2	:BYPASS REST OF TEST	
2554	006122	012737	006122	001110	MOV	#1\$,\$LPERR	:CHANGE LOOP ON ERROR ADDRESS	
2555	006130	004737	025606		JSR	PC,\$CLRP	:CLEAR THE RP11	
2556	006134	004737	024364		JSR	PC,\$SETBUF	:LOAD ONE'S PATTERN INTO THE BUFFER	
2557	006140	012737	000002	002504	MOV	#2,\$WC	:2 WORD TRANSFER	
2558	006146	112737	000003	002502	MOVB	#WRTSEK,\$DPB+2	:CHANGE COMMAND TO WRITE	
2559	006154	004737	022740		JSR	PC,\$RP11	:WRITE	
2560	006160	004737	023616		JSR	PC,\$CONRDY	:WAIT FOR CONTROLLER READY	
2561	006164	004737	023172		JSR	PC,\$SAVRP	:STORE THE REGISTERS	
2562	006170	032737	100000	001340	BIT	#ERR,\$RPCS	:ERROR ?	
2563	006176	001402			BEQ	2\$:BR IF NOT	
2564	006200	104007			ERROR	7	:ERROR WRITING THE PARTIAL SECTOR	
2565	006202	000472			BR	EXIT2	:BYPASS THE REST OF THE TEST	
2566	006204	012737	006204	001110	MOV	#2\$,\$LPERR	:CHANGE THE ERROR LOOP ADDRESS	
2567	006212	004737	025606		JSR	PC,\$CLRP	:CLEAR THE RP11	
2568	006216	012737	000400	002504	MOV	#256,\$WC	:CHANGE THE WORD COUNT	
2569	006224	012737	000005	002502	MOV	#RDSEK,\$DPB+2	:READ COMMAND	
2570	006232	004737	022740		JSR	PC,\$RP11	:READ THE SECTOR	
2571	006236	004737	023616		JSR	PC,\$CONRDY	:WAIT FOR CONTROLLER READY	
2572	006242	004737	023172		JSR	PC,\$SAVRP	:STORE THE REGISTERS	
2573	006246	032737	100000	001340	BIT	#ERR,\$RPCS	:DID AN ERROR OCCUR ?	
2574	006254	001402			BEQ	3\$:BR IF NOT	
2575	006256	104017			ERROR	17	:ERROR OCCURED READING SECTOR	
2576	006260	000443			BR	EXIT2	:BYPASS REST OF TEST	
2577	006262	022737	177777	043602	3\$:	CMP	#177777,\$BUFFER	:COMPARE FIRST WORD SHOULD BE ONES
2578	006270	001004			BNE	4\$:BR IF NOT	
2579	006272	022737	177777	043604	4\$:	CMP	#177777,\$BUFFER+2	:CHECK THE SECOND WORD
2580	006300	001407			BEQ	5\$:BR IF OK	
2581	006302	012737	177777	001124	4\$:	MOV	#177777,\$GDOAT	:EXPECTED DATA

```

2582 006310 013737 043602 001126      MOV    BUFFER,$BDDAT ;ACTUAL DATA
2583 006316 104011      ERROR  11           ;DATA COMPARE ERROR IN FIRST 2 WORDS
2584 006320 012700 043606      5$:  MOV    #BUFFER+4,R0 ;STARTING ADDRESS OF LAST PART OF THE BUFFER
2585 006324 012701 000376      MOV    #254.,R1      ;WORD COUNT
2586 006330 005720      6$:  TST    (R0)+       ;REMAINDER OF SECTOR SHOULD BE ZEROS
2587 006332 001003      BNE    7$           ;BRANCH IF NOT
2588 006334 005301      DEC    R1           ;DECREMENT THE CUNT
2589 006336 001374      BNE    6$           ;BR IF NOT FINISHED
2590 006340 000413      BR     EXIT2        ;EXIT
2591 006342 016037 177776 001126 7$:  MOV    -2(R0),$BDDAT ;INCORRECT DATA
2592 006350 005037 001124      CLR    $GDDAT       ;EXPECTED DATA (ZEROS)
2593 006354 010037 001122      MOV    R0,$BDADR    ;ADDRESS OF INCORRECT CHARACTER
2594 006360 162737 000002 001122  SUB    #2,$BDADR     ;DECREMENT THE ADDRESS
2595 006366 104012      ERROR  12           ;DATA FOUND IN AREA OF SECTOR
2596 006366 104012      ERROR  12           ;WHICH SHOULD HAVE BEEN CLEARED
2597 006366 104012      ERROR  12           ;BY A ONE WORD WRITE
2598 006370 000004      EXIT2: SCOPE

```

```

*****
;TEST 3 'EOP' TEST

```

```

;TEST THAT 'EOP' SETS WHEN THE CONTROLLER TRIES TO WRITE BEYOND THE
;LIMITS OF THE DRIVE. THE FIRST SECTOR OF THE PACK IS WRITTEN WITH
;ZEROS; THEN A TWO SECTOR WRITE OF ALL ONE'S IS ISSUED FOR THE MAXIMUM
;CYLINDER, HEAD 19, SECTOR 9. 'EOP' AND THE ERROR BITS IN 'RPCS' SHOULD
;BE SET. THE FIRST SECTOR OF THE PACK IS CHECKED TO MAKE SURE THAT IS
;STILL CONTAINS ZEROS.

```

```

*****
;TST3:

```

```

2612 006372      ;TST3:
2613 006372 033737 001406 001276      BIT    BITS+(STN*2),TSTNMS ;IS THIS TEST SELECTED
2614 006400 001062      BNE    .+6          ;BR IF YES
2615 006402 000137 007050      JMP    TST4         ;GO TO THE NEXT TEST
2616 006406 012737 006466 001110      MOV    #TEST3,$LPERR ;SETUP THE ERROR LOOP ADDRESS
2617 006414 012737 006372 001106      MOV    #TST3,$LPADR  ;SETUP THE SCOPE LOOP ADDRESS
2618 006422 012737 000003 001102      MOV    #3,$TSTNM    ;SETUP TEST NUMBER AND
2619 006430 013777 001102 172504      MOV    $TSTNM,$DISPLAY ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
2620 006436 012737 007046 001200      MOV    #EXIT3,BYPASS ;SETUP BYPASS ADDRESS
2621 006444 013704 001324      MOV    RPAOR,R4     ;RP11E BUS ADDRESS
2622 006450 105037 002501      CLRB  DPB+1         ;CLEAR ANY EXTENDED MEMORY BITS
2623 006454 105037 002503      CLRB  DPB+3         ;CLEAR 'MODE' & 'HDR' BITS
2624 006460 012737 000031 001162      MOV    #25.,$TIMES  ;DO 25. ITERATIONS
2625 006466 012706 001100      TEST3: MOV    #STACK,SP ;SETUP THE STACK POINTER
2626 006472 012737 006466 001110      MOV    #TEST3,$LPERR ;INITIAL LOOP ON ERRORR ADDRESS
2627 006500 004737 025606      JSR    PC,CLRP      ;CLEAR THE RP11
2628 006504 012737 000400 002504      MOV    #256.,$WC    ;SET WORD COUNT TO 1 SECTOR
2629 006512 012737 043602 002506      MOV    #BUFFER,$BUF ;SETUP OUTPUT BUFFER
2630 006520 005037 001204      CLR    PATNUM       ;SETUP FOR PATTERN ZERO
2631 006524 004737 024364      JSR    PC,SETBUF    ;LOAD ZEROS INTO THE BUFFER
2632 006530 005037 002512      CLR    $SEC         ;SET SECTOR & TRACK TO ZERO
2633 006534 005037 002510      CLR    $CYL         ;SET CYLINDER TO ZERO
2634 006540 112737 000003 002502      MOVB  #WRTSEK,DPB+2 ;WRITE COMMAND
2635 006546 004737 022740      JSR    PC,RP11     ;WRITE ZEROS IN CO, TO, SO
2636 006552 004737 023616      JSR    PC,CONRDY   ;WAIT FOR THE ORDER TO COMPLETE

```

2538	006556	004737	023172			JSR	PC,SAVRP	:STORE THE REGISTERS
2539	006562	032737	100000	001340		BIT	#ERR,\$RPCS	:DID AN ERROR OCCUR ?
2540	006570	001402				BEQ	15	:BR IF NOT
2541	006572	104004				ERROR	4	:ERROR OCCURED DURING SETUP
2542	006574	000524				BR	EXIT3	:BYPASS REST OF TEST
2543	006576	012737	006576	001110	15:	MOV	#15,\$LPERR	:CHANGE LOOP ON ERROR ADDRESS
2544	006604	004737	025606			JSR	PC,CLRP	:CLEAR THE RP11
2545	006610	012737	000015	001204		MOV	#13,\$PATNUM	:ONE'S PATTERN INDEX
2546	006616	004737	024364			JSR	PC,SETBUF	:LOAD THE PATTERN INTO THE BUFFER
2547	006622	012737	001000	002504		MOV	#52,\$WC	:CHANGE WORD COUNT TO 2 SECTORS
2548	006630	013737	001210	002510		MOV	MAXCYL,\$CYL	:SELECT THE DRIVE'S MAXIMUM CYLINDER
2549	006636	012737	000011	002512		MOV	#9,\$SEC	:SELECT SECTOR 9
2550	006644	112737	000023	002513		MOVB	#19,\$TRK	:SELECT HEAD 19
2551	006652	004737	022740			JSR	PC,RP11	:START THE READ
2552	006656	004737	023616			JSR	PC,CONRDY	:WAIT FOR CONTROLLER READY
2553	006662	004737	023172			JSR	PC,SAVRP	:STORE THE RP11 REGISTERS
2554	006666	032737	000002	001336		BIT	#BIT01,\$RPER	:DID EOP ERROR SET?
2555	006674	001002				BNE	25	:BRANCH IF SET
2556	006676	104013				ERROR	13	: 'EOP' DIDN'T SET
2557	006700	000405				BR	35	:CHECK RPCA
2558	006702	032737	100000	001340	25:	BIT	#ERR,\$RPCS	:DID THE ERROR FLAG SET?
2559	006710	001001				BNE	35	:BRANCH IF SET
2560	006712	104014				ERROR	14	:ERROR DID NOT SET AFTER GENERATING 'EOP'
2561	006714	005737	001346		35:	TST	\$RPCA	:SEE IF THE CYLINDER ADDRESS REGISTER WAS CLEARED
2562	006720	001401				BEQ	45	:BR IF CLEARED
2563	006722	104015				ERROR	15	: 'RPCA' NOT CLEARED BY 'EOP'
2564	006724	023737	001354	001210	45:	CMP	\$SUCA,MAXCYL	:SUCA STILL EQUAL TO THE MAXIMUM CLINDER
2565	006732	001401				BEQ	55	:BR IF IT IS
2566	006734	104016				ERROR	16	:SUCA NOT CORRECT AFTER EOP ERROR
2567	006736	012737	006736	001110	55:	MOV	#55,\$LPERR	:CHANGE LOOP ON ERROR ADDRESS
2568	006744	004737	025606			JSR	PC,CLRP	:CLEAR THE RP11
2569	006750	012737	000002	002504		MOV	#2,\$WC	:WORD COUNT
2570	006756	005037	002512			CLR	\$SEC	:CLEAR SECTOR/TRACK ADDRESS
2571	006762	005037	002510			CLR	\$CYL	:CLEAR THE CYLINDER ADDRESS
2572	006766	012737	000005	002502		MOV	#RDSEK,DPB+2	:CHANGE COMMAND TO READ
2573	006774	004737	022740			JSR	PC,RP11	:EXECUTE THE COMMAND
2574	007000	004737	023616			JSR	PC,CONRDY	:WAIT FOR CONTROLLER READY
2575	007004	004737	023172			JSR	PC,SAVRP	:STORE THE REGISTERS
2576	007010	032737	100000	001340		BIT	#ERR,\$RPCS	:WERE THERE ANY ERRORS?
2577	007016	001402				BEQ	65	:BRANCH IF NOT
2578	007020	104017				ERROR	17	:ERROR ENCOUNTERED ON 2 WORD READ
2579	007022	000411				BR	EXIT3	:OF FIRST SECTOR ON THE PACK
2580	007024	013737	043602	001126	65:	MOV	BUFFER,\$BDDAT	:GET FIRST WORD OF BUFFER
2581	007032	005737	001126			TST	\$BDDAT	:DOES 1ST SECTOR STILL CONTAIN ZEROS?
2582	007036	001403				BEQ	EXIT3	:BRANCH IF YES
2583	007040	005037	001124			CLR	\$GDDAT	:ZEROS WERE EXPECTED
2584	007044	104020				ERROR	20	:CONTENTS OF THE FIRST SECTOR OF THE
2585								:PACK CHANGED AFTER FORCING EOP
2586	007046	000004				EXIT3:	SCOPE	
2587								
2588								
2589								
2590								
2591								
2592								
2593								

: *TEST 4 TEST 'PROG' ERROR
: *VERIFY THAT THE RP11E GENERATES A 'PROG' ERROR IF A COMMAND IS ISSUED
: *WHILE THE CONTROLLER IS BUSY.

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

007050
007050 033737 001410 001276
007056 001002
007060 000137 007300
007064 012737 007144 001110
007072 012737 007050 001106
007100 012737 000004 001102

007106 013777 001102 172026
007114 012737 007276 001200
007122 013704 001324
007126 105037 002501
007130 105037 002503
007136 012737 000144 001162
007144 012706 001100
007150 004737 025606
007154 005037 002512
007160 005037 002510
007164 012737 043602 002506
007172 012737 002000 002504
007200 112737 000005 002502
007206 004737 022740
007212 012737 000001 001220
007220 004737 023134
007224 112764 000005 000004
007232 004737 023172
007236 032737 002000 001336
007244 001002
007246 104021

007250 000412
007252 032737 040000 001340
007260 001001
007262 104107
007264 032737 100000 001340
007272 001001
007274 104022
007276 000004

TST4: BIT BITS+(<STN*2>),TSTNMS :IS THIS TEST SELECTED
BNE +6 :BR IF YES
JMP TST5 :GO TO THE NEXT TEST
MOV #TEST4,\$LPERR :SETUP THE ERROR LOOP ADDRESS
MOV #TST4,\$LPADR :SETUP THE SCOPE LOOP ADDRESS
MOV #4,\$TSTNM :SETUP TEST NUMBER AND
:CLEAR THE ERROR FLAG (\$ERFLG)
MOV \$TSTNM,\$DISPLAY :LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
MOV #EXIT4,BYPASS :SETUP BYPASS ADDRESS
MOV RPAOR,R4 :RP11E BUS ADDRESS
CLR8 DPB+1 :CLEAR ANY EXTENDED MEMORY BITS
CLR8 DPB+3 :CLEAR 'MODE' & 'HOR' BITS
MOV #100,\$TIMES :DO 100. ITERATIONS
TEST4: MOV #STACK,\$P :SETUP THE STACK POINTER
JSR PC,CLR# :CLEAR THE RP11
CLR \$SEC :CLEAR THE SECTOR - TRACK ADDRESSES
CLR \$CYL :CLEAR THE CYLINDER ADDRESS
MOV #BUFFER,\$BUF :SETUP BUFFER ADDRESS
MOV #1024,\$WC :SETUP WORD COUNT
MOV#B #ROSEK,\$DPB+2 :READ COMMAND
JSR PC,RP11 :ISSUE THE COMMAND
MOV #1,\$STALL :LOAD STALL VALUE
JSR PC,STALL :STALL FOR 1 MILLISECOND
MOV#B #ROSEK,\$RPCS(R4) :ISSUE READ COMMAND WHILE BUSY
JSR PC,SAVRP :STORE THE REGISTERS
BIT #BIT10,\$RPER :DID PROGRAM ERROR SET?
BNE 18 :BRANCH IF SET
ERROR 21 :PROGRAM ERROR DID NOT SET WHEN A
:READ COMMAND WAS ISSUED WHILE
:THE DEVICE WAS BUSY
BR EXIT4 :BYPASS REST OF TEST
18: BIT #HE,\$RPCS :'HE' SET?
BNE 28 :BR IF NOT
ERROR 107 :'HE' DIDN'T SET WITH 'PROG'
28: BIT #ERR,\$RPCS :DID 'ERR' SET?
BNE EXIT4 :BRANCH IF SET
ERROR 22 :'ERR' DID NOT SET WITH PROGRAM ERROR
EXIT4: SCOPE

*TEST 5 'HEADER NOT FOUND' TEST
:UNFORMAT THE FIRST SECTOR ON THE PACK AND THEN READ IT BACK. VERIFY
:THAT READ AND WRITE HEADER OPERATIONS WILL TRANSFER DATA CORRECTLY.
:ISSUE A WRITE COMMAND TO SECTOR ZERO; THIS SHOULD CAUSE 'HEADER NOT
:FOUND' TO SET. REFORMAT THE SECTOR.

TST5: BIT BITS+(<STN*2>),TSTNMS :IS THIS TEST SELECTED
BNE +6 :BR IF YES
JMP TST6 :GO TO THE NEXT TEST

2750	007314	012737	007374	001110		MOV	#TEST5,\$LPERR	:SETUP THE ERROR LOOP ADDRESS
2751	007322	012737	007300	001106		MOV	#TST5,\$LPADR	:SETUP THE SCOPE LOOP ADDRESS
2752	007330	012737	000005	001102		MOV	#5,\$STSTM	:SETUP TEST NUMBER AND
2753								:CLEAR THE ERROR FLAG (SERFLG)
2754	007336	013777	001102	171576		MOV	\$STSTM,\$DISPLAY	:LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
2755	007344	012737	010022	001200		MOV	#EXITS,BYPASS	:SETUP BYPASS ADDRESS
2756	007352	013704	001324			MOV	RPAOR,\$R4	:RP11E BUS ADDRESS
2757	007356	105037	002501			CLRB	DPB+1	:CLEAR ANY EXTENDED MEMORY BITS
2758	007362	105037	002503			CLRB	DPB+3	:CLEAR 'MODE' & 'HDR' BITS
2759	007366	012737	000144	001162		MOV	#100,\$TIMES	:DO 100 ITERATIONS
2760	007374	012706	001100		TESTS:	MOV	#STACK,\$P	:SETUP THE STACK POINTER
2761	007400	012737	007374	001110		MOV	#TEST5,\$LPERR	:INITIAL LOOP ON ERROR ADDRESS
2762	007406	004737	025606			JSR	PC,CLRP	:CLEAR THE RP11
2763	007412	012737	000001	043602		MOV	#1,BUFFER	:SETUP INVALID HEADER IMAGE
2764	007420	012737	000001	043604		MOV	#1,BUFFER+2	:CYLINDER/TRACK
2765	007426	012737	000001	043606		MOV	#1,BUFFER+4	:SECTOR
2766	007434	012737	000003	002504		MOV	#3,\$WC	:LOAD WORD COUNT
2767	007442	005037	002512			CLR	\$SEC	:CLEAR SECTOR/TRACK ADDRESS
2768	007446	005037	002510			CLR	\$CYL	:CLEAR THE CYLINDER ADDRESS
2769	007452	012737	014000	002502		MOV	#MODE!HDR,DPB+2	: 'MODE' AND 'HEADER' BITS
2770	007460	112737	000003	002502		MOV	#WRITESEK,DPB+2	:WRITE ORDER
2771	007466	004737	022740			JSR	PC,RP11	:MISFORMAT SECTOR 0
2772	007472	004737	023616			JSR	PC,CONRDY	:WAIT FOR THE OPERATION TO COMPLETE
2773	007476	004737	023172			JSR	PC,SAVRP	:STORE THE REGISTERS
2774	007502	032737	100000	001340		BIT	#ERR,\$RPCS	:DID AN ERROR OCCUR ?
2775	007510	001402				BEQ	1\$:BR IF NOT
2776	007512	104023				ERROR	23	:ERROR OCCURED ATTEMPTING TO MISFORMAT SECTOR 0
2777	007514	000506				BR	TST5B	:BYPASS TEST & REFORMAT SECTOR 0
2778	007516	012737	007516	001110	1\$:	MOV	#1\$,\$LPERR	:CHANGE LOOP ON ERROR ADDRESS
2779	007524	004737	025606			JSR	PC,CLRP	:CLEAR THE RP11
2780	007530	112737	000005	002502		MOV	#RDSEK,DPB+2	:CHANGE COMMAND
2781	007536	004737	022740			JSR	PC,RP11	:EXECUTE THE COMMAND
2782	007542	004737	023616			JSR	PC,CONRDY	:WAIT FOR CONTROLLER READY
2783	007546	032737	100000	001340		BIT	#ERR,\$RPCS	:ANY ERRORS?
2784	007554	001401				BEQ	2\$:BRANCH IF NOT
2785	007556	104024				ERROR	24	:ERROR WHILE READING THE HEADER
2786								:ON SECTOR ZERO
2787	007560	012737	000001	001124	2\$:	MOV	#1,\$GDDAT	:EXPECTED RESULT
2788	007566	005000				CLR	RO	:CLEAR THE INDEX
2789	007570	023760	001124	043602	3\$:	CMF	\$GDDAT,BUFFER(RO)	:CHECK DATA READ BACK
2790	007576	001006				BNE	4\$:BRANCH ON NON COMPARE
2791	007600	062700	000002			ADD	#2,RO	:UPDATE MODIFIER
2792	007604	022700	000006			CMF	#6,RO	:END OF BUFFER?
2793	007610	001367				BNE	3\$:BR IF NOT
2794	007612	000404				BR	TST5A	:CONTINUE WITH TEST
2795	007614	016037	043602	001126	4\$:	MOV	BUFFER(RO),\$BDDAT	:GET BAD DATA
2796	007622	104025				ERROR	25	:DATA DID NOT VERIFY AFTER READING
2797								:THE HEADER OF SECTOR ZERO
2798	007624	012737	007624	001110	TST5A:	MOV	#TST5A,\$LPERR	:CHANGE LOOP ON ERROR ADDRESS
2799	007632	004737	025606			JSR	PC,CLRP	:CLEAR THE RP11
2800	007636	012737	000003	002502		MOV	#WRITESEK,DPB+2	:CHANGE COMMAND (AND CLEAR 'MODE' & 'HDR')
2801	007644	004737	022740			JSR	PC,RP11	:EXECUTE THE COMMAND
2802	007650	004737	023616			JSR	PC,CONRDY	:WAIT FOR CONTROLLER READY
2803	007654	004737	023172			JSR	PC,SAVRP	:STORE THE REGISTERS
2804	007660	032737	010000	001334	1\$:	BIT	#BIT12,\$R7DS	:DID HEADER NOT FOUND SE1
2805	007666	001002				BNE	2\$:BRANCH IF YES

2807	007670	104026				ERROR	26	;'HNF' DID NOT SET
2808	007672	000417				BR	TST58	: BYPASS REST OF TEST
2809	007674	032737	000001	001336	25:	BIT	#BIT00,SRPER	: DID DISK ERROR SET?
2810	007702	001001				BNE	38	: BRANCH IF YES
2811	007704	104027				ERROR	27	: 'DSK' DID NOT SET WITH 'HNF'
2812	007706	032737	040000	001340	35:	BIT	#HE,SRPCS	: DID HARD ERROR SET?
2813	007714	001001				BNE	45	: BRANCH IF YES
2814	007716	104030				ERROR	30	: HARD ERROR NOT SET AFTER 'HNF'
2815	007720	032737	100000	001340	45:	BIT	#ERR,SRPCS	: DID 'ERR' SET ?
2816	007726	001001				BNE	TST58	: BRANCH IF YES
2817	007730	104031				ERROR	31	: 'ERR' DID NOT SET WITH 'HNF'
2818	007732	012737	007732	001110	TST58:	MOV	#TST58,\$LPERR	: CHANGE LOOP ON ERROR ADDRESS
2819	007740	004737	025606			JSR	PC,CLRP	: CLEAR THE RP11
2820	007744	005037	043602			CLR	BUFFER	: SETUP HEADER IMAGE
2821	007750	005037	043604			CLR	BUFFER+2	: CYLINDER/TRACK = 0
2822	007754	005037	043606			CLR	BUFFER+4	: SECTOR = 0
2823	007760	012737	014000	002502		MOV	#MODE!HDR,DPB+2	: 'MODE' AND 'HDR' BITS
2824	007766	112737	000003	002502		MOV	#WRTSEK,DPB+2	: WRITE COMMAND
2825	007774	004737	022740			JSR	PC,RP11	: RESTORE THE HEADER
2826	010000	004737	023616			JSR	PC,CONRDY	: WAIT FOR THE OPERATION TO COMPLETE
2827	010004	004737	023172			JSR	PC,SAVRP	: STORE THE REGISTERS
2828	010010	032737	100000	001340		BIT	#ERR,SRPCS	: DID AN ERROR OCCUR ?
2829	010016	001557				BEG	EXIT6	: BR IF NOT
2830	010020	104032				ERROR	32	: ERROR ATTEMPTING TO RESTORE HEADER
2831	010022	000004				EXIT6		

EXITS: SCOPE

;TEST 6 TEST COMMAND ISSUED TO A SEEKING DRIVE

;ISSUE A SEEK COMMAND AND WAIT FOR DONE TO SET. THEN ISSUE A READ
;COMMAND WHILE THE HEADS ARE STILL MOVING. THE RP11E SHOULD
;HOLD THE WRITE COMMAND TILL THE SEEK IS COMPLETE.

TST6:

2841	010024					BIT	BITS+(\$TN*2),TSTNMS	: IS THIS TEST SELECTED
2842	010024	033737	001414	001276		BNE	+6	: BR IF YES
2843	010032	001002				JMP	TST7	: GO TO THE NEXT TEST
2844	010034	000137	010360			MOV	#TEST6,\$LPERR	: SETUP THE ERROR LOOP ADDRESS
2845	010040	012737	010120	001110		MOV	#TST6,\$LPAOR	: SETUP THE SCOPE LOOP ADDRESS
2846	010046	012737	010024	001106		MOV	#6,\$TSTNM	: SETUP TEST NUMBER AND
2847	010054	012737	000006	001102		MOV		: CLEAR THE ERROR FLAG (\$ERFLG)
2848						MOV	\$TSTNM,\$DISPLAY	: LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
2849	010062	013777	001102	171052		MOV	#EXIT6,BYPASS	: SETUP BYPASS ADDRESS
2850	010070	012737	010356	001200		MOV	RPAOR,R4	: RP11E BUS ADDRESS
2851	010076	013704	001324			CLR	DPB+1	: CLEAR ANY EXTENDED MEMORY BITS
2852	010102	105037	002501			CLR	DPB+3	: CLEAR 'MODE' & 'HDR' BITS
2853	010106	105037	002503			MOV	#25,\$TIMES	: DO 25 ITERATIONS
2854	010112	012737	000031	001162		MOV	#STACK,SP	: SETUP THE STACK POINTER
2855	010120	012706	001100		TEST6:	JSR	PC,CLRP	: CLEAR THE RP11
2856	010124	004737	025606			CLR	\$SEC	: SECTOR/TRACK 0
2857	010130	005037	002512			CLR	\$CYL	: CYLINDER 0
2858	010134	005037	002510			MOV	#HOMSEK,DPB+2	: HOME SEEK COMMAND
2859	010140	112737	000015	002502		JSR	PC,RP11	: DO THE COMMAND
2860	010146	004737	022740			JSR	PC,DRVPOY	: WAIT FOR DRIVE READY
2861	010152	004737	023500					

E05

```

2862 010156 012737 000300 002510      MOV      #192.,$CYL      ;CHANGE CYLINDER TO CYLINDER 192
2863 010164 112737 000011 002502      MOVB     #SEEK,DPB+2    ;SEEK COMMAND
2864 010172 004737 022740          JSR      PC,RP11       ;INITIATE THE SEEK
2865 010176 012737 000001 001220      MOV      #1,STALL     ;LOAD STALL VALUE
2866 010204 004737 023134          JSR      PC,STALL     ;STALL FOR 1 MILLISECOND
2867 010210 012764 177775 000006      MOV      #-3,RPWC(R4)  ;SET WORD COUNT TO THREE WORDS
2868 010216 012764 043602 000010      MOV      #BUFFER,RPBA(R4) ;BUFFER ADDRESS
2869 010224 052737 014000 002502      BIS      #MODE!HOR,DPB+2 ;SETUP TO READ THE HEADER
2870 010232 112764 000017 000004      MOVB     #READ,RPCC(R4) ;TRY TO DO A READ
2871 010240 004737 023500          JSR      PC,DRVRDY    ;WAIT FOR DRIVE READY
2872 010244 004737 023172          JSR      PC,SAVRP     ;STORE THE REGISTERS
2873 010250 032737 000200 001340      BIT      #RDY,$RPCS    ;DID CONTROLLER BECOME READY ALSO ?
2874 010256 001402          BEQ     1$           ;BR IF NOT
2875 010260 104033          ERROR   33          ;READ DIDN'T START DURING SEEK
2876 010262 000435          BR      EXIT6       ;EXIT
2877 010264 004737 023616          JSR      PC,CONRDY    ;WAIT FOR CONTROLLER READY
2878 010270 004737 023172          JSR      PC,SAVRP     ;STORE THE REGISTERS
2879 010274 032737 100000 001340      BIT      #ERR,$RPCS   ;ANY ERRORS?
2880 010302 001401          BEQ     2$           ;BRANCH IF NO
2881 010304 104034          ERROR   34          ;ERROR OCCURED DURING A READ ISSUED
2882 010306 012737 014000 001124      MOV      #(<192.*32.)$GDOAT ;EXPECTED CYLINDER VALUE
2883 010314 013737 043604 001126      MOV      BUFFER+2,$SBOAT ;RECEIVED DATA
2884 010322 023737 001124 001126      CMP      $GDOAT,$SBOAT ;CORRECT ?
2885 010330 001366          BNE     2$           ;BR IF NOT
2886 010332 005037 001124          CLR      $GDOAT      ;EXPECTED VALUE FOR SECTOR
2887 010336 013737 043606 001126      MOV      BUFFER+4,$SBOAT ;RECEIVED SECTOR
2888 010344 023737 001124 001126      CMP      $GDOAT,$SBOAT ;CORRECT ?
2889 010352 001401          BEQ     EXIT6       ;BR IF OK
2890 010354 104035          ERROR   35          ;HEADER DATA INCORRECT
2891 010356 000004      EXIT6:  SCOPE

;*****
;*TEST 7      TEST 'NXME' BIT
;*****
;*THE NON-EXISTENT MEMORY ERROR BIT IS TESTED BY ATTEMPTING TO READ 1
;*WORD INTO LOCATION 760000. 'NXME', 'HE', & 'ERR' SHOULD ALL BE SET.
;*****
TST7:
010360          BIT      BITS+(<$TN*2>),TSTNMS ;IS THIS TEST SELECTED
010360          BNE     +6         ;BR IF YES
010366          JMP     TST10      ;GO TO THE NEXT TEST
010370          MOV     #TEST7,$LPERR    ;SETUP THE ERROR LOOP ADDRESS
010374          MOV     #TST7,$LPADR    ;SETUP THE SCOPE LOOP ADDRESS
010402          MOV     #7,$TSTNM      ;SETUP TEST NUMBER AND
010410          MOV     000007,$SERFLG ;CLEAR THE ERROR FLAG (SERFLG)
010416          MOV     $TSTNM,$DISPLAY ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
010424          MOV     #EXIT7,BYPASS ;SETUP BYPASS ADDRESS
010432          MOV     RPAR,R4      ;RP11E BUS ADDRESS
010436          CLRB   DPB+1        ;CLEAR ANY EXTENDED MEMORY BITS
010442          CLRB   DPB+3        ;CLEAR 'MODE' & 'HOR' BITS
010446          MOV     #100,$TIMES  ;DO 100. ITERATIONS
010454          MOV     #STACK,SP    ;SETUP THE STACK POINTER
010460          JSR      PC,CLRP    ;CLEAR THE RP11

```

2918	010464	012737	000001	002504	MOV	#1, \$WC	:WORD COUNT
2919	010472	005037	002512		CLR	\$SEC	:SECTOR/TRACK ZERO
2920	010476	005037	002510		CLR	\$CYL	:CYLINDER ZERO
2921	010502	112737	000060	002501	MOV	#MEX1!MEX0, DPB+1	:LOAD BOTH EXTENDED MEMORY BITS
2922	010510	012737	160000	002506	MOV	#160000, \$BUF	:LOAD ADDR TO FORCE NON EX MEMORY
2923	010516	112737	000005	002502	MOV	#R0SEK, DPB+2	:TRY TO READ INTO THE LOCATION
2924	010524	004737	022740		JSR	PC, RP11	:INITIATE THE OPERATION
2925	010530	004737	023616		JSR	PC, CONRDY	:WAIT FOR CONTROLLER READY
2926	010534	004737	023172		JSR	PC, SAVRP	:STORE THE REGISTERS
2927	010540	032737	000004	001336	BIT	#BIT02, \$RPER	:DID NON EX MEMORY SET ?
2928	010546	001002			BNE	2\$:BRANCH IF SET
2929	010550	104036			ELFOR	36	: 'NXME' ERROR DIDN'T SET
2930	010552	000412			BR	EXIT7	:DON'T CHECK THE RPCS BITS
2931	010554	032737	040000	001340	2\$: BIT	#HE, \$RPCS	:DID 'HE' SET ON NON EX MEMORY
2932	010562	001001			BNE	3\$:BRANCH IF SET
2933	010564	104037			ERROR	37	: 'HE' DIDN'T SET
2934	010566	032737	100000	001340	3\$: BIT	#ERR, \$RPCS	:DID 'ERR' SET WITH NON EX MEMORY
2935	010574	001001			BNE	EXIT7	:BRANCH IF SET
2936	010576	104040			ERROR	40	: 'ERR' DIDN'T SET
2937	010600	000004			EXIT7:	SCOPE	

*TEST 10 SECTOR ADDRESSING TEST

*THE SECTOR ADDRESSING TEST TESTS THE CAPABILITY OF THE RP11E TO LOCATE
*A SECTOR BY USING THE 'SOT' COUNTER (FOR 'HOR' OPERATIONS) AND TO LOCATE
*A SECTOR BY COMPARING THE HEADER CONTENTS AGAINST THE CYLINDER AND TRACK/
*SECTOR ADDRESS REGISTERS (NORMAL MODE). THE TEST IS PERFORMED IN 3 PARTS.

1. WRITE ALL HEADERS ON TRACK 0, CYLINDER 0 IN ASCENDING SEQUENCE
FROM INDEX (0, 1, 2 ... 9).

2. READ EACH HEADER, BEGINNING WITH SECTOR 0, AND VERIFY THAT THE
SECTOR FIELD IN THE HEADER IS CORRECT FOR THE HEADER EXPECTED.

VERIFY THAT THE 'SOT' COUNTER AND THE SECTOR ADDRESS IN 'RPOA' ARE
EQUAL AND ARE THE EXPECTED VALUE. THE EXPECTED VALUE IS 1 GREATER
THAN THE SECTOR READ. THE SECTOR ADDRESS REGISTER AND THE 'SOT'
WILL BE INCREMENTED BY THE TIME THE PROGRAM IS ABLE TO READ THE
REGISTER.

WRITE THE SECTOR'S ADDRESS IN THE DATA FIELD. (FOR SECTOR 0, AN
OCTAL 12 IS WRITTEN.)

CONTINUE THIS SEQUENCE FOR THE REMAINING SECTORS ON THE TRACK.

3. READ THE DATA FROM EACH SECTOR USING A NORMAL (I.E., NON-HEADER)
READ AND VERIFY THAT THE DATA IS CORRECT FOR THE EXPECTED SECTOR.

*TST10:

2969	010602				BIT	BITS+(\$TN*2), TSTNMS	: IS THIS TEST SELECTED
2970	010602	033737	001420	001276	BNE	+6	: BR IF YES
2971	010610	001002			JMP	TST11	: GO TO THE NEXT TEST
2972	010612	000137	011522		MOV	#TEST10, \$LPERR	: SETUP THE ERROR LOOP ADDRESS
2973	010616	012737	010676	001110			

2974	010524	012737	010502	001106		MOV	#ST10, \$LPAOR	: SETUP THE SCOPE LOOP ADDRESS
2975	010632	012737	000010	001102		MOV	#10, \$ST1M	: SETUP TEST NUMBER AND
2976								: CLEAR THE ERROR FLAG (SERFLG)
2977	010640	013777	001102	170274		MOV	\$ST1M, \$DISPLAY	: LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
2978	010646	012737	011520	001200		MOV	#EXIT10, BYPASS	: SETUP BYPASS ADDRESS
2979	010654	013704	001324			MOV	RPAOR, R4	: RP11E BUS ADDRESS
2980	010660	105037	002501			CLRB	DPB+1	: CLEAR ANY EXTENDED MEMORY BITS
2981	010664	105037	002503			CLRB	DPB+3	: CLEAR 'MODE' & 'HDR' BITS
2982	010670	012737	000031	001162			#25, \$ST1ES	: DO 25 ITERATIONS
2983	010676	012706	001100		TEST10:	MOV	#STACK, SP	: RESET THE STACK POINTER
2984	010702	012737	010676	001110		MOV	#TEST10, \$LPERR	: LOOP ON ERROR ADDRESS
2985	010710	004737	025606			JSR	PC, CLRP	: CLEAR THE RP11
2986	010714	012737	043602	002506		MOV	#BUFFER, \$BUF	: SETUP BUFFER ADDRESS
2987	010722	012737	000036	002504		MOV	#30, \$WC	: SETUP WORD COUNT FOR 10 SECTORS
2988	010730	005037	002510			CLR	\$CYL	: SETUP FOR CYLINDER 0
2989	010734	005037	002512			CLR	\$SEC	: TRACK/SECTOR 0
2990	010740	004737	024236			JSR	PC, LOOSEC	: SETUP FORMAT BUFFER
2991	010744	012737	014000	002502		MOV	#MODE!HDR, DPB+2	: SETUP THE HEADER CONTROL BITS
2992	010752	112737	000003	002502		MOV	#WRTSEK, DPB+2	: THE COMMAND CODE
2993	010760	004737	022740			JSR	PC, RP11	: FORMAT THE TRACK
2994	010764	004737	023616			JSR	PC, CONRDY	: WAIT FOR THE ORDER TO COMPLETE
2995	010770	004737	023172			JSR	PC, SAVRP	: STORE THE REGISTERS
2996	010774	032737	100000	001340		BIT	#ERR, \$RPCS	: ERROR ?
2997	011002	001403				BEQ	1\$: BR IF NOT
2998	011034	104041				ERROR	41	: ERROR TRYING TO FORMAT THE TRACK
2999	011006	000137	011520			JMP	EXIT10	: BYPASS REST OF THE TEST
3000	011012	105037	002512		1\$:	CLRB	\$SEC	: START CHECK WITH SECTOR 0
3001	011016	012737	011016	001110	2\$:	MOV	#25, \$LPERR	: CHANGE THE LOOP ON ERROR ADDRESS
3002	011024	004737	025606			JSR	PC, CLRP	: CLEAR THE RP11
3003	011030	012737	014000	002502		MOV	#MODE!HDR, DPB+2	: SET 'MODE' AND 'HDR' BITS
3004	011036	112737	000005	002502		MOV	#RDSEK, DPB+2	: CHANGE THE OPERATION TO A READ
3005	011044	012737	000003	002504		MOV	#3, \$WC	: CHANGE THE WORD COUNT TO 3
3006	011052	004737	022740			JSR	PC, RP11	: READ THE SECTOR'S HEADER
3007	011056	004737	023616			JSR	PC, CONRDY	: WAIT FOR THE ORDER TO COMPLETE
3008	011062	004737	023172			JSR	PC, SAVRP	: STORE THE REGISTERS
3009	011066	012737	000000	001220		MOV	#0, \$STALL	: LOAD STALL VALUE
3010	011074	004737	023134			JSR	PC, STALL	: STALL FOR 0 MILLISECONDS
3011	011100	016437	000014	001350		MOV	RPAOR(R4), \$RPAOR	: GET THE CONTENTS OF THE ADDRESS REGISTER
3012	011106	032737	100000	001340		BIT	#ERR, \$RPCS	: ERROR ?
3013	011114	001402				BEQ	3\$: BR IF NOT
3014	011116	104042				ERROR	42	: ERROR READING THE HEADER
3015	011120	006457				BR	6\$: BYPASS DATA CHECK
3016	011122	005037	001124		3\$:	CLR	\$GDOAT	: CLEAR FOR EXPECTED DATA
3017	011126	113737	002512	001124		MOV	\$SEC, \$GDOAT	: MOVE SECTOR ADDRESS
3018	011131	023737	001124	043606		CMP	\$GDOAT, BUFFER+4	: CHECK THE SECTOR FIELD
3019	011142	001401				BEQ	4\$: BR IF CORRECT
3020	011144	104043				ERROR	43	: SECTOR FIELD FROM HEADER NOT CORRECT
3021	011146	175237	001124		4\$:	INC	\$GDOAT	: INCREMENT SECTOR ADDRESS FOR 'SOT' CHECK
3022	011152	123727	001124	000012		CMP	\$GDOAT, #10.	: EXCEED THE MAXIMUM ?
3023	011160	113402				BLO	.+6	: BR IF NOT
3024	011162	045037	001124			CLR	\$GDOAT	: RESET COMPARISON VALUE
3025	011166	013737	001350	001126		MOV	\$RPAOR, \$BDOAT	: CHECK THE 'SOT'
3026	011174	042737	177417	001126		BIC	#1C360, \$BDOAT	: CLEAR THE OTHER BITS
3027	011202	006237	001126			ASR	\$BDOAT	: SHIFT THE 'SOT'
3028	011206	006237	001126			ASR	\$BDOAT	: SHIFT THE 'SOT'
3029	011212	006237	001126			ASR	\$BDOAT	: SHIFT THE 'SOT'

3030	011216	006237	001126		ASR	\$BODAT	:SHIFT THE 'SOT'
3031	011222	013737	001350	001122	MOV	\$RPOA, \$BOADR	:CHECK THE SECTOR ADDRESS
3032	011230	042737	177760	001122	BIC	#17, \$BOADR	:LEAVE THE SECTOR ADDRESS
3033	011236	023737	001124	001126	CMP	\$GDDAT, \$BODAT	:IS THE 'SOT' CORRECT ?
3034	011244	001004			BNE	5\$:BR IF NOT
3035	011246	023737	001126	001122	CMP	\$BODAT, \$BOADR	:DOES THE SECTOR ADDRESS EQUAT THE 'SOT' ?
3036	011254	001401			BEQ	6\$:BR IF IT DOES
3037	011256	104045			ERROR	45	: 'SOT' OR SECTOR ADDRESS IS NOT CORRECT
3038	011260	012737	011260	001110	5\$: 6\$: MOV	#6\$, \$LPERR	:CHANGE THE LOOP ON ERROR ADDRESS
3039	011266	004737	025536		JSR	PC, CLRP	:CLEAR THE RP11
3040	011272	105037	006503		CLRB	DPB+3	:CLEAR 'MODE' & 'HDR' BITS
3041	011276	004737	024270		JSR	PC, FILSEC	:SETUP TO WRITE SECTOR ADDRESS AS DATA
3042	011302	112737	000003	002502	MOVB	#WRTSEK, DPB+2	:CHANGE OPERATION
3043	011310	004737	022740		JSR	PC, RP11	:WRITE THE SECTOR
3044	011314	004737	023616		JSR	PC, CONRDY	:WAIT FOR THE ORDER TO COMPLETE
3045	011320	004737	023172		JSR	PC, SAVRP	:STORE THE REGISTERS
3046	011324	032737	100000	001340	BIT	#ERR, \$RPCS	:ERROR ?
3047	011332	001401			BEQ	7\$:BR IF NOT
3048	011334	104044			ERROR	44	:ERROR TRYING TO WRITE SECTOR ADDRESS AS DATA
3049	011336	112737	000005	002502	7\$: MOVB	#RDSEK, DPB+2	:CHANGE THE OPERATION
3050	011344	105237	002512		INCB	\$SEC	:INCREMENT THE SECTOR
3051	011350	123727	002512	000012	CMPB	\$SEC, #10.	:AT MAXIMUM SECTOR ADDRESS ?
3052	011356	001217			BNE	2\$:BR IF NOT
3053	011360	012737	011414	001110	MOV	#8\$, \$LPERR	:CHANGE THE LOOP ON ERROR ADDRESS
3054	011366	012737	000400	002504	MOV	#256., \$WC	:CHANGE WORD COUNT
3055	011374	105037	002512		CLRB	\$SEC	:START WITH SECTOR ZERO
3056	011400	012737	000012	001124	MOV	#10., \$GDDAT	:SECTOR ZERO PATTERN
3057	011406	112737	000005	002502	MOVB	#RDSEK, DPB+2	:CHANGE OPERATION CODE
3058	011414	004737	025606		8\$: JSR	PC, CLRP	:CLEAR THE RP11
3059	011420	004737	022740		JSR	PC, RP11	:READ THE SECTOR
3060	011424	004737	023616		JSR	PC, CONRDY	:WAIT FOR THE ORDER TO COMPLETE
3061	011430	004737	023172		JSR	PC, SAVRP	:STORE THE REGISTERS
3062	011434	032737	100000	001340	BIT	#ERR, \$RPCS	:ERROR ?
3063	011442	001402			BEQ	9\$:BR IF NOT
3064	011444	104017			ERROR	17	:ERROR TRYING TO READ THE SECTOR
3065	011446	000412			BR	10\$:BYPASS DATA CHECK
3066	011450	005037	001120		9\$: CLR	\$GADR	:CLEAR FOR EXPECTED DATA
3067	011454	113737	002512	001120	MOVB	\$SEC, \$GADR	:EXPECTED DATA
3068	011462	023737	001124	043602	CMP	\$GDDAT, BUFFER	:CORRECT CONTENTS ?
3069	011470	001401			BEQ	10\$:BR IF CORRECT
3070	011472	104046			ERROR	46	:SECTOR PATTERN NOT CORRECT FOR SECTOR ADDRESS
3071	011474	105237	002512		10\$: INCB	\$SEC	:INCREMENT SECTOR ADDRESS
3072	011500	123727	002512	000012	CMPB	\$SEC, #10.	:FINISHED SECTOR NINE ?
3073	011506	001404			BEQ	EXIT10	:BR IF COMPLETED
3074	011510	113737	002512	001124	MOVB	\$SEC, \$GDDAT	:NEXT SECTOR
3075	011516	000736			BR	8\$:CONTINUE CHECKING
3076	011520	000004			EXIT10:	SCOPE	:LOOP ?

: #TEST 11 TRACK ADDRESSING TEST

: #TRACK ADDRESSING IS TESTED BY WRITING THE TRACK'S ADDRESS AS DATA IN
: #SECTOR 0 OF EACH TRACK IN CYLINDER 0. SECTOR 0 IS THEN READ BACK AND
: #THE DATA IS CHECKED TO VERIFY THAT THE PROPER HEAD WAS SELECTED.

3077
3078
3079
3080
3081
3082
3083
3084
3085


```

3096 011522 033737 001422 001276
3097 011522 001002
3098 011530 000137 012052
3099 011532 012737 011642 001110
3090 011536 012737 011522 001106
3091 011544 012737 000011 001102
3092 011552 013777 001102 167354
3093 011560 012737 012050 001200
3094 011574 013704 001324
3095 011600 105037 002501
3096 011604 105037 002503
3097 011610 012737 000031 001162
3098 011616 005037 002512
3099 011622 005037 002510
3100 011626 012737 000400 002504
3101 011634 012737 043602 002506
3102 011642 012706 001100
3103 011646 012737 011646 001110
3104 011654 004737 025606
3105 011660 112737 000003 002502
3106 011666 004737 024330
3107 011672 004737 022740
3108 011676 004737 023616
3109 011702 004737 023172
3110 011706 032737 100000 001340
3111 011714 001401
3112 011716 104004
3113 011720 105237 002513
3114 011724 123727 002513 000024
3115 011732 103745
3116 011734 005037 001124
3117 011740 005037 002512
3118 011744 012737 011744 001110
3119 011752 004737 025606
3120 011756 112737 000005 002502
3121 011764 004737 022740
3122 011770 004737 023616
3123 011774 004737 023172
3124 012000 032737 100000 001340
3125 012006 001402
3126 012010 104017
3127 012012 000405
3128 012014 123737 001125 043603
3129 012022 001401
3130 012024 104047
3131 012026 123727 002513 000023
3132 012034 001405
3133 012036 105237 002513
3134 012042 105237 001125
3135 012046 000736
3136 012050 000004
3137
3138
3139
3140
3141

```

```

TST11: BIT BITS+(<STN*2),1STNMS ; IS THIS TEST SELECTED
      BNE +6 ; BR IF YES
      JMP TST12 ; GO TO THE NEXT TEST
      MOV #TST11,$LPERR ; SETUP THE ERROR LOOP ADDRESS
      MOV #TST11,$LPADR ; SETUP THE SCOPE LOOP ADDRESS
      MOV #11,$1STNM ; SETUP TEST NUMBER AND
      ; CLEAR THE ERROR FLAG (SERFLG)
      MOV $1STNM,$DISP ; LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
      MOV #EXIT11,$BYPASS ; SETUP BYPASS ADDRESS
      MOV RP11,$R4 ; RP11E BUS ADDRESS
      CLRB DPB+1 ; CLEAR ANY EXTENDED MEMORY BITS
      CLRB DPB+3 ; CLEAR 'MODE' & 'HDR' BITS
      MOV #25,$TIMES ; DO 25 ITERATIONS
      CLR $SEC ; CLEAR SECTOR/TRACK ADDRESS
      CLR $CYL ; CLEAR CYLINDER ADDRESS
      MOV #256,$WC ; SET WORD COUNT TO ONE SECTOR
      MOV #BUFFER,$BUF ; SETUP THE BUFFER ADDRESS
      MOV #STACK,$SP ; SETUP THE STACK POINTER
TEST11: MOV #18,$LPERR ; CHANGE LOOP ON ERROR ADDRESS
18: JSR PC,$CLRP ; CLEAR THE RP11
      MOVB #WRITE,$DPB+2 ; CHANGE COMMAND TO WRITE
      JSR PC,$FILTRK ; FILL BUFFER WITH THE TRACK ADDRESS
      JSR PC,$RP11 ; WRITE THE TRACK'S ADDRESS
      JSR PC,$CONRDY ; WAIT FOR CONTROLLER TO FINISH
      JSR PC,$SAVRP ; STORE THE REGISTERS
      BIT #ERR,$RPCS ; ERROR ?
      BEQ 25 ; BR IF NOT
      ERROR 4 ; ERROR WRITTING TEST PATTERN
      INCB $TRK ; INCREMENT THE TRACK ADDRESS
      CMPB $TRK,#20 ; FINISHED ?
      BLO 18 ; BR IF NOT FINISHED
      CLR $GDDAT ; SETUP EXTPECTED VALUE
      CLR $SEC ; START WITH ZERO
      MOV #45,$LPERR ; CHANGE THE LOOP ON ERROR ADDRESS
      JSR PC,$CLRP ; CLEAR THE RP11
      MOVB #ROSEK,$DPB+2 ; CHANGE COMMAND
      JSR PC,$RP11 ; READ THE SECTOR
      JSR PC,$CONRDY ; WAIT FOR THE CONTROLLER TO BE READY
      JSR PC,$SAVRP ; STORE THE REGISTERS
      BIT #ERR,$RPCS ; ERROR ?
      BEQ 55 ; BR IF NOT
      ERROR 17 ; ERROR READING THE SECTOR
      BR 65 ; BYPASS THE DATA CHECK
      CMPB $GDDAT+1,$BUFFER+1 ; CORRECT VALUE ?
      BEQ 65 ; BR IF OK
      ERROR 47 ; CONTENTS WRONG
      CMPB $TRK,#19 ; FINISHED ?
      BEQ EXIT11 ; BR IF FINISHED
      INCB $TRK ; INCREMENT THE TRACK ADDRESS
      INCB $GDDAT+1 ; COMPARISON WORD
      BR 45 ; CONTINUE
EXIT11: SCOPE LOOP ?

```

3142
3143
3144
3145
3146
3147
3148
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3190
3191
3192
3193
3194
3195
3196
3197

012052
012052 033737 001424 001276
012060 001002
012062 000137 012710
012066 012737 012172 001110
012074 012737 012052 001106
012102 012737 000012 001102
012110 013777 001102 167024
012116 012737 012702 001200
012124 013704 001324
012130 105037 002501
012134 105037 002503
012140 012737 000144 001162
012146 005737 022202
012152 100402
012154 000137 012710
012160 005737 001232
012164 001002
012166 000137 012710
012172 012706 001100
012176 012737 012172 001110
012204 004737 025606
012210 005037 043602
012214 012737 177777 043604
012222 012737 043602 002506
012230 012737 000002 002504
012236 005037 002512
012242 005037 002510
012246 012737 000003 002502
012254 004737 022740
012260 004737 023616
012264 004737 023172
012270 032737 100000 001340
012276 001403

```
*****
*TEST 12 EXTENDED MEMORY ADDRESS TEST
*****
*THIS TEST TESTS THE OPERATION OF THE EXTENDED MEMORY ADDRESS BITS. IF THE
*SYSTEM DOES NOT HAVE MEMORY MANAGEMENT OR HAS MEMORY MANAGEMENT AND ONLY
*32K, THE TEST WILL NOT BE PERFORMED.
*
* 1. THE PROGRAM WRITES A 2 WORD TEST SECTOR OF ALL ONES.
*
* 2. EXTENDED ADDRESS BIT 'MEX00' IS TESTED BY CLEARING LOCATION
* 200000 AND READING THE TEST SECTOR INTO LOCATION 177776. LOCATION
* 200000 IS CHECKED TO VERIFY THAT THE DATA IS CORRECT (ONES). THE
* PROGRAM VERIFIES THAT 'MEX00' HAS SET AND THAT 'MEX01' IS NOT SET.
*
* 3. IF THE SYSTEM HAS AT LEAST 64K, 'MEX01' IS TESTED. LOCATION
* 400000 IS CLEARED AND THE TEST SECTOR IS READ INTO LOCATION 377776
* ('MEX00' IS SET FOR THE READ). LOCATION 400000 IS CHECKED FOR THE
* PROPER CONTENTS (ONES). 'MEX00' SHOULD HAVE RESET AND 'MEX01' SHOULD
* BE SET.
*****
TST12:
BIT BITS+(<STN*2>),TSTNMS ;IS THIS TEST SELECTED
BNE .+6 ;BR IF YES
JMP TST13 ;GO TO THE NEXT TEST
MOV #TEST12,$LPERR ;SETUP THE ERROR LOOP ADDRESS
MOV #TST12,$LPAOR ;SETUP THE SCOPE LOOP ADDRESS
MOV #12,$STNM ;SETUP TEST NUMBER AND
;CLEAR THE ERROR FLAG (SERFLG)
MOV $STNM,DISP ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
MOV #EXIT12,BYPASS ;SETUP BYPASS ADDRESS
MOV RPAOR,R4 ;RP11E BUS ADDRESS
CLRB DPB+1 ;CLEAR ANY EXTENDED MEMORY BITS
CLRB DPB+3 ;CLEAR 'MODE' & 'HDR' BITS
MOV #100,$TIMES ;DO 100. ITERATIONS
TST $KT11 ;MEMORY MANAGEMENT ON THE SYSTEM ?
BMI IS ;BR IF ON SYSTEM
JMP TST13 ;GO TO THE NEXT TEST
IS: TST MAXMEM+2 ;ENOUGH MEMORY FOR THIS TEST ?
BNE TEST12 ;BR IF YES
TEST12: MOV #STACK,SP ;GO TO THE NEXT TEST
MOV #TEST12,$LPERR ;SETUP THE STACK POINTER
JSR PC,CLRP ;INITIAL ERROR LOOP ADDRESS
CLR BUFFER ;CLEAR THE RP11
MOV #-1,BUFFER+2 ;TEST PATTERN
MOV #BUFFER,$BUF ;SECOND WORD OF PATTERN
MOV #2,$WC ;SETUP THE BUFFER ADDRESS
CLR $SEC ;WORD COUNT OF 2
CLR $CYL ;SECTOR/TRACK ZERO
MOV #WRTSEK,DPB+2 ;CYLINDER ZERO
JSR PC,RP11 ;WRITE COMMAND
JSR PC,CONRDY ;INITIATE THE COMMAND
JSR PC,SAVRP ;WAIT FOR CONTROLLER READY
BIT #ERR,$APCS ;STORE THE REGISTERS
BEG IS ;DID OPERATION TERMINATE WITH AN ERROR ?
;BR IF NOT
```


3198	012300	104004			ERROR	4	: ERROR AFTER 2 WORD WRITE
3199	012302	000137	012702		JMP	EXIT12	: BYPASS REST OF TEST
3200	012306	012737	012306	001110	1S: MOV	#15, \$LPERR	: CHANGE LOOP ON ERROR ADDRESS
3201	012314	004737	025606		JSR	PC, CLR	: CLEAR THE RP11
3202	012320	012737	007600	172306	MOV	#7600, KIPAR7	: OPEN I/O REGISTERS
3203	012326	005037	172340		CLR	KIPAR0	: ENABLE FIRST 4K
3204	012332	012737	000200	172342	MOV	#200, KIPAR1	: ENABLE SECOND 4K
3205	012340	012737	002000	172344	MOV	#2000, KIPAR2	: ENABLE FIRST TEST LOCATION
3206	012346	012737	077406	172300	MOV	#77406, KIPDR0	: LOAD DESCRIPTOR REGISTER 0
3207	012354	012737	077406	172302	MOV	#77406, KIPDR1	: LOAD DESCRIPTOR REGISTER 1
3208	012362	012737	077406	172304	MOV	#77406, KIPDR2	: LOAD DESCRIPTOR REGISTER 2
3209	012370	012737	077406	172316	MOV	#77406, KIPDR7	: LOAD DESCRIPTOR REGISTER 7
3210	012376	012737	000001	177572	MOV	#1, @#5A0	: TURN ON MEMORY MANAGEMENT
3211	012404	005037	040000		CLR	@#40000	: CLEAR LOCATION 200000
3212	012410	012737	177776	002506	MOV	#177776, \$BUF	: SETUP BUS ADDR
3213	012416	112737	000005	002502	MOVB	#RDSEK, DPB+2	: CHANGE COMMAND TO READ
3214	012424	004737	022740		JSR	PC, RP11	: START THE COMMAND
3215	012430	004737	023616		JSR	PC, CONRDY	: WAIT FOR CONTROLLER READY
3216	012434	004737	023172		JSR	PC, SAVRP	: STORE THE REGISTERS
3217	012440	032737	100000	001340	BIT	#ERR, \$RPCS	: ANY ERRORS?
3218	012446	001401			BEQ	2\$: BRANCH IF NOT
3219	012450	104050			ERROR	50	: ERROR AFTER READING 2 WORDS
3220	012452	032737	000020	001340	2S: BIT	#MEX0, \$RPCS	: DID 'MEX0' SET ?
3221	012460	001001			BNE	3\$: BRANCH IF IT DID
3222	012462	104051			ERROR	51	: 'MEX0' DID NOT SET AFTER 2 WORD READ
3223							: STARTING AT LOCATION 177776
3224	012464	032737	000040	001340	3S: BIT	#MEX1, \$RPCS	: 'MEX1' SHOULD NOT BE SET
3225	012472	001401			BEQ	4\$: BRANCH IF NOT SET
3226	012474	104052			ERROR	52	: 'MEX1' IS SET - SHOULD NOT BE
3227	012476	022737	177777	040000	4S: CMP	#177777, @#40000	: WAS DATA READ INTO LOCATION
3228	012504	001407			BEQ	TST128	: 200000 CORRECTLY? - BRANCH IF YES
3229	012506	012737	177777	001124	MOV	#177777, \$GDOAT	: EXPECTED CONTENTS
3230	012514	013737	040000	001126	MOV	@#40000, \$BDOAT	: CONTENTS OF LOCATION 200000
3231	012522	104053			ERROR	53	: DATA COMPARE ERROR AT LOC 200000
3232	012524	023727	001232	000002	TST128: CMP	MAXMEM+2, #2	: ENOUGH MEMORY FOR THE NEXT PART OF THE TEST ?
3233	012532	103463			BLO	EXIT12	: BR IF NOT
3234	012534	012737	004000	172344	MOV	#4000, KIPAR2	: CHANGE TEST LOCATION
3235	012542	012737	012542	001110	1S: MOV	#15, \$LPERR	: CHANGE THE LOOP ON ERROR ADDRESS
3236	012550	004737	025606		JSR	PC, CLR	: CLEAR THE RP11
3237	012554	005037	040000		CLR	@#40000	: CLEAR LOCATION 400000
3238	012560	012737	177776	002506	MOV	#177776, \$BUF	: BUFFER ADDRESS
3239	012566	112737	000020	002501	MOVB	#MEX0, DPB+1	: SETUP EXTENDED MEMORY ADDRESS BIT
3240	012574	112737	000005	002502	MOVB	#RDSEK, DPB+2	: READ COMMAND
3241	012602	004737	022740		JSR	PC, RP11	: START THE COMMAND
3242	012606	004737	023616		JSR	PC, CONRDY	: WAIT FOR CONTROLLER READY
3243	012612	004737	023172		JSR	PC, SAVRP	: STORE THE REGISTERS
3244	012616	032737	100000	001340	BIT	#ERR, \$RPCS	: ANY ERRORS?
3245	012624	001401			BEQ	2\$: BRANCH IF NOT
3246	012626	104054			ERROR	54	: ERROR WHILE READING TWO WORDS
3247	012630	032737	000020	001340	2S: BIT	#MEX0, \$RPCS	: IS 'MEX0' SET ?
3248	012636	001401			BEQ	3\$: BRANCH IF NOT
3249	012640	104055			ERROR	55	: 'MEX0' DID NOT CLEAR AFTER 2 WORD
3250							: READ STARTING AT LOCATION 377776
3251	012642	032737	000040	001340	3S: BIT	#MEX1, \$RPCS	: DID 'MEX1' SET ?
3252	012650	001001			BNE	4\$: BR IF SET
3253	012652	104056			ERROR	56	: 'MEX1' DID NOT SET AFTER 2 WORD TRANSFER

3255	012654	022737	177777	040000	45:	CMP	#177777, #40000	: STARTING AT LOCATION 377776
3256	012662	001407				BEQ	EXIT12	: WAS DATA READ INTO LOCATION 400000
3257	012664	012737	177777	001124		MOV	#177777, \$GDDAT	: CORRECTLY ? - BRANCH IF YES
3258	012672	013737	040000	001126		MOV	#40000, \$BDDAT	: EXPECTED DATA
3259	012700	104057				ERROR	57	: DATA FROM LOCATION 400000
3260	012702	005037	177572		EXIT12:	CLR	#SR0	: DATA COMPARE ERROR AT LOC 400000
3261	012706	000004				SCOPE		: TURN OFF MEMORY MANAGEMENT
3262								: LOOP ?

3263
3264
3265
3266
3267
3268
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292

: *TEST 13 DATA BUFFER REGISTER BIT TEST

: *THE DATA BUFFER REGISTER TEST VERIFIES THAT ALL 36 BITS OF THE DATA
: *BUFFER REGISTER CAN BE SET AND CLEARED.

: * 1. THE TEST FIRST WRITES THE FOLLOWING TEST PATTERN IN CYLINDER 0,
: * TRACK 0, SECTOR 0 USING 10/15 MODE ('MODE' BIT SET).

: * 052525
: * 052525
: * 000005
: * 125252
: * 125252
: * 000012
: * 052525
: * 052525
: * 000005
: * 125252
: * .
: * .
: * ETC

: * 2. THE TEST SECTOR IS THEN READ AND THE DATA IS COMPARED. AT THE
: * COMPLETION OF THE TEST, THE PROGRAM RESTORES THE TEST SECTOR TO
: * PDP-11 MODE.

: *TST13:

3293	012710					BIT	BITS+(<\$TN*2>), TSTNMS	: IS THIS TEST SELECTED
3294	012710	033737	001426	001276		BNE	.+6	: BR IF YES
3295	012716	001002				JMP	TST14	: GO TO THE NEXT TEST
3296	012720	000137	013564			MOV	#TST13, \$LPERR	: SETUP THE ERROR LOOP ADDRESS
3297	012724	012737	013004	001110		MOV	#TST13, \$LPADR	: SETUP THE SCOPE LOOP ADDRESS
3298	012732	012737	012710	001106		MOV	#13, \$TSTNM	: SETUP TEST NUMBER AND
3299	012740	012737	000013	001102				: CLEAR THE ERROR FLAG (\$ERFLG)
3300						MOV	\$TSTNM, \$DISPLAY	: LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
3301	012746	013777	001102	166166		MOV	#EXIT13, BYPASS	: SETUP BYPASS ADDRESS
3302	012754	012737	013562	001200		MOV	RPAOR, R4	: RP11E BUS ADDRESS
3303	012762	013704	001324			CLRB	DPB+1	: CLEAR ANY EXTENDED MEMORY BITS
3304	012766	105037	002501			CLRB	DPB+3	: CLEAR 'MODE' & 'HOR' BITS
3305	012772	105037	002503			MOV	#100, \$TIMES	: DO 100. ITERATIONS
3306	012776	012737	000144	001162	TEST13:	MOV	#STACK, SP	: SETUP THE STACK POINTER
3307	013004	012706	001100			MOV	#BUFFER, R1	: BUFFER ADDRESS
3308	013010	012701	043602			MOV	#128., R2	: SIZE
3309	013014	012702	000200					

M05

MD-11-DZRPY-B, RP11-E FUNCTIONAL LOGIC & R/W TEST
 DZRPYB.CMB T13 DATA BUFFER REGISTER BIT TEST

MACY11 27(732) 02-NOV-76 14:50 PAGE 65

3310	013020	012721	052525		1S:	MOV	#52525, (R1)+	: PATTERN FOR BITS 20 - 35
3311	013024	012721	052525			MOV	#52525, (R1)+	: PATTERN FOR BITS 4 - 19
3312	013030	012721	000005			MOV	#5, (R1)+	: PATTERN FOR BITS 0 - 3
3313	013034	005302				DEC	R2	: AT THE END ?
3314	013036	001410				BEQ	2S	: BR IF END
3315	013040	012721	125252			MOV	#125252, (R1)+	: PATTERN FOR BITS 20 - 35
3316	013044	012721	125252			MOV	#125252, (R1)+	: PATTERN FOR BITS 4 - 19
3317	013050	012721	000012			MOV	#12, (R1)+	: PATTERN FOR BITS 0 - 3
3318	013054	005302				DEC	R2	: AT THE END ?
3319	013056	001360				BNE	1S	: BR IF NOT
3320	013060	012737	013060	001110	2S:	MOV	#2S, \$LPERR	: LOOP ON ERROR ADDRESS
3321	013066	004737	025606			JSR	PC, CLR	: CLEAR THE RP11

3322	013072	005037	002512			CLR	\$SEC	:SECTOR/TRACK ADDRESS OF ZERO
3323	013076	005037	002510			CLR	\$CYL	:CYLINDER ZERO
3324	013102	012737	000600	002504		MOV	#394,\$WC	:1 SECTOR IN 10/15 MODE
3325	013110	012737	043602	002506		MOV	#BUFFER,\$BUF	:BUFFER ADDRESS
3326	013116	012737	010000	002502		MOV	#MODE,DPB+2	:SET THE 'MODE' BIT
3327	013124	112737	000003	002502		MOVB	#WRTSEK,DPB+2	:WRITE COMMAND
3328	013132	004737	022740			JSR	PC,RP11	:DO THE COMMAND
3329	013136	004737	023616			JSR	PC,CONRDY	:WAIT FOR THE COMMAND TO COMPLETE
3330	013142	004737	023172			JSR	PC,SAVRP	:SAVE THE REGISTERS
3331	013146	032737	100000	001340		BIT	#ERR,\$RPCS	:OPERATION GO OK ?
3332	013154	001402				BEQ	3\$:BR IF IT DID
3333	013156	104004				ERROR	4	:SETUP OPERATION TERMINATED WITH AN ERROR
3334	013160	000541				BR	TST13A	:CORRECT THE SECTOR AND BYPASS REST OF TEST
3335	013162	012737	013162	001110	3\$:	MOV	#3\$, \$LPERR	:CHANGE THE LOOP ON ERROR ADDRESS
3336	013170	004737	025606			JSR	PC,CLRP	:CLEAR THE RP11
3337	013174	012701	043602			MOV	#BUFFER,R1	:CLEAR THE INPUT BUFFER
3338	013200	063701	002504			ADD	\$WC,R1	:BEGINNING ADDRESS
3339	013204	013702	002504			MOV	\$WC,R2	:SIZE
3340	013210	005001			4\$:	CLR	R1	:CLEAR THE BUFFER
3341	013212	005302				DEC	R2	:FINISHED ?
3342	013214	001375				BNE	4\$:BR IF NOT
3343	013216	013737	002504	002506		MOV	\$WC,\$BUF	:CHANGE BUFFER ADDRESS
3344	013224	063737	002504	002506		ADD	\$WC,\$BUF	:DOUBLE THE WORD COUNT
3345	013232	062737	043602	002506		ADD	#BUFFER,\$BUF	:ADD BUFFER BASE ADDRESS
3346	013240	112737	000005	002502		MOVB	#RDSEK,DPB+2	:CHANGE COMMAND
3347	013246	004737	022740			JSR	PC,RP11	:READ THE SECTOR
3348	013252	004737	023616			JSR	PC,CONRDY	:WAIT FOR THE OPERATION TO COMPLETE
3349	013256	004737	023172			JSR	PC,SAVRP	:STORE THE REGISTERS
3350	013262	032737	100000	001340		BIT	#ERR,\$RPCS	:DID AN ERROR OCCUR ?
3351	013270	001402				BEQ	5\$:BR IF NOT
3352	013272	104017				ERROR	17	:ERROR OCCURED READING THE TEST SECTOR
3353	013274	000473				BR	TST13A	:BYPASS THE COMPARSION
3354	013276	012701	043602		5\$:	MOV	#BUFFER,R1	:GOOD DATA
3355	013302	013702	002504			MOV	\$WC,R2	:SIZE
3356	013306	013703	002504			MOV	\$WC,R3	:FORM ADDRESS OF TEST DATA
3357	013312	063703	002504			ADD	\$WC,R3	:DOUBLE THE WORD COUNT
3358	013316	062703	043602			ADD	#BUFFER,R3	:ADD THE BUFFER START ADDRESS
3359	013322	005037	013462			CLR	10\$:CLEAR THE COMPARSION ERROR COUNTER
3360	013326	032777	000010	165604	6\$:	BIT	#SW03,\$SWR	:COMPARE THE DATA ?
3361	013334	001053				BNE	TST13A	:BR IF NOT
3362	013336	022123				CMP	(R1)+,(R3)+	:COMPARE THE DATA
3363	013340	001445				BEQ	9\$:BR IF OK
3364	013342	016137	177776	001124		MOV	-2(R1),\$SDAT	:GOOD DATA
3365	013350	016337	177776	001126		MOV	-2(R3),\$SDAT	:BAD DATA
3366	013356	010337	001122			MOV	R3,\$BDADR	:FORM BAD ADDRESS
3367	013362	162737	000002	001122		SUB	#2,\$BDADR	:DECREMENT IT
3368	013370	005737	013462			TST	10\$:ANY COMPARE ERROR ALREADY ?
3369	013374	001010				BNE	7\$:BR IF SOME
3370	013376	012737	013376	001116		MOV	#,\$ERRPC	:PC FOR ERROR MESSAGE
3371	013404	112737	000102	001114		MOVB	#102,\$ITEMB	:ERROR TABLE INDEX
3372	013412	004737	017232			JSR	PC,TYPERR	:REPORT COMPARSION ERROR
3373	013416	112737	000104	001114	7\$:	MOVB	#104,\$ITEMB	:INDEX OF DATA DISPLAY MESSAGE
3374	013424	004737	017232			JSR	PC,TYPERR	:TYPE THE MESSAGE
3375	013430	005237	013462		8\$:	INC	10\$:INCREMENT COMPARSION ERROR COUNT
3376	013434	023737	001320	013462		CMP	CMP.CT,10\$:COMPARE AGAINST THE LIMIT
3377	013442	103004				BHIS	9\$:BR IF NOT GREATER

```

3378 013444 032777 000020 165466 BIT #SW04,SWR ;CONTINUE COMPARING ANYWAY ?
3379 013452 001404 BEQ TST13A ;BR IF NOT
3380 013454 005302 98: DEC R2 ;FINISHED COMPARING ?
3381 013456 001323 BNE 68 ;BR IF NOT
3382 013460 000401 BR TST13A ;CHANGE MODE OF THE SECTOR
3383 013462 000000 108: .WORD 0 ;COMPARISON ERROR COUNTER
3384 013464 012737 013464 DU1110 TST13A: MOV #TST13A,SLPERR ;CHANGE LOOP ON ERROR ADDRESS
3385 013472 004737 025506 JSR PC,CLRP ;CLEAR THE RP11
3386 013476 105037 002503 CLR8 DPB+3 ;CLEAR THE 'MODE' & 'HDR' BITS
3387 013502 012737 000002 002504 MOV #2,SWC ;CHANGE WORD COUNT
3388 013510 112737 000003 002502 MOV8 #WATSEK,DPB+2 ;CHANGE COMMAND
3389 013516 005037 043602 CLR BUFFER ;SET DATA TO ZERO
3390 013522 005037 043604 CLR BUFFER+2 ;OTHER DATA WORD
3391 013526 012737 043602 002506 MOV #BUFFER,$BUF ;BUFFER ADDRESS
3392 013534 004737 022740 JSR PC,RP11 ;RE-WRITE THE SECTOR
3393 013540 004737 023616 JSR PC,CONROY ;WAIT FOR THE OPERATION TO COMPLETE
3394 013544 004737 023172 JSR PC,SAVAP ;STORE THE REGISTERS
3395 013550 032737 100000 001340 BIT #ERR,$RPCS ;DID THE WRITE FINISH WITHOUT ERROR ?
3396 013556 001401 BEQ EXIT13 ;BR IF IT DID
3397 013560 104060 ERROR 60 ;ERROR RE-WRITING THE SECTOR
3398 013562 000004 EXIT13: SCOPE ;LOOP ?

```

```

*****
;TEST 14 DATA STORAGE/RETRIEVAL TEST

```

```

;THE DATA TEST PERFORMS DATA STORAGE AND RETRIEVAL TESTING USING THE CONTROLLER
;AND THE CURRENTLY SELECTED DRIVE. UNLESS ALTERED BY THE OPERATOR, THE
;TEST WILL WRITE AND WRITE CHECK THE ENTIRE DISK PACK USING ALL 16 DATA
;PATTERNS IN A ROTATING MANNER. THE BUFFER SIZE USED FOR THE WRITE AND
;WRITE CHECK ORDERS IS DETERMINED BY THE AVAILABLE MEMORY ON THE SYSTEM
;(MINUS THE SPACE REQUIRED BY THE PROGRAM LOADER). THE PROGRAM WILL USE
;THE AVAILABLE MEMORY SIZE OR BK, WHICHEVER IS LESS, AS THE BUFFER
;SIZE.
;AFTER THE PACK HAS BEEN WRITTEN, THE TEST WILL READ AND COMPARE THE
;DATA. THE BUFFER SIZE USED FOR THE READ IS THE SAME SIZE AS THAT USED
;FOR THE WRITE/WRITE CHECK PART OF THE TEST. FOR THE READ, THE TEST WILL
;ROTATE THE READ BUFFER THROUGH MEMORY SO THAT ALL OF THE AVAILABLE MEMORY
;IS USED BY THE READ AND COMPARE OPERATIONS. THE TEST USES MEMORY MANAGEMENT
;ON SYSTEMS WITH MORE THAN 28K. (MEMORY MANAGEMENT WILL NOT BE USED IF
;ON MEMORY MANAGEMENT SYSTEMS WITH 28K OR LESS.)

```

```

;THE DATA PATTERNS USED BY THE DATA TEST ARE GIVEN BELOW:

```

PAT 0	PAT 1	PAT 2	PAT 3	PAT 4	PAT 5	PAT 6	PAT 7
000000	000001	177776	030000	000000	052525	007417	026455
000000	000003	177774	000000	010421	052525	007417	026455
000000	000007	177770	000000	021042	052525	007417	026455
000000	000017	177760	177777	031463	125252	170360	151322
000000	000037	177740	177777	042104	125252	170360	151322
000000	000077	177700	177777	052525	125252	170360	151322
000000	000177	177600	000000	063146	052525	007417	026455
000000	000377	177400	000000	073567	052525	007417	026455
000000	000777	177000	177777	104210	125252	170360	151322
000000	001777	176000	177777	114631	125252	170360	151322

```

3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433

```

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

000000	003777	174000	000000	125252	052525	007417	026455
000000	007777	170000	177777	135673	125252	170360	151322
000000	017777	160000	000000	146314	052525	007417	026455
000000	037777	140000	177777	156735	125252	170360	151322
000000	077777	100000	177777	167356	125252	170360	151322
000000	177777	000000	000000	177777	052525	007417	026455
PAT 8	PAT 9	PAT 10	PAT 11	PAT 12	PAT 13	PAT 14	PAT 15
077577	000001	177776	172666	077777	177777	000000	177777
077577	000002	177775	155555	137777	177777	000000	177777
077577	000004	177773	172666	157777	177777	177777	000000
077577	000010	177767	155555	167777	177777	177777	000000
077577	000020	177757	172666	173777	177777	177777	000000
077577	000040	177737	155555	175777	177777	177777	000000
077577	000100	177677	172666	176777	177777	177777	000000
077577	000200	177577	155555	177377	177777	177777	000000
077577	000400	177377	172666	177577	177777	177777	000000
077577	001000	176777	155555	177677	177777	177777	000000
077577	002000	175777	172666	177377	177777	177777	000000
077577	004000	177377	172666	177577	177777	177777	000000
077577	001000	176777	155555	177677	177777	177777	000000
077577	002000	175777	172666	177377	177777	177777	000000
077577	004000	177377	172666	177577	177777	177777	000000
077577	010000	176777	155555	177677	177777	177777	000000
077577	020000	175777	172666	177377	177777	177777	000000
077577	040000	177377	172666	177577	177777	177777	000000
077577	010000	176777	155555	177677	177777	177777	000000
077577	020000	175777	172666	177377	177777	177777	000000
077577	040000	177377	172666	177577	177777	177777	000000
077577	100000	077777	155555	177776	177777	177777	000000

TST14:

```

BIT      BITS+(<STN*2>),TSTNMS      : IS THIS TEST SELECTED
BNE      +6                          : BR IF YES
JMP      TST15                        : GO TO THE NEXT TEST
MOV      @TEST14,$LPERR               : SETUP THE ERROR LOOP ADDRESS
MOV      @TST14,$LPAOR               : SETUP THE SCOPE LOOP ADDRESS
MOV      @14,$TSTNM                  : SETUP TEST NUMBER AND
                                        : CLEAR THE ERROR FLAG ($ERFLG)
MOV      $TSTNM,$DISPLAY              : LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
MOV      @EXIT14,BYPASS              : SETUP BYPASS ADDRESS
MOV      RPAOR,R4                    : RP11E BUS ADDRESS
CLRB     DPB+1                        : CLEAR ANY EXTENDED MEMORY BITS
CLRB     DPB+3                        : CLEAR 'MODE' & 'HOR' BITS
MOV      @STACK,SP                   : SETUP THE STACK POINTER
BIT      @SW06,$SWR                  : USE MEMORY MANAGEMENT ?
BNE      2$                          : BR IF NOT
TST      $KT11                       : MEMORY MANAGEMENT AVAILABLE ?
BPL      2$                          : BR IF NOT
MOV      @-1,$MMACTV                 : SET MEMORY MANAGEMENT ACTIVE INDICATOR
JSR      PC,MOVLDR                   : MOVE THE LOADER
MOV      (SP)+,$LOMEM                : BEGINNING ADDRESS OF BUFFER SPACE
MOV      $MAXMEM,$HMEM               : HIGHEST MEMORY ADDRESS USED BY THE TEST
MOV      $MAXMEM+2,$HMEM+2          : UPPER ADDRESS BITS
BR       3$                          : CONTINUE
CLR      $MMACTV                     : CLEAR MEMORY MANAGEMENT ACTIVE INDICATOR
CLR      $HMEM+2                     : CLEAR UPPER 'HMEM' ADDRESS BITS
MOV      $MEMSIZ,$HMEM               : HIGHEST NON MEMORY MANAGEMENT LOCATION
MOV      @BUFFER,$LOMEM              : DEFAULT BUFFER STARTING ADDRESS

```

1\$:

2\$:

013564			
013564	033737	001430	001276
013572	001002		
013574	000137	014620	
013600	012737	014056	001110
013606	012737	013564	001106
013614	012737	000014	001102
013622	013777	001102	165312
013630	012737	014606	001200
013636	013704	001324	
013642	105037	002501	
013646	105037	002503	
013650	012706	001100	
013656	032777	000100	165254
013664	001021		
013666	005737	022202	
013672	100016		
013674	012737	177777	001222
013702	004737	023300	
013706	012637	001240	
013712	013737	001230	001234
013720	013737	001232	001236
013726	000412		
013730	005037	001222	
013734	005037	001236	
013740	013737	001302	001234
013746	012737	043602	001240

3490	013754	005037	001242		35:	CLR	L0MEM+2	: CLEAR UPPER STORAGE WORD
3491	013760	013737	001240	001224		MOV	L0MEM,ACTMEM	: STARTING ADDRESS OF FIRST BUFFER
3492	013766	013737	001242	001226		MOV	L0MEM+2,ACTMEM+2	: UPPER ADDRESS BITS
3493	013774	013737	001256	001254		MOV	WC14,BLKS14	: CONVERT THE WORD COUNT TO A SECTOR COUNT
3494	014002	000337	001254			SWAB	BLKS14	: RIGHT JUSTIFY THE SECTOR COUNT
3495	014006	105037	001255			CLRB	BLKS14+1	: CLEAR THE RESIDUE
3496	014012	005237	001254			INC	BLKS14	: CHANGE TO ABSOLUTE COUNT
3497	014016	013746	001254			MOV	BLKS14,-(SP)	: CONVERT BLOCK COUNT TO SECTOR/TRACK INCREMENTS
3498	014022	005037	001252			CLR	INCTRK	: CLEAR TRACK STORAGE
3499	014026	005037	001250			CLR	INCSEC	: CLEAR SECTOR STORAGE
3500	014032	162716	000012		45:	SUB	#10.,(SP)	: SUBTRACT A FULL SECTOR
3501	014036	100403				BMI	55	: BR IF MINUS
3502	014040	005237	001252			INC	INCTRK	: COUNT THE TRACK
3503	014044	000772				BR	45	: CONTINUE
3504	014046	062716	000012		55:	ADD	#10.,(SP)	: CORRECT THE SECTOR RESIDUE
3505	014052	012637	001250			MOV	(SP)+,INCSEC	: SECTOR INCREMENT VALUE
3506	014056	012706	001100		TEST14:	MOV	#STACK,SP	: LOAD THE STACK POINTER
3507	014062	004737	023734			JSR	PC,SECNBR	: GET THE NUMBER OF SECTORS ON THE PACK
3508								: BEING USED
3509	014066	113737	001272	002512	15:	MOVSB	SSEC,\$SEC	: STARTING SECTOR ADDRESS
3510	014074	113737	001270	002513		MOVSB	STRK,\$TRK	: STARTING TRACK ADDRESS
3511	014102	013737	001266	002510		MOV	SCYL,\$CYL	: STARTING CYLINDER ADDRESS
3512	014110	013737	001240	002506		MOV	L0MEM,\$BUF	: BUFFER ADDRESS
3513	014116	013737	001256	002504		MOV	WC14,\$WC	: STARTING WORD COUNT
3514	014124	012737	000017	001204		MOV	#15,PATNUM	: PRESET THE PATTERN INDEX
3515	014132	005737	001260			TST	OPRSEL	: USING AN OPERATOR SPECIFIED ADDRESS ?
3516	014136	001005				BNE	25	: BR IF YES
3517	014140	163737	001254	001244		SUB	BLKS14,PAKSIZ	: DECREMENT THE SECTOR COUNT FOR THE
3518	014146	005637	001246			SBC	PAKSIZ+2	: FIRST OPERATION
3519	014152	004737	024002		25:	JSR	PC,PATSEL	: SELECT THE PATTERN
3520	014156	012737	014156	001110	35:	MOV	#35,\$LPERR	: LOOP ON ERROR ADDRESS
3521	014164	004737	025606			JSR	PC,CLRP	: CLEAR THE RP11
3522	014170	017746	164744			MOV	@SWR,-(SP)	: INHIBIT WRITE AND
3523	014174	005116				COM	(SP)	: WRITE CHECK ?
3524	014176	032726	000003			BIT	#SW00!SW01,(SP)+	: YES--BRANCH
3525	014202	001453				BEQ	TST14A	: YES--BRANCH
3526	014204	004737	024364			JSR	PC,SETBUF	: MOVE DATA PATTERN INTO THE BUFFER
3527	014210	032777	000001	164722		BIT	#SW00,@SWR	: INHIBIT WRITE ?
3528	014216	001016				BNE	45	: YES--BRANCH
3529	014220	112737	000003	002502		MOVSB	#WRTSEK,DPB+2	: COMMAND=WRITE
3530	014226	004737	022740			JSR	PC,RP11	: DO THE COMMAND
3531	014232	004737	023616			JSR	PC,CONRDY	: WAIT FOR THE COMMAND TO COMPLETE
3532	014236	004737	023172			JSR	PC,SAVRP	: STORE THE REGISTERS
3533	014242	032737	100000	001340		BIT	#ERR,\$RPCS	: ERROR OCCURED ?
3534	014250	001401				BEQ	45	: BR IF NOT
3535	014252	104061				ERROR	61	: REPORT THE ERROR
3536	014254	032777	000002	164656	45:	BIT	#SW01,@SWR	: INHIBIT WRITE CHECK ?
3537	014262	001020				BNE	55	: YES--BRANCH
3538	014264	004737	025606			JSR	PC,CLRP	: CLEAR THE RP11
3539	014270	112737	000007	002502		MOVSB	#WRTCHK,DPB+2	: COMMAND=WRITE CHECK
3540	014276	004737	022740			JSR	PC,RP11	: DO THE WRITE CHECK
3541	014302	004737	023616			JSR	PC,CONRDY	: WAIT FOR THE COMMAND TO FINISH
3542	014306	004737	023172			JSR	PC,SAVRP	: STORE THE REGISTERS
3543	014312	032737	100000	001340		BIT	#ERR,\$RPCS	: ERROR ?
3544	014320	001401				BEQ	55	: BR IF NOT
3545	014322	104005				ERROR	5	: REPORT THE WRITE CHECK ERROR

E06

NO-11-DZRPY-B, RP11-E FUNCTIONAL LOGIC & R/W TEST
DZRPYB.CMB T14 DATA STORAGE/RETRIEVAL TEST

MACY11 27(732) 02-NOV-76 14:50 PAGE 70

```

3546 014324 004737 024046 58: JSR PC,INCAUR ; INCREMENT THE ADDRESS
3547 014330 000710 BR ; CONTINUE
3548 014332 004737 023734 TST14A: JSR PC,SECNR ; GET NUMBER OF SECTORS ON THE DRIVE
3549 ; BEING TESTED
3550 014236 113737 001272 002512 MOVB SSEC,$SEC ; STARTING SECTOR
3551 014344 113737 001270 002513 MOVB STRK,STRK ; STARTING TRACK
3552 014352 013737 1266 002510 MOV SCYL,$CYL ; STARTING CYLINDER
3553 014360 013737 001256 002504 MOV MC14,$MC ; INITIAL WORD COUNT
3554 014366 012737 000017 001204 MOV #15,PATNUM ; PRESET PATTERN NUMBER
3555 014374 005737 001260 TST OPSEL ; USING AN OPERATOR SPECIFIED PATTERN
3556 014400 001005 BNE IS ; BR IF YES
3557 014402 163737 001254 001244 SUB BLKS14,PAKSIZ ; DECREMENT THE SECTOR COUNT FOR
3558 014410 005637 001246 SBC PAKSIZ+2 ; THE FIRST OPERATION
3559 014414 005037 001314 18: CLR RETRY ; CLEAR THE RETRY COUNTER
3560 014420 004737 024002 JSR PC,PATSEL ; SELECT A PATTERN
3561 014424 004737 024434 JSR PC,GETBUF ; GENERATE A BUFFER ADDRESS
3562 014430 012737 014430 001110 28: MOV #25,$LPERR ; LOOP ON ERROR ADDRESS
3563 014436 004737 025606 JSR PC,CLRP ; CLEAR THE RP11
3564 014442 032777 000004 164470 BIT #SW02,$SWR ; INHIBIT READ DATA AND SOFTWARE COMPARE ?
3565 014450 001056 BNE EXIT14 ; YES--BRANCH
3566 014452 112737 000005 002502 MOVB #ROSEK,DPB+2 ; LOAD READ COMMAND
3567 014460 004737 022740 JSR PC,RP11 ; START THE READ
3568 014464 004737 023616 JSR PC,CONRDY ; WAIT FOR THE READ TO COMPLETE
3569 014470 004737 023172 JSR PC,SAVRP ; STORE THE REGISTERS
3570 014474 032737 100000 001340 BIT #ERR,$RPCS ; ERROR ?
3571 014502 001420 BEQ 58 ; BR IF NOT
3572 014504 005737 001314 TST RETRY ; RETRY ATTEMPT ?
3573 014510 001404 BEQ 38 ; BR IF NOT
3574 014512 032777 000040 164420 BIT #SW05,$SWR ; DISPLAY ALL RETRY ATTEMPTS
3575 014520 001401 BEQ 48 ; BR IF NOT
3576 014522 104062 38: ERROR 62 ; REPORT THE ERROR
3577 014524 005237 001314 48: INC RETRY ; INCREMENT THE RETRY COUNT
3578 014530 023737 001314 001316 CMP RETRY,LRETRY ; RETRY LIMIT ?
3579 014536 101734 BLOS 28 ; ER IF NOT
3580 014540 104110 ERROR 110 ; REPORT UNSUCCESSFUL RETRY
3581 014542 000410 BR 68 ; CONTINUE
3582 014544 005737 001314 58: TST RETRY ; COMING FROM A SUCCESSFUL RETRY ?
3583 014550 001405 BEQ 68 ; ER IF NOT
3584 014552 112737 000111 001114 MOVB #111,$ITEMB ; RETRY COUNT MESSAGE NUMBER
3585 014560 004737 017232 JSR PC,TYPERR ; REPORT THE RETRY COUNT
3586 014564 032777 000010 164346 68: BIT #SW03,$SWR ; COMPARE THE DATA ?
3587 014572 001002 BNE 78 ; NO--BRANCH
3588 014574 004737 024546 JSR PC,DATCMP ; YES--DO IT
3589 014600 004737 024046 78: JSR PC,INCAUR ; INCREMENT THE DISK ADDRESS
3590 014604 000703 BR 18 ; CONTINUE
3591 014606 000004 EXIT14: SCOPE ; LOOP ?
3592 014610 004737 023404 JSR PC,RESLDR ; RESTORE THE LOADER IF IT WAS MOVED
3593 014614 000137 016606 JMP SEOP ; GO TO END OF TEST

```

; *TEST 15 POWER FAIL TEST

; *TEST THE ABILITY OF THE RP11E TO SENSE POWER FAILURE AND FOR THE DRIVES
; *TO RETRACT THEIR HEADS. WHEN POWER IS TURNED ON AGAIN
; *THE CYLINDER ADDRESS IS TESTED FOR ZERO. AFTER TYPING THE MESSAGE
; *REQUESTING POWER TO BE TURNED OFF, THE PROGRAM GOES INTO A LOOP

3601

F06

MD-11-DZRPY-B, RP11-E FUNCTIONAL LOGIC & R/W TEST
DZRPYB.CMB T15 POWER FAIL TEST

MACY11 27(732) 02-NOV-76 14:50 PAGE 71

;*READING FROM THE SELECTED DISK DRIVE. AFTER POWER IS RESTORED,
;*MEMORY IS CHECKED TO SEE THAT THE DISK DID NOT TRANSFER ANY DATA
;*TO MEMORY WHILE POWER WAS GOING DOWN.

::*****

```

TST15: BIT BITS+(<STN*2>),TSTNMS ;IS THIS TEST SELECTED
      BNE +6 ;BR IF YES
      JMP TST16 ;GO TO THE NEXT TEST
      MOV #TST15,$LPERR ;SETUP THE ERROR LOOP ADDRESS
      MOV #TST15,$LPAOR ;SETUP THE SCOPE LOOP ADDRESS
      MOV #15,$TSTNM ;SETUP TEST NUMBER AND
      ;CLEAR THE ERROR FLAG (SERFLG)
      MOV $TSTNM,$DISPLAY ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
      MOV #EXIT15,BYPASS ;SETUP BYPASS ADDRESS
      MOV RPAOR,R4 ;RPI1E BUS ADDRESS
      CLRB DPB+1 ;CLEAR ANY EXTENDED MEMORY BITS
      CLRB DPB+3 ;CLEAR 'MODE' & 'HOR' BITS
      TYPE MSG15A ;'POWER FAIL TEST'
      JSR PC,MOVLDR ;MOVE THE LOADER
      MOV (SP)+,LOMEM ;BUFFER STARTING ADDRESS
      CLR LOMEM+2 ;CLEAR THE UPPER BITS JUST TO BE NICE
TEST15: MOV #STACK,SP ;SETUP THE STACK POINTER
      JSR PC,CLRP ;CLEAR THE RPI1
      MOV #TST15B,$R24 ;SET UP POWER FAIL VECTOR
      MOV #R7,$R26 ;LOCKOUT INTERRUPTS
      MOV LOMEM,R0 ;START OF FREE MEMORY SPACE
      IS: MOV #125252,(R0)+ ;FILL MEMORY WITH CHECKERBOARD
      CMP R0,MEMSIZ ;PATTERN
      BLOS IS ;BR IF NOT FINISHED
      MOV MAXCYL,$CYL ;LOAD MAXIMUM CYLINDER
      CLR $SEC ;CLEAR SECTOR/TRACK
      MOV #256,$WC ;WORD COUNT = 1 SECTOR
      MOV LOMEM,$BUF ;BUFFER ADDRESS
      MOVB #WRTSEK,DPB+2 ;WRITE OPERATION
      JSR PC,RPI1 ;START THE COMMAND
      JSR PC,CONRDY ;WAIT FOR THE CONTROLLER TO BE READY
      JSR PC,SAVRP ;STORE THE REGISTERS
      BIT #ERR,$RPCS ;ERROR DURING OPERATION ?
      BEQ 2$ ;BR IF NOT
      ERROR 4 ;REPORT THE ERROR
      TYPE MSG15B ;'TURN CPU POWER OFF AFTER MESSAGE'
      JSR PC,CLRP ;CLEAR THE RPI1
      MOVB #R0SEK,DPB+2 ;READ COMMAND
      JSR PC,RPI1 ;INITIATE THE READ
      JSR PC,CONRDY ;WAIT FOR CONTROLLER READY
      BR 3$ ;LOOP, READING THE DISK
  
```

;*AFTER MACHINE IS POWERED DOWN AND UP, CONTROL IS TRANSFERRED HERE.

```

TST15A: JSR PC,RPINIT ;INITIALIZE THE RPI1E
      TYPE MSG15D ;'PROGRAM WILL LOOP, WAITING FOR DRIVE TO COME ONLINE'
      IS: MOV #15,$LPERR ;SETUP LOOP ON ERROR ADDRESS
      MOV #2500,$STALL ;LOAD STALL VALUE
      JSR PC,STALL ;STALL FOR 2500 MILLISECONDS
      MOV CHKDRV,R1 ;DRIVE ADDRESS
  
```

```

3602
3603
3604
3605
3606
3607 014620
3608 014620 033737 001432 001276
3609 014626 001002
3610 014630 000137 015362
3611 014634 012737 014726 001110
3612 014642 012737 014620 001106
3613 014650 012737 000015 001102
3614
3615 014656 013777 001102 164256
3616 014664 012737 015272 001200
3617 014672 013704 001324
3618 014676 105037 002501
3619 014702 105037 002503
3620 014706 104401 027357
3621 014712 004737 023300
3622 014716 012637 001240
3623 014722 005037 001242
3624 014726 012706 001100
3625 014732 004737 025606
3626 014736 012737 015304 000024
3627 014744 012737 000340 000026
3628 014752 013700 001240
3629 014756 012720 125252
3630 014762 020037 001302
3631 014766 101773
3632 014770 013737 001210 002510
3633 014776 005037 002512
3634 015002 012737 000256 002504
3635 015010 013737 001240 002506
3636 015016 112737 000003 002502
3637 015024 004737 022740
3638 015030 004737 023616
3639 015034 004737 023172
3640 015040 032737 100000 001340
3641 015046 001401
3642 015050 104004
3643 015052 104401 027402
3644 015056 004737 027406
3645 015072 112737 000005 002502
3646 015070 004737 027404
3647 015074 004737 023616
3648 015100 000766
3649
3650
3651
3652 015102 004737 022536
3653 015106 104401 027576
3654 015112 012737 015112 001110
3655 015120 012737 002500 001220
3656 015126 004737 023134
3657 015132 013701 001176
  
```

```

3658 015136 004737 022636 JSR PC,DRVINT ;CHECK THE DRIVE'S STATUS
3659 015142 105761 001370 TSTB DRVSTA(R1) ;IS THE DRIVE ONLINE ?
3660 015146 001761 BEQ 1$ ;CONTINUE WAITING FOR DRIVE TO COME ONLINE
3661 015150 003002 BGT 2$ ;BR IF IT IS ONLINE
3662 015152 104112 ERROR 112 ;DRIVE UNSAFE AFTER POWER FAIL
3663 015154 000446 BR EXIT15 ;DRIVE UNSAFE, EXIT
3664 015156 113764 001176 000005 2$: MOVB CHKDRV,RPCS+1(R4) ;SELECT THE DRIVE
3665 015164 004737 023172 JSR PC,SAVRP ;STORE THE REGISTERS
3666 015170 005737 001354 TST $SUCA ;SELECTED CYLINDER SHOULD BE ZERO
3667 015174 001404 BEQ 3$ ;BR IF IT DID
3668 015176 012737 015206 001164 MOV #3$,SESCAPE ;ESCAPE TO 3$ ON ERROR
3669 015204 104063 ERROR 63 ;DRIVE DID NOT RETURN TO CYLINDER 0
3670 ;ON POWER FAILURE
3671 015206 3$:
3672 015206 012737 015266 001164 MOV #6$,SESCAPE ;ESCAPE TO 6$ ON ERROR
3673 015214 013700 001240 MOV LOMEM,R0 ;BUFFER ADDRESS
3674 015220 022720 125252 4$: CMP #125252,(R0)+ ;DID MEMORY RETAIN PATTERN ON POWER FAIL ?
3675 015224 001004 BNE 5$ ;BRANCH IF NOT
3676 015226 020037 001302 CMP R0,MEMSIZ ;END OF MEMORY ?
3677 015232 001372 BNE 4$ ;BR IF NOT
3678 015234 000414 BR 6$ ;EXIT
3679 015236 012737 125252 001124 5$: MOV #125252,$GDDAT ;EXPECTED MEMORY CONTENTS
3680 015244 016037 177776 001126 MOV -2(R0),$BODAT ;ACTUAL MEMORY CONTENTS
3681 015252 010037 001122 MOV R0,$BODADR ;ADDRESS
3682 015256 162737 000002 001122 SUB #2,$BODADR ;CORRECT IT
3683 015264 104064 ERROR 64 ;CONTENTS OF MEMORY CHANGED AFTER POWER FAIL
3684 015266 104401 027543 6$: TYPE ,MSG15C ;'END OF POWER FAIL TEST'
3685 015272 000004 EXIT15: SCOPE ;LOOP ?
3686 015274 004737 023404 JSR PC,RESLDR ;RESTORE THE LOADER
3687 015300 000137 003676 JMP START2 ;RETURN TO 'CONVERSATION MODE' ENTRANCE
3688
3689 ;POWER FAIL TRAP HANDLER
3690
3691 015304 012737 015314 000024 TST15B: MOV #TST15C,#24 ;SET UP POWER UP VECTOR
3692 015312 000774 BR TST15B ;LOOP
3693
3694 ;POWER UP TRAP HANDLER
3695
3696 015314 012737 000026 000024 TST15C: MOV #26,#24 ;RESTORE TRAP CATCHER
3697 015322 005037 000026 CLR #26 ;CATCHER HALT
3698 015326 012706 001100 MOV #STACK,SP ;SETUP STACK POINTER
3699 015332 005037 177776 CLR PS ;RETURN TO PRIORITY ZERO
3700 015336 012737 011610 001220 MOV #5000,STALLT ;LOAD STALL VALUE
3701 015344 004737 023134 JSR PC,STALL ;STALL FOR 5000. MILLISECONDS
3702 015350 004737 020374 JSR PC,$TKINT ;INITIALIZE THE TTY KEYBOARD
3703 015354 013704 001324 MOV RPAOR,R4 ;RPIIE ADDRESS
3704 015360 000650 BR TST15A ;VERIFY POWER DOWN SEQUENCE
3705
3706
3707 ;*****
3708 ;*TEST 16 HEAD ALIGNMENT ROUTINE
3709
3710 ;*THIS ROUTINE PERFORMS HEAD SELECTION AS DIRECTED FROM THE KEYBOARD.
3711 ;*THE ALIGNMENT OF THE SELECTED HEAD MAY BE CHECKED OR ADJUSTED.
3712
3713 ;*****

```

```

3714 015362          TST16:
3715 015362 033737 001434 001276 BIT      BITS+(<STN*2>),TSTNMS  IS THIS TEST SELECTED
3716 015370 001002          .+6          ;BR IF YES
3717 015372 000137 015614          JMP      TST17          ;GO TO THE NEXT TEST
3718 015376 012737 015450 001110 MOV     @TEST16,$LPERR  ;SETUP THE ERROR LOOP ADDRESS
3719 015404 012737 015362 001106 MOV     @TST16,$LPADR  ;SETUP THE SCOPE LOOP ADDRESS
3720 015412 012737 000016 001102 MOV     @16,$TSTNM    ;SETUP TEST NUMBER AND
3721          ;CLEAR THE ERROR FLAG (SERFLG)
3722 015420 013777 001102 163514 MOV     $TSTNM,@DISPLAY ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
3723 015426 012737 015610 001200 MOV     @EXIT16,BYPASS ;SETUP BYPASS ADDRESS
3724 015434 013704 001324          MOV     RPAADR,R4     ;RP11E BUS ADDRESS
3725 015440 105037 002501          CLR     DPB+1        ;CLEAR ANY EXTENDED MEMORY BITS
3726 015444 105037 002503          CLR     DPB+3        ;CLEAR 'MODE' & 'WDR' BITS
3727 015450 012706 001100          TEST16: MOV    @STACK,SP    ;SETUP THE STACK POINTER
3728 015454 1C4401 027667          TYPE    ,MSG16A      ;'HEAD ALIGNMENT ROUTINE'
3729 015460 000410          BR      3$          ;GET THE HEAD ADDRESS WITHOUT THE SW07 NONSENSE
3730 015462 032777 000200 163450 1$: BIT    @SW07,@SWR    ;SWITCH 7 SET ?
3731 015470 001774          BEQ     1$          ;BR IF NOT
3732 015472 032777 000200 163440 2$: BIT    @SW07,@SWR    ;SWITCH 7 RESET ?
3733 015500 001374          BNE     2$          ;BR IF NOT
3734 015502 104401 030201 3$: TYPE    ,MSG16B      ;'ENTER HEAD ADDRESS (IN OCTAL)'
3735 015506 104411          RDLIN          ;GET THE ENTRY
3736 015510 012601          MOV     (SP)+,R1     ;STARTING ADDRESS OF INPUT ASCII STRING
3737 015512 004037 027062          JSR     RO,CK.NUM    ;CHECK THE NUMBER
3738 015516 015502          3$          ;ILLEGAL INPUT
3739 015520 015524          4$          ;TERMINATED WITH A 'CR'
3740 015522 015502          3$          ;NO ENTRY - 'CR' ONLY
3741 015524 122702 000024 4$: CMPB  @20.,R2     ;VALID ?
3742 015530 101764          BLOS   3$          ;BR IF NOT
3743 015532 110237 002513          MOVB   R2,$TRK      ;LOAD NEW HEAD ADDRESS
3744 015536 136237 001360 001304 BITB   ATABIT(R2),DRVTP ;RPO2 OR RPO3 ?
3745 015544 001004          BNE     5$          ;BR IF RPO3
3746 015546 012737 000111 002510 MOV     @73.,$CYL    ;RPO2 ALIGNMENT CYLINDER
3747 015554 000403          BR      6$          ;CONTINUE
3748 015556 012737 000222 002510 5$: MOV   @146.,$CYL    ;RPO3 ALIGNMENT CYLINDER
3749 015564 112737 000011 002502 6$: MOVB  @SEEK,DPB+2  ;SEEK COMMAND
3750 015572 004737 025606          JSR    PC,CLAP      ;CLEAR THE RP11
3751 015576 004737 022740          JSR    PC,RP11     ;SEEK AND LOAD THE HEAD
3752 015602 004737 023500          JSR    PC,DRVROY   ;WAIT FOR THE DRIVE TO BECOME READY
3753 015606 000725          BR      1$          ;WAIT FOR SWITCH 7 TO BE TOGGLED
3754 015610 000000          EXIT16: HALT      ;'EXIT' FOR COMPATABILITY WITH OTHER TESTS
3755 015612 000776          BR      -2         ;LOCK THE HALT
3756
3757
3758          ;*****
3759          ;*TEST 17      RP11E/DRIVE CONTROL PANEL SWITCH TEST
3760
3761          ;*THIS TEST TEST THE 'ENABLE/DISABLE' AND 'READ ONLY' SWITCHES ON
3762          ;*THE DRIVE CONTROL PANEL AND TESTS THE 'WRITE LOCKOUT' AND 'LOR'
3763          ;*SWITCHES ON THE RP11E/DRIVE CONTROL PANEL.
3764
3765          ;*****
3766 015614          †TST17:
3767 015614 033737 001436 001276 BIT      BITS+(<STN*2>),TSTNMS  IS THIS TEST SELECTED
3768 015622 001002          BNE     .+6          ;BR IF YES
3769 015624 000137 016606          JMP     @SEOP        ;GO TO END OF TEST

```

```

3770 015630 012737 015706 001110      MOV      #TEST17,$LPERR      ;SETUP THE ERROR LOOP ADDRESS
3771 015636 012737 015614 001106      MOV      #TST17,$LPADR      ;SETUP THE SCOPE LOOP ADDRESS
3772 015644 012737 000017 001102      MOV      #17,$STSTNM        ;SETUP TEST NUMBER AND
3773                                     ;CLEAR THE ERROR FLAG ($ERFLG)
3774 015652 013777 001102 163262      MOV      $STSTNM,$DISPLAY    ;LOAD THE TEST NUMBER INTO THE DISPLAY REGISTER
3775 015660 012737 016574 001200      MOV      #EXIT17,BYPASS      ;SETUP BYPASS ADDRESS
3776 015666 013704 001324             MOV      RPAOR,R#A           ;RP11E BUS ADDRESS
3777 015672 105037 002501             CLRB     DPB+1              ;CLEAR ANY EXTENDED MEMORY BITS
3778 015676 105037 002503             CLRB     DPB+3              ;CLEAR 'MODE' & 'MOR' BITS
3779 015702 104401 030243             TYPE     ,MSG17A            ;'RP11E CONTROL PANEL SWITCH TEST'
3780 015706 012706 001100      TEST17: MOV     $STACK,SP      ;SETUP THE STACK POINTER
3781
3782                                     ;CHECK THE ENABLE/DISABLE SWITCH ON THE DRIVE
3783
3784 015712 104401 030306      2$:      TYPE     ,MSG17B ;'TYPE A CR TO START TEST AFTER EACH SETUP'
3785 015716 104401 030365      TYPE     ,MSG17C           ;'DISABLE DRIVE #'
3786 015722 013746 001176      MOV      CHKDRV,-(SP)      ;SAVE CHKDRV FOR TYPEOUT
3787                                     ;TYPE DRIVE NUMBER
3788 015726 104403             TYPOS                    ;GO TYPE--OCTAL ASCII
3789 015730             .BYTE     1              ;TYPE 1 DIGIT(S)
3790 015731             .BYTE     1              ;TYPE LEADING ZEROS
3791 015732 104411             ROLIN                    ;WAIT FOR THE 'CR'
3792 015734 012737 015734 001110      3$:      MOV      #3$,$LPERR      ;CHANGE LOOP ON ERROR ADDRESS
3793 015742 004737 025606             JSR      PC,CLRP           ;CLEAR THE RP11
3794 015746 113764 001176 000005      MOV      CHKDRV,RPCS+1(R#) ;SELECT DRIVE
3795 015754 004737 023172             JSR      PC,SAVRP          ;SAVE THE RP11 REGISTERS
3796 015760 032737 040000 001334      BIT      #SUOL,$R#PDS      ;IS THE UNIT ON LINE?
3797 015766 001401             BEQ     4$                ;BRANCH IF NOT
3798 015770 104065             ERROR    65               ;SELECTED UNIT ON LINE WITH THE
3799                                     ;DRIVE DISABLED
3800 015772 104401 030410      4$:      TYPE     ,MSG17D           ;'ENABLE DRIVE #'
3801 015776 013746 001176      MOV      CHKDRV,-(SP)      ;SAVE CHKDRV FOR TYPEOUT
3802                                     ;TYPE DRIVE NUMBER
3803 016002 104403             TYPOS                    ;GO TYPE--OCTAL ASCII
3804 016004             .BYTE     1              ;TYPE 1 DIGIT(S)
3805 016005             .BYTE     1              ;TYPE LEADING ZEROS
3806 016006 104411             ROLIN                    ;WAIT FOR THE 'CR'
3807 016010 012737 016010 001110      5$:      MOV      #5$,$LPERR      ;CHANGE THE LOOP ON ERROR ADDRESS
3808 016016 004737 025606             JSR      PC,CLRP           ;CLEAR THE RP11
3809 016022 113764 001176 000005      MOV      CHKDRV,RPCS+1(R#) ;RESELECT THE UNIT
3810 016030 004737 023172             JSR      PC,SAVRP          ;SAVE THE RP11 REGISTERS
3811 016034 032737 040000 001334      BIT      #SUOL,$R#PDS      ;IS THE UNIT ON LINE?
3812 016042 001001             BNE     ROSW              ;BR IF ONLINE
3813 016044 104066             ERROR    66               ;SELECTED UNIT NOT ON LINE WITH
3814                                     ;THE DRIVE ENABLED
3815
3816                                     ;CHECK THE DRIVE'S WRITE LOCKOUT SWITCH
3817
3818 016046 104401 030432      ROSW:   TYPE     ,MSG17E           ;'SET READ ONLY ON DRIVE'
3819 016052 013746 001176      MOV      CHKDRV,-(SP)      ;SAVE CHKDRV FOR TYPEOUT
3820                                     ;TYPE DRIVE NUMBER
3821 016056 104403             TYPOS                    ;GO TYPE--OCTAL ASCII
3822 016060             .BYTE     1              ;TYPE 1 DIGIT(S)
3823 016061             .BYTE     1              ;TYPE LEADING ZEROS
3824 016062 104411             ROLIN                    ;WAIT FOR 'CR'
3825 016064 012737 016064 001110      2$:      MOV      #2$,$LPERR      ;LOAD LOOP ON ERROR ADDRESS
    
```

```

3826 016072 004737 025606 JSR PC,CLRP ;CLEAR THE RP11
3827 016076 113764 001176 000005 MOVB CHKDRV,RPCS+1(R4) ;SELECT THE DRIVE
3828 016104 004737 023172 JSR PC,SAVRP ;SAVE THE RP11 REGISTERS
3829 016110 032737 000400 001334 BIT #SUMP,SRPDS ;IS SELECTED UNIT WRITE PROTECTED ?
3830 016116 001001 BNE 38 ;BR IF IT IS
3831 016120 104067 ERROR 67 ;SELECTED UNIT WRITE PROTECT DID
3832 ;NOT SET WITH READ ONLY SET
3833 016122 104401 030466 38: TYPE ,MSG17F ;SET READ/WRITE ON DRIVE
3834 016126 013746 001176 MOV CHKDRV,-(SP) ;SAVE CHKDRV FOR TYPEOUT
3835 ;TYPE DRIVE NUMBER
3836 016132 104403 TYPOS ;GO TYPE--OCTAL ASCII
3837 016134 001 .BYTE 1 ;TYPE 1 DIGIT(S)
3838 016135 001 .BYTE 1 ;TYPE LEADING ZEROS
3839 016136 104411 ROLIN ;WAIT FOR 'CR'
3840 016140 012737 016140 001110 48: MOV #48,SLPERR ;CHANGE THE LOOP ON ERROR ADDRESS
3841 016146 004737 025606 JSR PC,CLRP ;CLEAR THE RP11
3842 016152 113764 001176 000005 MOVB CHKDRV,RPCS+1(R4) ;RESELECT THE DRIVE
3843 016160 004737 023172 JSR PC,SAVRP ;SAVE THE RP11 REGISTERS
3844 016164 032737 000400 001334 BIT #SUMP,SRPDS ;IS SELECTED UNIT WRITE PROTECTED ?
3845 016172 001401 BEQ WRSW ;BR IF NOT
3846 016174 104070 ERROR 70 ;SELECTED UNIT WRITE PROTECT SET
3847 ;WITH DRIVE WRITE ENABLED
3848
3849 ;TEST THE SETTING OF WRITE LOCKOUT AND THE 'LOA' SWITCHES ON THE
3850 ;RP11E CONTROL PANEL.
3851
3852 016176 104401 030523 WRSW: TYPE ,MSG17G ;'SET WRITE LOCKOUT', ETC
3853 016202 104411 ROLIN ;WAIT FOR 'CR' FROM THE OPERATOR
3854 016204 012737 016204 001110 18: MOV #18,SLPERR ;CHANGE LOOP ON ERROR ADDRESS
3855 016212 005064 000012 CLR RPCA(R4) ;CLEAR THE CYLINDER REGISTER
3856 016216 004737 023172 JSR PC,SAVRP ;SAVE THE RP11 REGISTERS
3857 016222 032737 000400 001334 BIT #SUMP,SRPDS ;IS SELECTED UNIT WRITE PROTECT SET ?
3858 016230 001001 BNE 28 ;BR IF SET
3859 016232 104071 ERROR 71 ;SELECTED UNIT WRITE PROTECT SHOULD
3860 ;BE SET - RPCA EQUAL TO ZERO
3861 016234 012764 000002 000012 28: MOV #2,RPCA(R4) ;LOAD A 2 INTO RPCA
3862 016242 004737 023172 JSR PC,SAVRP ;SAVE THE RP11 REGISTERS
3863 016246 032737 000400 001334 BIT #SUMP,SRPDS ;IS SELECTED UNIT WRITE PROTECTED ?
3864 016254 001401 BEQ WRSW1 ;BR IF NOT PROTECTED
3865 016256 104072 ERROR 72 ;SELECTED UNIT WRITE PROTECT IS SET
3866 ;WITH EQUAL TO 2.
3867
3868 ;CHECK THE CYLINDER 'LOA' SWITCHES
3869
3870 016260 012700 000002 WRSW1: MOV #2,RO ;INITIALIZE SWITCH PATTERN
3871 016264 104401 030640 TYPE ,MSG17H ;'SET FOLLOWING VALUES IN THE
3872 ;CYLINDER LOA SWITCHES'
3873 016270 104401 030740 18: TYPE ,MSG17I ;'SET'
3874 016274 006200 ASR RO ;SETUP PATTERN FOR TYP0UT
3875 016276 010046 MOV RO,-(SP) ;SAVE RO FOR TYPEOUT
3876 ;TYPE THE SWITCH VALUE
3877 016300 104403 TYPOS ;GO TYPE--OCTAL ASCII
3878 016302 003 .BYTE 3 ;TYPE 3 DIGIT(S)
3879 016303 001 .BYTE 1 ;TYPE LEADING ZEROS
3880 016304 006300 ASL RO ;RESTORE THE PATTERN
3881 016306 104411 ROLIN ;WAIT FOR 'CR' FROM THE OPERATOR
    
```

```

3892 016310 012737 016310 001110 2$: MOV #2$, $LPERR ;CHANGE LOOP ON ERROR ADDRESS
3893 016316 010064 000012 000012 3$: MOV R0, RPCA(R4) ;LOAD PATTERN INTO RPCA
3894 016322 004737 023172 000012 4$: JSR PC, SAVRP ;SAVE THE RP11 REGISTERS
3895 016326 032737 000400 001334 BIT #SUMP, $RPDS ;IS UNIT WRITE PROTECT SET ?
3896 016334 001001 000400 001334 BNE 3$ ;BR IF SET
3897 016336 104073 000400 001334 ERROR 73 ;UNIT WRITE PROTECT IS NOT SET
3898 016336 104073 000400 001334 ;RPCA EQUALS CONTENTS OF BOUNDARY
3899 016340 062764 000002 000012 3$: ADD #2, RPCA(R4) ;INCREMENT CYLINDER ADDRESS
3900 016346 012737 016346 001110 4$: MOV #4$, $LPERR ;CHANGE THE LOOP ON ERROR ADDRESS
3901 016354 004737 023172 000012 4$: JSR PC, SAVRP ;SAVE THE RP11 REGISTERS
3902 016360 032737 000400 001334 BIT #SUMP, $RPDS ;IS UNIT WRITE PROTECT SET ?
3903 016366 001401 000400 001334 BEQ 5$ ;BR IF NOT
3904 016370 104074 000400 001334 ERROR 74 ;UNIT WRITE PROTECT IS SET WITH
3905 016372 006300 000400 001334 ;RPCA ONE GREATER THAN BOUNDARY
3906 016374 032700 001000 001000 5$: ASL R0 ;UPDATE TEST PATTERN
3907 016400 001733 001000 001000 BIT #BIT09, R0 ;IS PATTERN EXCEEDED?
3908 016400 001733 001000 001000 BEQ 1$ ;BRANCH IF NOT
3909 ;CHECK DRIVE 'LOA' SWITCHES
3910 WRSW2: CLR CHKDRV ;START WITH DRIVE 0
3911 016402 005037 001176 001176 TYPE ,MSG17J ;'RESET THE CYLINDER LOA SWITCHES
3912 016406 104401 030763 030763 ;'SET THE FOLLOWING VALUES IN THE DRIVE
3913 ;'LOA SWITCHES'
3914 016412 012764 000002 000012 MOV #2, RPCA(R4) ;LOAD CYLINDER ADDRESS
3915 016420 113764 001176 000005 1$: MOV#B CHKDRV, RPCS+1(R4) ;SELECT THE FIRST DRIVE NUMBER
3916 016426 104401 030740 030740 TYPE MSG17I ;'SET SWITCHES TO '
3917 016432 013746 001176 001176 MOV CHKDRV, -(SP) ;SAVE CHKDRV FOR TYPEOUT
3918 016436 104403 000000 000000 TYPOS ;TYPE SWITCH SETTING
3919 016440 001 000000 000000 .BYTE 1 ;GO TYPE--OCTAL ASCII
3920 016441 001 000000 000000 .BYTE 1 ;TYPE 1 DIGIT(S)
3921 016442 104411 000000 000000 ROLIN ;TYPE LEADING ZEROS
3922 016444 012737 016444 001110 2$: MOV #2$, $LPERR ;WAIT FOR THE OPERATOR TO ENTER A 'CR'
3923 016452 004737 023172 001110 2$: JSR PC, SAVRP ;LOOP ON ERROR ADDRESS
3924 016456 032737 000400 001334 BIT #SUMP, $RPDS ;STORE THE RP11 REGISTERS
3925 016464 001401 000400 001334 BEQ 3$ ;IS THE SELECTED DRIVE WRITE PROTECTED ?
3926 016466 104075 000400 001334 ERROR 75 ;BR IF NOT
3927 016470 005737 001176 001176 3$: TST CHKDRV ;'SUMP' SET WHEN DRIVE LOA SWITCHES
3928 016474 001004 000001 001176 4$: BNE 4$ ;'EQUAL THE SELECTED DRIVE'
3929 016476 012737 000001 001176 MOV #1, CHKDRV ;THROUGH HERE THE FIRST TIME ?
3930 016504 000406 000001 001176 BR 5$ ;BR IF NOT
3931 016506 006337 001176 001176 4$: ASL CHKDRV ;SET FOR DRIVE 1
3932 016512 032737 000010 001176 BIT #BIT03, CHKDRV ;CONTINUE
3933 016520 001025 000010 001176 BNE EXIT17 ;SHIFT TEST ADDRESS
3934 016522 104401 030740 001176 5$: TYPE MSG17I ;FINISHED WITH ADDRESSES ?
3935 016526 013746 001176 001176 MOV CHKDRV, -(SP) ;BR IF FINISHED
3936 016532 104403 000000 000000 TYPOS ;'SET SWITCHES TO '
3937 016534 001 000000 000000 .BYTE 1 ;SAVE CHKDRV FOR TYPEOUT
3938 016535 001 000000 000000 .BYTE 1 ;TYPE SWITCH SETTING
3939 016536 104411 000000 000000 ROLIN ;GO TYPE--OCTAL ASCII
3940 016540 012737 016540 001110 6$: MOV #6$, $LPERR ;TYPE 1 DIGIT(S)
;TYPE LEADING ZEROS
;WAIT FOR THE OPERATOR TO ENTER A 'CR'
;CHANGE ERROR LOOP ADDRESS
    
```

```

3938 016546 004737 023172 JSR PC,SAVRP ;STORE THE REGISTERS
3939 016552 032737 000400 001334 BIT #SUMP,SAPDS ;IS THE DRIVE WRITE PROTECTED ?
3940 016560 001001 BNE 7$ BR IF IT IS
3941 016562 104113 ERROR 113 ;'SUMP' NOT SET WHEN DRIVE LOA SWITCHES
3942 ;GREATER THAN SELECTED DRIVE NUMBER
3943 016564 113764 001176 000005 7$: MOVB CHKDRV,RPCS+1(R4) ;SELECT THE NEXT DRIVE
3944 016572 000724 BR 2$ ;CONTINUE WITH THE TEST
3945 016574 000004 EXIT17: SCOPE ;LOOP ?
3946 016576 104401 031122 TYPE MSG17K ;'END OF CONTROL PANEL TEST - RETURN SWITCHES TO NORMAL'
3947 016602 000137 003676 JMP START2 ;RETURN TO THE 'CONVERSATION MODE' ENTRANCE

```

.SBTTL END OF PASS ROUTINE

```

*****
;INCREMENT THE PASS NUMBER ($PASS)
;INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
;IF THERES A MONITOR GO TO IT
;IF THERE ISN'T JUMP TO RESTART

```

```

3957 016606 SEOP:
3958 016606 104401 016614 TYPE 65$ ;:TYPE ASCIZ STRING
3959 016612 000410 BR 64$ ;:GET OVER THE ASCIZ
3960 ;:65$: .ASCIZ <15><12><12>/END OF PASS/
3961 016634 64$:
3962 016634 005737 001312 TST @#DRVSEL ;ANY DRIVES SELECTED
3963 016640 001436 BEQ 1$ ;NO, BRANCH
3964 016642 104401 016650 TYPE 67$ ;:TYPE ASCIZ STRING
3965 016646 000407 BR 66$ ;:GET OVER THE ASCIZ
3966 ;:67$: .ASCIZ / USING DRIVE/
3967 66$:
3968 016666 MOV @#CHKDRV,-(SP) ;:SAVE @#CHKDRV FOR TYPEOUT
3969 016672 104403 TYPOS ;:GO TYPE--OCTAL ASCII
3970 016674 002 ;:TYPE 2 DIGIT(S)
3971 016675 000 ;:SUPPRESS LEADING ZEROS
3972 016676 104401 016704 TYPE 69$ ;:TYPE ASCIZ STRING
3973 016702 000412 BR 68$ ;:GET OVER THE ASCIZ
3974 ;:69$: .ASCIZ / ERRORS DETECTED=/
3975 68$:
3976 016730 MOV @#SERTTL,-(SP) ;:SAVE @#SERTTL FOR TYPEOUT
3977 016734 104402 TYPOC ;:GO TYPE--OCTAL ASCII(ALL DIGITS)
3978 016736 005037 001112 CLR @#SERTTL ;ZERO ERROR TOTAL
3979 016742 005037 001102 CLR $STNM ;ZERO THE TEST NUMBER
3980 016746 005037 001162 CLR $TIMES ;ZERO THE NUMBER OF ITERATIONS
3981 016752 005237 001100 INC $PASS ;INCREMENT THE PASS NUMBER
3982 016756 042737 100000 001100 BIC #100000,$PASS ;DON'T ALLOW A NEG. NUMBER
3983 016764 005327 DEC (PC)+ ;:LOOP?
3984 016766 000001 SEOPCT: .WORD 1
3985 016770 003027 BGT $DOAGN ;:YES
3986 016772 012737 MOV (PC)+,@(PC)+ ;:RESTORE COUNTER
3987 016774 000001 SENDCT: .WORD 1
3988 016776 016766 SEOPCT
3989 017000 104401 017006 TYPE 65$ ;:TYPE ASCIZ STRING
3990 017004 000407 BR 64$ ;:GET OVER THE ASCIZ
3991 ;:65$: .ASCIZ <15><12>/END OF TEST/
3992 64$:
3993 017024 104401 017054 TYPE ,SENULL ;:TYPE NULL CHARACTER

```



```

3994 017030 013700 000042 $GET42: MOV @#42,RO ;:GET MONITOR ADDRESS
3995 017034 001405 BEQ $DOAGN ;:BRANCH IF NO MONITOR
3996 017036 000005 RESET ;:CLEAR THE WORLD
3997 017040 004710 SENDAD: JSR PC,(RO) ;:GO TO MONITOR
3998 017042 000240 NOP ;:SAVE ROOM
3999 017044 000240 NOP ;:FOR
4000 017046 000240 NOP ;:ACT11
4001 017050 $DOAGN: ;:
4002 017050 000137 JMP @PC+ ;:RETURN
4003 017052 004752 SRTNAD: .WORD RESTART
4004 017054 377 377 000 SENULL: .BYTE -1,-1,0 ;:NULL CHARACTER STRING
4005 017060 .EVEN

.SBTTL *** SUBROUTINES ***

.SBTTL ERROR HANDLER ROUTINE

;:*****
;:THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
;:SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
;:AND GO TO TYPERR ON ERROR
;:THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
;:SW15=1 HALT ON ERROR
;:SW13=1 INHIBIT ERROR TYPEOUTS
;:SW10=1 BELL ON ERROR
;:SW09=1 LOOP ON ERROR
;:CALL
;: * ERROR N ;:ERROR=EMT AND N=ERROR ITEM NUMBER

$ERROR:
4025 017060
4026 017060 104407
4027 017062 105237 001103 7$: CKSWR ;:TEST FOR CHANGE IN SOFT-SWR
INCB $ERFLG ;:SET THE ERROR FLAG
BEQ 7$ ;:DON'T LET THE FLAG GO TO ZERO
MOV $STNM,@DISPLAY ;:DISPLAY TEST NUMBER AND ERROR FLAG
BIT #BIT10,@SWR ;:BELL ON ERROR?
BEQ 1$ ;:NO - SKIP
TYPE ,SBELL ;:RING BELL
4032 017106 104401 001166 1$: INC $ERTTL ;:COUNT THE NUMBER OF ERRORS
MOV (SP),$ERRPC ;:GET ADDRESS OF ERROR INSTRUCTION
4033 017112 005237 001112
4034 017116 011637 001116
4035 017122 162737 000002 001116 SUB #2,$ERRPC
4036 017130 117737 161762 001114 MOVB @ERRPC,$ITEMB ;:STRIP AND SAVE THE ERROR ITEM CODE
4037 017136 032777 020000 161774 BIT #BIT13,@SWR ;:SKIP TYPEOUT IF SET
4038 017144 001004 BNE 20$ ;:SKIP TYPEOUTS
4039 017146 004737 017232 JSR PC,TYPERR ;:GO TO USER ERROR ROUTINE
4040 017152 104401 001173 TYPE ,SCRLF

20$:
4041 017156
4042 017156 005777 161756 2$: TST @SWR ;:HALT ON ERROR
BPL 3$ ;:SKIP IF CONTINUE
HALT ;:HALT ON ERROR!
4045 017166 104407 CKSWR ;:TEST FOR CHANGE IN SOFT-SWR
4046 017170 032777 001000 161742 3$: BIT #BIT09,@SWR ;:LOOP ON ERROR SWITCH SET?
BEQ 4$ ;:BR IF NO
4048 017200 013716 001110 MOV $LPERR,(SP) ;:FUDGE RETURN FOR LOOPING
4049 017204 005737 001164 4$: TST $ESCAPE ;:CHECK FOR AN ESCAPE ADDRESS

```



```

4050 017210 001402          BEQ      5$          ;;BR IF NONE
4051 017212 013716 001164   MOV      $ESCAPE,(SP) ;;FUDGE RETURN ADDRESS FOR ESCAPE
4052 017216                5$:          CMP      #SENDAD,2#42 ;;ACT-11 AUTO-ACCEPT?
4053 017216 022737 017040 000042 BNE      6$          ;;BRANCH IF NO
4054 017224 001001                HALT          ;;YES
4055 017226 000000                6$:          RTI          ;;RETURN
4056 017230
4057 017230 000002
4058
4059                ;;*****
4060
4061                ;THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE
4062                ;WHICH ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR
4063                ;TABLE" ($ERRTB), AND REPORTS THE APPROPRIATE INFORMATION
4064                ;CONCERNING THE ERROR.
4065                ;CALL
4066                ;      JSR      PC,TYPERR
4067                ;      RETURN
4068
4069 017232 104412          TYPERR: SAVREG          ;SAVE R0-R5
4070 017234 032777 020000 161676 BIT      #SW13,2$WR    ;INHIBIT TYPEOUTS ?
4071 017242 001105          BNE      20$          ;BR IF YES
4072 017244 005037 001160   CLR      $TMPD        ;CLEAR LOCATION FOR TEST NUMBER
4073 017250 113737 001102 001160 MOVB     $STNM,$TMPD  ;LOAD TEST NUMBER FOR TYPEOUT
4074 017256 005000          CLR      R0           ;CLEAR R0 FOR ERROR NUMBER
4075 017260 113700 001114   MOVB     $ITEMB,R0   ;ERROR NUMBER
4076 017264 005300          DEC      R0           ;FORM INDEX FOR ERROR TABLE
4077 017266 006300          ASL      R0
4078 017270 006300          ASL      R0
4079 017272 006300          ASL      R0
4080 017274 062700 002514   1$:      ADD      #ERRTB,R0 ;FORM ADDRESS
4081 017300 012037 017314   MOV      (R0)+,2$    ;GET ERROR MESSAGE (EM) POINTER
4082 017304 001404          BEQ      3$          ;BRANCH IF THERE ISN'T ONE
4083 017306 104401 001173   TYPE    ,SCRLF      ;"CARRIAGE RETURN - LINE FEED
4084 017312 104401
4085 017314 000000          2$:      .WORD    0      ;"EM" POINTER GOES HERE
4086 017316 012037 017332   3$:      MOV      (R0)+,4$    ;PICK UP DATA HEADER (DH) POINTER
4087 017322 001404          BEQ      5$          ;BRANCH IF NONE
4088 017324 104401 001173   TYPE    ,SCRLF      ;CARRIAGE RETURN-LINE FEED
4089 017330 104401
4090 017332 000000          4$:      .WORD    0      ;"DH" POINTER GOES HERE
4091 017334 012001 5$:      MOV      (R0)+,R1    ;PICKUP DATA TABLE (DT) POINTER
4092 017336 001447          BEQ      20$        ;BRANCH IF NONE
4093 017340 005005          CLR      R5          ;SET INDENT SWITCH
4094 017342 012000          MOV      (R0)+,R0    ;DATA FORMAT (DF) POINTER
4095 017344 012002          MOV      (R0)+,R2    ;NUMBER OF DH'S TO TYPE
4096 017346 001443          BEQ      20$        ;BRANCH IF DH NUMBER IS 0
4097 017350 005105          COM      R5          ;NO INDENT
4098 017352 112003 10$:     MOVB     (R0)+,R3    ;NUMBER OF DATA WORDS TO TYPE
4099 017354 001440          BEQ      20$        ;BR IF NO WORDS TO TYPE
4100 017356 112004          MOVB     (R0)+,R4    ;AND HOW TO TYPE THEM
4101 017360 104401 001173   TYPE    ,SCRLF      ;CR-LF
4102 017364 005705          TST      R5          ;INDENT ?
4103 017366 001002          BNE      11$        ;BR IF NOT
4104 017370 104401 027354   TYPE    ,BLNKS2     ;BLANKS
4105 017374 006004          11$:     ROR      R4          ;OCTAL OR DECIMAL?

```

```

4106 017376 103403          BCS      12$      ;DECIMAL--BRANCH
4107 017400 013146          MOV      2(R1)+,-(SP) ;SAVE 2(R1)+ FOR TYPEOUT
4108 017402 104402          TYPOC          ;GO TYPE--OCTAL ASCII(ALL DIGITS)
4109 017404 000402          BR       13$
4110 017406
4111 017406 013146          MOV      2(R1)+,-(SP) ;SAVE 2(R1)+ FOR TYPEOUT
4112 017410 104405          TYPOS          ;GO TYPE--DECIMAL ASCII WITH SIGN
4113 017412 005303          DEC      R3        ;MORE NUMBERS TO TYPE?
4114 017414 001403          BEQ      14$      ;NO--BRANCH
4115 017416 104401 027354          TYPE    ,BLNKS2    ;YES--TYPE SEPARATORS
4116 017422 000764          BR       11$      ;LOOP
4117 017424 005302          DEC      R2        ;MORE DH'S?
4118 017426 003413          BLE     20$      ;NO--BRANCH
4119 017430 104401 001173          TYPE    ,CRLF      ;YES--START A NEW LINE
4120 017434 005105          COM      R5        ;INDENT?
4121 017436 001002          BNE     15$      ;NO--BRANCH
4122 017440 104401 027354          TYPE    ,BLNKS2    ;YES--TYPE SPACES
4123 017444 012037 017452          MOV      (R0)+,16$ ;GET NEXT DH
4124 017450 104401          TYPE          ;NEXT TYPE IT
4125 017452 000000          .WORD   0          ;DH POINTER GOES HERE
4126 017454 000736          BR       10$      ;LOOP
4127 017456 104413          RLSREG          ;RESTORE R0-R5
4128 017460 000207          RTS      PC        ;RETURN

```

.SBTTL TYPE ROUTINE

```

4130
4131
4132
4133
4134
4135
4136
4137
4138
4139
4140
4141
4142
4143
4144
4145
4146
4147 017462 105737 001157          STYPE:  TSTB     $TPFLG ; IS THERE A TERMINAL?
4148 017466 100002          BPL     1$        ; BR IF YES
4149 017470 000000          HALT          ; HALT HERE IF NO TERMINAL
4150 017472 000407          BR       3$
4151 017474 010046          1$:      MOV      R0,-(SP) ; SAVE R0
4152 017476 017600 000002          MOV      22(SP),R0 ; GET ADDRESS OF ASCIZ STRING
4153 017502 112046          2$:      MOVVB   (R0)+,-(SP) ; PUSH CHARACTER TO BE TYPED ONTO STACK
4154 017504 001005          BNE     4$        ; BR IF IT ISN'T THE TERMINATOR
4155 017506 005726          TST     (SP)+     ; IF TERMINATOR POP IT OFF THE STACK
4156 017510 012600          60$:    MOV      (SP)+,R0 ; RESTORE R0
4157 017512 062716 000002          3$:      ADD      2,(SP)   ; ADJUST RETURN PC
4158 017516 000002          RTI          ; RETURN
4159 017520 122716 000011          4$:      CMPB    #HT,(SP) ; BRANCH IF <HT>
4160 017524 001430          BFG      8$
4161 017526 122716 000200          CMPB    #CRLF,(SP) ; BRANCH IF NOT <CRLF>

```

```

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

```

```

4162 017532 001006 BNE 5$
4163 017534 005726 TST (SP)+
4164 017536 104401 TYPE
4165 017540 001173 SCRLF
4166 017542 105037 017676 CLRB $CHARCNT
4167 017546 000755 BR 2$
4168 017550 004737 017632 5$: JSR PC,$TYPEC
4169 017554 123726 001155 6$: CMPB $FILLC,(SP)+
4170 017560 001350 BNE 2$
4171 017562 013746 001154 MOV $NULL,-(SP)
4172
4173 017566 105366 000001 7$: DECB 1(SP)
4174 017572 002770 BLT 6$
4175 017574 004737 017632 JSR PC,$TYPEC
4176 017600 105337 017676 DECB $CHARCNT
4177 017604 000770 BR 7$

```

```

::POP (CR)<LF> EQUIV
::TYPE A CR AND LF
::CLEAR CHARACTER COUNT
::GET NEXT CHARAC R
::GO TYPE THIS CHARACTER
::IS IT TIME FOR FILLER CHARS.?
::IF NO GO GET NEXT CHAR.
::GET # OF FILLER CHARS. NEEDED
AND THE NULL CHAR.
::DOES A NULL NEED TO BE TYPED?
BR IF NO--GO POP THE NULL OFF OF STACK
::GO TYPE A NULL
::DO NOT COUNT AS A COUNT
::LOOP

```

;HORIZONTAL TAB PROCESSOR

```

4181 017606 112716 000040 8$: MOVB 8'(SP)
4182 017612 004737 017632 9$: JSR PC,$TYPEC
4183 017616 132737 000007 017676 BITB 8', $CHARCNT
4184 017624 001372 BNE 9$
4185 017626 005726 TST (SP)+
4186 017630 000724 BR 2$
4187 017632 105777 161312 $TYPEC: TSTB 2$TPS
4188 017636 100375 BPL $TYPEC
4189 017640 116677 000002 161304 MOVB 2(SP), 2$TPB
4190 017646 122766 000015 000002 CMPB $CR, 2(SP)
4191 017654 001003 BNE 1$
4192 017656 105037 017676 CLRB $CHARCNT
4193 017662 000406 BR $TYPEX
4194 017664 122766 000012 000002 1$: CMPB $LF, 2(SP)
4195 017672 001402 BEQ $TYPEX
4196 017674 105227 INCB (PC)+
4197 017676 000000 $CHARCNT: .WORD 0
4198 017700 000207 $TYPEX: RTS PC

```

```

::REPLACE TAB WITH SPACE
::TYPE A SPACE
::BRANCH IF NOT AT
TAB STOP
::POP SPACE OFF STACK
::GET NEXT CHARACTER
::WAIT UNTIL PRINTER IS READY
::LOAD CHAR TO BE TYPED INTO DATA REG.
::IS CHARACTER A CARRIAGE RETURN?
BRANCH IF NO
YES--CLEAR CHARACTER COUNT
EXIT
::IS CHARACTER A LINE FEED?
BRANCH IF YES
COUNT THE CHARACTER
CHARACTER COUNT STORAGE

```

.SBTTL BINARY TO OCTAL (ASCII) AND TYPE

```

4201
4202
4203
4204
4205
4206
4207
4208
4209
4210
4211
4212
4213
4214
4215
4216
4217

```

```

*****
*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
*OCTAL (ASCII) NUMBER AND TYPE IT.
*STYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
*CALL:
*   MOV     NUM,-(SP)      ::NUMBER TO BE TYPED
*   TYPOS   ::CALL FOR TYPEOUT
*   .BYTE  N              ::N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
*   .BYTE  M              ::M=1 OR 0
*                               ::1=TYPE LEADING ZEROS
*                               ::0=SUPPRESS LEADING ZEROS
*STYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
*STYPOS OR $TYPOC
*CALL:

```

```

4218          *      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
4219          *      TYPON                      ;;CALL FOR TYPEOUT
4220          *
4221          *STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
4222          *CALL:
4223          *      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
4224          *      TYPYC                      ;;CALL FOR TYPEOUT
4225          *
4226 017702 017646 000000          STYPOS: MOV      2(SP),-(SP)      ;;PICKUP THE MODE
4227 017706 116637 000001 020125  MOVB     1(SP),SOFILL      ;;LOAD ZERO FILL SWITCH
4228 017714 112637 020127          MOVB     (SP)+,SOMODE+1    ;;NUMBER OF DIGITS TO TYPE
4229 017720 062716 000002          ADD      #2,(SP)        ;;ADJUST RETURN ADDRESS
4230 017724 000406          BR      STYPON
4231 017726 112737 000001 020125  STYPOC: MOVB     #1,SOFILL      ;;SET THE ZERO FILL SWITCH
4232 017734 112737 000006 020127  MOVB     #6,SOMODE+1    ;;SET FOR SIX(6) DIGITS
4233 017742 112737 000005 020124  STYPON: MOVB     #5,SOCNT      ;;SET THE ITERATION COUNT
4234 017750 010346          MOV      R3,-(SP)      ;;SAVE R3
4235 017752 010446          MOV      R4,-(SP)      ;;SAVE R4
4236 017754 010546          MOV      R5,-(SP)      ;;SAVE R5
4237 017756 113704 020127          MOVB     SOMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
4238 017762 005404          NEG      R4
4239 017764 062704 000006          ADD      #6,R4          ;;SUBTRACT IT FOR MAX. ALLOWED
4240 017770 110437 020126          MOVB     R4,SOMODE      ;;SAVE IT FOR USE
4241 017774 113704 020125          MOVB     SOFILL,R4      ;;GET THE ZERO FILL SWITCH
4242 020000 016605 000012          MOV      12(SP),R5     ;;PICKUP THE INPUT NUMBER
4243 020004 005003          CLR      R3            ;;CLEAR THE OUTPUT WORD
4244 020006 006105          1$: ROL     R5          ;;ROTATE MSB INTO "C"
4245 020010 000404          BR      3$            ;;GO DO MSB
4246 020012 006105          2$: ROL     R5          ;;FORM THIS DIGIT
4247 020014 006105          ROL     R5
4248 020016 006105          ROL     R5
4249 020020 010503          MOV      R5,R3
4250 020022 006103          3$: ROL     R3          ;;GET LSB OF THIS DIGIT
4251 020024 105337 020126          DECB     SOMODE         ;;TYPE THIS DIGIT?
4252 020030 100016          CPL      7$            ;;BR IF NO
4253 020032 042703 177770          BIC      #177770,R3    ;;GET RID OF JUNK
4254 020036 001002          BNE     4$            ;;TEST FOR 0
4255 020040 005704          TST     R4            ;;SUPPRESS THIS 0?
4256 020042 001403          BEQ     5$            ;;BR IF YES
4257 020044 005204          4$: INC     R4          ;;DON'T SUPPRESS ANYMORE 0'S
4258 020046 052703 000060          BIS     #'0,R3        ;;MAKE THIS DIGIT ASCII
4259 020052 052703 000040          5$: BIS     #' ,R3     ;;MAKE ASCII IF NOT ALREADY
4260 020056 110337 020122          MOVB     R3,R5        ;;SAVE FOR TYPING
4261 020062 104401 020122          TYPE    2$            ;;GO TYPE THIS DIGIT
4262 020066 105337 020124          7$: DECB     SOCNT      ;;COUNT BY 1
4263 020072 003347          BGT     2$            ;;BR IF MORE TO DO
4264 020074 012402          BLT     6$            ;;BR IF DONE
4265 020076 007204          INC     R4            ;;INSURE LAST DIGIT ISN'T A BLANK
4266 020100 000744          BR      2$            ;;GO DO THE LAST DIGIT
4267 020102 012605          6$: MOV     (SP)+,R5     ;;RESTORE R5
4268 020104 012604          MOV     (SP)+,R4      ;;RESTORE R4
4269 020106 012603          MOV     (SP)+,R3      ;;RESTORE R3
4270 020110 016666 000002 000004  MOV     2(SP),4(SP)    ;;SET THE STACK FOR RETURNING
4271 020116 012616          MOV     (SP)+,(SP)
4272 020120 000002          RTI
4273 020122 000          8$: .BYTE 0          ;;RETURN
;;STORAGE FOR ASCII DIGIT

```

E07

MD-11-DZRPY-B, R011-E FUNCTIONAL LOGIC & R/W TEST
 DZRPYB.CMB BINARY TO OCTAL (ASCII) AND TYPE

MACY11 27(732) 02-NOV-76 14:50 PAGE 83

4274 020123 000
 4275 020124 000
 4276 020125 000
 4277 020126 000000
 4278
 4279
 4280
 4281
 4282
 4283
 4284
 4285
 4286
 4287
 4288
 4289
 4290
 4291 020130
 4292 020130 010046
 4293 020132 010146
 4294 020134 010246
 4295 020136 010346
 4296 020140 010546
 4297 020142 012746 020200
 4298 020146 016605 000020
 4299 020152 100004
 4300 020154 005405
 4301 020156 112766 000055 000001
 4302 020164 005000
 4303 020166 012703 020344
 4304 020172 112723 000040
 4305 020176 005002
 4306 020200 016001 020334
 4307 020204 160105
 4308 020206 002402
 4309 020210 005202
 4310 020212 000774
 4311 020214 060105
 4312 020216 005702
 4313 020220 001002
 4314 020222 105716
 4315 020224 100407
 4316 020226 106316
 4317 020230 103003
 4318 020232 116663 000001 177777
 4319 020240 052702 000060
 4320 020244 052702 000040
 4321 020250 110223
 4322 020252 005720
 4323 020254 020027 000010
 4324 020260 002746
 4325 020262 003002
 4326 020264 010502
 4327 020266 000764
 4328 020270 105726
 4329 020272 100003

```

.SOCNT: .BYTE 0          ;; TERMINATOR FOR TYPE ROUTINE
.SOFILL: .BYTE 0         ;; OCTAL DIGIT COUNTER
.SOMODE: .WORD 0         ;; ZERO FILL SWITCH
                          ;; NUMBER OF DIGITS TO TYPE
  
```

.SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

```

*****
*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
*REPLACED WITH SPACES.
*CALL:
*   MOV     NUM,-(SP)      ;; PUT THE BINARY NUMBER ON THE STACK
*   TYPDS          ;; GO TO THE ROUTINE
  
```

```

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

```

4330 020274 116663 177777 177777:   MOVB   -1(SP),-2(R3)   ;; YES--SET THE SIGN FOR TYPING
4331 020302 105013   9S:   CLRB   (R3)         ;; SET THE TERMINATOR
4332 020304 012605   MOV    (SP)+,R5       ;; POP STACK INTO R5
4333 020306 012603   MOV    (SP)+,R3       ;; POP STACK INTO R3
4334 020310 012602   MOV    (SP)+,R2       ;; POP STACK INTO R2
4335 020312 012601   MOV    (SP)+,R1       ;; POP STACK INTO R1
4336 020314 012600   MOV    (SP)+,R0       ;; POP STACK INTO R0
4337 020316 104401 020344   TYPE   $DBLK         ;; NOW TYPE THE NUMBER
4338 020322 016666 000002 000004   MOV    2(SP),4(SP)   ;; ADJUST THE STACK
4339 020330 012616   MOV    (SP)+,(SP)
4340 020332 000002   RTI
4341 020334 023420   SDBL: 10000.         ;; RETURN TO USER
4342 020336 001750   1000.
4343 020340 000144   100.
4344 020342 000012   10.
4345 020344 000004   SDBLK: .BLKW 4
4346
4347   .SBTTL  TTY INPUT ROUTINE
4348
4349   ;*****
4350   .ENABL  LSB
4351 020354 000000   $TKCNT: .WORD 0      ;; NUMBER OF ITEMS IN QUEUE
4352 020356 000000   $TKQIN: .WORD 0     ;; INPUT POINTER
4353 020360 000000   $TKQOUT: .WORD 0    ;; OUTPUT POINTER
4354 020362 000012   $TKQSRV: .BLKB 10.  ;; TTY KEYBOARD QUEUE
4355   $TKQEND=.
4356
4357   ;*TK INITIALIZE ROUTINE
4358   ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
4359   ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
4360
4361   ;*CALL:
4362   ;*   JSR   PC,$TKINT
4363   ;*   RETURN
4364
4365 020374 005037 020354   $TKINT: CLR    $TKCNT   ;; CLEAR COUNT OF ITEMS IN QUEUE
4366 020400 012737 020362 020356   MOV    $TKQSRV,$TKQIN ;; MOVE THE STARTING ADDRESS OF THE
4367 020406 013737 020356 020360   MOV    $TKQIN,$TKQOUT ;; QUEUE INTO THE INPUT & OUTPUT POINTERS.
4368 020414 012737 020444 000060   MOV    $TKSRV,$TKVEC  ;; INITIALIZE THE KEYBOARD VECTOR
4369 020422 012737 000200 000062   MOV    #200,$TKVEC+2  ;; "BR" LEVEL 4
4370 020430 005777 160512 000062   TST   $TKB           ;; CLEAR DONE FLAG
4371 020434 012777 000100 160502   MOV    #100,$TKS     ;; ENABLE TTY KEYBOARD INTERRUPT
4372 020442 000207   RTS    PC            ;; RETURN TO CALLER
4373
4374   ;*TK SERVICE ROUTINE
4375   ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
4376   ;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
4377   ;*IT IN THE QUEUE.
4378   ;*IF THE CHARACTER IS A "CONTROL-C" (1C) $TKINT IS CALLED AND
4379   ;*UPON RETURN EXIT IS MADE TO THE "CONTROL-C" RESTART ADDRESS (START2)
4380
4381 020444 117746 160476   $TKSRV: MOVB   $TKB,-(SP) ;; PICKUP THE CHARACTER
4382 020450 042716 177600   BIC   #1C177,(SP)    ;; STRIP THE JUNK
4383 020454 021627 000003   CMP   (SP),#3        ;; IS IT A CONTROL C?
4384 020460 001007   BNE   IS             ;; BRANCH IF NO
4385 020462 104401 021562   TYPE  ,SCNTLC        ;; TYPE A CONTROL-C (1C)

```

```

4386 020466 004737 020374 JSR PC,STKINT      ;: INIT THE KEYBOARD
4387 020472 005726 TST (SP)+        ;: CLEAN UP STACK
4388 020474 000137 003676 JMP START2       ;: CONTROL C RESTART
4389 020500 021627 000007 15: CMP (SP),#7     ;: IS IT A CONTROL G?
4390 020504 001004 BNE 25          ;: BRANCH IF NO
4391 020506 022737 000176 001140 CMP #SWREG,SWR  ;: IS SOFT-SWR SELECTED?
4392 020514 001500 BEQ 65          ;: GO TO SWR CHANGE
4393
4394 020516
4395 020516 022737 000012 020354 25: CMP #10.,STKCNT ;: IS THE QUEUE FULL?
4396 020524 001004 BNE 35          ;: BRANCH IF NO
4397 020526 104401 001166 TYPE $BELL      ;: RING THE TTY BELL
4398 020532 005726 TST (SP)+        ;: CLEAN CHARACTER OFF OF STACK
4399 020534 000451 BR 55           ;: EXIT
4400 020536 021627 000023 35: CMP (SP),#23    ;: IS IT A CONTROL-S?
4401 020542 001021 BNE 32$        ;: BRANCH IF NO
4402 020544 005077 160374 CLR @STKS      ;: DISABLE TTY KEYBOARD INTERRUPTS
4403 020550 005726 TST (SP)+        ;: CLEAN CHAR OFF STACK
4404 020552 105777 160366 31$: TSTB @STKS    ;: WAIT FOR A CHAR
4405 020556 100375 BPL 31$        ;: LOOP UNTIL ITS THERE
4406 020560 117746 160362 MOVB @STKB, -(SP) ;: GET THE CHARACTER
4407 020564 042716 177600 BIC #177, (SP)  ;: MAKE IT 7-BIT ASCII
4408 020570 022627 000021 CMP (SP)+, #21  ;: IS IT A CONTROL-Q?
4409 020574 001366 BNE 31$        ;: BRANCH IF NO
4410 020576 012777 000100 160340 MOV #100, @STKS ;: REENABLE TTY KEYBOARD INTERRUPTS
4411 020604 000002 RTI                ;: RETURN
4412 020606 005237 020354 32$: INC STKCNT      ;: COUNT THIS CHARACTER
4413 020612 021627 000140 CMP (SP), #140  ;: IS IT UPPER CASE?
4414 020616 002405 BLT 45          ;: BRANCH IF YES
4415 020620 021627 000175 CMP (SP), #175  ;: IS IT A SPECIAL CHAR?
4416 020624 003002 BGT 45          ;: BRANCH IF YES
4417 020626 042716 000040 BIC #40, (SP)  ;: MAKE IT UPPER CASE
4418 020632 112677 177520 MOVB (SP)+, @STKQIN ;: AND PUT IT IN QUEUE
4419 020636 005237 020356 INC STKQIN      ;: UPDATE THE POINTER
4420 020642 023727 020356 020374 CMP STKQIN, #STKQEND ;: GO OFF THE END?
4421 020650 001003 BNE 55          ;: BRANCH IF NO
4422 020652 012737 020362 020356 MOV #STKQSR, STKQIN ;: RESET THE POINTER
4423 020660 000002 55: RTI                ;: RETURN
4424
4425
4426
4427
4428
4429
4430 020662 022737 000176 001140 $CKSWR: CMP #SWREG,SWR ;: IS THE SOFT-SWR SELECTED
4431 020670 001124 BNE 15$        ;: EXIT IF NOT
4432 020672 105777 160246 TSTB @STKS    ;: IS A CHAR WAITING?
4433 020676 100121 BPL 15$        ;: IF NOT, EXIT
4434 020700 117746 160242 MOVB @STKB, -(SP) ;: YES
4435 020704 042716 177600 BIC #177, (SP)  ;: MAKE IT 7-BIT ASCII
4436 020710 021627 000007 CMP (SP), #7   ;: IS IT A CONTROL-G?
4437 020714 001300 BNE 25          ;: IF NOT, PUT IT IN THE TTY QUEUE
4438
4439
4440
4441

```

```

;: *****
;: #SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
;: #ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
;: #SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
;: #CALL WHEN OPERATING IN TTY INTERRUPT MODE.

```

```

;: *****
;: #CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE

```



```

4442 ;#ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
4443 ;#CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
4444 020716 123727 001134 000001 65:  CMB  $AUT08,01  :: ARE WE RUNNING IN AUTO-MODE?
4445 020724 001674  BEQ  25  :: BRANCH IF YES
4446 020726 005726  TST  (SP)+  :: CLEAR CONTROL-G OFF STACK
4447 020730 004737 020374  JSR  PC,$TKINT  :: FLUSH THE TTY INPUT QUEUE
4448 020734 005077 160204  CLR  $TKS  :: DISABLE TTY KEYBOARD INTERRUPTS
4449 020740 112737 000001 001135  MOVB  #1,$INTAG  :: SET INTERRUPT MODE INDICATOR
4450
4451 020746 104401 021574  SGTSWR: TYPE  ,SCNTLG  :: ECHO THE CONTROL-G (1G)
4452 020752 104401 021601  TYPE  ,SMSWR  :: TYPE CURRENT CONTENTS
4453 020756 013746 000176  MOV  SWREG,-(SP)  :: SAVE SWREG FOR TYPEOUT
4454 020762 104402  TYPEOC  :: GO TYPE--OCTAL ASCII(ALL DIGITS)
4455 020764 104401 021612  TYPE  ,SMNEW  :: PROMPT FOR NEW SWR
4456 020770 005046 195:  CLR  -(SP)  :: CLEAR COUNTER
4457 020772 005046  CLR  -(SP)  :: THE NEW SWR
4458 020774 105777 160144  75:  TSTB  $TKS  :: CHAR THERE?
4459 021000 100375  BPL  75  :: IF NOT TRY AGAIN
4460
4461 021002 117746 160140  MOVB  $TKB,-(SP)  :: PICK UP CHAR
4462 021006 042716 177600  BIC  #1C177,(SP)  :: MAKE IT 7-BIT ASCII
4463
4464 021012 021627 000003  CMP  (SP),#3  :: IS IT A CONTROL-C?
4465 021016 001015  BNE  95  :: BRANCH IF NOT
4466 021020 104401 021562  TYPE  ,SCNTLC  :: YES, ECHO CONTROL-C (1C)
4467 021024 062706 000006  ADD  #6,SP  :: CLEAN UP STACK
4468 021030 123727 001135 000001  CMPB  $INTAG,#1  :: REENABLE TTY KEYBOARD INTERRUPTS?
4469 021036 001003  BNE  85  :: BRANCH IF NO
4470 021040 012777 000100 160076  MOV  #100,$TKS  :: ALLOW TTY KEYBOARD INTERRUPTS
4471 021046 000137 003676  85:  JMP  START2  :: CONTROL-C RESTART
4472
4473
4474 021052 021627 000025  95:  CMP  (SP),#25  :: IS IT A CONTROL-U?
4475 021056 001005  BNE  105  :: BRANCH IF NOT
4476 021060 104401 021567  TYPE  ,SCNTLU  :: YES, ECHO CONTROL-U (1U)
4477 021064 062706 000006  205:  ADD  #6,SP  :: IGNORE PREVIOUS INPUT
4478 021070 000737  BR  195  :: LET'S TRY IT AGAIN
4479
4480
4481 021072 021627 000015  105:  CMP  (SP),#15  :: IS IT A (CR)?
4482 021076 001022  BNE  165  :: BRANCH IF NO
4483 021100 005766 000004  TST  4(SP)  :: YES, IS IT THE FIRST CHAR?
4484 021104 001403  BEQ  115  :: BRANCH IF YES
4485 021106 016677 000002 160024  MOV  2(SP),$SWR  :: SAVE NEW SWR
4486 021114 062706 000006  115:  ADD  #6,SP  :: CLEAN UP STACK
4487 021120 104401 001173  145:  TYPE  ,SCRLF  :: ECHO (CR) AND (LF)
4488 021124 123727 001135 000001  CMPB  $INTAG,#1  :: RE-ENABLE TTY KBD INTERRUPTS?
4489 021128 001003  BNE  155  :: BRANCH IF NOT
4490 021134 012777 000100 160002  MOV  #100,$TKS  :: RE-ENABLE TTY KBD INTERRUPTS
4491 021142 000002  155:  RTI  :: RETURN
4492 021144 004737 017632  165:  JSR  PC,$TYPEC  :: ECHO CHAR
4493 021150 021627 000060  CMP  (SP),#60  :: CHAR < 0?
4494 021154 002420  BLT  185  :: BRANCH IF YES
4495 021156 021627 000067  CMP  (SP),#67  :: CHAR > 7?
4496 021162 003015  BGT  185  :: BRANCH IF YES
4497 021164 042726 000060  BIC  #60,(SP)+  :: STRIP-OFF ASCII

```



```

4498 021170 005766 000002          TST      2(SP)          ;; IS THIS THE FIRST CHAR
4499 021174 001403                   BEQ      17$          ;; BRANCH IF YES
4500 021176 006316                   RSL     (SP)          ;; NO, SHIFT PRESENT
4501 021200 006316                   RSL     (SP)          ;; CHAR OVER TO MAKE
4502 021202 006316                   RSL     (SP)          ;; ROOM FOR NEW ONE.
4503 021204 005266 000002          17$: INC      2(SP)          ;; KEEP COUNT OF CHAR
4504 021210 056616 177776          BT      -2(SP), (SP)  ;; SET IN NEW CHAR
4505 021214 000667 177776          BR      7$           ;; GET THE NEXT ONE
4506 021216 104401 001172          18$: TYPE   $QUES     ;; TYPE '(CR)<LF>
4507 021222 000720                   BR      20$          ;; SIMULATE CONTROL-U
4508                                     .DSABL  LSB
4509
4510
4511                                     ;; *****
4512                                     ;; THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
4513                                     ;; CALL:
4514                                     ;;      RDCHR          ;; GET A CHARACTER FROM THE QUEUE
4515                                     ;;      RETURN HERE   ;; CHARACTER IS ON THE STACK
4516                                     ;; *****
4517                                     ;; WITH PARITY BIT STRIPPED OFF
4518
4519 021224 011646                   SRDCHR: MOV     (SP), -(SP)  ;; PUSH DOWN THE PC AND
4520 021226 016666 000004 000002    MOV     4(SP), 2(SP)  ;; THE PS
4521 021234 005066 000004          CLR     4(SP)          ;; GET READY FOR A CHARACTER
4522 021240 005046                   CLR     -(SP)          ;; PUT NEW PS ON STACK
4523 021242 012746 021250          MOV     #64$, -(SP)   ;; PUT NEW PC ON STACK
4524 021246 000002                   RTI                    ;; POP NEW PC AND PS
4525 021250
4526 021250 005737 020354          64$:  TST     $TKCNT     ;; WAIT ON A CHARACTER
4527 021254 001775                   1$:   BEQ      1$           ;;
4528 021256 005337 020354          DEC     $TKCNT        ;; DECREMENT THE COUNTER
4529 021262 117766 177072 000004    MOV     2$TKQOUT, 4(SP) ;; GET ONE CHARACTER
4530 021270 005237 020360          INC     $TKQOUT       ;; UPDATE THE POINTER
4531 021274 023727 020360 020374    CMP     $TKQOUT, #2$TKQEND ;; DID IT GO OFF OF THE END?
4532 021302 001003                   BNE     2$           ;; BRANCH IF NO
4533 021304 012737 020362 020360    MOV     #2$TKQSRST, $TKQOUT ;; RESET THE POINTER
4534 021312 000002                   2$:   RTI                    ;; RETURN
4535                                     ;; *****
4536                                     ;; THIS ROUTINE WILL INPUT A STRING FROM THE TTY
4537                                     ;; CALL:
4538                                     ;;      ROLIN          ;; INPUT A STRING FROM THE TTY
4539                                     ;;      RETURN HERE   ;; ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
4540                                     ;; *****
4541                                     ;; TERMINATOR WILL BE A BYTE OF ALL 0'S
4542 021314 010346                   SRDOLIN: MOV    R3, -(SP)  ;; SAVE R3
4543 021316 005046                   CLR     -(SP)          ;; CLEAR THE RUBOUT KEY
4544 021320 012703 021550          1$:   MOV     #2$TTYIN, R3  ;; GET ADDRESS
4545 021324 022703 021562          2$:   CMP     #2$TTYIN+10., R3 ;; BUFFER FULL?
4546 021330 101456                   BLOS   4$           ;; BR IF YES
4547 021332 104410                   RDCHR   4$           ;; GO READ ONE CHARACTER FROM THE TTY
4548 021334 112613                   MOV     (SP)+, (R3)    ;; GET CHARACTER
4549 021336 122713 000177          10$:  CMP     #177, (R3)   ;; IS IT A RUBOUT
4550 021342 001022                   BNE     5$           ;; BR IF NO
4551 021344 005716                   TST     (SP)          ;; IS THIS THE FIRST RUBOUT?
4552 021346 001007                   BNE     6$           ;; BR IF NO
4553 021350 112737 000134 021546    MOV     #' \, 9$      ;; TYPE A BACK SLASH

```

```

4554 021356 104401 021546 TYPE 95
4555 021362 012716 177777 MOV 1, (SP)
4556 021366 005303 6S: DEC R3
4557 021370 020327 021550 CMP R3, #STTYIN
4558 021374 103434 BLO 45
4559 021376 111337 021546 MOVB (R3), 95
4560 021402 104401 021546 TYPE 95
4561 021406 000746 BR 25
4562 021410 005716 5S: TST (SP)
4563 021412 001406 BEQ 75
4564 021414 112737 000134 021546 MOVB #' \, 95
4565 021422 104401 021546 TYPE 95
4566 021426 005016 CLR (SP)
4567 021430 122713 000025 7S: CMPB #25, (R3)
4568 021434 001003 BNE 85
4569 021436 104401 021567 TYPE , SCNTLU
4570 021442 000726 BR 15
4571 021444 122713 000022 8S: CMPB #22, (R3)
4572 021450 001011 BNE 35
4573 021452 105013 CLR (R3)
4574 021454 104401 001173 TYPE , SCRLF
4575 021460 104401 021550 TYPE , STTYIN
4576 021464 000717 BR 25
4577 021466 104401 001172 4S: TYPE , SQUES
4578 021472 000712 BR 15
4579 021474 111337 021546 3S: MOVB (R3), 95
4580 021500 104401 021546 TYPE 95
4581 021504 122723 000015 CMPB #15, (R3)+
4582 021510 001305 BNE 25
4583 021512 105063 177777 CLR (R3)
4584 021516 104401 001174 TYPE , SLF
4585 021522 005726 TST (SP)+
4586 021524 012603 MOV (SP)+, R3
4587 021526 011646 MOV (SP), -(SP)
4588 021530 016666 000004 000002 MOV 4(SP), 2(SP)
4589 021536 012766 021550 000004 MOV #STTYIN, 4(SP)
4590 021544 000002 RTI
4591 021546 000 9S: .BYTE 0
4592 021547 000 .BYTE 0
4593 021550 000012 $TTYIN: .BLKB 10.
4594 021562 041536 005015 000 SCNTLC: .ASCIZ /C/<15><12>
4595 021567 136 006525 000012 SCNTLU: .ASCIZ /U/<15><12>
4596 021574 043536 005015 000 SCNTLG: .ASCIZ /G/<15><12>
4597 021601 015 051412 051127 SMSMR: .ASCIZ <15><12>/SMR = /
4598 021606 036440 000040 SHNEW: .ASCIZ / NEW = /
4599 021612 020040 042516 020127
4600 021620 020075 000
4601 021624 .EVEN
4602
4603 .SBTTL SCOPE HANDLER ROUTINE
4604
4605 *****
4606 ; THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
4607 ; AND LOAD THE TEST NUMBER($STNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
4608 ; AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
4609 ; THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:

```

K07

```

4610      ;*SW14=1      LOOP ON TEST
4611      ;*SW11=1      INHIBIT ITERATIONS
4612      ;*SW09=1      LOOP ON ERROR
4613      ;*CALL
4614      ;*          SCOPE          ;;SCOPE=IOT
4615
4616      $SCOPE:
4617      021624      104407      CKSWR          ;;TEST FOR CHANGE IN SOFT-SWR
4618      021626      032777      040000      157304      1$: BIT      #BIT14,2SWR      ;;LOOP ON PRESENT TEST?
4619      021634      001076      BNE      $OVER      ;;YES IF SW14=1
4620      ;*****START OF CODE FOR THE XOR TESTER*****
4621      021636      000416      $XTSTR: BR      6$      ;;IF RUNNING ON THE "XOR" TESTER CHANGE
4622      ;THIS INSTRUCTION TO A "NOP" (NOP=240)
4623      021640      013746      000004      MOV      2#ERRVEC,-(SP)      ;;SAVE THE CONTENTS OF THE ERROR VECTOR
4624      021644      012737      021664      000004      MOV      #5$,$#ERRVEC      ;;SET FOR TIMEOUT
4625      021652      005737      177060      TST      2#177060      ;;TIME OUT ON XOR?
4626      021656      012637      000004      MOV      (SP)+,2#ERRVEC      ;;RESTORE THE ERROR VECTOR
4627      021662      000450      BR      $SVLAD      ;;GO TO THE NEXT TEST
4628      021664      022626      5$: CMP      (SP)+,(SP)+      ;;CLEAR THE STACK AFTER A TIME OUT
4629      021666      012637      000004      MOV      (SP)+,2#ERRVEC      ;;RESTORE THE ERROR VECTOR
4630      021672      000413      BR      7$      ;;LOOP ON THE PRESENT TEST
4631      021674
4632      021674      105737      001103      6$:;*****END OF CODE FOR THE XOR TESTER*****
4633      021700      001421      2$: TSTB     $ERFLG      ;;HAS AN ERROR OCCURRED?
4634      021702      123737      001115      001103      BEQ      3$      ;;BR IF NO
4635      021710      101015      CMPB     $ERMAX,$ERFLG      ;;MAX. ERRORS FOR THIS TEST OCCURRED?
4636      021712      032777      001000      157220      BHI      3$      ;;BR IF NO
4637      021720      001404      BIT      #BIT09,2SWR      ;;LOOP ON ERROR?
4638      021722      013737      001110      001106      7$: BEQ      4$      ;;BR IF NO
4639      021730      000440      MOV      $LPERR,$LPADR      ;;SET LOOP ADDRESS TO LAST SCOPE
4640      021732      105037      001103      4$: BR      $OVER
4641      021736      005037      001162      CLRB     $ERFLG      ;;ZERO THE ERROR FLAG
4642      021742      000412      CLR      $TIMES      ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
4643      021744      032777      004000      157166      3$: BR      1$      ;;ESCAPE TO THE NEXT TEST
4644      021752      001006      BIT      #BIT11,2SWR      ;;INHIBIT ITERATIONS?
4645      021754      005237      001104      BNE      1$      ;;BR IF YES
4646      021760      023737      001162      001104      INC      $ICNT      ;;INCREMENT ITERATION COUNT
4647      021766      002021      CMP      $TIMES,$ICNT      ;;CHECK THE NUMBER OF ITERATIONS MADE
4648      021770      012737      000001      001104      BGE      $OVER      ;;BR IF MORE ITERATION REQUIRED
4649      021776      013737      022046      001162      1$: MOV      #1,$ICNT      ;;REINITIALIZE THE ITERATION COUNTER
4650      022004      105237      001102      $SVLAD: MOV      $MXCNT,$TIMES      ;;SET NUMBER OF ITERATIONS TO DO
4651      022010      011637      001106      INCB     $STNM      ;;COUNT TEST NUMBERS
4652      022014      011637      001110      MOV      (SP),$LPADR      ;;SAVE SCOPE LOOP ADDRESS
4653      022020      005037      001164      MOV      (SP),$LPERR      ;;SAVE ERROR LOOP ADDRESS
4654      022024      112737      000001      001115      CLP      $ESCAPE      ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
4655      022032      013777      001102      157102      $OVER: MOV      #1,$ERMAX      ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
4656      022040      013716      001106      MOV      $STNM,$DISPLAY      ;;DISPLAY TEST NUMBER
4657      022044      000002      RTI      ;;FUDGE RETURN ADDRESS
4658      022046      000001      $MXCNT: 1      ;;FIXES PS
4659
4660      .SBTTL SAVE AND RESTORE RD-R5 ROUTINES
4661
4662      ;*****
4663      ;*SAVE RD-R5
4664      ;*CALL:
4665      ;* SAVREG

```

4666
4667
4668
4669
4670
4671
4672
4673
4674
4675
4676
4677 022050
4678 022050 010046
4679 022052 010146
4680 022054 010246
4681 022056 010346
4682 022060 010446
4683 022062 010546
4684 022064 016646 000022
4685 022070 016646 000022
4686 022074 016646 000022
4687 022100 016646 000022
4688 022104 000002
4689
4690
4691
4692
4693 022106
4694 022106 012666 000022
4695 022112 012666 000022
4696 022116 012666 000022
4697 022122 012666 000022
4698 022126 012605
4699 022130 012604
4700 022132 012603
4701 022134 012602
4702 022136 012601
4703 022140 012600
4704 022142 000002
4705
4706
4707
4708
4709
4710
4711
4712
4713
4714
4715
4716
4717
4718
4719
4720
4721

:UPON RETURN FROM \$SAVREG THE STACK WILL LOOK LIKE:

```

*
*TOP---(+16)
* +2---(+18)
* +4---R5
* +6---R4
* +8---R3
*+10---R2
*+12---R1
*+14---R0

```

\$SAVREG:

```

MOV R0, -(SP)      ;; PUSH R0 ON STACK
MOV R1, -(SP)      ;; PUSH R1 ON STACK
MOV R2, -(SP)      ;; PUSH R2 ON STACK
MOV R3, -(SP)      ;; PUSH R3 ON STACK
MOV R4, -(SP)      ;; PUSH R4 ON STACK
MOV R5, -(SP)      ;; PUSH R5 ON STACK
MOV 22(SP), -(SP)  ;; SAVE PS OF MAIN FLOW
MOV 22(SP), -(SP)  ;; SAVE PC OF MAIN FLOW
MOV 22(SP), -(SP)  ;; SAVE PS OF CALL
MOV 22(SP), -(SP)  ;; SAVE PC OF CALL
RTI

```

:RESTORE RO-R5

:CALL:

\$RESREG: RESREG

```

MOV (SP)+, 22(SP)  ;; RESTORE PC OF CALL
MOV (SP)+, 22(SP)  ;; RESTORE PS OF CALL
MOV (SP)+, 22(SP)  ;; RESTORE PC OF MAIN FLOW
MOV (SP)+, 22(SP)  ;; RESTORE PS OF MAIN FLOW
MOV (SP)+, R5      ;; POP STACK INTO R5
MOV (SP)+, R4      ;; POP STACK INTO R4
MOV (SP)+, R3      ;; POP STACK INTO R3
MOV (SP)+, R2      ;; POP STACK INTO R2
MOV (SP)+, R1      ;; POP STACK INTO R1
MOV (SP)+, R0      ;; POP STACK INTO R0
RTI

```

.SBTTL ROUTINE TO SIZE MEMORY

:CALL:

```

JSR PC, $SIZE
RETURN
*SLSTAD WILL CONTAIN:
* WITH KT11 OPTION -- LAST VIRTUAL ADDRESS OF THE LAST BANK
* WITHOUT KT11 OPTION -- LAST ABSOLUTE ADDRESS OF AVAILABLE MEMORY
*SLSTBK WILL CONTAIN THE LAST BANK AS A SAF
*SKT11 IS THE MEMORY MANAGEMENT KEY
*BIT07 = 0 DON'T USE MEMORY MANAGEMENT
* MUST BE SETUP BEFORE THE CALL
*BIT15 = 0 DON'T HAVE MEMORY MANAGEMENT OPTION
* DETERMINED BY ROUTINE

```

M07

4722	022144	010046				\$SIZE: MOV	R0,-(SP)	::	SAVE R0 ON THE STACK
4723	022146	010146				MOV	R1,-(SP)	::	SAVE R1 ON THE STACK
4724	022150	010246				MOV	R2,-(SP)	::	SAVE R2 ON THE STACK
4725	022152	010346				MOV	R3,-(SP)	::	SAVE R3 ON THE STACK
4726	022154	013746	000004			MOV	@ERRVEC,-(SP)	::	SAVE PRESENT ERROR VECTOR PS & PC
4727	022160	013746	000006			MOV	@ERRVEC+2,-(SP)	::	
4728	022164	010600				MOV	SP,R0	::	SAVE THE STACK POINTER
4729						;;SET THE ERRVEC PS TO THE PRESENT PS			
4730	022166	104400				TRAP		::	PUSH OLD PSH AND PC ON STACK
4731	022170	012637	000006			MOV	(SP)+,@ERRVEC+2	::	SAVE THE PSH IN @ERRVEC+2
4732	022174	012701	003776			MOV	#3776,R1	::	SETUP ADDRESS
4733	022200	105727				TSTB	(PC)+	::	USE MEMORY MANAGEMENT?
4734	022202	000200				\$KT11: .WORD			200
4735	022204	100062				BPL	\$CORE	::	BR IF NO
4736	022206	012737	022344	000004		MOV	#\$KTNEX,@ERRVEC	::	SET FOR TIMEOUT
4737	022214	005737	177572			TST	@SR0	::	KT11 ARE YOU THERE?
4738	022220	052737	100000	022202		BIS	#10000,\$KT11	::	YES--SET KT11 KEY
4739	022226	005046				CLR	-(SP)	::	INITIALIZE FOR "PAR" LOADING
4740	022230	012702	172340			MOV	#KIPAR0,R2	::	ADDRESS OF FIRST "PAR"
4741	022234	012703	000010			MOV	#108,R3	::	LOAD EIGHT "PAR.'S" AND EIGHT "PDR.'S"
4742	022240	012762	077406	177740	1\$:	MOV	#77406,-40(R2)	::	PDR = 4K UP, READ/WRITE
4743	022246	011622				MOV	(SP),(R2)+	::	LOAD "PAR"
4744	022250	062716	000200			ADD	#200,(SP)	::	UPDATE FOR NEXT "PAR"
4745	022254	077307				SOB	R3,1\$::	LOOP UNTIL ALL EIGHT ARE LOADED
4746	022256	012742	177600			MOV	#177600,-(R2)	::	SETUP KIPAR7 FOR I/O
4747	022262	005042				CLR	-(R2)	::	SETUP KIPAR6 FOR TESTING
4748	022264	012737	022302	000004		MOV	#2\$,@ERRVEC	::	CATCH TIMEOUT IF NO SR3
4749	022272	012737	000020	172516		MOV	#20,@SR3	::	ENABLE 22 BIT MODE
4750	022300	000401				BR	3\$::	THIS PDP-11 HAS A SR3 REGISTER
4751	022302	022626			2\$:	CMP	(SP)+,(SP)+	::	CLEAN OFF THE STACK--NO SR3
4752	022304	005237	177572		3\$:	INC	@SR0	::	TURN ON MEMORY MANAGEMENT
4753	022310	012737	022334	000004		MOV	#\$KTOUT,@ERRVEC	::	SET FOR TIME OUT
4754	022316	005737	143776		4\$:	TST	@143776	::	TRAP ON NON-EX-MEM
4755	022322	062712	000040			ADD	#40,(R2)	::	MAKE A 1K STEP
4756	022326	023712	172356			CMP	@KIPAR7,(R2)	::	LAST ONE?
4757	022332	101371				BHI	4\$::	NO--TRY IT
4758	022334	011202				MOV	(R2),R2	::	GET LAST BANK+1
4759	022336	005037	177572			CLR	@SR0	::	TURN OFF MEMORY MANAGEMENT
4760	022342	000421				BR	\$SIZEX	::	
4761	022344	042737	100000	022202	\$KTNEX:	BIC	#10000,\$KT11	::	KT11 NON-EXISTENT
4762	022352	012737	022402	000004	\$CORE:	MOV	#\$SCROUT,@ERRVEC	::	SET FOR TIMEOUT
4763	022360	005002				CLR	R2	::	SET UP BANK
4764	022362	062701	004000		1\$:	ADD	#4000,R1	::	INCREMENT BY 1K
4765	022366	062702	000040			ADD	#40,R2	::	1K STEP
4766	022372	005711				TST	(R1)	::	TRAP ON TIME OUT
4767	022374	022701	177776			CMP	#177776,R1	::	LAST ONE
4768	022400	001370				BNE	1\$::	NO--TRY AGAIN
4769	022402	162701	004000		\$SCROUT:	SUB	#4000,R1	::	
4770	022406	162702	000040		\$SIZEX:	SUB	#40,R2	::	DROP BACK
4771	022412	010006				MOV	R0,SP	::	RESTORE THE STACK
4772	022414	012637	000006			MOV	(SP)+,@ERRVEC+2	::	RESTORE ERROR VECTOR
4773	022420	012637	000004			MOV	(SP)+,@ERRVEC	::	
4774	022424	010137	022446			MOV	R1,\$LSTAD	::	LAST ADDRESS
4775	022430	010237	022450			MOV	R2,\$LSTBK	::	LAST BANK
4776	022434	012603				MOV	(SP)+,R3	::	RESTORE R3
4777	022436	012602				MOV	(SP)+,R2	::	RESTORE R2

4778 022440 012601
 4779 022442 012600
 4780 022444 000207
 4781 022446 000000
 4782 022450 000000
 4783
 4784
 4785
 4786
 4787
 4788
 4789
 4790
 4791
 4792 022452 010046
 4793 022454 016600 000002
 4794 022460 005740
 4795 022462 111000
 4796 022464 006300
 4797 022466 016000 022506
 4798 022472 000200
 4799
 4800
 4801
 4802
 4803 022474 011646
 4804 022476 016666 000004 000002
 4805 022504 000002
 4806
 4807
 4808
 4809
 4810
 4811
 4812
 4813
 4814 022506 022474
 4815 022510 017462
 4816 022512 017726
 4817 022514 017702
 4818 022516 017742
 4819 022520 020130
 4820
 4821 022522 020752
 4822
 4823 022524 020662
 4824 022526 021224
 4825 022530 021314
 4826 022532 022050
 4827 022534 022106
 4828
 4829
 4830
 4831
 4832
 4833 022536 104412

```

MOV (SP)+,R1 ;;RESTORE R1
MOV (SP)+,R0 ;;RESTORE R0
RTS PC
SLSTAD: .WORD 0 ;;CONTAINS THE LAST ADDRESS
SLSTBK: .WORD 0 ;;CONTAINS THE LAST BANK

```

.SBTTL TRAP DECODER

```

*****
*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
*GO TO THAT ROUTINE.

```

```

$TRAP: MOV RO, -(SP) ;;SAVE RO
MOV 2(SP),RO ;;GET TRAP ADDRESS
TST -(RO) ;;BACKUP BY 2
MOVB (RO),RO ;;GET RIGHT BYTE OF TRAP
ASL RO ;;POSITION FOR INDEXING
MOV $TRPAD(RO),RO ;;INDEX TO TABLE
RTS RO ;;GO TO ROUTINE

```

;;THIS IS USE TO HANDLE THE "GETPRI" MACRO

```

$TRAP2: MOV (SP), -(SP) ;;MOVE THE PC DOWN
MOV 4(SP), 2(SP) ;;MOVE THE PSW DOWN
RTI ;;RESTORE THE PSW

```

.SBTTL TRAP TABLE

```

*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
*BY THE "TRAP" INSTRUCTION.

```

ROUTINE	STARTING ADDRESS	CALL	DESCRIPTION
TRAP+1(104401)	022474	CALL=TYPE	TTY TYPEOUT ROUTINE
TRAP+2(104402)	017462	CALL=TYPOC	TYPE OCTAL NUMBER (WITH LEADING ZEROS)
TRAP+3(104403)	017726	CALL=TYPOS	TYPE OCTAL NUMBER (NO LEADING ZEROS)
TRAP+4(104404)	017742	CALL=TYPON	TYPE OCTAL NUMBER (AS PER LAST CALL)
TRAP+5(104405)	020130	CALL=TYPDS	TYPE DECIMAL NUMBER (WITH SIGN)
TRAP+6(104406)	020752	CALL=GT\$WR	GET SOFT-SWR SETTING
TRAP+7(104407)	020662	CALL=CK\$WR	TEST FOR CHANGE IN SOFT-SWR
TRAP+10(104410)	021224	CALL=RDCHR	TTY TYPEIN CHARACTER ROUTINE
TRAP+11(104411)	021314	CALL=RDLIN	TTY TYPEIN STRING ROUTINE
TRAP+12(104412)	022050	CALL=SAVREG	SAVE RO-R5 ROUTINE
TRAP+13(104413)	022106	CALL=RESREG	RESTORE RO-R5 ROUTINE

```

*THIS ROUTINE CLEARS THE RP11 AND DETERMINES WHICH DRIVES ARE AVAILABLE.
*THE TABLE 'DRVSTA' IS LOADED TO REFLECT THE SYSTEM STATUS.

```

```

RPINIT: SAVREG ;SAVE RO-R5

```

4834	022540	013701	001326		MOV	RPVEC,R1	: VECTOR ADDRESS	
4835	022544	005021			CLR	(R1)+	: SET INTERRUPT ADDRESS TO ZERO	
4836	022546	013711	001332		MOV	RPPRIO,(R1)	: RP11 PRIORITY	
4837	022552	005037	001304		CLR	DRVSTP	: CLEAR DRIVE TYPE STORAGE	
4838	022556	005037	001370		CLR	DRVSTA	: SET DRIVE STATUS TO OFFLINE	
4839	022562	005037	001372		CLR	DRVSTA+2	: SET DRIVE STATUS TO OFFLINE	
4840	022566	005037	001374		CLR	DRVSTA+4	: SET DRIVE STATUS TO OFFLINE	
4841	022572	005037	001376		CLR	DRVSTA+6	: SET DRIVE STATUS TO OFFLINE	
4842	022576	013704	001324		MOV	RPAOR,R4	: PUT RP11 ADDRESS INTO R4	
4843	022602	012764	000001	000004	MOV	#CLEAR,RPCS(R4)	: CLEAR THE RP11	
4844	022610	012764	000001	000004	MOV	#CLEAR,RPCS(R4)	: CLEAR THE RP11 AGAIN	
4845	022616	012701	000007		MOV	#7,R1	: 'DRVSTA' TABLE INDEX	
4846	022622	004737	022636	15:	JSR	PC,DRVINT	: CHECK THE DRIVE'S STATUS	
4847	022626	005301			DEC	R1	: DECREMENT THE TABLE INDEX	
4848	022630	100374			BFL	15	: BR IF NOT FINISHED	
4849	022632	104413			RESREG		: RESTORE R0-R5	
4850	022634	000207			RTS	PC	: RETURN	
4851								
4852							: ROUTINE TO CHECK THE DRIVE'S STATUS. DRIVE NUMBER MUST BE IN R1	
4853								
4854	022636	110164	000005		DRVINT: MOV	R1,RPCS+1(R4)	: SELECT DRIVE	
4855	022642	032764	001000	000000	BIT	#SUFU,RPDS(R4)	: SEE IF DRIVE UNSAFE	
4856	022650	001404			BEQ	15	: BR IF NOT UNSAFE	
4857	022652	112761	177777	001370	MOVB	#-1,DRVSTA(R1)	: SET INDICATOR TO 'UNSAFE'	
4858	022660	003417			BR	35	: EXIT	
4859	022662	032764	040000	000000	15:	BIT	#SUOL,RPDS(R4)	: IS DRIVE ONLINE ?
4860	022670	001422			BEQ	45	: BR IF NOT	
4861	022672	032764	004000	000000	BIT	#SUSI,RPDS(R4)	: SEEK INCOMPLETE SET ?	
4862	022700	001004			BNE	25	: BR IF SET	
4863	022702	032764	100000	000000	BIT	#SUDY,RPDS(R4)	: IS DRIVE READY ?	
4864	022710	001412			BEQ	45	: BR IF NOT	
4865	022712	112761	000001	001370	25:	MOVB	#1,DRVSTA(R1)	: SET DRIVE INDICATOR TO 'ONLINE'
4866	022720	032764	020000	000000	35:	BIT	#SURP03,RPDS(R4)	: IS THE DRIVE AN RPO3 ?
4867	022726	001403			BEQ	45	: BR IF NOT	
4868	022730	155137	001360	001304	45:	BISB	ATABIT(R1),DRVSTP	: SET RPO3 INDICATOR
4869	022736	000207			RTS	PC	: RETURN	
4870								
4871							: DO THE REQUESTED COMMAND. THE RP11 ADDRESS MUST BE IN R4	
4872								
4873	022740	104412			RP11: SAVREG		: SAVE R0 - R5	
4874	022742	113764	002500	000005	MOVB	DPB,RPCS+1(R4)	: SELECT THE DRIVE	
4875	022750	005001			CLR	R1	: CLEAR R1	
4876	022752	113701	002500		MOVB	DPB,R1	: DRIVE ADDRESS	
4877	022756	004737	023052		JSR	PC,CIR	: SEE IF DRIVE IS READY	
4878	022762	013746	002504		MOV	SAC,-(SP)	: COMPLEMENT THE WORD COUNT	
4879	022766	005416			NEG	(SP)	: COMPLEMENT	
4880	022770	012664	000006		MOV	(SP)+,RPMC(R4)	: LOAD THE WORD COUNT REGISTER	
4881	022774	013764	002506	000010	MOV	#BUF,RPBA(R4)	: BUFFER ADDRESS	
4882	023002	013764	002510	000012	MOV	#CYL,RPCA(R4)	: CYLINDER ADDRESS	
4883	023010	013764	002512	000014	MOV	#SEC,RPDA(R4)	: SECTOR/TRACK ADDRESS	
4884	023016	113764	002500	000005	MOVB	DPB,RPCS+1(R4)	: SELECT THE DRIVE	
4885	023024	153764	002503	000005	BISB	DPB+3,RPCS+1(R4)	: SET 'MODE' & 'HDR' BITS	
4886	023032	013746	002502		MOV	DPB+2,-(SP)	: COMMAND CODE	
4887	023036	153716	002501		BISB	DPB+1,(SP)	: 'MEX' BITS	
4888	023042	112664	000004		MOVB	(SP)+,RPCS(R4)	: START THE COMMAND	
4889	023046	104413			RESREG		: RESTORE R0 - R5	


```

4930 023050 000207          RTS      PC          ;RETURN
4931
4932          ;SEE IF DRIVE IS STILL ONLINE
4933
4934          CIR:
4935          MOV      #28, $ESCAPE      ;; ESCAPE TO 28 ON ERROR
4936          JSR      PC, DRVINT        ;; CHECK ON THE DRIVE'S STATUS
4937          JSR      PC, SAVRP        ;; STORE RP11 REGISTERS IN CASE OF ERROR
4938          TSTB     DRVSTA(R1)       ;; IS THE DRIVE STILL ONLINE ?
4939          BGT      38                ;; BR IF IT IS
4940          BLT      18                ;; BR IF DRIVE IS UNSAFE
4941          ERROR    76                ;; DRIVE HAS GONE OFFLINE
4942          ERROR    77                ;; DRIVE HAS BECOME UNSAFE
4943          BIT      #SH09, 2$WR      ;; LOOP ON THE ERROR ?
4944          BNE      CIR              ;; BR IF LOOP
4945          BISB     ATABIT(R1), DRVBA0 ;; SETUP TO DEASSIGN THE DRIVE
4946          JMP      28$EOP           ;; GO TO THE END OF PASS ROUTINE
4947          CLR      $ESCAPE          ;; CLEAR THE 'ESCAPE' FLAG
4948          RTS      PC              ;; RETURN
4949
4950          ;THIS ROUTINE WILL PROVIDE A STALL IN MILLISECONDS FOR A SPECIFIC
4951          ;AMOUNT OF TIME. STALL VALU. IS STORED IN 'STALL'.
4952          ;
4953          JSR      PC, STALL
4954
4955          STALL: MOV      STALL, -(SP) ;; PICKUP THE STALL TIME
4956          CLR      -(SP)             ;; CLEAR TEMP. LOCATION
4957          18:     SUB      #1, 2(SP)  ;; MORE STALL REQUIRED?
4958          BLO      38                ;; NO--BRANCH
4959          MOV      #100., (SP)      ;; STALL FOR ABOUT 1 MILLISECOND
4960          28:     TST      R0         ;; NOP TO KILL TIME
4961          DEC      (SP)             ;; COUNT
4962          BNE      28                ;; LOOP IF MORE COUNTS NEEDED
4963          BR      18
4964          38:     CMP      (SP)+, (SP)+ ;; CLEAN OFF THE STACK
4965          RTS      PC              ;; EXIT
4966
4967          ;THIS ROUTINE STORES THE RP11E REGISTERS FOR USE BY THE PROGRAM.
4968          ;THE PROGRAM DOES NOT USE THE 'LIVE' REGISTERS FOR ERROR DETERMINATION.
4969          ;CALL
4970          ;
4971          JSR      PC, SAVRP
4972          RETURN
4973
4974          SAVRP: MOV      R4, -(SP)   ;; SAVE R4
4975          MOV      RPAOR, R4         ;; RP11 ADDRESS
4976          MOV      RPOS(R4), $RPOS   ;; STORE RPOS
4977          MOV      RPER(R4), $RPER   ;; STORE RPER
4978          MOV      RPCS(R4), $RPCS   ;; STORE RPCS
4979          MOV      RPWC(R4), $RPWC   ;; STORE RPWC
4980          MOV      RPBA(R4), $RPBA   ;; STORE RPBA
4981          MOV      RPCA(R4), $RPCA   ;; STORE RPCA
4982          MOV      RPOA(R4), $RPOA   ;; STORE RPOA
4983          MOV      RPMI(R4), $RPMI   ;; STORE RPMI
4984          MOV      SUCA(R4), $SUCA   ;; STORE SUCA
4985          MOV      SILO(R4), $SILO   ;; STORE SILO
4986          MOV      (SP)+, R4         ;; RESTORE R4
4987          RTS      PC              ;; RETURN

```

```

4946
4947
4948
4949
4950 023300 105737 000041
4951 023304 001005
4952 023306 012702 000144
4953 023312 012701 044112
4954 023316 000404
4955 023320 012702 002734
4956 023324 012701 051472
4957 023330 012703 157776
4958 023334 005737 022202
4959 023340 100402
4960 023342 013703 022446
4961 023346 005046
4962 023350 016616 000002
4963 023354 010166 000002
4964 023360 005737 001322
4965 023364 001006
4966 023366 014341
4967 023370 005302
4968 023372 001375
4969 023374 012737 177777 001322
4970 023402 000207
4971
4972
4973
4974 023404 104412
4975 023406 005737 001322
4976 023412 001430
4977 023414 105737 000041
4978 023420 001005
4979 023422 012702 000144
4980 023426 012701 044112
4981 023432 000404
4982 023434 012702 002734
4983 023440 012701 051472
4984 023444 012703 157776
4985 023450 005737 022202
4986 023454 100402
4987 023456 013703 022446
4988 023462 014143
4989 023464 005302
4990 023466 001375
4991 023470 005037 001322
4992 023474 104413
4993 023476 000207
4994
4995
4996 023500 032764 100000 000000
4997 023506 001042
4998 023510 013746 001212
4999 023514 005046
5000 023516 162766 000001 000002
5001 023524 001411

```

;ROUTINE TO MOVE THE LOADER FROM HIGH MEMORY TO JUST ABOVE THE PROGRAM

```

MOVLDL: TSTB 41 ;WHO LOADED THE PROGRAM
        BNE 1$ ;BR IF NOT 'ABS'
        MOV #100.,R2 ;'ABS' LOADER SIZE
        MOV #BUFFER+200.,R1 ;RELOCATE TO HERE
        BR 2$ ;FINISH
1$: MOV #1500.,R2 ;'XXOP' SIZE
    MOV #BUFFER+3000.,R1 ;RELOCATE TO HERE (UPPER BOUNDARY)
2$: MOV #157776,R3 ;TRY THIS UPPER MEMORY ADDRESS
    TST SKT11 ;MEMORY MANAGEMENT ?
    BMI 3$ ;BR IF YES
    MOV SLSTAD,R3 ;USE ACTUAL HIGH ADDRESS
3$: CLR -(SP) ;CLEAR THE STACK
    MOV 2(SP), (SP) ;MOVE THE RETURN ADDRESS DOWN THE STACK
    MOV R1, 2(SP) ;BEGINNING ADDRESS OF BUFFER
    TST LDRFLG ;LOADER ALREADY RELOCATED ?
    BNE 5$ ;BR IF YES
4$: MOV -(R3), -(R1) ;MOVE A WORD
    DEC R2 ;COUNT IT
    BNE 4$ ;BR IF MORE TO GO
5$: MOV #-1, LDRFLG ;SET THE LOADER MOVED FLAG
    RTS PC ;RETURN

```

;ROUTINE TO RELOCATE THE LOADER TO UPPER MEMORY

```

RESLDL: SAVREG ;SAVE R0 - R5
        TST LDRFLG ;LOADER MOVED ?
        BEQ 5$ ;BR IF NOT
        TSTB 41 ;'ABS' LOADER
        BNE 1$ ;BR IF NOT
        MOV #100.,R2 ;'ABS' LOADER SIZE
        MOV #BUFFER+200.,R1 ;RELOCATE FROM HERE
        BR 3$ ;FINISH
1$: MOV #1500.,R2 ;'XXOP' SIZE
    MOV #BUFFER+3000.,R1 ;RELOCATE FROM HERE (UPPER BOUNDARY)
3$: MOV #157776,R3 ;TRY THIS UPPER MEMORY ADDRESS
    TST SKT11 ;MEMORY MANAGEMENT ?
    BMI 4$ ;BR IF YES
    MOV SLSTAD,R3 ;USE ACTUAL HIGH ADDRESS
4$: MOV -(R1), -(R3) ;MOVE A WORD
    DEC R2 ;COUNT IT
    BNE 4$ ;BR IF MORE TO GO
5$: CLR LDRFLG ;CLEAR THE LOADER MOVED FLAG
    RESPEG ;RESTORE R0 - R5
    RTS PC ;RETURN

```

;ROUTINE TO WAIT FOR DRIVE READY

```

DRVROY: BIT #SURDY,RPDS(R4) ;DRIVE READY ?
        BNE 6$ ;BR IF READY
        MOV TIMEOUT, -(SP) ;WAIT TIME: APPROX 1 SEC
        CLR -(SP) ;CLEAR WORKING LOCATION
1$: SUB #1, 2(SP) ;ANY TIME LEFT IN WAIT LOOP ?
    BEQ 3$ ;BR IF NOT

```

```

5002 023526 012716 000144          MOV      #100, (SP)      ;VALUE FOR APPROX 1 MS
5003 023532 032764 100000 000000 2$:  BIT      #SURDY, RPS(R4) ;DRIVE READY ?
5004 023540 001023          BNE     5$              ;BR IF READY
5005 023542 005316          DEC     (SP)           ;COUNT SOME TIME
5006 023544 001372          BNE     2$              ;BR IF NOT FINISHED
5007 023546 000763          BR      1$              ;COUNT MAJOR TIME AGAIN
5008 023550 004737 023172          JSR     PC, SAVRP      ;STORE RP11 REGISTERS FOR ERROR MESSAGE
5009 023554 012737 023564 001164 3$:  MOV     #4$, $ESCAPE   ;ESCAPE TO 4$ ON ERROR
5010 023562 104100          ERROR   100           ;DRIVE TIMED OUT
5011 023564 005037 001164          CLR     $ESCAPE       ;CLEAR THE 'ESCAPE' FLAG
5012 023570 062706 000004          ADD     #4, SP        ;CORRECT THE STACK POINTER
5013 023574 032777 001000 155336          BIT     #S109, $SWR   ;LOOP ON THE ERROR ?
5014 023602 001336          BNE     DRYRDY        ;LOOP IF SET
5015 023604 000177 155370          JMP     @BYPASS       ;GO THE 'BYPASS' ADDRESS
5016 023610 062706 000004          ADD     #4, SP        ;CORRECT THE STACK POINTER
5017 023614 000207          RTS     PC            ;RETURN
5018
5019
5020
5021          ;ROUTINE TO WAIT FOR CONTROLLER READY
5022 023616 032764 000200 000004  CONRDY: BIT      #RDY, RPS(R4)  ;CONTROLLER READY ?
5023 023624 001042          BNE     6$              ;BR IF READY
5024 023626 013746 001212          MOV     TIMEOUT, -(SP) ;WAIT TIME: APPROX 1 SEC
5025 023632 005046          CLR     -(SP)         ;CLEAR WORKING LOCATION
5026 023634 162766 000001 000002  1$:  SUB     #1, 2(SP)     ;ANY TIME LEFT IN WAIT LOOP ?
5027 023642 001411          BEQ     3$              ;BR IF NOT
5028 023644 012716 000144          MOV     #100, (SP)    ;VALUE FOR APPROX 1 MS
5029 023650 032764 000200 000004  2$:  BIT      #RDY, RPS(R4) ;CONTROLLER READY ?
5030 023656 001023          BNE     5$              ;BR IF READY
5031 023658 005316          DEC     (SP)           ;COUNT SOME TIME
5032 023662 001372          BNE     2$              ;BR IF NOT FINISHED
5033 023664 000763          BR      1$              ;COUNT MAJOR TIME AGAIN
5034 023666 004737 023172          JSR     PC, SAVRP      ;STORE RP11 REGISTERS FOR ERROR MESSAGE
5035 023672 012737 023702 001164 3$:  MOV     #4$, $ESCAPE   ;ESCAPE TO 4$ ON ERROR
5036 023700 104101          ERROR   101           ;CONTROLLER TIMED OUT
5037 023702 005037 001164          CLR     $ESCAPE       ;CLEAR THE 'ESCAPE' FLAG
5038 023706 062706 000004          ADD     #4, SP        ;CORRECT THE STACK POINTER
5039 023712 032777 001000 155220          BIT     #S109, $SWR   ;LOOP ON THE ERROR ?
5040 023720 001336          BNE     CONRDY        ;BR IF LOOPING
5041 023722 000177 155252          JMP     @BYPASS       ;'BYPASS' THE TEST
5042 023726 062706 000304          ADD     #4, SP        ;CORRECT THE STACK POINTER
5043 023732 000207          RTS     PC            ;RETURN
5044
5045          ;ROUTINE TO DETERMINE THE NUMBER OF SECTORS ON THE DRIVE BEING TESTED
5046
5047 023734 010046          SECNR: MOV     RO, -(SP)  ;SAVE RO
5048 023736 012737 117230 001244          MOV     #40600, PAKSIZ ;START WITH THE NUMBER OF SECTORS ON
5049 023744 005037 001246          CLR     PAKSIZ+2      ;AN RPO2
5050 023750 005000          CLR     RO            ;USE RO AS AN INDEX
5051 023752 113700 002500          MOV     DPB, RO       ;GET THE DRIVE NUMBER
5052 023756 136037 001360 001304          BIT     ATABIT(RO), DRV TYP ;IS THE DRIVE AN RPO3 ?
5053 023764 001404          BEQ     1$              ;BR IF NOT
5054 023766 006337 001244          ASL     PAKSIZ        ;DOUBLE THE SECTOR COUNT
5055 023772 006137 001246          ROL     PAKSIZ+2      ;GET THE CARRY
5056 023776 012600          MOV     (SP)+, RO     ;RESTORE RO
5057 024000 000207          RTS     PC            ;RETURN

```

5058
5059
5060
5061
5062
5063
5064
5065
5066
5067
5068
5069
5070
5071
5072
5073
5074
5075
5076
5077
5078
5079
5080
5081
5082
5083
5084
5085
5086
5087
5088
5089
5090
5091
5092
5093
5094
5095
5096
5097
5098
5099
5100
5101
5102
5103
5104
5105
5106
5107
5108
5109
5110
5111
5112
5113

024002 010046
024004 005237 001204
024010 022737 000020 001204
024016 101002
024020 005037 001204
024024 013700 001204
024030 006300
024036 036037 001400 001300
024040 001761
024042 012600
024044 000207

024046 005737 001260
024052 001066
024054 005737 001246
024060 003017
024062 005737 001244
024066 001460
024070 023737 001244 001254
024076 103010
024100 013737 001244 002504
024106 000337 002504
024112 005037 001244
024116 000410
024120 013737 001256 002504
024126 163737 001254 001244
024134 005637 001246
024140 005046
024142 113716 002512
024146 063716 001250
024152 121627 000012
024156 002404
024160 162716 000012
024164 105237 002513
024170 111637 002512
024174 113716 002513
024200 063716 001252
024204 121627 000024
024210 002404
024212 162716 000024
024216 005237 002510
024222 112637 002513
024226 000402
024230 062716 000002
024234 000207

;ROUTINE TO SELECT A PATTERN

```
PATSEL: MOV    RO, -(SP)      ; SAVE RO
1$:      INC    PATNUM       ; INCREMENT THE PATTERN NUMBER
        CMP    #16., PATNUM ; NUMBER AT MAXIMUM
        BHI   2$           ; BR IF NOT
        CLR    PATNUM       ; START AT ZERO AGAIN
2$:      MOV    PATNUM, RO   ; SEE IF THE PATTERN IS IN USE
        RSL   RO           ; CONVERT INTO A TABLE INDEX
        BIT   BITS(RO), PATTRN ; IN USE INDICATOR SET FOR THIS PATTERN ?
        BEQ   1$          ; BR IF NOT
        MOV   (SP)+, RO     ; RESTORE RO
        RTS   PC          ; RETURN
```

;ROUTINE TO INCREMENT THE DISK ADDRESS
; RETURN + 2 WHEN THE END OF THE PACK IS REACHED

```
INCADR: TST    OPRSEL       ; USING AN OPERATOR SPECIFIED ADDRESS ?
        BNE   5$           ; BR IF YES
        TST   PAKSIZ+2     ; SECTORS LEFT ?
        BGT   1$           ; BR IF AN UPPER COUNT
        TST   PAKSIZ       ; LOWER COUNT AT ZERO ?
        BEQ   5$           ; BR IF IT IS
        CMP   PAKSIZ, BLKS14 ; ENOUGH SECTORS LEFT FOR A FULL TRANSFER ?
        BHI   1$           ; BR IF THERE ARE
        MOV   PAKSIZ, SMC   ; USE THE RESIDUE COUNT FOR THIS TRANSFER
        SWAB SMC           ; CONVERT TO A WORD COUNT
        CLR   PAKSIZ       ; FORCE COUNT TO ZERO
        BR   2$           ; GO AND INCREMENT THE ADDRESS
        MOV   WC14, SMC    ; USE THE STANDARD WORD COUNT
        SUB   BLKS14, PAKSIZ ; DECREMENT THE TOTAL SECTORS AVAILABLE
        SBC   PAKSIZ+2     ; SUBTRACT ANY CARRY
        CLR   -(SP)       ; CLEAR THE STACK
2$:      MOVB  $SEC, (SP)   ; GET CURRENT SECTOR
        ADD   INCSEC, (SP) ; ADD THE SECTOR INCREMENT
        CMPB  (SP), #10.   ; EXCEED THE MAXIMUM ?
        BLT   3$           ; BR IF NOT
        SUB   #10., (SP)   ; CORRECT THE RESIDUE
        INCB  $TRK         ; INCREMENT THE TRACK
3$:      MOVB  (SP), $SEC   ; NEW SECTOR ADDRESS
        MOVB  $TRK, (SP)   ; GET CURRENT TRACK VALUE
        ADD   INCTRK, (SP) ; ADD THE TRACK INCREMENT
        CMPB  (SP), #20.   ; EXCEED THE MAXIMUM ?
        BLT   4$           ; BR IF NOT
        SUB   #20., (SP)   ; CORRECT THE SIZE
        INC   $CYL         ; INCREMENT THE CYLINDER ADDRESS
4$:      MOVB  (SP)+, $TRK ; NEW TRACK ADDRESS
        BR   6$           ; EXIT
5$:      ADD   #2, (SP)    ; INCREMENT RETURN - AT END
6$:      RTS   PC          ; RETURN
```

;ROUTINE TO SETUP A BUFFER WITH THE SEQUENTIAL HEADERS FOR CYLINDER 0
; TRACK 0

```
L0DSEC: SAVREG ; SAVE RO - R5
```

5114	024240	013701	002506		MOV	\$BUF,R1	: BUFFER ADDRESS
5115	024244	005003			CLR	R3	: SECTOR COUNTER
5116	024246	005021		1\$:	CLR	(R1)+	: UPPER WORD OF HEADER
5117	024250	005021			CLR	(R1)+	: CYLINDER/TRACK WORD OF HEADER
5118	024252	010321			MOV	R3,(R1)+	: SECTOR ADDRESS
5119	024254	005203			INC	R3	: INCREMENT THE SECTOR ADDRESS
5120	024256	022703	000012		CMP	#10.,R3	: FINISHED ?
5121	024258	001371			BNE	1\$: BR IF NOT
5122	024264	104413			RESREG		: RESTORE R0 - R5
5123	024266	000207			RTS	PC	: RETURN
5124							
5125							
5126							
5127	024270	104412					
5128	024272	005003					
5129	024274	113703	002512				
5130	024300	001002					
5131	024302	012703	000012				
5132	024306	013702	002504	1\$:	MOV	\$WC,R2	: WORD COUNT
5133	024312	013701	002506		MOV	\$BUF,R1	: BUFFER START ADDRESS
5134	024316	010321		2\$:	MOV	R3,(R1)+	: MOVE DATA INTO THE BUFFER
5135	024320	005302			DEC	R2	: DECREMENT THE COUNTER
5136	024322	001375			BNE	2\$: BR IF MORE TO DO
5137	024324	104413			RESREG		: RESTORE R0-R5
5138	024326	000207			RTS	PC	: RETURN
5139							
5140							
5141							
5142	024330	104412					
5143	024332	012701	043603				
5144	024336	013702	002504				
5145	024342	005003					
5146	024344	113711	002513	1\$:	MOV	\$TRK,(R1)	: MOVE TRACK ADDRESS INTO THE BUFFER
5147	024350	022701	000002		ADD	#2,R1	: INCREMENT THE BUFFER POINTER
5148	024354	005302			DEC	R2	: DECREMENT THE COUNTER
5149	024356	001372			BNE	1\$: BR IF MORE TO DO
5150	024360	104413			RESREG		: RESTORE R0-R5
5151	024362	000207			RTS	PC	: RETURN
5152							
5153							
5154							
5155							
5156	024364	104412					
5157	024366	013700	001204				
5158	024372	006300					
5159	024374	013701	002506				
5160	024400	013702	002504				
5161	024404	016003	001440	1\$:	MOV	PAT.PT(R0),R3	: PICKUP PATTERN POINTER
5162	024410	012704	000020		MOV	#16.,R4	: PATTERN COUNTER
5163	024414	012321		2\$:	MOV	(R3)+,(R1)+	: MOVE WORD 'N' INTO DATA BUFFER
5164	024416	005302			DEC	R2	: DECREMENT THE COUNTER
5165	024420	001403			BEQ	3\$: BR IF DONE
5166	024422	005304			DEC	R4	: DECREMENT THE PATTERN COUNTER
5167	024424	001767			BEQ	1\$: START PATTERN AGAIN
5168	024426	000772			BR	2\$: CONTINUE
5169	024430	104413		3\$:	RESREG		: RESTORE R0 - R5

;FILL THE BUFFER WITH THE SECTOR ADDRESS (CONTAINED IN '\$SEC')

FILSEC: SAVREG ;STORE R0-R5
 CLR R3 ;CLEAR THE INDEX
 MOV \$SEC,R3 ;SECTOR ADDRESS
 BNE 1\$;BR IF NOT SECTOR ZERO
 MOV #10.,R3 ;USE 10. AS VALUE FOR SECTOR ZERO
 1\$: MOV \$WC,R2 ;WORD COUNT
 MOV \$BUF,R1 ;BUFFER START ADDRESS
 2\$: MOV R3,(R1)+ ;MOVE DATA INTO THE BUFFER
 DEC R2 ;DECREMENT THE COUNTER
 BNE 2\$;BR IF MORE TO DO
 RESREG ;RESTORE R0-R5
 RTS PC ;RETURN

;ROUTINE TO FILL THE BUFFER WITH THE TRACK ADDRESS (CONTAINED IN '\$TRK')

FILTRK: SAVREG ;STORE R0-R5
 MOV #BUFFER+1,R1 ;BUFFER ADDRESS
 MOV \$WC,R2 ;WORD COUNT
 CLR R3 ;CLEAR THE DATA STORAGE
 1\$: MOV \$TRK,(R1) ;MOVE TRACK ADDRESS INTO THE BUFFER
 ADD #2,R1 ;INCREMENT THE BUFFER POINTER
 DEC R2 ;DECREMENT THE COUNTER
 BNE 1\$;BR IF MORE TO DO
 RESREG ;RESTORE R0-R5
 RTS PC ;RETURN

;THIS ROUTINE WILL MOVE THE 16 WORDS OF THE
 ;DESIRED PATTERN INTO THE DATA BUFFER.

SETBUF: SAVREG ;SAVE R0 - R5
 MOV PATNUM,R0 ;GET THE PATTERN NUMBER
 ASL R0 ;CONVERT NUMBER TO INDEX
 MOV \$BUF,R1 ;FIRST ADDRESS
 MOV \$WC,R2 ;WORD COUNT
 1\$: MOV PAT.PT(R0),R3 ;PICKUP PATTERN POINTER
 MOV #16.,R4 ;PATTERN COUNTER
 2\$: MOV (R3)+,(R1)+ ;MOVE WORD 'N' INTO DATA BUFFER
 DEC R2 ;DECREMENT THE COUNTER
 BEQ 3\$;BR IF DONE
 DEC R4 ;DECREMENT THE PATTERN COUNTER
 BEQ 1\$;START PATTERN AGAIN
 BR 2\$;CONTINUE
 3\$: RESREG ;RESTORE R0 - R5

```

S170 024432 000207          RTS      PC          ;RETURN
S171
S172          ;THIS ROUTINE LOADS THE PHYSICAL TRANSFER ADDRESS INTO 'SBUF'
S173
S174 024434 013737 001224 002506 GETBUF: MOV      ACTMEM,SBUF      ;LOWER 16 BITS OF PHYSICAL ADDRESS
S175 024442 013746 001226          MOV      ACTMEM+2,-(SP)    ;UPPER ADDRESS BITS ON THE STACK
S176 024446 042716 177774          BIC      BIC3,(SP)        ;LEAVE ONLY THE LOWER 2 BITS
S177 024452 006316          ASL      (SP)              ;ALIGN UPPER ADDRESS BITS
S178 024454 006316          ASL      (SP)              ;ALIGN UPPER ADDRESS BITS
S179 024456 006316          ASL      (SP)              ;ALIGN UPPER ADDRESS BITS
S180 024460 006316          ASL      (SP)              ;ALIGN UPPER ADDRESS BITS
S181 024462 112637 002501          MOVW    (SP)+,DPB+1       ;LOAD 'HEX' BITS
S182 024466 013746 002504          MOV      SWC,-(SP)        ;PUT WORD COUNT ON THE STACK
S183 024472 006316          ASL      (SP)              ;DOUBLE THE WORD COUNT
S184 024474 006337 001224          ADD     (SP)+,ACTMEM      ;FORM NEXT STARTING ADDRESS
S185 024480 006537 001226          ADC     ACTMEM+2          ;ADD ANY CARRY
S186 024504 023737 001226 001236          CMP     ACTMEM+2,HIMEM+2  ;EXCEEDING HIGHEST BANK ADDRESS
S187 024512 101005          BHI     1$                ;BR IF HIGHER
S188 024514 103413          BLO     2$                ;BR IF LOWER
S189 024516 023737 001224 001234          CMP     ACTMEM,HIMEM     ;CHECK LOWER 16 BITS
S190 024524 103407          BLO     2$                ;BR IF LOWER
S191 024526 013737 001240 001224 1$: MOV     LOMEM,ACTMEM      ;RESET ADDRESS
S192 024534 013737 001242 001226          MOV     LOMEM+2,ACTMEM+2 ;UPPER ADDRESS BITS
S193 024542 000734          BR      GETBUF           ;TRY AGAIN
S194 024544 000207          2$:   RTS      PC          ;RETURN
S195
S196          ;THIS ROUTINE COMPARES A 16 WORD DATA PATTERN AGAINST THE
S197          ;DATA BUFFER. THE PATTERN NUMBER IS IN 'PATNUM'
S198
S199 024546 104412          DATCMP: SAVREG          ;SAVE R0-R5
S200 024550 005737 001222          TST     MMACTV           ;MEMORY MANAGEMENT ACTIVE ?
S201 024554 001510          BEQ     2$                ;CR IF NOT

```

5202	024556	005037	172340		CLR	KIPAR0	: LOAD ADDRESS REGISTER 0
5203	024562	012737	000200	172342	MOV	#200, KIPAR1	: LOAD ADDRESS REGISTER 1
5204	024570	012737	000400	172344	MOV	#400, KIPAR2	: LOAD ADDRESS REGISTER 2
5205	024576	012737	177600	172356	MOV	#177600, KIPAR7	: ACCESS TO I/O PAGE
5206	024604	005046			CLR	-(SP)	: MAKE ROOM ON THE STACK
5207	024606	005046			CLR	-(SP)	: MAKE ROOM ON THE STACK
5208	024610	013756	002506	000002	MOV	\$BUF, 2(SP)	: LOWER 16 BITS OF STARTING ADDRESS
5209	024616	113716	002501		MOVB	DPB+1, (SP)	: UPPER ADDRESS BITS
5210	024622	006216			ASR	(SP)	: RIGHT JUSTIFY THE ADDRESS BITS
5211	024624	006216			ASR	(SP)	: RIGHT JUSTIFY THE ADDRESS BITS
5212	024626	006216			ASR	(SP)	: RIGHT JUSTIFY THE ADDRESS BITS
5213	024630	006216			ASR	(SP)	: RIGHT JUSTIFY THE ADDRESS BITS
5214	024632	012703	000006		MOV	#6, R3	: LOAD COUNTER
5215	024636	006216			ASR	(SP)	: CONVERT ADDRESS TO BLOCK ADDRESS
5216	024640	006066	000002		ROR	2(SP)	: LOWER 16 BITS
5217	024644	005303			DEC	R3	: DECREMENT THE COUNTER
5218	024646	001373			BNE	1\$: CONTINUE UNTIL FINISHED
5219	024650	042766	000177	000002	BIC	#177, 2(SP)	: CLEAR LOWER BLOCK ADDRESS BITS
5220	024656	016637	000002	172346	MOV	2(SP), KIPAR3	: LOAD ADDRESS REGISTER 3
5221	024664	062766	000200	000002	ADD	#200, 2(SP)	: INCREMENT BLOCK ADDRESS BY 4K
5222	024672	016637	000302	172350	MOV	2(SP), KIPAR4	: LOAD ADDRESS REGISTER # 4
5223	024700	012737	077406	172300	MOV	#77406, KIPDR0	: LOAD DESCRIPTOR REGISTER # 0
5224	024706	012737	077406	172302	MOV	#77406, KIPDR1	: LOAD DESCRIPTOR REGISTER # 1
5225	024714	012737	077406	172304	MOV	#77406, KIPDR2	: LOAD DESCRIPTOR REGISTER # 2
5226	024722	012737	077406	172306	MOV	#77406, KIPDR3	: LOAD DESCRIPTOR REGISTER # 3
5227	024730	012737	077406	172310	MOV	#77406, KIPDR4	: LOAD DESCRIPTOR REGISTER # 4
5228	024736	012737	077406	172316	MOV	#77406, KIPDR7	: LOAD DESCRIPTOR REGISTER # 7
5229	024744	023727	002504	010000	CMP	\$WC, #4096.	: WAS TRANSFER GREATER THAN 4K ?
5230	024752	101411			BLOS	2\$: BR IF NOT
5231	024754	062766	000200	000002	ADD	#200, 2(SP)	: INCREMENT THE ADDRESS BY 4K
5232	024762	016637	000002	172352	MOV	2(SP), KIPAR5	: LOAD ADDRESS REGISTER # 5
5233	024770	012737	077406	172312	MOV	#77406, KIPDR5	: LOAD DESCRIPTOR REGISTER # 5
5234	024776	013701	002506		MOV	\$BUF, R1	: LOAD STARTING ADDRESS
5235	025002	005737	001222		TST	MMACTV	: MEMORY MANAGEMENT ACTIVE ?
5236	025006	001425			BEQ	CMPAR	: BR IF NOT
5237	025010	042701	140000		BIC	#140000, R1	: CHANGE PHYSICAL BASE TO VIRTUAL BASE
5238	025014	052701	060000		BIS	#50000, R1	: START AT VIRTUAL 12K
5239	025020	062706	000304		ADD	#4, SP	: CORRECT THE STACK POINTER
5240	025024	012737	025042	000004	MOV	#3\$, ERRVEC	: CHANGE THE ERROR VECTOR
5241	025032	012737	000020	172516	MOV	#20, #SR3	: ENABLE 22 BIT ADDRESSING MODE
5242	025040	000402			BR	4\$: BR IF AN 11/70
5243	025042	062706	000304		ADD	#4, SP	: CORRECT THE STACK POINTER
5244	025046	012737	000006	000004	MOV	#ERRVEC+2, ERRVEC	: RESTORE THE ERROR VECTOR
5245	025054	012737	000001	177572	MOV	#1, #SR0	: ENABLE MEMORY MANAGEMENT
5246	025062	013737	001342	025352	CMPAR:	\$RPWC, CMCNT	: WORD COUNT TO WORKING LOCATION
5247	025070	063737	002504	025352	ADD	\$WC, CMCNT	: CALCULATE ACTUAL WORDS TRANSFERRED
5248	025076	013737	002510	025354	MOV	\$CYL, CMCYL	: CYLINDER ADDRESS
5249	025104	113737	002513	025360	MOVB	\$TRK, CMTRK	: TRACK ADDRESS
5250	025112	113737	002512	025356	MOVB	\$SEC, CMSEC	: SECTOR ADDRESS
5251	025120	013737	001320	025350	CMSTR:	CMP.CT, LIMIT	: DISPLAY LIMIT
5252	025126	005237	025350		INC	LIMIT	: CONVERT PARAMETER INTO LIMIT VALUE
5253	025132	010137	025362		MOV	R1, CMBUF	: STARTING ADDRESS OF SECTOR BUFFER
5254	025136	005037	025344		CLR	FRSTER	: CLEAR 'FIRST ERROR' INDICATOR
5255	025142	005037	025346		CLR	ERCTR	: CLEAR ERROR COUNTER
5256	025146	023727	025352	000400	CMP	CMCNT, #256.	: IS BUFFER SIZE GREATER THAN ONE SECTOR ?
5257	025154	101003			BHI	1\$: BR IF IT IS

5258	025156	013702	025352		MOV	CMCNT,R2		; LESS THAN, USE REMAINING BUFFER
5259	025162	000402			BR	ZS		
5260	025164	012702	000400	1S:	MOV	#256,R2		; COMPARE SECTOR
5261	025170	160237	025352	2S:	SUB	R2,CMCNT		; DECREMENT WORD COUNT
5262	025174	013704	001204		MOV	PAT,R4		; PATTERN NUMBER
5263	025200	006304			ASL	R4		; GENERATE INTO AN INDEX
5264	025202	012703	000020	CMDAT:	MOV	#16,R3		; R3 IS PATTERN POSITION COUNTER
5265	025206	016405	001440		MOV	PAT.PT(R4),R5		; PATTERN ADDRESS
5266	025212	022125		1S:	CMP	(R1)+,(R5)+		; COMPARE BUFFER WITH PATTERN
5267	025214	001406			BEQ	ZS		; BR IF EQUAL
5268	025216	032777	000010	153714	BIT	#SW03,@SWR		; SWITCH 3 SET ?
5269	025224	001002			BNE	ZS		; BR IF NOT
5270	025226	004737	025364		JSR	PC,CMPRT		; TYPE THE ERROR
5271	025232	005302		2S:	DEC	R2		; DECREMENT SIZE COUNT
5272	025234	001403			BEQ	ZS		; BR WHEN AT END
5273	025236	005303			DEC	R3		; DECREMENT PATT POS COUNT
5274	025240	001364			BNE	1S		; BR IF NOT AT END OF PATT
5275	025242	000757			BR	CMDAT		; RESTART THE PATTERN
5276	025244	004737	025562	3S:	JSR	PC,ENDCMP		; PRINT LAST LINE (IF ERRORS)
5277	025250	005737	025352		TST	CMCNT		; AT END OF BUFFER
5278	025254	003424			BLE	CMPRX		; BR IF AT END
5279	025256	005237	025356		INC	CMSEC		; INCREMENT SECTOR
5280	025262	023727	025356	000012	CMP	CMSEC,#10.		; SECTOR GREATER THAN MAX ?
5281	025270	103713			BLO	CMSTR		; BR IF NOT GREATER THAN MAX
5282	025272	005037	025356		CLR	CMSEC		; CLEAR SECTOR ADDRESS
5283	025276	005237	025360		INC	CMTRK		; INCREMENT TRACK
5284	025302	023727	025360	000024	CMP	CMTRK,#20.		; TRACK GREATER THAN MAX ?
5285	025310	103703			BLO	CMSTR		; BR IF NOT GREATER
5286	025312	005037	025360		CLR	CMTRK		; RESET TRACK ADDRESS
5287	025316	005237	025354		INC	CMCYL		; INCREMENT CYLINDER ADDRESS
5288	025322	000137	025120		JMP	CMSTR		; CONTINUE WITH COMPARE
5289	025326	104413		CMPRX:	RESREG			; RESTORE R0 - R5
5290	025330	005737	001222		TST	MMACTV		; MEMORY MANAGEMENT ACTIVE ?
5291	025334	001402			BEQ	1S		; BR IF NOT
5292	025336	005037	177572		CLR	@#SRO		; TURN OFF MEMORY MANAGEMENT
5293	025342	000207		1S:	RTS	PC		
5294								
5295	025344	000000		FRSTER:	.WORD	0		; FIRST ERROR INDICATOR
5296	025346	000000		ERCTR:	.WORD	0		; NUMBER OF ERRORS
5297	025350	000000		LIMIT:	.WORD	0		; DISPLAY LIMIT
5298	025352	000000		CMCNT:	.WORD	0		; WORD COUNT
5299	025354	000000		CMCYL:	.WORD	0		; CYLINDER ADDRESS
5300	025356	000000		CMSEC:	.WORD	0		; SECTOR ADDRESS
5301	025360	000000		CMTRK:	.WORD	0		; TRACK ADDRESS
5302	025362	000000		CMBUF:	.WORD	0		; BEGINNING ADDRESS OF SECTOR
5303								
5304								
5305								
5306	025364	005737	025344		CMPRX:	TST	FRSTER	; FIRST ERROR?
5307	025370	001037			BNE	CMPT1		; BR IF NOT
5308	025372	005737	001222		TST	MMACTV		; MEMORY MANAGEMENT ACTIVE ?
5309	025376	001011			BNE	1S		; BR IF ACTIVE
5310	025400	012737	025400	001116	MOV	#.SERRPC		; PC FOR ERROR MESSAGE
5311	025406	112737	000102	001114	MOV	#102,\$ITEMB		; COMPARISON ERROR MESSAGE NUMBER
5312	025414	004737	017232		JSR	PC,TYPERR		; REPORT THE ERROR
5313	025420	000412			BR	ZS		; CONTINUE

;TYPE DATA COMPARE ERRORS

```

5314 025422 005037 177572          1$: CLR      20SRO          ;TURN OFF MEMORY MANAGEMENT
5315 025426 012737 025426 001116  MOV      0, $ERRPC      ;PC FOR ERROR MESSAGE
5316 025434 112737 000103 001114  MOVB     #103, $ITEMB   ;COMPARISON ERROR MESSAGE WITH MM REGISTERS
5317 025442 004737 017232          JSR      PC, TYPERR     ;REPORT THE COMPARISON ERROR
5318 025446 012737 177777 025344  2$: MOV      #-1, FRSTER  ;SET FIRST ERROR INDICATOR
5319 025454 005737 001222          TST      MMACTV        ;MEMORY MANAGEMENT ACTIVE ?
5320 025460 001403          BEQ      CMPRT1        ;BR IF 0
5321 025462 012737 000001 177572  MOV      #1, 20SRO      ;ENABLE MEMORY MANAGEMENT AGAIN
5322 025470 005737 025350          CMPRT1: TST      LIMIT   ;TYPEOUT LIMIT REACHED ?
5323 025474 001403          DEQ      1$           ;BR IF IT HAS
5324 025476 005337 025350          DEC      LIMIT        ;DECREMENT LIMIT COUNTER
5325 025502 001004          BNE      2$           ;BR IF NOT AT LIMIT
5326 025504 032777 000020 153426  1$: BIT      #SW04, 20SWR ;PRINT ALL DATA COMPARE ERRORS ?
5327 025512 001420          BEQ      3$           ;BR IF NOT
5328 025514 010137 001122          MOV      R1, $B0ADR    ;ADDRESS OF BAD WORD
5329 025520 162737 000002 001122  SUB      #2, $B0ADR     ;ADJUST ADDRESS
5330 025526 016537 177776 001124  MOV      -2(R5), $G0DAT ;GOOD DATA
5331 025534 016137 177776 001126  MOV      -2(R1), $B0DAT ;BAD DATA
5332 025542 112737 000104 001114  MOVB     #104, $ITEMB  ;ERROR TABLE INDEX
5333 025550 004737 017232          JSR      PC, TYPERR     ;DISPLAY WORD THAT DIDN'T COMPARE
5334 025554 005237 025346          3$: INC      ERCTR      ;COUNT THE ERROR
5335 025560 000207          RTS      PC            ;RETURN
5336
5337
5338          ;LAST LINE OF COMPARE ERROR REPORTING
5339 025562 005737 025346          ENDCMP: TST      ERCTR  ;SEE IF ANY ERRORS
5340 025566 001406          BEQ      2$           ;BR IF NONE
5341 025570 012737 025600 001164  MOV      #1$, $ESCAPE  ;ESCAPE TO 1$ ON ERROR
5342 025576 104105          ERROR   105          ;NUMBER OF COMPARE ERRORS
5343 025600 005037 001164          1$: CLR      $ESCAPE   ;CLEAR THE ERROR ESCAPE ADDRESS
5344 025604 000207          2$: RTS      PC            ;RETURN
5345
5346          ;ROUTINE TO CLEAR THE RP11E
5347
5348          ;THE FIRST 'CLEAR' TERMINATES ANY ORDER WHICH IS IN PROGRESS. THE SECOND
5349          ;'CLEAR' CLEARS THE 'PROGRAM ERROR' WHICH MAY HAVE BEEN SET BY THE FIRST
5350          ;'CLEAR'.
5351
5352          ;NOTE: R4 MUST CONTAIN THE BASE ADDRESS OF THE RP11E
5353
5354 025606 012764 000001 000004  CLRP:  MOV      #CLEAR, RPCS(R4) ;CLEAR THE RP11
5355 025614 012764 000001 000004  MOV      #CLEAR, RPCS(R4) ;CLEAR THE RP11
5356 025622 000207          RTS      PC            ;RETURN
5357
5358          ;ROUTINE TO SIZE MEMORY AND TO SETUP FOR TEST 14
5359
5360 025624 012737 000200 022202  SIZMEM: MOV      #BIT07, $KT11 ;TELL THE 'SIZE' ROUTINE TO USE MM
5361 025632 004737 022144          JSR      PC, $SIZE     ;FIND OUT HOW MUCH MEMORY AND IF THE
5362          ;SYSTEM HAS A KT11
5363 025636 005737 022202          TST      $KT11        ;DOES THE SYSTEM HAVE A KT11 ?
5364 025642 100060          BPL      2$           ;BR IF NOT
5365 025644 023727 022450 001540  CMP      $LSTBK, #1540 ;MORE THAN 28K ON THE SYSTEM ?
5366 025652 101003          BHI      1$           ;BR IF MORE THAN 28K
5367 025654 005037 022202          CLR      $KT11        ;CLEAR MEMORY MANAGEMENT INDICATOR
5368 025660 000761          BR       SIZMEM        ;DO IT AGAIN
5369 025662 012737 157776 001302  1$: MOV      #157776, MEMSIZ ;ADDRESS OF LAST NON MEMORY MANAGEMENT LOCATION

```

5370	025670	013737	022450	001230		MOV	\$LSTBK,MAXMEM	:HIGH MEMORY BANK ADDRESS
5371	025676	005037	001232			CLR	MAXMEM+2	:CLEAR UPPER MEMORY ADDRESS BITS
5372	025702	006337	001230			ASL	MAXMEM	:CONVERT THE BANK COUNT INTO AN
5373	025706	006137	001232			ROL	MAXMEM+2	:ABSOLUTE MEMORY ADDRESS
5374	025712	006337	001230			ASL	MAXMEM	:CONVERT THE BANK COUNT INTO AN
5375	025716	006137	001232			ROL	MAXMEM+2	:ABSOLUTE MEMORY ADDRESS
5376	025722	006337	001230			ASL	MAXMEM	:CONVERT THE BANK COUNT INTO AN
5377	025726	006137	001232			ROL	MAXMEM+2	:ABSOLUTE MEMORY ADDRESS
5378	025732	006337	001230			ASL	MAXMEM	:CONVERT THE BANK COUNT INTO AN
5379	025736	006137	001232			R.L.	MAXMEM+2	:ABSOLUTE MEMORY ADDRESS
5380	025742	006337	001230			ASL	MAXMEM	:CONVERT THE BANK COUNT INTO AN
5381	025746	006137	001232			ROL	MAXMEM+2	:ABSOLUTE MEMORY ADDRESS
5382	025752	006337	001230			ASL	MAXMEM	:CONVERT THE BANK COUNT INTO AN
5383	025756	006137	001232			ROL	MAXMEM+2	:ABSOLUTE MEMORY ADDRESS
5384	025762	063737	022446	001230		ADD	\$LSTAD,MAXMEM	:ADD THE SIZE OF THE LAST BANK
5385	025770	005537	001232			ADC	MAXMEM+2	:HANDLE ANY CARRY
5386	025774	012737	020000	001264		MOV	#8192.,MAXWC	:SETUP FOR MAXIMUM TRANSFER SIZE
5387	025802	000433				BR	SS	:EXIT
5388	026004	013737	022446	001302	2\$:	MOV	\$LSTAD,MEMSIZ	:HIGH NON MEMORY MANAGEMENT ADDRESS
5389	026012	013737	022446	001230		MOV	\$LSTAD,MAXMEM	:HIGH NON MEMORY MANAGEMENT ADDRESS
5390	026020	005037	001232			CLR	MAXMEM+2	:DON'T NEED THE UPPER WORD
5391	026024	013737	001230	001264		MOV	MAXMEM,MAXWC	:CONVERT MEMORY SIZE TO WORD COUNT
5392	026032	162737	043602	001264		SUB	#BUFFER,MAXWC	:SUBTRACT STARTING ADDRESS OF BUFFER
5393	026040	000241				CLC		:CLEAR THE 'C' BIT
5394	026042	006037	001264			ROR	MAXWC	:CHANGE TO WORD COUNT
5395	026046	105737	000041			TSTB	41	:SEE WHICH LOADER
5396	026052	001004				BNE	4\$:BR IF 'XXDP'
5397	026054	162737	000144	001264	3\$:	SUB	#100.,MAXWC	: 'ABS' LOADER SIZE (PLUS A LITTLE BIT)
5398	026062	000403				BR	5\$:EXIT
5399	026064	162737	002734	001264	4\$:	SUB	#1500.,MAXWC	: 'XXDP' SIZE
5400	026072	070207			5\$:	RTS	PC	:RETURN

5401
5402 :THIS ROUTINE IS USED TO ENSURE THAT THE BUS ADDRESS
5403 :OF THE RP11 IS SETUP TO READ THE PROPER VALUE.
5404 :IT WILL ALSO READ THE ADDRESS FROM THE TTY IF
5405 :REQUIRED.
5406 :NOTE: THIS ROUTINE DESTROYS R0 - R4
5407

5408	026074	005737	001216			GETADR: TST	BUSADR	:INPUT FROM TTY REQUESTED?
5409	026100	001425				BEQ	3\$:NO--BRANCH
5410	026102	005037	001216			CLR	BUSADR	:YES--CLEAR THE REQUEST FLAG
5411	026106	104401	001172		1\$:	TYPE	,SCLF	:CR-LF
5412	026112	104401	032106			TYPE	,RPADR	: 'RPADR='
5413	026116	013746	001324			MOV	RPADR,-(SP)	:SAVE RPAADR FOR TYPEOUT
5414								:RP11 ADDRESS
5415	026122	104403				TYPOS		:GO TYPE--OCTAL ASCII
5416	026124	006				.BYTE	6	:TYPE 6 DIGIT(S)
5417	026125	001				.BYTE	1	:TYPE LEADING ZEROS
5418	026126	104401	031246			TYPE	,SLASH	: '/'
5419	026132	104411				RDLIN		:GET THE ENTRY
5420	026134	012601				MOV	(SP)+,R1	:STARTING ADDRESS OF INPUT ASCII STRING
5421	026136	004037	027062			JSR	R0,CK.NUM	:CHECK THE NUMBER
5422	026142	026106				1\$:ILLEGAL INPUT
5423	026144	026150				2\$:TERMINATED WITH A 'CR'
5424	026146	026154				3\$:NO ENTRY - 'CR' ONLY
5425	026150	010237	001324		2\$:	MOV	R2,RPADR	:STORE THE ADDRESS

M08

MD-11-DZRPY-B, RP11-E FUNCTIONAL LOGIC & R/W TEST
DZRPYB.CMB TRAP TABLE

MACY11 27(732) 02-NOV-76 14:50 PAGE 104

```

5426 026154 013701 000004          3S:  MOV   ERRVEC,R1      ;SAVE THE ERROR VECTOR
5427 026160 012737 026200 000004    MOV   #4S,ERRVEC    ;SETUP FOR TRAP
5428 026166 005777 153132          TST   JRPADR        ;CHECK FOR RP11
5429 026172 010137 000004          MOV   R1,ERRVEC     ;RESTORE ERROR VECTOR
5430 026176 000207                RTS   PC             ;RETURN
5431 026200 010137 000004          4S:  MOV   R1,ERRVEC     ;RESTORE ERROR VECTOR
5432 026204 022626                CMP   (SP)+,(SP)+   ;CLEAN OFF THE STACK
5433 026206 012737 026216 001164    MOV   #5S,SESCAPE   ;ESCAPE TO 5S ON ERROR
5434 026214 104001                ERROR 1             ;REPORT THE ERROR
5435 026216 005037 001164          5S:  CLR   SESCAPE      ;CLEAR ERROR ESCAPE ADDRESS
5436 026222 005737 000042          TST   #42           ;IS THERE A MONITOR?
5437 026226 001727                BEQ   1S            ;NO--GO ASK FOR ADDRESS
5438 026230 005037 016766          CLR   SEOPCT       ;NO PASSES
5439 026234 000137 016606          JMP   SEOP          ;GO TO END OF PROGRAM

5440
5441 ;THIS ROUTINE ALLOWS THE OPERATOR TO CONTROL THE PROGRAM
5442 ; (CONVERSATION MODE)
5443
5444 026240 104401 031250          ENTPRM: TYPE ,ENTNM ;'ENTER TEST NUMBER'
5445 026244 104411                RDLIN              ;GET THE ENTRY
5446 026246 012601                MOV   (SP)+,R1     ;STARTING ADDRESS OF INPUT ASCII STRING
5447 026250 004037 027062          JSR   RD,CK.NUM    ;CHECK THE NUMBER
5448 026254 026240                ENTPRM            ;ILLEGAL INPUT
5449 026256 026262                1S               ;TERMINATED WITH A 'CR'
5450 026260 026306                2S               ;NO ENTRY - 'CR' ONLY
5451 026262 022702 000020          1S:  CMP   #17777,R2   ;VALID NUMBER?
5452 026266 101764                BLOS ENTprm       ;BR IF NOT
5453 026270 006302                ASL   R2           ;CONVERT INTO A BIT INDEX
5454 026272 016237 001400 001276    MOV   BITS(R2),TSTNMS ;TEST SELECTION BIT
5455 026300 006202                ASR   R2           ;CHANGE BACK INTO A WORD COUNT
5456 026302 010246                MOV   R2,-(SP)     ;SAVE THE TEST NUMBER ENTRY
5457 026304 000405                BR   ENTPRM       ;GET THE DRIVE NUMBER
5458 026306 012737 017777 001276    2S:  MOV   #17777,TSTNMS ;SELECT ALL TESTS
5459 026314 012746 177777          MOV   #-1,-(SP)   ;ALL TESTS SELECTED
5460 026320 005037 001312          ENTPR1: CLR  DRVSEL ;CLEAR DRIVE SELECTION WORD
5461 026324 004737 022536          JSR   PC,RPINIT   ;CHECK THE DRIVE STATUS
5462 026330 012737 000312 001210    MOV   #202,MAXCYL ;ASSUME ONLY RPO2'S
5463 026336 005737 001304          TST   DRVSTYP     ;SEE WHICH TYPE
5464 026340 001403                BEQ   1S           ;BR IF ONLY RPO2'S
5465 026344 012737 000625 001210    MOV   #405,MAXCYL ;RPO3'S
5466 026352 104401 031276          1S:  TYPE ,DRVNM      ;ENTER DRIVE NUMBER
5467 026356 104411                RDLIN              ;GET THE ENTRY
5468 026360 012601                MOV   (SP)+,R1     ;STARTING ADDRESS OF INPUT ASCII STRING
5469 026362 004037 027062          JSR   RD,CK.NUM    ;CHECK THE NUMBER
5470 026366 026352                1S               ;ILLEGAL INPUT
5471 026370 026374                2S               ;TERMINATED WITH A 'CR'
5472 026372 026450                5S               ;NO ENTRY - 'CR' ONLY
5473 026374 022702 000010          2S:  CMP   #8.,R2     ;VALID ENTRY?
5474 026400 101764                1S               ;BR IF NOT
5475 026402 116237 001360 001312    MOVB  ATABIT(R2),DRVSEL ;DRIVE SELECTION BIT
5476 026410 105762 001370          TSTB  DRVSTA(R2)  ;IS DRIVE ONLINE
5477 026414 003035                BGT   8S           ;BR IF ONLINE
5478 026416 104401 031323          TYPE ,DRIVE       ;'DRIVE'
5479 026422 105762 001370          TSTB  DRVSTA(R2)  ;CHECK DRIVE STATUS AGAIN
5480 026426 100403                BMI   3S           ;BR IF UNSAFE
5481 026430 104401 027236          TYPE ,OFFLIN      ;'OFFLINE'

```

Address	Offset	Value	Label	Operation	Comment
5482	026434	000402		BR	4S CONTINUE
5483	026436	104401	027222	3S: TYPE	'UNSAFE'
5484	026442	104401	001173	4S: TYPE	'\$CRLF
5485	026446	000724		BR	ENTPR1
5486	026450	005001		CLR	R1
5487	026452	005716		TST	(SP)
5488	026454	100403		BMI	6S
5489	026456	022716	000015	CMP	#15,(SP)
5490	026462	101733		BLOS	1S
5491	026464	105761	001370	6S: TSTB	DRVSTA(R1)
5492	026470	003403		BLE	7S
5493	026472	156137	001360 001312	BISB	ATABIT(R1),DRVSEL
5494	026500	005201		7S: INC	R1
5495	026502	020127	000010	CMP	R1,#8
5496	026506	103766		BLO	6S
5497	026510	022726	000014	8S: CMP	#14,(SP)+
5498	026514	001137		BNE	ENTPRX
5499	026516	104401	031332	ENTPR2: TYPE	'CHNGPM
5500	026522	104411		ROLIN	
5501	026524	123627	000131	CMPB	#(SP)+,#'Y
5502	026530	001131		BNE	ENTPRX
5503	026532	104401	031373	TYPE	'MXWRDS
5504	026536	013746	001264	MOV	MAXWC,-(SP)
5505					
5506	026542	104403		TYPOS	
5507	026544	005		.BYTE	5
5508	026545	000		.BYTE	0
5509	026546	104401	001173	TYPE	'\$CRLF
5510	026552	104401	031447	1S: TYPE	'ENTWC
5511	026556	104411		ROLIN	
5512	026560	012601		MOV	(SP)+,R1
5513	026562	004037	027062	JSR	RD,CK.NUM
5514	026566	026552		1S	
5515	026570	026574		2S	
5516	026572	026614		ENTPR3	
5517	026574	020237	001264	2S: CMP	R2,MAXWC
5518	026600	101364		BHI	1S
5519	026602	020237	001262	CMP	R2,LMTBK
5520	026606	101361		BHI	1S
5521	026610	010237	001256	MOV	R2,WC14
5522	026614	104401	031511	ENTPR3: TYPE	'CHNGPT
5523	026620	104411		ROLIN	
5524	026622	012601		MOV	(SP)+,R1
5525	026624	004037	027062	JSR	RD,CK.NUM
5526	026630	026614		ENTPR3	
5527	026632	026636		1S	
5528	026634	026642		ENTPR4	
5529	026636	010237	001300	1S: MOV	R2,PATRN
5530	026642	104401	031564	ENTPR4: TYPE	'CHNGAD
5531	026646	104411		ROLIN	
5532	026650	123627	000131	CMPB	#(SP)+,#'Y
5533	026654	001057		BNE	ENTPRX
5534	026656	104401	031637	TYPE	'ONEPAT
5535					
5536	026662	104401	032000	1S: TYPE	'ENTCYL
5537	026666	104411		ROLIN	

5538	026670	012601		MOV	(SP)+,R1	: STARTING ADDRESS OF INPUT ASCII STRING
5539	026672	004037	027062	JSR	RO,CK.NUM	: CHECK THE NUMBER
5540	026676	026662		1\$: ILLEGAL INPUT
5541	026700	026704		2\$: TERMINATED WITH A 'CR'
5542	026702	026662		1\$: NO ENTRY - 'CR' ONLY
5543	026704	020237	001210	2\$:	CMP R2,MAXCYL	: TOO LARGE ?
5544	026710	101364		1\$: BR IF TOO LARGE
5545	026712	010237	001266	MOV	R2,SCYL	: LOAD CYLINDER ADDRESS
5546	026716	104401	032031	3\$:	TYPE ,ENTRX	: 'ENTER TRACK ADDRESS'
5547	026722	104411			RDLIN	: GET THE ENTRY
5548	026724	012601		MOV	(SP)+,R1	: STARTING ADDRESS OF INPUT ASCII STRING
5549	026726	004037	027062	JSR	RO,CK.NUM	: CHECK THE NUMBER
5550	026732	026716		3\$: ILLEGAL INPUT
5551	026734	026740		4\$: TERMINATED WITH A 'CR'
5552	026736	026716		3\$: NO ENTRY - 'CR' ONLY
5553	026740	020227	000023	4\$:	CMP R2,#19.	: VALID ENTRY ?
5554	026744	101364		3\$: BR IF NOT
5555	026746	010237	001270	MOV	R2,STRK	: TRACK ADDRESS
5556	026752	104401	032057	5\$:	TYPE ,ENTSEC	: 'ENTER SECTOR ADDRESS'
5557	026756	104411			RDLIN	
5558	026760	012601		MOV	(SP)+,R1	: STARTING ADDRESS OF INPUT ASCII STRING
5559	026762	004037	027062	JSR	RO,CK.NUM	: CHECK THE NUMBER
5560	026766	026752		5\$: ILLEGAL INPUT
5561	026770	026774		6\$: TERMINATED WITH A 'CR'
5562	026772	026752		5\$: NO ENTRY - 'CR' ONLY
5563	026774	020227	000011	6\$:	CMP R2,#9.	: VALID ENTRY ?
5564	027000	101364		5\$: BR IF NOT
5565	027002	010237	001272	MOV	R2,SECT	: LOAD SECTOR ADDRESS
5566	027006	012737	177777 001260	MOV	#-1,OPRSEL	: SET ADDRESS FLAG
5567	027014	000207		ENTPRX:	RTS PC	: RETURN

: THIS ROUTINE IS USED TO CHECK IF AN ASCII CHARACTER IS A DIGIT BETWEEN 0 AND 7.

5571				MOV	#ADR,R1	: ADDRESS OF ASCII CHARACTER
5572				JSR	RO,CK.OCT	: CHECK THE CHARACTER
5573				RETURN1		: CHARACTER IS IN R2 AS A
5574						: OCTAL DIGIT
5575				RETURN2		: CHARACTER IS NOT BETWEEN 0-7

5578	027016	121127	000060	CK.OCT:	CMPB (R1),#0	: LESS THAN ZERO?
5579	027022	103407		BLO	1\$: YES -- BRANCH
5580	027024	121127	000067	CMPB	(R1),#7	: GREATER THAN SEVEN?
5581	027030	101004		BHI	1\$: YES -- BRANCH
5582	027032	111102		MOVB	(R1),R2	: GET THE CHARACTER
5583	027034	042702	177770	BIC	#1C7,R2	: STRIP AWAY THE ASCII
5584	027040	000401		BR	2\$: BYPASS RETURN ADJUST
5585	027042	005720		1\$:	TST (R0)+	: ADJUST FOR RETURN
5586	027044	000200		2\$:	RTS R0	: RETURN

: THIS ROUTINE WILL CHECK AN ASCII CHARACTER TO DETERMINE WHAT IT IS.

5589				MOV	#ADR,R1	: ADDRESS OF ASCII CHARACTER
5590				JSR	RO,CK.CHR	: CHECK CHARACTER
5591				RETURN	ADR1	: UNKNOWN CHARACTER

```

5594 ; RETURN ADR2 ;CARRIAGE RETURN * (R1)=ADR+1
5595
5596 027046 105711 CK.CHR: TSTB (R1) ;"CARRIAGE RETURN"?
5597 027050 001002 BNE 1$ ;NO -- BRANCH
5598 027052 005720 TST (R0)+ ;CARRIAGE RETURN
5599 027054 005201 INC R1 ;MOVE POINTER TO NEXT CHARACTER
5600 027056 011000 1$: MOV (R0),R0 ;UNKNOWN CHARACTER
5601 027060 000200 RTS R0 ;RETURN
5602
5603
5604 ; THIS ROUTINE CHECKS AN ASCIZ STRING FOR LEGAL CHARACTERS
5605 ; AND FORMS AN OCTAL NUMBER IN R2
5606
5607 :
5608 : MOV @ADR,R1 ; ADDRESS OF ASCIZ STRING
5609 : JSR R0,CK.NUM ; GO FORM THE NUMBER
5610 : RETURN ADR1 ; ILLEGAL CHARACTER IN THE INPUT STRING
5611 : RETURN ADR2 ; "CR" --R2=NUMBER
5612 : RETURN ADR3 ; 'CR' WAS FIRST ENTRY
5613
5614 027062 010346 CK.NUM: MOV R3,-(SP) ;SAVE R3
5615 027064 005003 CLR R3 ;START NUMBER AT ZERO
5616 027066 004037 027016 JSR R0,CK.OCT ;OCTAL DIGIT?
5617 027072 000404 BR 1$ ;YES--BRANCH
5618 027074 004037 027046 JSR R0,CK.CHR ;CHECK ONE CHARACTER
5619 027100 027154 6$ ;ILLEGAL CHARACTER
5620 027102 027146 8$ ;CARRIAGE RETURN
5621 027104 005201 1$: INC R1 ;MOVE TO NEXT CHARACTER
5622 027106 006303 ASL R3 ;FOR THE OCTAL NUMBER IN R3
5623 027110 103421 BCS 6$ ;DON'T LET IT GET TO BIG
5624 027112 006303 ASL R3
5625 027114 103417 BCS 6$
5626 027116 006303 ASL R3
5627 027120 103415 BCS 6$
5628 027122 060203 R00 R2,R3
5629 027124 004037 027016 JSR R0,CK.OCT ; IS THIS AN OCTAL DIGIT?
5630 027130 000765 BR 1$ ; YES--MAKE IT PART OF THE NUMBER
5631 027132 010302 2$: MOV R3,R2 ;SAVE THE OCTAL NUMBER
5632 027134 005003 CLR R3 ;START WITH ZERO INDEX
5633
5634 027136 004037 027046 3$: JSR R0,CK.CHR ;CHECK ONE CHARACTER
5635 027142 027154 6$ ;ILLEGAL CHARACTER
5636 027144 027150 5$ ;CARRIAGE RETURN
5637 027146 005723 8$: TST (R3)+ ;'CR' ONLY
5638 027150 005723 5$: TST (R3)+ ;"CR"
5639 027152 060300 R00 R3,R0 ;YES--SAVE THE OCTAL NUMBER
5640 027154 012603 6$: MOV (SP)+,R3 ;RESTORE R3
5641 027156 011000 MOV (R0),R0 ;PICKUP EXIT ADDRESS
5642 027160 000200 RTS R0 ;RETURN
5643
5644 .SBTTL TELETYPE MESSAGES
5645 027162 005015 042012 044522 DRSTAT: .ASCIZ <15><12><12>DRIVE STATUS:0<15><12><12>
5646 027170 042526 051440 040524
5647 027176 052524 035123 005015
5648 027204 000012
5649 027206 020040 020040 047440 ONLINE: .ASCIZ @ ONLINE@

```


5650	027214	046116	047111	000105		
5651	027222	020040	020040	052440	UNSAFE: .ASCIZ	UNSAFE
5652	027230	051516	043101	000105		
5653	027238	020040	020040	047440	OFFLIN: .ASCIZ	OFFLINE
5654	027244	043106	044514	042516		
5655	027252	000000				
5656	027260	000062	051040	030120	RP02: .ASCIZ	RP02
5657	027268	020040	050122	031460	RP03: .ASCIZ	RP03
5658	027276	000000				
5659	027284	000000				
5660	027292	000000				
5661	027300	051450	044522	042526	DRVTST: .ASCIZ	DRIVE(S) TO BE USED
5662	027308	041040	020051	047524		
5663	027316	041040	020105	051525		
5664	027324	047516	000040			
5665	027332	051000	042516	000	NONE: .ASCIZ	NONE
5666	027340	051015	000		COMMA: .ASCIZ	,
5667	027348	052123	005012	042524	TSTING: .ASCIZ	<15><12><12>TESTING WITH DRIVE
5668	027356	044527	047111	020107		
5669	027364	044527	044124	042040		
5670	027372	044522	042526	000040		
5671	027380	020040	000		BLNKS2: .ASCIZ	
5672	027388	015	005012	047520	MSG15A: .ASCIZ	<15><12><12>POWER FAIL TEST
5673	027396	042527	020122	040506		
5674	027404	046111	052040	051505		
5675	027412	000124				
5676	027420	005015	052012	051125	MSG15B: .ASCII	<15><12><12>TURN CPU POWER OFF AFTER THIS MESSAGE, WAIT<15><12>
5677	027428	020116	050103	020125		
5678	027436	047520	042527	020122		
5679	027444	047440	043040	043040		
5680	027452	020040	043101	042524		
5681	027460	020122	044124	051511		
5682	027468	046440	051505	040523		
5683	027476	042507	020054	040527		
5684	027484	052111	005015			
5685	027492	020065	042523	047503	.ASCIZ	25 SECONDS, THEN TURN CPU POWER ON AGAIN<15><12>
5686	027500	042116	026123	052040		
5687	027508	042510	020116	052524		
5688	027516	047122	041440	052520		
5689	027524	050040	053517	051105		
5690	027532	020040	020117	020116		
5691	027540	040440	040507	047111		
5692	027548	005015	000			
5693	027556	015	042412	042116	MSG15C: .ASCIZ	<15><12>END OF POWER FAIL TEST<15><12>
5694	027564	047440	020106	047520		
5695	027572	042527	020122	040506		
5696	027580	046111	052040	051505		
5697	027588	006524	000012			
5698	027596	005015	050012	047522	MSG15D: .ASCIZ	<15><12><12>PROGRAM WILL LOOP, WAITING FOR DRIVE TO COME ONLINE<15><12>
5699	027604	051107	046501	053440		
5700	027612	046111	020114	047514		
5701	027620	050117	020054	040527		
5702	027628	052111	047111	020107		
5703	027636	047506	020122	051104		
5704	027644	053111	020105	047524		
5705	027652	041440	046517	020105		

5706	027656	047117	044514	042516
5707	027664	005015	000	
5708				
5709	027667	015	005012	042510
5710	027674	042101	040440	044514
5711	027702	047107	042515	052116
5712	027710	051040	052517	044524
5713	027716	042516		
5714	027720	005015	041412	047117
5715	027726	042516	052103	040440
5716	027734	045040	046525	042520
5717	027742	020122	042502	053524
5718	027750	042505	020116	044520
5719	027756	051516	045040	031461
5720	027764	031115	023040	045040
5721	027772	031461	031103	047440
5722	030000	020116	044124	020105
5723	030006	050122	030461	020105
5724	030014	040502	045503	046120
5725	030022	047101	105	
5726	030025	015	020012	051050
5727	030032	043105	051105	052040
5728	030040	020117	044124	020105
5729	030046	050101	046120	041511
5730	030054	041101	042514	046440
5731	030062	044501	052116	047105
5732	030070	047101	042503	046440
5733	030076	047101	040523	020114
5734	030104	047506	020122	044124
5735	030112	020105	051104	053111
5736	03012	020105	042523	052524
5737	030126	027120	051	
5738	030131	015	005012	047524
5739	030136	043507	042514	051440
5740	030144	044527	041524	020110
5741	030152	020067	047506	020122
5742	030160	042510	042101	051440
5743	030166	046105	041505	044524
5744	030174	047117	005015	000
5745	030201	015	042412	052116
5746	030206	051105	044040	040505
5747	030214	020104	042101	051104
5748	030222	051505	020123	044450
5749	030230	020116	041517	040524
5750	030236	024514	020072	000
5751				
5752	030243	015	005012	050122
5753	030250	030461	020105	047503
5754	030256	052116	047522	020114
5755	030264	040520	042516	020114
5756	030272	053523	052111	044103
5757	030300	052040	051505	000124
5758	030306	005015	054524	042520
5759	030314	040440	023440	051103
5760	030322	020047	047524	051440
5761	030330	040524	052122	052040

MSG16A: .ASCII <15><12><12>@HEAD ALIGNMENT ROUTINE@

.ASCII <15><12><12>@CONNECT A JUMPER BETWEEN PINS J13M2 & J13C2 ON THE RP11E BA

.ASCII <15><12>@ (REFER TO THE APPLICABLE MAINTENANCE MANUAL FOR THE DRIVE SETU

.ASCIZ <15><12><12>@TOGGLE SWITCH 7 FOR HEAD SELECTION@<15><12>

MSG16B: .ASCIZ <15><12>@ENTER HEAD ADDRESS (IN OCTAL): @

MSG17A: .ASCIZ <15><12><12>@RP11E CONTROL PANEL SWITCH TEST@

MSG17B: .ASCIZ <15><12>@TYPE A 'CR' TO START TEST AFTER EACH SETUP@<15><12>

5762	030336	051505	020124	043101
5763	030344	042524	020122	040505
5764	030352	044103	051440	052105
5765	030360	050125	005015	000
5766	030368	015	042012	051511
5767	030376	041101	042514	042040
5768	030400	044522	042526	021440
5769	030406	000040		
5770	030410	05015	047105	041101
5771	030416	042514	042040	044522
5772	030424	042526	021440	000040
5773	030432	005015	042523	020124
5774	030440	051047	040505	020104
5775	030446	047117	054514	020047
5776	030454	047117	042040	044522
5777	030462	042526	000040	
5778	030466	005015	042523	020124
5779	030474	051047	040505	026504
5780	030502	051127	052111	023505
5781	030510	047440	020116	051104
5782	030516	053111	020105	000
5783	030523	015	051413	052105
5784	030530	023440	051127	052111
5785	030536	020105	047514	045503
5786	030544	052517	023524	040440
5787	030552	042116	041440	042514
5788	030560	051101	052040	042510
5789	030566	046040	040517	051440
5790	030574	044527	041524	042510
5791	030502	123		
5792	030603	015	047412	020116
5793	030610	044124	020105	050122
5794	030616	030461	020105	047503
5795	030624	052116	047522	020114
5796	030632	040520	042516	000114
5797	030640	005015	042523	020124
5798	030646	044124	020105	047506
5799	030654	046114	053517	047111
5800	030662	020107	041517	040524
5801	030670	020114	040526	052514
5802	030676	051505	044440	020116
5803	030704	044124	020105	054503
5804	030712	044514	042116	051105
5805	030720	046040	040517	051440
5806	030726	044527	041524	042510
5807	030734	035123	000040	
5808	030740	05015	042523	020124
5809	030746	053523	052111	044103
5810	030754	051505	052040	020117
5811	030762	000		
5812	030763	015	051012	051505
5813	030770	052105	052040	042510
5814	030776	041440	046131	047111
5815	031004	042504	020122	047514
5816	031012	020101	053523	052111
5817	031020	044103	051505	

MSG17C: .ASCIZ <15><12>DISABLE DRIVE 0 0

MSG17D: .ASCIZ <15><12>ENABLE DRIVE 0 0

MSG17E: .ASCIZ <15><12>SET 'READ ONLY' ON DRIVE 0

MSG17F: .ASCIZ <15><12>SET 'READ-WRITE' ON DRIVE 0

MSG17G: .ASCII <15><12>SET 'WRITE LOCKOUT' AND CLEAR THE LOA SWITCH 00

.ASCIZ <15><12>ON THE RP11E CONTROL PANEL0

MSG17H: .ASCIZ <15><12>SET THE FOLLOWING OCTAL VALUES IN THE CYLINDER LOA SWITCHES: 0

MSG17I: .ASCIZ <15><12>SET SWITCHES TO 0

MSG17J: .ASCII <15><12>RESET THE CYLINDER LOA SWITCHES0

5818	031024	005015	051412	052105
5819	031032	052040	042510	043040
5820	031040	046117	047514	044527
5821	031046	043516	047440	052103
5822	031054	046101	053040	046101
5823	031062	042525	020123	047111
5824	031070	052040	042510	042040
5825	031076	044522	042526	046040
5826	031104	040517	051440	044527
5827	031112	041524	042510	035123
5828	031120	000040		
5829	031122	005015	042412	042116
5830	031130	047440	020106	047503
5831	031136	051116	046117	050040
5832	031144	047101	046105	051440
5833	031152	044527	041524	020110
5834	031160	042524	052123	
5835	031164	005015	042522	052524
5836	031172	047122	051040	030520
5837	031200	042461	041440	047117
5838	031206	051124	046117	050040
5839	031214	047101	046105	051440
5840	031222	044527	041524	042510
5841	031230	020123	047524	047040
5842	031236	051117	040515	006514
5843	031244	000012		
5844				
5845	031246	000057		
5846	031250	005015	047105	042524
5847	031256	020122	042524	052123
5848	031264	047040	046525	042502
5849	031272	035122	000040	
5850	031276	047105	042524	020122
5851	031304	051104	053111	020105
5852	031312	052516	041115	051105
5853	031320	020072	000	
5854	031323	104	044522	042526
5855	031330	000040		
5856	031332	044103	047101	042507
5857	031340	050040	051101	046501
5858	031346	052105	051105	020123
5859	031354	047506	020122	042524
5860	031362	052123	030440	020064
5861	031370	020077	000	
5862	031373	115	054101	046511
5863	031400	046525	053440	051117
5864	031406	020104	047503	047125
5865	031414	020124	047506	020122
5866	031422	042524	052123	030440
5867	031430	020064	044450	020116
5868	031436	041517	040524	024514
5869	031444	020072	000	
5870	031447	105	052116	051105
5871	031454	047040	053505	053440
5872	031462	051117	020104	047503
5873	031470	047125	020124	044450

.ASCIZ <15><12><12>@SET THE FOLLOWING OCTAL VALUES IN THE DRIVE LOA SWITCHES: @

MSG17K: .ASCII <15><12><12>@END OF CONROL PANEL SWITCH TEST@

.ASCIZ <15><12>@RETURN RP11E CONTROL PANEL SWITCHES TO NORMAL@<15><12>

SLASH: .ASCIZ @/@
ENTNM: .ASCIZ <15><12>@ENTER TEST NUMBER: @

DRVM: .ASCIZ @ENTER DRIVE NUMBER: @

DRIVE: .ASCIZ @DRIVE @

CHNGPM: .ASCIZ @CHANGE PARAMETERS FOR TEST 14 ? @

MXWRDS: .ASCIZ @MAXIMUM WORD COUNT FOR TEST 14 (IN OCTAL): @

ENTWC: .ASCIZ @ENTER NEW WORD COUNT (IN OCTAL): @

5874 031476 020116 041517 040524
5875 031504 024514 020072 000
5876 031511 105 052116 051105
5877 031516 050040 052101 042524
5878 031524 047122 051440 046105
5879 031532 041505 044524 047117
5880 031540 041440 042117 020105
5881 031546 047506 020122 042524
5882 031554 052123 030440 035064
5883 031562 000040
5884 031564 051525 020105 020101
5885 031572 050123 041505 043111
5886 031600 041511 042040 051511
5887 031606 020113 042101 051104
5888 031614 051505 020123 047506
5889 031622 020122 042524 052123
5890 031630 030440 020064 020077
5891 031636 000
5892 031637 040 040450 051440
5893 031644 047111 046107 020105
5894 031652 040504 040524 050040
5895 031660 052101 042524 047122
5896 031666 046440 051525 020124
5897 031674 040510 042526 041040
5898 031702 042505 020116 050123
5899 031710 041505 043111 042511
5900 031716 006504 012
5901 031721 040 043111 047040
5902 031726 052117 020054 040520
5903 031734 052124 051105 020116
5904 031742 020060 040450 046114
5905 031750 055040 051105 051517
5906 031755 020051 044527 046114
5907 031764 041040 020105 051525
5908 031772 042105 006451 000012
5909 032000 047105 042524 020122
5910 032006 054503 044514 042116
5911 032014 051105 040440 042104
5912 032022 042522 051523 020072
5913 032030 000
5914 032031 105 052116 051105
5915 032036 052040 040522 045503
5916 032044 040440 042104 042522
5917 032052 051523 020072 000
5918 032057 105 052116 051105
5919 032064 051440 041505 047524
5920 032072 020122 042101 051104
5921 032100 051505 035123 000040
5922
5923 032106 050122 042101 020122
5924 032114 020075 000
5925
5926
5927
5928 032117 122 030520 020061
5929 032124 044504 047104 052047

CHNGPT: .ASCIZ 2ENTER PATTERN SELECTION CODE FOR TEST 14: 2

CHNGAD: .ASCIZ 2USE A SPECIFIC DISK ADDRESS FOR TEST 14 ? 2

ONEPAT: .ASCII 2 (A SINGLE DATA PATTERN MUST HAVE BEEN SPECIFIED)2<15><12>

.ASCIZ 2 IF NOT, PATTERN 0 (ALL ZEROS) WILL BE USED)2<15><12>

ENTCYL: .ASCIZ 2ENTER CYLINDER ADDRESS: 2

ENTRK: .ASCIZ 2ENTER TRACK ADDRESS: 2

ENTSEC: .ASCIZ 2ENTER SECTOR ADDRESS: 2

MRPADR: .ASCIZ 2RPADR = 2

.SBTTL ERROR MESSAGES

EMI: .ASCIZ 2RP11 DIDN'T RESPOND TO ADDRESSING2

5930	032132	051040	051505	047520		
5931	032140	042116	052040	020117		
5932	032146	042101	051104	051505		
5933	032154	044523	043516	000		
5934	032161	103	047101	052047	EM2:	.ASCIZ @CAN'T START READ COMMAND@
5935	032166	051440	040524	052122		
5936	032174	051040	040505	020104		
5937	032202	047503	046515	047101		
5938	032210	000104				
5939	032212	041447	042514	051101	EM3:	.ASCIZ @'CLEAR' COMMAND DIDN'T TERMINATE DATA TRANSFER OPERATIONS@
5940	032220	020047	047503	046515		
5941	032226	047101	020104	044504		
5942	032234	047104	052047	052040		
5943	032242	051105	044515	040516		
5944	032250	042524	042040	052101		
5945	032256	020101	051124	047101		
5946	032264	043123	051105	047440		
5947	032272	042520	040522	044524		
5948	032300	047117	000			
5949	032303	105	051122	051117	EM4:	.ASCIZ @ERROR WRITING TEST SECTOR(S)@
5950	032310	053440	044522	044524		
5951	032316	043516	052040	051505		
5952	032324	020124	042523	052103		
5953	032332	051117	051450	000051		
5954	032340	051127	052111	020105	EM5:	.ASCIZ @WRITE CHECK ERROR CHECKING TEST SECTOR(S)@
5955	032346	044103	041505	020113		
5956	032354	051105	047522	020122		
5957	032362	044103	041505	044513		
5958	032370	043516	052040	051505		
5959	032376	020124	042523	052103		
5960	032404	051117	051450	000051		
5961	032412	054105	042520	052103	EM6:	.ASCIZ @EXPECTED 'WCE' DIDN'T OCCUR@
5962	032420	042105	023440	041527		
5963	032426	023505	042040	042111		
5964	032434	023516	020124	041517		
5965	032442	052503	000122			
5966	032446	051105	047522	020122	EM7:	.ASCIZ @ERROR WRITING LESS THAN A FULL SECTOR@
5967	032454	051127	052111	047111		
5968	032462	020107	042514	051523		
5969	032470	052040	040510	020116		
5970	032476	020101	052506	046114		
5971	032504	051440	041505	047524		
5972	032512	000122				
5973	032514	047117	020105	043117	EM11:	.ASCIZ @ONE OF 2 WORDS WRITTEN IN THE PARTIAL WRITE DIDN'T COMPARE@
5974	032522	031040	053440	051117		
5975	032530	051504	053440	044522		
5976	032536	052124	047105	044440		
5977	032544	020116	044124	020105		
5978	032552	040520	052122	040511		
5979	032560	020114	051127	052111		
5980	032566	020105	044504	047104		
5981	032574	052047	041440	046517		
5982	032602	040520	042522	000		
5983	032607	116	047117	055055	EM12:	.ASCII @NON-ZERO DATA IN PART OF SECTOR WHICH SHOULD HAVE@<15><12>
5984	032614	051105	020117	040504		
5985	032622	040524	044440	020116		

5996	032630	040520	052122	047440	
5987	032636	020106	042523	052103	
5988	032644	051117	053440	044510	
5989	032652	044103	051440	047510	
5990	032660	046125	020104	040510	
5991	032666	042526	005015		
5992	032672	042502	047105	055040	.ASCIZ @BEEN ZERO FILLED DURING PARTIAL WRITE@
5993	032700	051105	020117	044506	
5994	032706	046114	042105	042040	
5995	032714	051125	047111	020107	
5996	032722	040520	052122	040511	
5997	032730	020114	051127	052111	
5998	032736	000105			
5999	032740	042447	050117	020047	EM13: .ASCII @'EOP' DIDN'T SET DURING A 2 SECTOR WRITE BEGINNING@
6000	032746	044504	047104	052047	
6001	032754	051440	052105	042040	
6002	032762	051125	047111	020107	
6003	032770	020101	020062	042523	
6004	032776	052103	051117	053440	
6005	033004	044522	042524	041040	
6006	033012	043505	047111	044516	
6007	033020	043516			
6008	033022	005015	052101	052040	.ASCIZ <15><12>@AT THE LAST SECTOR OF THE PACK@
6009	033030	042510	046040	051501	
6010	033036	020124	042523	052103	
6011	033044	051117	047440	020106	
6012	033052	044124	020105	040520	
6013	033060	045503	000		
6014	033063	047	051105	023522	EM14: .ASCIZ @'ERR' DIDN'T SET WITH 'EOP'@
6015	033070	042040	042111	023516	
6016	033076	020124	042523	020124	
6017	033104	044527	044124	023440	
6018	033112	047505	023520	000	
6019	033117	047	047505	023520	EM15: .ASCIZ @'EOP' DIDN'T CLEAR RPCA@
6020	033124	042040	042111	023516	
6021	033132	020124	046103	040505	
6022	033140	020122	050122	040503	
6023	033146	000			
6024	033147	123	041525	020101	EM16: .ASCIZ @SUCA NOT CORRECT AFTER 'EOP'@
6025	033154	047516	020124	047503	
6026	033162	051122	041505	020124	
6027	033170	043101	042524	020122	
6028	033176	042447	050117	000047	
6029	033204	051105	047522	020122	EM17: .ASCIZ @ERROR READING TEST SECTOR@
6030	033212	042522	042101	047111	
6031	033220	020107	042524	052123	
6032	033226	051440	041505	047524	
6033	033234	000122			
6034	033236	047503	052116	047105	EM20: .ASCII @CONTENTS OF FIRST SECTOR ON PACK CHANGED AFTER@<15><12>
6035	033244	051524	047440	020106	
6036	033252	044506	051522	020124	
6037	033260	042523	052103	051117	
6038	033266	047440	020116	040520	
6039	033274	045503	041440	040510	
6040	033302	043516	042105	040440	
6041	033310	052106	051105	005015	

6042	033316	047506	041522	047111		.ASCIZ @FORCING 'EOP' DURING WRIT@
6043	033324	020107	042447	050117		
6044	033332	020047	052504	044522		
6045	033340	043516	053440	044522		
6046	033346	042524	000			
6047	033351	047	051120	043517	EM21:	.ASCII @'PROG' ERROR DIDN'T SET WHEN READ COMMAND ISSUED@
6048	033356	020047	051105	047522		
6049	033364	020122	044504	047104		
6050	033372	052047	051440	052105		
6051	033400	053440	042510	020116		
6052	033406	042522	042101	041440		
6053	033414	046517	040515	042116		
6054	033422	044440	051523	042525		
6055	033430	104				
6056	033431	015	053412	044510		.ASCIZ <15><12>@WHILE CONTROLLER BUSY@
6057	033436	042514	041440	047117		
6058	033444	051124	046117	042514		
6059	033452	020122	052502	054523		
6060	033460	000				
6061	033461	047	051105	023522	EM22:	.ASCIZ @'ERR' DIDN'T SET WITH 'PROG'@
6062	033466	042040	042111	023516		
6063	033474	020124	042523	020124		
6064	033502	044527	04 24	023440		
6065	033510	051120	043517	000047		
6066	033516	051105	047522	020122	EM23:	.ASCIZ @ERROR ATTEMPTING TO MISFORMAT SECTOR @
6067	033524	052101	042524	050115		
6068	033532	044524	043516	052040		
6069	033540	020117	044515	043123		
6070	033546	051117	040515	020124		
6071	033554	042523	052103	051117		
6072	033562	030040	000			
6073	033565	105	051122	051117	EM24:	.ASCIZ @ERROR VERIFYING THE MISFORMATTED TEST HEADER@
6074	033572	053040	051105	043111		
6075	033600	044531	043516	052040		
6076	033606	042510	046440	051511		
6077	033614	047506	046522	052101		
6078	033622	042524	020104	042524		
6079	033630	052123	044040	040505		
6080	033636	042504	000122			
6081	033642	044515	043123	051117	EM25:	.ASCIZ @MISFORMATTED TEST HEADER IS NOT CORRECT@
6082	033650	040515	052124	042105		
6083	033656	052040	051505	020124		
6084	033664	042510	042101	051105		
6085	033672	044440	020123	047516		
6086	033700	020124	047503	051122		
6087	033706	041505	000124			
6088	033712	044047	043116	020047	EM26:	.ASCIZ @'HNF' DIDN'T SET WHEN SEARCHING FOR MISFORMATTED SECTOR@
6089	033720	044504	047104	052047		
6090	033726	051440	052105	053440		
6091	033734	042510	020116	042523		
6092	033742	051101	044103	047111		
6093	033750	020107	047506	020122		
6094	033756	044515	043123	051117		
6095	033764	040515	052124	042105		
6096	033772	051440	041505	047524		
6097	034000	000122				

6098	034002	042047	045523	020047	EM27:	.ASCIZ @'DSK' NOT SET WITH 'HNF'@
6099	034010	047516	020124	042523		
6100	034016	020124	044527	044124		
6101	034024	023440	047110	023506		
6102	034032	000				
6103	034033	047	042510	020047	EM30:	.ASCIZ @'HE' DIDN'T SET WITH 'HNF'@
6104	034040	044504	047104	052047		
6105	034046	051440	052105	053440		
6106	034054	052111	020110	044047		
6107	034062	043116	000047			
6108	034066	042447	051122	020047	EM31:	.ASCIZ @'ERR' NOT SET WITH 'HNF'@
6109	034074	047516	020124	042523		
6110	034102	020124	044527	044124		
6111	034110	023440	047110	023506		
6112	034116	000				
6113	034117	105	051122	051117	EM32:	.ASCIZ @ERROR WHILE RESTORING MISFORMATTED TEST HEADERS@
6114	034124	053440	044510	042514		
6115	034132	051040	051505	047524		
6116	034140	044522	043516	046440		
6117	034146	051511	047506	046522		
6118	034154	052101	042524	020104		
6119	034162	042524	052123	044040		
6120	034170	040505	042504	000122		
6121	034176	047503	052116	047522	EM33:	.ASCII @CONTROLLER DIDN'T BECOME BUSY WHEN READ COMMAND@
6122	034204	046114	051105	042040		
6123	034212	042111	023516	020124		
6124	034220	042502	047503	042515		
6125	034226	041040	051525	020131		
6126	034234	044127	047105	051040		
6127	034242	040505	020104	047503		
6128	034250	046515	047101	104		
6129	034255	015	044412	051523	.ASCIZ	<15><12>@ISSUED TO SEEKING DRIVE@
6130	034262	042525	020104	047524		
6131	034270	051440	042505	044513		
6132	034276	043516	042040	044522		
6133	034304	042526	000			
6134	034307	105	051122	051117	EM34:	.ASCIZ @ERROR DURING READ COMMAND WHICH WAS ISSUED TO SEEKING DRIVE@
6135	034314	042040	051125	047111		
6136	034322	020107	042522	042101		
6137	034330	041440	046517	040515		
6138	034336	042116	053440	044510		
6139	034344	044103	053440	051501		
6140	034352	044440	051523	042525		
6141	034360	020104	047524	051440		
6142	034366	042505	044513	043516		
6143	034374	042040	044522	042526		
6144	034402	000				
6145	034403	104	052101	020101	EM35:	.ASCIZ @DATA INCORRECT FROM HEADER READ COMMAND ISSUED TO A SEEKING DRIVE@
6146	034410	047111	047503	051122		
6147	034416	041505	020124	051106		
6148	034424	046517	044040	040505		
6149	034432	042504	020122	042522		
6150	034440	042101	041440	046517		
6151	034446	040515	042116	044440		
6152	034454	051523	042525	020104		
6153	034462	047524	040440	051440		

6154	034470	042505	044513	043516	
6155	034476	042040	044522	042526	
6156	034504	000			
6157	034505	047	054116	042515	EM36: .ASCIZ @'NXME' DIDN'T SET WHEN LOCATION 760000 ADDRESSED@
6158	034512	020047	044504	047104	
6159	034520	052047	051440	052105	
6160	034526	053440	042510	020116	
6161	034534	047514	040503	044524	
6162	034542	047117	033440	030066	
6163	034550	030060	020060	042101	
6164	034556	051104	051505	042523	
6165	034564	000104			
6166	034566	044047	023505	042040	EM37: .ASCIZ @'HE' DIDN'T SET WITH 'NXME'@
6167	034574	042111	023516	020124	
6168	034602	042523	020124	044527	
6169	034610	044124	023440	054116	
6170	034616	042515	000047		
6171	034622	042447	051122	020047	EM40: .ASCIZ @'ERR' DIDN'T SET WITH 'NXME'@
6172	034630	044504	047104	052067	
6173	034636	051440	052105	053440	
6174	034644	052111	020110	047047	
6175	034652	046530	023505	000	
6176	034657	105	051122	051117	EM41: .ASCIZ @ERROR FORMATTING TRACK @
6177	034664	043040	051117	040515	
6178	034672	052124	047111	020107	
6179	034700	051124	041501	020113	
6180	034706	030060			
6181	034710	051105	047522	020122	EM42: .ASCIZ @ERROR READING THE HEADER FROM THE TEST SECTOR@
6182	034716	042522	042101	047111	
6183	034724	020107	044124	020105	
6184	034732	042510	042101	051105	
6185	034740	043040	047522	020115	
6186	034746	044124	020105	042524	
6187	034754	052123	051440	041505	
6188	034762	047524	000122		
6189	034766	042523	052103	051117	EM43: .ASCIZ @SECTOR FIELD FROM HEADER IS NOT CORRECT@
6190	034774	043040	042511	042114	
6191	035002	043040	047522	020115	
6192	035010	042510	042101	051105	
6193	035016	044440	020123	047516	
6194	035024	020124	047503	051122	
6195	035032	041505	000124		
6196	035036	051105	047522	020122	EM44: .ASCIZ @ERROR WRITING SECTOR ADDRESS IN DATA FIELD@
6197	035044	051127	052111	047111	
6198	035052	020107	042523	052103	
6199	035060	051117	040440	042104	
6200	035066	042522	051523	044440	
6201	035074	020116	040504	040524	
6202	035102	043040	042511	042114	
6203	035110	000			
6204	035111	047	047523	023524	EM45: .ASCIZ @'SOT' OR SECTOR ADDRESS REGISTER IS NOT CORRECT@
6205	035116	047440	020122	042523	
6206	035124	052103	051117	040440	
6207	035132	042104	042522	051523	
6208	035140	051040	044505	051507	
6209	035146	042524	020122	051511	

6210	035154	047040	052117	041440	
6211	035162	051117	042522	052103	
6212	035170	000			
6213	035171	104	052101	020101	EM46: .ASCIZ @DATA READ FROM THE SECTOR IS NOT CORRECT@
6214	035176	042522	042101	043040	
6215	035204	047522	020115	044124	
6216	035212	020105	042523	052103	
6217	035220	051117	044440	020123	
6218	035226	047516	020124	047503	
6219	035234	051122	041505	000124	
6220	035242	042523	052103	051117	EM47: .ASCII @SECTOR CONTENTS WRONG, DATA SHOULD BE THE@<15><12>
6221	035250	041440	047117	042524	
6222	035256	052116	020123	051127	
6223	035264	047117	026107	042040	
6224	035272	052101	020101	044123	
6225	035300	052517	042114	041040	
6226	035306	020105	044124	006505	
6227	035314	012			
6228	035315	124	040522	045503	.ASCIZ @TRACK NUMBER OF THE CURRENTLY SELECTED HEAD@
6229	035322	047040	046525	042502	
6230	035330	020122	043117	052040	
6231	035336	042510	041440	051125	
6232	035344	042522	052116	054514	
6233	035352	051440	046105	041505	
6234	035360	042524	020104	042510	
6235	035366	042101	000		
6236	035371	105	051122	051117	EM50: .ASCIZ @ERROR AFTER 2 WORD READ STARTING AT LOC 177776@
6237	035376	040440	052106	051105	
6238	035404	031040	053440	051117	
6239	035412	020104	042522	042101	
6240	035420	051440	040524	052122	
6241	035425	047111	020107	052101	
6242	035434	046040	041517	030440	
6243	035442	033467	033467	000066	
6244	035450	046447	054105	023460	EM51: .ASCII @'MEX0' DIDN'T SET AFTER 2 WORD READ STARTING@<15><12>
6245	035456	042040	042111	023516	
6246	035464	020124	042523	020124	
6247	035472	043101	042524	020122	
6248	035500	020062	047527	042122	
6249	035506	051040	040505	020104	
6250	035514	052123	051101	044524	
6251	035522	043516	005015		
6252	035526	052101	046040	041517	.ASCIZ @AT LOC 177776@
6253	035534	030440	033467	033467	
6254	035542	000066			
6255	035544	046447	054105	023461	EM52: .ASCIZ @'MEX1' SET AFTER 2 WORD READ STARTING AT LOC 177776@
6256	035552	051440	052105	040440	
6257	035560	052106	051105	031040	
6258	035566	053440	051117	020104	
6259	035574	042522	042101	051440	
6260	035602	040524	052122	047111	
6261	035610	020107	052101	046040	
6262	035616	041517	030440	033467	
6263	035624	033467	000066		
6264	035630	047503	052116	047105	EM53: .ASCII @CONTENTS OF LOC 200000 WRONG AFTER 2 WORD READ@<15><12>
6265	035636	051524	047440	020106	

6266	035644	047514	020103	030062
6267	035652	030060	030060	053440
6268	035660	047522	043516	040440
6269	035666	052106	051105	031040
6270	035674	053440	051117	020104
6271	035702	042522	042101	005015
6272	035710	052123	051101	044524
6273	035716	043516	040440	020124
6274	035724	047514	020103	033461
6275	035732	033467	033067	000
6276	035737	105	051122	051117
6277	035744	040440	052106	051105
6278	035752	031040	053440	051117
6279	035760	020104	042522	042101
6280	035766	051440	040524	052122
6281	035774	047111	020107	052101
6282	036002	046040	041517	031440
6283	036010	033467	033467	000066
6284	036016	046447	054105	023460
6285	036024	042040	042111	023516
6286	036032	020124	046103	040505
6287	036040	020122	043101	042524
6288	036046	020122	020062	047527
6289	036054	042122	051440	040524
6290	036062	052122	047111	020107
6291	036070	052101	046040	041517
6292	036076	031440	033467	033467
6293	036104	000066		
6294	036106	046447	054105	023461
6295	036114	042040	042111	023516
6296	036122	020124	042524	020124
6297	036130	043101	042524	020122
6298	036136	020062	047527	042122
6299	036144	051040	040505	020104
6300	036152	052123	051101	044524
6301	036160	043516	040440	020124
6302	036166	047514	020103	033463
6303	036174	033467	033067	000
6304	036201	103	047117	042524
6305	036206	052116	020123	043117
6306	036214	046040	041517	032040
6307	036222	030060	030060	020060
6308	036230	051127	047117	020107
6309	036236	043101	042524	020122
6310	036244	020062	047527	042122
6311	036252	051040	040505	006504
6312	036260	012		
6313	036261	123	040524	052122
6314	036256	047111	020107	052101
6315	036274	046040	041517	052101
6316	036302	047511	020116	033463
6317	036310	033467	033067	000
6318	036315	105	051122	051117
6319	036322	051040	052105	051125
6320	036330	044516	043516	051440
6321	036336	041505	047524	020122

.ASCIZ @STARTING AT LOC 1777763

EM54: .ASCIZ @ERROR AFTER 2 WORD READ STARTING AT LOC 3777763

EM55: .ASCIZ @'MEXO' DIDN'T CLEAR AFTER 2 WORD STARTING AT LOC 3777763

EM56: .ASCIZ @'MEXI' DIDN'T SET AFTER 2 WORD READ STARTING AT LOC 3777763

EM57: .ASCII @CONTENTS OF LOC 40000 WRONG AFTER 2 WORD READ@15><12>

EM60: .ASCII @ERROR RETURNING SECTOR TO PDP-11 MODE USING @2<15><12>

6322	036344	047524	050040	050104
6323	036352	030455	020061	047515
6324	036360	042504	052440	044523
6325	036366	043516	040440	005015
6326	036374	020062	047527	042122
6327	036402	053440	044522	042524
6328	036410	000		
6329	036411	105		
6330	036416	053440	051123	051117
6331	036424	043516	044523	044524
6332	036432	020124	052040	051505
6333	036440	051117	042523	002103
6334	036446	051105	047522	000051
6335	036454	042522	042101	020122
6336	036462	020107	042524	052123
6337	036470	051440	041505	047524
6338	036476	04122	024523	000
6339	036503	104	044522	042526
6340	036510	042040	042111	023516
6341	036516	020124	042522	052524
6342	036524	047122	052040	020117
6343	036532	054503	020114	020060
6344	036540	051106	046517	052040
6345	036546	042510	046440	054101
6346	036554	046511	046523	005015
6347	036562	054503	020114	047117
6348	036570	041440	047117	051124
6349	036576	046117	042514	020123
6350	036604	047520	042527	020123
6351	036612	040506	046111	000
6352	036617	103	047117	042524
6353	036624	052116	020123	043117
6354	036632	046440	046503	051117
6355	036640	020131	044103	047101
6356	036646	042507	020104	052504
6357	036654	044522	043516	050040
6358	036662	053517	051106	005015
6359	036670	040506	046111	053440
6360	036676	044510	042514	051040
6361	036704	040505	044504	043516
6362	036712	052040	042510	042040
6363	036720	051511	000113	
6364	036724	051447	047525	023514
6365	036732	051440	052105	053440
6366	036740	052111	020110	051104
6367	036746	053111	020105	044504
6368	036754	040523	046102	042105
6369	036762	000		
6370	036763	047	052523	046117
6371	036770	020047	047516	020124
6372	036776	042523	020124	044527
6373	037004	044124	042040	044523
6374	037012	042526	042440	040516
6375	037020	046102	042105	000
6376	037025	047	052523	050127
6377	037032	020047	047516	020124

.ASCIZ 22 WORD WRITED

EM61: .ASCIZ 2ERROR WRITING TEST SECTOR(S)2

EM62: .ASCIZ 2ERROR READING TEST SECTOR(S)2

EM63: .ASCII 2ORIVE DIDN'T RETURN TO CYL 0 FROM THE MAXIMUM2(15)<(12)

.ASCIZ 2CYL ON CONTROLLER POWER FAIL2

EM64: .ASCII 2CONTENTS OF MEMORY CHANGED DURING POWER2(15)<(12)

.ASCIZ 2FAIL WHILE READING THE DISK2

EM65: .ASCIZ 2'SUOL' SET WITH DRIVE DISABLED2

EM66: .ASCIZ 2'SUOL' NOT SET WITH DRIVE ENABLED2

EM67: .ASCIZ 2'SUMP' NOT SET WITH DRIVE SET TO 'READ ONLY'2

6378	037040	042523	020124	044527
6379	037045	044124	042040	044522
6380	027054	042526	051440	052105
6381	037062	052105	020117	051047
6382	037070	044525	020104	047117
6383	037076	054514	000047	
6384	037102	051447	053525	023520
6385	037110	051440	052105	053440
6386	037116	052111	020110	051104
6387	037124	053111	020105	042523
6388	037132	020124	047524	023440
6389	037140	042522	042101	053457
6390	037146	044523	042524	000047
6391	037154	051447	053525	023520
6392	037162	051440	052105	053440
6393	037170	052111	020110	050122
6394	037176	030461	023440	051127
6395	037204	052111	020105	047514
6396	037212	045503	052517	023524
6397	037220	051440	052105	020054
6398	037226	054503	044514	042116
6399	037234	051105	005015	
6400	037240	047514	023501	020123
6401	037246	020075	026060	040440
6402	037254	042116	051040	041520
6403	037262	020101	020075	000060
6404	037270	051447	053525	023520
6405	037276	051440	052105	053440
6406	037304	052111	020110	050122
6407	037312	040503	036440	031040
6408	037320	023040	041440	046131
6409	037326	046040	040517	051447
6410	037334	036440	030040	000
6411	037341	047	052523	050127
6412	037346	020047	047516	020124
6413	037354	042523	035524	051040
6414	037362	041520	020101	020075
6415	037370	040526	052514	020105
6416	037376	047111	041440	046131
6417	037404	046040	040517	051447
6418	037412	000		
6419	037413	047	052523	050127
6420	037420	020047	042523	020124
6421	037426	044527	044124	051040
6422	037434	041520	020101	047117
6423	037442	020105	051107	040505
6424	037450	042524	020122	044124
6425	037456	047101	041440	046131
6426	037464	046040	040517	053040
6427	037472	046101	042525	000
6428	037477	047	052523	050127
6429	037504	020047	042523	020124
6430	037512	044527	044124	042040
6431	037520	044522	042526	046040
6432	037526	040517	051440	044527
6433	037534	041524	042510	020123

EM70: .ASCIZ @'SUMP' SET WITH DRIVE SET TO 'READ/WRITE'@

EM71: .ASCII @'SUMP' SET WITH RP11 'WRITE LOCKOUT' SET, CYLINDER@<15><12>

.ASCIZ @LOA'S = 0, AND RPCA = 0@

EM72: .ASCIZ @'SUMP' SET WITH RPCA = 2 & CYL LOA'S = 0@

EM73: .ASCIZ @'SUMP' NOT SET; RPCA = VALUE IN CYL LOA'S@

EM74: .ASCIZ @'SUMP' SET WITH RPCA ONE GREATER THAN CYL LOA VALUE@

EM75: .ASCIZ @'SUMP' SET WITH DRIVE LOA SWITCHES EQUAL TO SELECTED DRIVE@

6434	037542	050505	040525	020114	
6435	037550	047524	051440	046105	
6436	037556	041505	042524	020104	
6437	037564	051104	053111	000105	
6438	037572	051104	053111	020105	EM76: .ASCIZ @DRIVE HAS GONE OFFLINE@
6439	037600	040510	020123	047507	
6440	037606	042516	047440	043106	
6441	037614	044514	042516	000	
6442	037621	104	044522	042526	EM77: .ASCIZ @DRIVE IS UNSAFE@
6443	037626	044440	020123	047123	
6444	037634	040523	042506	000	
6445	037641	104	044522	042526	EM100: .ASCIZ @DRIVE TIMED OUT@
6446	037646	052040	046511	042105	
6447	037654	047440	052125	000	
6448	037661	103	047117	051124	EM101: .ASCIZ @CONTROLLER TIMED OUT@
6449	037666	046117	042514	020122	
6450	037674	044524	042515	020104	
6451	037702	052517	000124		
6452	037706	040504	040524	041440	EM102: .ASCIZ @DATA COMPARE ERROR@
6453	037714	046517	040520	042522	
6454	037722	042440	051122	051117	
6455	037730	000			
6456	037731	104	052101	020101	EM103: .ASCIZ @DATA COMPARE ERROR (MEMORY MANAGEMENT ENABLED)@
6457	037736	047503	050115	051101	
6458	037744	020105	051105	047522	
6459	037752	020122	046450	046505	
6460	037760	051117	020131	040515	
6461	037766	040516	042507	042515	
6462	037774	052116	042440	040516	
6463	040002	046102	042105	000051	
6464	040010	042447	051122	020047	EM106: .ASCIZ @'ERR' DIDN'T SET WITH 'WCE'@
6465	040016	044504	047104	052047	
6466	040024	051440	052105	053440	
6467	040032	052111	020110	053447	
6468	040040	042503	000047		
6469	040044	044047	023505	042040	EM107: .ASCIZ @'HE' DIDN'T SET WITH 'PROG'@
6470	040052	042111	023516	020124	
6471	040050	042523	020124	044527	
6472	040056	044124	023440	051120	
6473	040074	043517	000047		
6474	040100	051104	053111	020105	EM112: .ASCIZ @DRIVE UNSAFE AFTER POWER FAIL@
6475	040106	047125	040523	042506	
6476	040114	040440	052106	051105	
6477	040122	050040	053517	051105	
6478	040130	043040	044501	000114	
6479	040136	051447	053525	023520	EM113: .ASCIZ @'SUMP' NOT SET WITH DRIVE LOR SWITCHES GREATER THAN SELECTED DRIVE@
6480	040144	047040	052117	051440	
6481	040152	052105	053440	052111	
6482	040160	020110	051104	053111	
6483	040166	020105	047514	020101	
6484	040174	053523	052111	044103	
6485	040202	051505	043440	042522	
6486	040210	052101	051105	052040	
6487	040216	040510	020116	040523	
6488	040224	042514	042103	042105	
6489	040232	042040	044522	042526	

6490 040240 000
6491
6492
6493
6494 040241 122 040520 051104
6495 040246 000
6496 040247 124 051505 020124
6497 040254 020043 042440 051122
6498 040252 050040 020103 042040
6499 040270 053122 021440 020040
6500 040276 051040 042120 020123
6501 040304 020040 051040 042520
6502 040312 020122 020040 051040
6503 040320 041520 020123 020040
6504 040326 051040 053520 000103
6505 040334 050122 040502 020040
6506 040342 020040 050122 040503
6507 040350 020040 020040 050122
6508 040356 040504 020040 020040
6509 040364 050122 030515 020040
6510 040372 020040 052523 040503
6511 040400 020040 020040 044523
6512 040406 047514 000
6513 040411 124 051505 020124
6514 040416 020043 042440 051122
6515 040424 050040 020103 042040
6516 040432 053122 021440 020040
6517 040440 050040 052101 047122
6518 040446 021440 051040 042120
6519 040454 020123 020040 051040
6520 040462 042520 020122 020040
6521 040470 051040 041520 000123
6522 040476 050122 041527 020040
6523 040504 020040 050122 040502
6524 040512 020040 020040 050122
6525 040520 040503 020040 020040
6526 040526 050122 040504 020040
6527 040534 020040 050122 030515
6528 040542 020040 020040 052523
6529 040550 040503 020040 020040
6530 040556 044523 047514 000
6531 040563 124 051505 020124
6532 040570 020043 042440 051122
6533 040576 050040 020103 042040
6534 040604 053122 021440 020040
6535 040612 042440 050130 023524
6536 040620 020104 051461 020124
6537 040626 047527 042122 031040
6538 040634 042116 053440 051117
6539 040642 000104
6540 040644 042524 052123 021440
6541 040652 020040 051105 020122
6542 040660 041520 020040 051104
6543 040666 020126 020043 020040
6544 040674 051105 052120 042047
6545 040702 020040 042522 053103

;'DH' MESSAGE LINES

DH1: .ASCIZ @RPAORA
DH2: .ASCIZ @TEST @ ERR PC DRV @ RPDS RPER RPCS RPWC@
DH2A: .ASCIZ @RPBA RPCA RPOA RPM1 SUCA SILO@
DH5: .ASCIZ @TEST @ ERR PC DRV @ PATRN @ RPDS RPER RPCS@
DH5A: .ASCIZ @RPWC RPBA RPCA RPOA RPM1 SUCA SILO@
DH11: .ASCIZ @TEST @ ERR PC DRV @ EXPT'D 1ST WORD 2ND WORD@
DH12: .ASCIZ @TEST @ ERR PC DRV @ EXPT'D RECV'D LOC@

6602	041400	051040	042120	000101																	
6603	041406	042510	042101	051105	DM43B:	.ASCIZ	2	HEADER	CONTENTS												
6604	041414	041440	047117	042524																	
6605	041422	052116	000123		DM45:	.ASCII	2					EXPT'D	'SOT'		SECTORA	(15)	(12)				
6606	041426	020040	020040	020040																	
6607	041434	020040	020040	020040																	
6608	041442	020040	020040	020040																	
6609	041450	020040	020040	020040																	
6610	041456	054105	052120	042047																	
6611	041464	020040	051447	052117																	
6612	041472	020047	020040	042523																	
6613	041500	052103	051117	005015																	
6614	041506	042524	052123	021440		.ASCIZ	2	TEST	ERR PC	DRV	VALUE	COUNTER	ADDR								
6615	041514	020040	051105	020122																	
6616	041522	041520	020040	051104																	
6617	041530	020126	020043	020040																	
6618	041536	040526	052514	020105																	
6619	041544	041440	052517	052116																	
6620	041552	051105	020040	042101																	
6621	041558	051104	000																		
6622	041563	122	042120	020123	DM45A:	.ASCIZ	2	RPDS	RPER	RPCS	RPDA										
6623	041570	020040	051040	042520																	
6624	041576	020122	020040	051040																	
6625	041604	041520	020123	020040																	
6626	041612	051040	042120	000101	DM46:	.ASCII	2					EXPT'D	RECV'D	(15)	(12)						
6627	041620	020040	020040	020040																	
6628	041626	020040	020040	020040																	
6629	041634	020040	020040	020040																	
6630	041642	020040	020040	020040																	
6631	041650	020040	020040	020040																	
6632	041656	020040	054105	052120																	
6633	041664	042047	020040	042522																	
6634	041672	053103	042047	005015																	
6635	041700	042524	052123	021440		.ASCIZ	2	TEST	ERR PC	DRV	SECTOR	DATA	DATA								
6636	041706	020040	051105	020122																	
6637	041714	041520	020040	051104																	
6638	041722	020126	020043	020040																	
6639	041730	042523	052103	051117																	
6640	041736	020040	040504	040524																	
6641	041744	020040	020040	040504																	
6642	041752	040524	000																		
6643	041755	124	051505	020124	DM47:	.ASCIZ	2	TEST	ERR PC	DRV	EXPT'D	RECV'D									
6644	041762	020043	042440	051122																	
6645	041770	050040	020103	042040																	
6646	041776	053122	021440	020040																	
6647	042004	042440	050130	023524																	
6648	042012	020104	051040	041505																	
6649	042020	023526	060104																		
6650	042024	050122	051504	020040	DM47A:	.ASCIZ	2	RPDS	RPER	RPCS	RPCA	RPDA									
6651	042032	020040	050122	051105																	
6652	042040	020040	020040	050122																	
6653	042046	051503	020040	020040																	
6654	042054	050122	040503	020040																	
6655	042062	020040	050122	040504																	
6656	042070	000																			
6657	042071	124	051505	020124	DM51:	.ASCIZ	2	TEST	ERR PC	DRV	RPDS	RPER	RPCS	RPWC	RPBA						

6658	042076	020043	042440	051122
6659	042104	050040	020103	042040
6660	042112	053122	021440	020040
6661	042120	051040	042120	020123
6662	042128	020040	051040	042520
6663	042134	020122	020040	051040
6664	042142	041520	020123	020040
6665	042150	051040	053520	020103
6666	042156	020040	051040	041120
6667	042164	000101		
6668	042166	042524	052123	021440
6669	042174	020040	051105	020122
6670	042202	041520	020040	051104
6671	042210	020126	020043	020040
6672	042216	050122	051504	020040
6673	042224	020040	050122	051105
6674	042232	020040	020040	050122
6675	042240	051503	020040	020040
6676	042246	052523	040503	000
6677	042253	124	051505	020124
6678	042260	020043	042440	051122
6679	042266	050040	020103	042040
6680	042274	053122	021440	020040
6681	042302	051040	042120	020123
6682	042310	020040	051040	042520
6683	042316	020122	020040	051040
6684	042324	041520	020123	020040
6685	042332	051040	041520	000101
6686	042340	042524	052123	021440
6687	042348	020040	051105	020122
6688	042356	041520	020040	051104
6689	042364	020123	020043	051440
6690	042372	041506	051440	051124
6691	042378	020124	054503	020114
6692	042404	020040	020040	051124
6693	042412	020113	020040	020040
6694	042420	042523	020103	020040
6695	042428	020040	040520	052124
6696	042434	051105	000116	
6697	042440	020040	020040	020040
6698	042446	020040	047507	042117
6699	042454	020040	020040	040502
6700	042462	006504	012	
6701	042465	101	042104	020122
6702	042472	020040	042040	052101
6703	042500	020101	020040	042040
6704	042506	052101	000101	
6705	042512	044513	040520	031522
6706	042520	020040	044513	040520
6707	042526	032122	020040	044513
6708	042534	040520	032522	020040
6709	042542	044513	042120	031522
6710	042550	020040	044513	042120
6711	042556	032122	020040	044513
6712	042564	042120	032522	000
6713	042571	126	051111	052524

DH63: .ASCIZ @TEST # ERR PC DRV # RPDS RPER RPCS SUCA@

DH71: .ASCIZ @TEST # ERR PC DRV # RPDS RPER RPCS RPCA@

DH102: .ASCIZ @TEST # ERR PC DRV # SEC STRT CYL TRK SEC PATTERN@

DH102A: .ASCII @ GOOD BAD@<15><12>

.ASCIZ @DOOR DATA DATA@

DH103: .ASCIZ @KIPAR3 KIPAR4 KIPARS KIPDR3 KIPDR4 KIPDR5@

DH103A: .ASCII @VIRTUAL GOOD BAD@<15><12>

6714	042576	046101	043440	047517					
6715	042604	020104	020040	041040					
6716	042612	042101	005015						
6717	042616	042101	051104	020040	.ASCIZ	QADDR	DATA	DATAQ	
6718	042624	020040	040504	040524					
6719	042632	020040	020040	040504					
6720	042640	040524	000						
6721	042643	124	052117	046101	DM105:	.ASCIZ	QTOTAL COMPARE ERRORS:Q		
6722	042650	041440	046517	040520					
6723	042656	042522	042440	051122					
6724	042664	051117	035123	000					
6725	042671	125	051516	042503	DM110:	.ASCIZ	QUNSUCCESSFUL AFTER 3 RETRIESQ		
6726	042676	051523	052506	020114					
6727	042704	043101	042524	020122					
6728	042712	042063	042522	051124					
6729	042720	042511	000123						
6730	042724	042522	051124	020131	DM111:	.ASCII	QRETRY SUCCESSFULQ<15><12>		
6731	042732	052523	042503	051523					
6732	042740	052506	006514	012					
6733	042745	122	052105	054522	.ASCIZ	QRETRY COUNT:Q			
6734	042752	041440	052517	052116					
6735	042760	000072							
6736									
6737					.EVEN				
6738									
6739					; 'DT' WORDS				
6740									
6741	042762	001324			DT1:	.WORD	RPADR		
6742	042764	001160	001116	001176	DT2:	.WORD	STMPD, SERRPC, CHKDRV, SRPDS, SRPER, SRPCS, SRPWC		
6743	042772	001334	001336	001340					
6744	043000	001342							
6745	043002	001344	001346	001350	.WORD	SRPBA, SRPCA, SRPDA, SRPM1, SSUCA, SSILO			
6746	043010	001352	001354	001356					
6747	043016	001160	001116	001176	DT5:	.WORD	STMPD, SERRPC, CHKDRV, PATNUM, SRPDS, SRPER, SRPCS		
6748	043024	001204	001334	001336					
6749	043032	001340							
6750	043034	001342	001344	001346	.WORD	SRPWC, SRPBA, SRPCA, SRPDA, SRPM1, SSUCA, SSILO			
6751	043042	001350	001352	001354					
6752	043050	001356							
6753	043052	001160	001116	001176	DT11:	.WORD	STMPD, SERRPC, CHKDRV, SGDDAT, BUFFER, BUFFER+2		
6754	043060	001124	043602	043604					
6755	043066	001160	001116	001176	DT12:	.WORD	STMPD, SERRPC, CHKDRV, SGDDAT, SBDDAT, SBDDADR		
6756	043074	001124	001126	001122					
6757	043102	001160	001116	001176	DT14:	.WORD	STMPD, SERRPC, CHKDRV, SRPDS, SRPER, SRPCS		
6758	043110	001334	001336	001340					
6759	043116	001160	001116	001176	DT16:	.WORD	STMPD, SERRPC, CHKDRV, MAXCYL, SSUCA		
6760	043124	001210	001354						
6761	043130	001160	001116	001176	DT25:	.WORD	STMPD, SERRPC, CHKDRV, SGDDAT, SBDDAT		
6762	043136	001124	001126						
6763	043142	043602	043604	043606	.WORD	BUFFER, BUFFER+2, BUFFER+4			
6764	043150	001160	001116	001176	DT43:	.WORD	STMPD, SERRPC, CHKDRV, SGDDAT, BUFFER+4		
6765	043156	001124	043606						
6766	043162	001334	001336	001340	.WORD	SRPDS, SRPER, SRPCS, SRPWC, SRPBA, SRPCA, SRPDA			
6767	043170	001342	001344	001346					
6768	043176	001350							
6769	043200	043602	043604	043606	.WORD	BUFFER, BUFFER+2, BUFFER+4			

6770	043206	001160	001116	001176	DT45:	.WORD	STMPD, SERRPC, CHKDRV, SGDDAT, SBDDAT, SBDADR
6771	043214	001124	001126	001122			
6772	043222	001334	001336	001340		.WORD	SRPDS, SRPER, SRPCS, SRPDA
6773	043230	001350					
6774	043232	001160	001116	001176	DT46:	.WORD	STMPD, SERRPC, CHKDRV, SGDAOR, SGDDAT, SBDDAT
6775	043240	001120	001124	001126			
6776	043246	001160	001116	001176	DT47:	.WORD	STMPD, SERRPC, CHKDRV, SGDDAT, BUFFER
6777	043254	001124	043602				
6778	043260	001334	001336	001340		.WORD	SRPDS, SRPER, SRPCS, SRPCA, SRPDA
6779	043266	001346	001350				
6780	043272	001160	001116	001176	DT51:	.WORD	STMPD, SERRPC, CHKDRV, SRPDS, SRPER, SRPCS, SRPWC, SRPBA
6781	043300	001334	001336	001340			
6782	043306	001342	001344				
6783	043312	001160	001116	001176	DT63:	.WORD	STMPD, SERRPC, CHKDRV, SRPDS, SRPER, SRPCS, \$SUCA
6784	043320	001334	001336	001340			
6785	043326	001354					
6786	043330	001160	001116	001176	DT71:	.WORD	STMPD, SERRPC, CHKDRV, SRPDS, SRPER, SRPCS, SRPCA
6787	043336	001334	001336	001340			
6788	043344	001346					
6789	043346	001160	001116	001176	DT102:	.WORD	STMPD, SERRPC, CHKDRV, CMBUF, CMCYL, CMTRK, CMSEC, PATNUM
6790	043354	025362	025354	025360			
6791	043362	025356	001204				
6792	043366	001122	001124	001126		.WORD	SBDADR, SGDDAT, SBDDAT
6793	043374	001160	001116	001176	DT103:	.WORD	STMPD, SERRPC, CHKDRV, CMBUF, CMCYL, CMTRK, CMSEC, PATNUM
6794	043402	025362	025354	025360			
6795	043410	025356	001204				
6796	043414	172346	172350	172352		.WORD	KIPAR3, KIPAR4, KIPARS, KIPDR3, KIPDR4, KIFDRS
6797	043422	172306	172310	172312			
6798	043430	001122	001124	001126	DT104:	.WORD	SBDADR, SGDDAT, SBDDAT
6799	043436	025346			DT105:	.WORD	ERCTR
6800	043440	001314			DT111:	.WORD	RETRY
6801							
6802							
6803							
6804	043442	000001			DF1:	.WORD	1
6805	043444	001				.BYTE	1
6806	043445	000				.BYTE	0
6807							
6808	043446	000002			DF2:	.WORD	2
6809	043450	007				.BYTE	7
6810	043451	000				.BYTE	0
6811	043452	040334				.WORD	DH2A
6812	043454	006				.BYTE	6
6813	043455	000				.BYTE	0
6814							
6815	043456	000002			DF5:	.WORD	2
6816	043460	007				.BYTE	7
6817	043461	010				.BYTE	10
6818	043462	040476				.WORD	DH5A
6819	043464	007				.BYTE	7
6820	043465	000				.BYTE	0
6821							
6822	043466	000001			DF11:	.WORD	1
6823	043470	006				.BYTE	6
6824	043471	000				.BYTE	0
6825							

6826	043472	000002	DF25:	.WORD	2
6827	043474	005		.BYTE	0
6828	043475	000		.BYTE	0
6829	043476	041162		.WORD	0H25A
6830	043500	003		.BYTE	0
6831	043501	000		.BYTE	0
6832					
6833	043502	000003	DF43:	.WORD	3
6834	043504	006		.BYTE	6
6835	043505	000		.BYTE	0
6836	043506	041321		.WORD	0H43A
6837	043510	005		.BYTE	0
6838	043511	000		.BYTE	0
6839	043512	041406		.WORD	0H43B
6840	043514	003		.BYTE	3
6841	043515	000		.BYTE	0
6842					
6843	043516	000002	DF45:	.WORD	2
6844	043520	006		.BYTE	6
6845	043521	000		.BYTE	0
6846	043522	041563		.WORD	0H45A
6847	043524	004		.BYTE	4
6848	043525	000		.BYTE	0
6849					
6850	043526	000002	DF47:	.WORD	2
6851	043530	005		.BYTE	5
6852	043531	000		.BYTE	0
6853	043532	042024		.WORD	0H47A
6854	043534	005		.BYTE	5
6855	043535	000		.BYTE	0
6856					
6857	043536	000001	DF51:	.WORD	1
6858	043540	010		.BYTE	8
6859	043541	000		.BYTE	0
6860					
6861	043542	000001	DF53:	.WORD	1
6862	043544	005		.BYTE	5
6863	043545	000		.BYTE	0
6864					
6865	043546	000001	DF63:	.WORD	1
6866	043550	007		.BYTE	7
6867	043551	000		.BYTE	0
6868					
6869	043552	000002	DF102:	.WORD	2
6870	043554	010		.BYTE	8
6871	043555	200		.BYTE	200
6872	043556	042440		.WORD	0H102A
6873	043560	000		.BYTE	0
6874	043561	000		.BYTE	0
6875					
6876	043562	000003	DF103:	.WORD	3
6877	043564	010		.BYTE	8
6878	043565	200		.BYTE	200
6879	043566	042512		.WORD	0H103
6880	043570	006		.BYTE	6
6881	043571	000		.BYTE	0

M10

MD-11-DZRPY-B, RP11-E FUNCTIONAL LOGIC & R/W TEST
DZRPYB.CMB ERROR MESSAGES

MACY11 27(732) 02-NOV-76 14:50 PAGE 130

6882	043572	042571			.WORD	DH103A
6883	043574	000			.BYTE	0
6884	043575	000			.BYTE	0
6885						
6886	043576	000001				
6887	043600	003	DF104:		.WORD	1
6888	043601	000			.BYTE	3
6889					.BYTE	0
6890						
6891					.EVEN	
6892						
6893		043602			BUFFER=.	;BUFFER STARTS HERE
6894						
6895	043602	005015	046412	044501	TITLE: .ASCII	<15><12><12>@MAINDEC-11-DZRPY-B@
6896	043610	042116	041505	030455		
6897	043616	026461	055104	050122		
6898	043624	026531	102			
6899	043627	015	051012	030520	.ASCIIZ	<15><12>@RP11-E FUNCTIONAL LOGIC & READ/WRITE TEST@<15><12><12>
6900	043634	026461	020105	052506		
6901	043642	041516	044524	047117		
6902	043650	046101	046040	043517		
6903	043656	041511	023040	051040		
6904	043664	040505	027504	051127		
6905	043672	052111	020105	042524		
6906	043700	052123	005015	000012		
6907		000001			.END	

ACTMEM	001224	1236#	3491*	3492*	5174	5175	5184*	5185*	5186	5189	5191*	5192*		
AIE =	020000	1175#												
ATABIT	001360	1289#	2307	2328	3744	4868	4905	5052	5475	5493				
BITS	001400	1315#	2398	2445	2524	2613	2697	2747	2842	2903	2970	3088	3164	3294
		3463	3608	3715	3767	5068	5454							
BIT0 =	000001	1080#												
BIT00 =	000001	1070#	1080	1315	2808									
BIT01 =	000002	1069#	1079	1316	2654									
BIT02 =	000004	1068#	1078	1317	2927									
BIT03 =	000010	1067#	1077	1318	2499	3928								
BIT04 =	000020	1066#	1076	1319										
BIT05 =	000040	1065#	1075	1320										
BIT06 =	000100	1064#	1074	1321										
BIT07 =	000200	1063#	1073	1322	5360									
BIT08 =	000400	1062#	1072	1323										
BIT09 =	001000	1061#	1071	1324	3899	4046	4636							
BIT1 =	000002	1079#												
BIT10 =	002000	1060#	1325	2722	4030									
BIT11 =	004000	1059#	1326	4643										
BIT12 =	010000	1058#	1327	2804										
BIT13 =	020000	1057#	1328	4037										
BIT14 =	040000	1056#	1329	4618										
BIT15 =	100000	1055#	1330											
BIT2 =	000004	1078#												
BIT3 =	000010	1077#												
BIT4 =	000020	1076#												
BIT5 =	000040	1075#												
BIT6 =	000100	1074#												
BIT7 =	000200	1073#												
BIT8 =	000400	1072#												
BIT9 =	001000	1071#												
BLKS14	001254	1243#	3493*	3494*	3495*	3496*	3497	3517	3557	5082	5089			
BLNKS2	027354	4104	4115	4122	5670#									
BPTVEC=	000014	1087#												
BUFFER=	043602	2413	2464	2489*	2540	2577	2579	2582	2584	2630	2680	2714	2763*	2764*
		2765*	2789	2795	2819*	2820*	2821*	2868	2884	2888	2986	3018	3068	3104
		3131	3186*	3187*	3188	3308	3325	3337	3345	3354	3358	3389*	3390*	3391
		3489	4952	4955	4979	4982	5143	5392	6753	6763	6764	6769	6776	6893#
BUSAOR	001216	1233#	2199*	2201*	2204*	2206*	5408	5410*						
BYPASS	001200	1226#	2275*	2406*	2453*	2532*	2621*	2705*	2755*	2850*	2911*	2978*	3096*	3172*
		3302*	3471*	3616*	3723*	3775*	5015	5041						
CHKDRV	001176	1225#	2362*	2366*	2372	2374	3657	3664						
		3834	3842	3904*	3909	3911	3923	3925*	3786	3794	3801	3809	3819	3827
		6747	6753	6755	6757	6759	6761	6764	3927*	3928	3931	3943	3968	6742
		6789	6793						6770	6774	6776	6780	6783	6786
CHNGAD	031564	5530	5884#											
CHNGPH	031332	5499	5856#											
CHNGPT	031511	5522	5876#											
CIR	023052	4877	4894#	4904										
CKSMR =	104407	4026	4045	4617	4823#									
CK.CHR	027046	5596#	5617	5633										
CK.NUM	027062	3737	5421	5447	5469	5513	5525	5539	5549	5559	5613#			
CK.CCT	027016	5578#	5615	5628										
CLEAR =	000001	1148#	2424	2425	4843	4844	5354	5355						
CLRP	025006	2381	2412	2466	2475	2483	2495	2539	2555	2567	2628	2644	2668	2711
		2762	2779	2799	2818	2856	2917	2985	3002	3039	3058	3107	3122	3185

		2201	3236	3721	3336	3385	3521	3538	3563	3625	3644	3750	3793	3808
CHBUF	025362	5253	5302	5254	6789	6793								
CHCMT	025373	5246	5247	5256	5258	5261	5277	5298						
CHC21	025384	5248	5287	5299	6789	6793								
CHC31	025395	5254	5275											
CHC41	025406	5298	5246											
CHC51	025417	5298	5246											
CHC61	025428	5298	5246											
CHC71	025439	5298	5246											
CHC81	025450	5307	5320	5322										
CHC91	025461	5278	5299											
CHC01	025472	5278	5299											
CHC11	025483	1265	3376	5251										
CHC21	025494	5250	5279	5280	5282	5300	6789	6793						
CHC31	025505	5251	5281	5285	5288									
CHC41	025516	5249	5283	5284	5286	5301	6789	6793						
CHC51	025527	1233	2202	2207	2319	2339	2361							
CHC61	027323	2157	5655											
CHC71	023616	2469	2478	2497	2548	2560	2571	2637	2652	2674	2772	2782	2802	2825
		2877	2825	2794	3007	3044	3060	3111	3175	3194	3215	3242	3329	3348
		3333	3531	3541	3568	3638	3647	5022	5040					
		995	4190	4200										
		996	4181	4200										
		3588	5199											
		1002	1208	2234										
		1655	2146	2174	6804									
		2125	6879											
		2132	6876											
		2139	6886											
		1711	1719	1734	1763	1771	1778	1792	1813	1820	1827	1871	1878	1920
		2026	2033	2040	2047	2054	2090	2153	2160	2188	6822			
		1662	1669	1676	1697	1727	1741	1755	1785	1806	1834	1842	1849	1864
		1825	1892	1906	1935	1965	1996	2097	2104	2111	2118	2181	6808	
		1749	1777	6826										
		1899	6810											
		1913	6843											
		1928	6810											
		1833	1890	2003	2010	6815								
		1943	1950	1973	1980	6857								
		1748	1958	1988	6861									
		2018	2022	2069	2076	2083	6865							
		1653	6494											
		2123	2130	6686										
		6697	6872											
		6705	6879											
		6713	6812											
		2144	6721											
		1709	1761	6531										
		2165	6725											
		2172	6730											
		1717	2024	6540										
		1732	1769	1776	1790	1811	1818	1825	1869	1876	2031	2038	2045	2052
		2088	2151	2158	2186	6548								
		1746	6556											
		1660	1667	1674	1695	1725	1739	1753	1783	1804	1832	1840	1847	1862
		1833	1890	1904	1933	1963	1994	2095	2102	2109	2116	2179	6496	
		6505	6811											
		1797	1855	1956	1986	6570								

DT5	043016	1682	1689	2002	2009	6747#
DT51	043372	1942	1949	1972	1979	6770#
DT63	043312	2017	6783#			
DT71	043330	2061	2068	2075	2082	6786#
EMTVEC=	000030	1090#	2219*	2220*		
EM1	032117	1652	5929#			
EM100	037641	2108	6445#			
EM101	037661	2115	6448#			
EM102	037706	2122	6452#			
EM103	037731	2129	6456#			
EM106	040010	2150	6464#			
EM107	040044	2157	6469#			
EM11	032514	1708	5973#			
EM112	040100	2178	6474#			
EM113	040136	2185	6479#			
EM12	032607	1715	5983#			
EM13	032740	1723	5999#			
EM14	033063	1731	6014#			
EM15	033117	1738	6019#			
EM16	033147	1745	6024#			
EM17	033204	1752	6029#			
EM2	032161	1659	5934#			
EM20	033236	1759	6034#			
EM21	033351	1767	6047#			
EM22	033461	1775	6061#			
EM23	033516	1782	6066#			
EM24	033565	1789	6073#			
EM25	033642	1796	6081#			
EM26	033712	1803	6089#			
EM27	034002	1810	6098#			
EM3	032212	1666	5939#			
EM30	034033	1817	6103#			
EM31	034066	1824	6108#			
EM32	034117	1831	6113#			
EM33	034176	1838	6121#			
EM34	034307	1846	6134#			
EM35	034403	1853	6145#			
EM36	034505	1861	6157#			
EM37	034566	1868	6166#			
EM4	032303	1673	5949#			
EM40	034622	1875	6171#			
EM41	034657	1882	6176#			
EM42	034710	1889	6181#			
EM43	034766	1896	6189#			
EM44	035036	1903	6196#			
EM45	035111	1910	6204#			
EM46	035171	1917	6213#			
EM47	035242	1924	6220#			
EM5	032340	1680	5954#			
EM50	035371	1932	6236#			
EM51	035450	1939	6244#			
EM52	035544	1947	6255#			
EM53	035630	1954	6264#			
EM54	035737	1962	6276#			
EM55	036016	1970	6284#			
EM56	036106	1977	6294#			

SIZMEM 025624
 SLASH 031246
 SP =X000006

2259	5360*	5368											
5418	5845*												
1013*	2215*	2231*	2239*	2243	2296*	2350*	2374*	2411*	2458*	2537*	2626*	2710*	
2760*	2855*	2916*	2983*	3105*	3183*	3307*	3475*	3482	3497*	3500*	3504*	3505	
3506*	3522*	3523*	3524	3622	3624*	3658*	3727*	3736	3780*	3786*	3801*	3819*	
3834*	3875*	3911*	3931*	3968*	3976*	4034	4048*	4051*	4107*	4111*	4151*	4152	
4153*	4155	4156	4157*	4159	4161	4163	4169	4171*	4173*	4181*	4185	4189	
4190	4194	4226*	4227	4228	4229*	4234*	4235*	4236*	4242	4267	4268	4269	
4270*	4271*	4292*	4293*	4294*	4295*	4296*	4297*	4298	4301*	4314	4316*	4318	
4328	4330	4332	4333	4334	4335	4336	4338*	4339*	4381*	4382*	4383	4387	
4389	4398	4400	4403	4406*	4407*	4408	4413	4415	4417*	4418	4434*	4435*	
4436	4446	4453*	4456*	4457*	4461*	4462*	4464	4467*	4474	4477*	4481	4483	
4485	4486*	4493	4495	4497*	4498	4500*	4501*	4502*	4503*	4504*	4519*	4520*	
4521*	4522*	4523*	4529*	4542*	4543*	4548	4551	4555*	4562	4566*	4585	4526	
4587*	4588*	4589*	4623*	4625	4628	4629	4651	4652	4656*	4678*	4679*	4620*	
4681*	4682*	4683*	4684*	4695*	4686*	4687*	4694*	4695*	4696*	4697*	4698	4699	
4700	4701	4702	4703	4722*	4723*	4724*	4725*	4726*	4727*	4728	4731	4739*	
4743	4744*	4751	4771*	4772	4773	4776	4777	4778	4779	4792*	4793	4803*	
4804*	4878*	4879*	4880	4886*	4887*	4888	4914*	4915*	4916*	4918*	4920*	4923	
4932*	4944	4960*	4961*	4962*	4998*	4999*	5000*	5002*	5005*	5012*	5015*	5024*	
5025*	5026*	5028*	5031*	5038*	5042*	5047*	5056	5061*	5070	5091*	5092*	5093*	
5094	5096*	5098	5099*	5100*	5101	5103*	5105	5107*	5175*	5176*	5177*	5178*	
5179*	5180*	5181	5182*	5183*	5184	5206*	5207*	5208*	5209*	5210*	5211*	5212*	
5213*	5215*	5216*	5219*	5220	5221*	5222	5231*	5232	5239*	5243*	5413*	5420	
5432	5446	5456*	5459*	5468	5487	5489	5497	5501	5504*	5512	5524	5532	
5538	5548	5558	5613*	5639									
1103*	3210*	3260*	4737	4752*	4759*	5245*	5292*	5314*	5321*				
1104*													
1105*													
1106*	4749*	5241*											
1250*	2264*	3509	3550	5565*									
988*	2215	2411	2458	2537	2626	2710	2760	2855	2916	2983	3105	3183	
3507	3475	3506	3624	3698	3727	3780							
2719	2866	3010	3656	3701	4914*								
1234*	2718*	2865*	3009*	3655*	3700*	4914							
2203	2208*												
977	2201*												
979	2206*	3687	3947	4388	4471								
981	2199*												
983	2204*												
2246	2248*												
2281*													
2291	2319*												
999*													
1249*	2263*	3510	3551	5555*									
2200	2202*												
2205	2207*												
1143*	4942												
1161*	4855												
1164*	3796	3811	4859										
1165*	4863	4996	5003										
1163*	4866												
1162*	4861												
1160*	3829	3844	3857	3863	3885	3893	3919	3939					
1207*	2213	2233*	2235	2241*	2252	2290	3360	3378	3476	3522	3527	3536	
3564	3574	3586	3730	3732	4030	4037	4042	4046	4070	4391	4430	4485*	

SRO = 177572
 SR1 = 177574
 SR2 = 177576
 SR3 = 172516
 SSEC = 001272
 STACK = 001100

 STALL 023134
 STALLT 001220
 START 003710
 START1 003654
 START2 003676
 START3 003644
 START4 003666
 START5 004136
 START6 004350
 START7 004534
 STKLMT= 177774
 STRX 001270
 STRT1A 003660
 STRT2A 003702
 SUCA = 000024
 SUFU = 001000
 SUOL = 040000
 SUPDY = 100000
 SURP03= 020000
 SJSI = 004000
 SUMP = 000400
 SWR 001140

TST10	010602	2905	2969#	2974										
TST11	011522	2972	3087#	3092										
TST12	012052	3090	3163#	3168										
TST12B	012524	3228	3232#											
TST13	012710	3166	3179	3182	3293#	3298								
TST13A	013464	3334	3353	3361	3379	3382	3384#							
TST14	013564	3296	3462#	3467										
TST14B	014332	3525	3543#											
TST15	014620	3465	3607#	3612										
TST15A	015102	3652#	3704											
TST15B	015304	3626	3691#	3692										
TST15C	015314	3691	3696#											
TST16	015362	3610	3714#	3719										
TST17	015614	3717	3766#	3771										
TST2	005714	2447	2523#	2528										
TST3	006372	2526	2612#	2617										
TST4	007050	2615	2696#	2701										
TST5	007300	2699	2746#	2751										
TST5A	007624	2794	2798#											
TST5B	007732	2777	2807	2815	2817#									
TST6	010024	2749	2841#	2846										
TST7	010360	2844	2902#	2907										
TYPOS =	104405	4112	4819#											
TYPE =	104401	2247	2292	2304	2306	2309	2311	2313	2314	2318	2332	2337	2357	2373
		2379	3620	3643	3653	3684	3728	3734	3779	3784	3785	3800	3818	3833
		3852	3871	3873	3905	3910	3930	3946	3958	3964	3972	3989	3993	4032
		4040	4083	4084	4088	4089	4101	4104	4115	4119	4122	4124	4164	4261
		4337	4385	4397	4451	4452	4455	4466	4476	4487	4506	4554	4560	4565
		4569	4574	4575	4577	4580	4584	4815#	5411	5412	5418	5444	5466	5478
		5481	5483	5484	5499	5503	5509	5510	5522	5530	5534	5536	5546	5556
TYPERR	017232	3372	3374	3595	4039	4069#	5312	5317	5333					
TYPOC =	104402	3977	4108	4454	4816#									
TYPON =	104404	4818#												
TYPOS =	104403	2298	2352	2376	3788	3803	3821	3836	3877	3913	3933	3969	4817#	5415
		5506												
UNSAFE	027222	2306	5483	5651#										
WC14	001256	1244#	2265#	2268#	3493	3513	3553	5088	5521#					
WRITE =	000013	1153#												
WFSW	016176	3845	3852#											
WFSW1	016260	3864	3870#											
WFSW2	016402	3904#												
WRTCHK=	000007	1151#	2476	3539										
WRTSEK=	000003	1149#	2467	2546	2558	2635	2770	2800	2823	2992	3042	3108	3192	3327
		3388	3529	3636										
\$AUTOB	001134	1204#	2256#	4444	4601									
\$BOADR	001122	1199#	2593#	2594#	3031#	3032#	3035	3366#	3367#	3681#	3682#	5328#	5329#	6755
		6770	6792	6798										
\$BOOAT	001126	1201#	2582#	2591#	2680#	2681	2795#	2894#	2885	2888#	2889	3025#	3026#	3027#
		3028#	3029#	3030#	3033	3035	3230#	3258#	3365#	3680#	5331#	6755	6761	6770
		6774	6792	6798										
\$BELL	001166	1220#	4032	4058	4397	4594								
\$BUF	002506	1630#	2413#	2464#	2540#	2630#	2714#	2922#	2986#	3104#	3188#	3212#	3238#	3325#
		3343#	3344#	3345#	3391#	3512#	3635#	4881	5114	5133	5159	5174#	5208	5234
\$CHARC	017676	4166#	4176#	4183	4192#	4197#								
\$CKSWR	020662	4430#	4823											
\$CNTAG	001100	1187#	2210	2211	2219	2223	2224	2225						

\$MXCNT	022046	4649	4658												
\$NULL	001154	1213	4171	4200											
\$NWTST=	000001	2390	2392	2436	2439	2513	2515	2601	2603	2689	2691	2737	2739	2833	
		2835	2895	2897	2940	2942	3079	3081	3142	3144	3264	3266	3400	3402	
		3595	3597	3707	3709	3758	3760								
\$OCNT	020124	4233	4262	4275											
\$OMODE	020126	4228	4232	4237	4240	4251	4277								
\$OVER	022032	4619	4639	4647	4655										
\$PASS	001100	1188	3981	3982	4004										
\$QUES	001172	1221	4058	4200	4506	4577	4594								
\$ROCHR	021224	4519	4824												
\$RODEC	***** U	4826													
\$ROLIN	021314	4542	4825												
\$RODOCT=	***** U	4826													
\$ROSZ =	000012	4535													
\$RESRE	022106	4693	4827												
\$RPBA	001344	1280	4938	6745	6750	6766	6780								
\$RPCA	001346	1281	2661	4939	6745	6750	6766	6778	6786						
\$RPCS	001340	1278	2427	2471	2480	2503	2550	2562	2573	2639	2658	2676	2728	2731	
		2774	2783	2811	2814	2827	2873	2879	2931	2934	2996	3012	3046	3062	
		3113	3127	3196	3217	3220	3224	3244	3247	3251	3331	3350	3395	3533	
		3543	3570	3640	4936	6742	6747	6757	6766	6772	6778	6780	6783	6786	
\$RPDA	001350	1282	3011	3025	3031	4940	6745	6750	6766	6772	6778	6780	6783	6786	
\$RPDS	001334	1276	2804	3796	3811	3829	3844	3857	3863	3885	3893	3919	3939	4934	
		6742	6747	6757	6765	6772	6778	6780	6783	6786					
\$RPER	001336	1277	2499	2654	2722	2808	2927	4935	6742	6747	6757	6766	6772	6778	
		6780	6783	6786											
\$RPM1	001352	1283	4941	6745	6750										
\$RPWC	001342	1279	4937	5246	6742	6750	6766	6780							
\$RTNAD	017052	4003													
\$R2A =	***** U	4828													
\$SAVRE	022050	4677	4826												
\$SCOPE	021624	2217	4616												
\$SEC	002512	1632	2416	2459	2542	2633	2649	2670	2712	2767	2857	2919	2989	3000	
		3017	3050	3051	3055	3067	3071	3072	3074	3101	3120	3190	3322	3509	
		3550	3633	4883	5092	5098	5129	5250							
\$SETUP=	000127	2191	2216	2217	2219	2221	2223	2224	2225	2227	2249	3979	4026	4045	
		4053	4389	4394	4395	4425	4601	4617							
\$SILO	001356	1285	4943	6745	5750										
\$SIZE	022144	4722	5361												
\$SIZEX	022406	4760	4770												
\$STUP =	177777	2191													
\$SUCA	001354	1284	2664	3666	4942	6745	6750	6759	6783						
\$SVLAD	022004	4627	4650												
\$SVPC =	000200	966	971												
\$SWR =	167000	921	931	935	936	937	938	939	940	941	1218	1219	1220	2224	
		2225	2227	2228	2410	2457	2536	2625	2709	2759	2854	2915	2982	3100	
		3176	3306	3475	3620	3727	3779	3954	3980	3996	4002	4004	4017	4018	
		4019	4020	4021	4030	4037	4042	4046	4058	4509	4610	4611	4612	4613	
		4618	4630	4632	4633	4634	4641	4642	4643	4652	4655	4658			
\$SWR* =	000000	4613													
\$TIMES	001162	1219	2224	2273	2410	2457	2536	2625	2709	2759	2854	2915	2982	3100	
		3176	3306	3980	4641	4646	4649	4658							
\$TKB	001146	1210	4350	4370	4381	4406	4434	4461							
\$TKCNT	020354	4351	4365	4395	4412	4526	4528								
\$TKINT	020374	2248	2432	3702	4365	4386	4447								

STXQEN# 020974	4355#	4420	4531														
STXQIN 020956	4352#	4366#	4367	4418#	4419#	4420	4422#										
STXQOU 020960	4353#	4367#	4529	4530#	4531	4533#											
STXQSR 020962	4354#	4366	4422	4533													
STXS 001144	1279#	4350	4371#	4402#	4404	4410#	4432	4448#	4458	4470#	4490#						
STYSRV 020444	4368	4381#															
STMPO 001160	1217#	4072#	4073#	6742	6747	6753	6755	6757	6759	6761	6764	6770	6774				
	6776	6780	6793	6786	6789	6793											
STN = 000020	921#	931	2390	2397	2398	2410#	2436	2444	2445	2457#	2513	2523	2524				
	2536#	2601	2612	2613	2625#	2689	2696	2697	2709#	2737	2746	2747	2759#				
	2833	2871	2842	2854#	2895	2902	2903	2915#	2940	2969	2970	2982#	3079				
	3087	3088	3100#	3142	3163	3164	3176#	3264	3293	3294	3306#	3400	3462				
	3463	3475#	3595	3607	3608	3620#	3707	3714	3715	3727#	3758	3766	3767				
	3779#	5451															
STPB 001152	1212#	4189#	4200														
STPFLG 001157	1216#	4147	4200														
STPS 001150	1211#	4187	4200														
STRAP 022452	2221	4792#															
STRAP2 022474	4803#	4814															
STRK 002513	1633#	2650#	3116#	3117	314	3136#	3510#	3551#	3743#	5097#	5099	5105#	5146				
	5249																
STRP = 000014	4807#	4816#	4817#	4818#	4819#	4820#	4821	4822#	4823	4824#	4825#	4826#	4827#				
	4629#																
STRPAD 022506	4797	4814#															
STSTNH 001102	1189#	2403#	2405	2450#	2452	2529#	2531	2618#	2620	2702#	2704	2752#	2754				
	2847#	2849	2908#	2910	2975#	2977	3093#	3095	3169#	3171	3299#	3301	3468#				
	3470	3613#	3615	3720#	3722	3772#	3774	3979#	4029	4058	4073	4608	4650#				
	4655	4659															
STTYIN 021550	4544	4545	4557	4575	4589	4593#											
STYBEN# ***** U	4820																
STYPOS 020130	4291#	4819															
STYPE 017462	4147#	4807	4815														
STYPEC 017632	4168	4175	4182	4187#	4188	4492											
STYPEX 017700	4193	4195	4198#														
STYPOC 017726	4231#	4816															
STYPOH 017742	4230	4233#	4818														
STYPOS 017702	4226#	4817															
SWC 002504	1629#	2414#	2461#	2541#	2557#	2568#	2629#	2647#	2669#	2715#	2766#	2918#	2987#				
	3005#	3054#	3103#	3189#	3324#	3338	3339	3343	3344	3355	3356	3357	3387#				
	3513#	3553#	3634#	4878	5084#	5085#	5088#	5132	5144	5160	5182	5229	5247				
	4621#																
SXTSTR 021636	3946#																
SSGET4= 000000	4227#	4231#	4241	4276#													
SOFILL 020125	4039	4618															
SLOCAT= ***** U	954#	958#	966	967#	969#	971#	975#	1186#	1224	2214	2227	2228	2278				
	2279	2399	2446	2525	2614	2698	2748	2843	2904	2971	3023	3089	3165				
	3295	3370	3464	3609	3716	3755	3768	3961#	3967#	4004	4005#	4058	4200				
	4345#	4350	4354#	4355	4356	4593#	4594	4601#	4658	4659	5310	5315	6893				

.EQUAT	10	9210	985						
.HEADE	10	9210							
.KTI1	10	9210	1095						
.SETUP	10	9210	2191						
.SAPHI	10	9210	931						
.SAPLO	9410	944	945	946	947	948	949	950	
.SCTI	10	9210	962						
.SPTB	10								
.SPTH	10								
.SPTY	10								
.SSTA	10								
.SCATC	10	9210	952						
.SCHTA	10	9210	1180						
.SDB2D	10								
.SDB2O	10								
.SDIV	10								
.SEOP	10	9210	3949						
.SEPO	10	9210	4011						
.SEHRT	10								
.SEULT	10								
.SPOME	10								
.SPANO	10	9210							
.SPOOE	10								
.SPOOC	10								
.SEERO	10	9210	4347						
.SEEAR	10								
.SSAVE	10	9210	4660						
.SSB2D	10								
.S1020	10								
.SSCOP	10	9210	4603						
.SSIZE	10	9210	4706						
.SSUPR	10								
.STRAP	10	9210	4784						
.STYPB	10								
.STYPD	10	9210	4279						
.STYPE	10	9210	4130						
.STYPO	10	9210	4201						
.S40CA	10								
.1170	10								

ADC	5185	5385																
ADD	2492	2791	3338	3344	3345	3357	3358	3504	3890	4080	4157	4229	4239	4311	4467			
	4477	4488	4744	4755	4764	4765	5012	5016	5038	5042	5093	5100	5107	5147	5184			
ASL	5221	5231	5239	5243	5247	5384	5627	5638										
	3890	3898	3927	4077	4078	4079	4500	4501	4502	4796	5054	5067	5158	5177	5178			
	5179	5180	5183	5263	5372	5374	5376	5378	5380	5382	5453	5621	5623	5625				
ASLB	2330	2367	4316															
ASRA	2345	2347	3027	3028	3029	3030	3874	5210	5211	5212	5213	5215	5455					
BCC	2348	2368	4317															
BOS	4106	5622	5624	5626														
BEO	2296	2289	2302	2320	2342	2356	2420	2472	2481	2491	2507	2551	2563	2574	2580			
	2640	2662	2665	2677	2682	2775	2784	2828	2874	2880	2890	2997	3013	3019	3036			
	3047	3063	3069	3073	3114	3128	3132	3135	3197	3218	3225	3228	3245	3248	3256			
	3314	3332	3351	3363	3379	3396	3525	3534	3544	3571	3573	3575	3583	3641	3660			
	3667	3731	3797	3845	3864	3894	3900	3920	3963	3995	4028	4031	4047	4050	4082			
	4087	4092	4096	4099	4114	4160	4195	4256	4392	4445	4484	4499	4527	4563	4633			
	4637	4856	4860	4864	4867	4975	5001	5027	5053	5069	5081	5165	5167	5201	5236			
	5267	5272	5291	5320	5323	5327	5340	5409	5437	5464								
BGE	4647																	
BGT	3661	3985	4263	4325	4416	4496	4899	5079	5477									
BHI	4635	4757	5064	5187	5257	5666	5518	5520	5544	5554	5564	5581						
BHIS	3377	5083																
BIC	2369	3026	3032	3982	4253	4382	4407	4417	4435	4462	4497	4761	5176	5219	5237			
	5133																	
BIS	2869	4258	4259	4319	4320	4504	4738	5238										
BISB	2328	4868	4865	4937	4905	5493												
BIT	2290	2364	2398	2419	2427	2445	2471	2480	2499	2503	2524	2550	2562	2573	2613			
	2639	2654	2658	2676	2697	2722	2728	2731	2747	2774	2783	2804	2808	2811	2814			
	2827	2842	2873	2879	2903	2927	2931	2934	2970	2996	3012	3046	3062	3088	3113			
	3127	3164	3196	3217	3220	3224	3244	3247	3251	3294	3331	3350	3360	3378	3395			
	3463	3476	3524	3527	3533	3536	3543	3564	3570	3574	3586	3608	3640	3715	3730			
	3732	3767	3796	3811	3829	3844	3857	3863	3885	3893	3899	3919	3928	3939	4030			
	4037	4046	4070	4618	4636	4643	4855	4859	4861	4863	4866	4903	4996	5003	5013			
	5022	5029	5039	5068	5268	5326												
BITB	2307	3744	4183	5052														
BLE	2327	4118	4278	4393														
BLOS	3023	3118	3233	4558	4917	5188	5190	5281	5285	5496	5579							
BLOT	2267	3679	4263	4374	4546	5230	5452	5474	5490									
BMT	4174	4264	4308	4474	4514	4494	4900	5095	5102									
BT	2303	3178	3501	4316	4358	4985	5480	5468										
BNE	2214	2216	2217	2218	2219	2220	2221	2222	2223	2224	2225	2226	2227	2228	2229	2230	2231	2232
	2233	2234	2235	2236	2237	2238	2239	2240	2241	2242	2243	2244	2245	2246	2247	2248	2249	2250
	2251	2252	2253	2254	2255	2256	2257	2258	2259	2260	2261	2262	2263	2264	2265	2266	2267	2268
	2269	2270	2271	2272	2273	2274	2275	2276	2277	2278	2279	2280	2281	2282	2283	2284	2285	2286
	2287	2288	2289	2290	2291	2292	2293	2294	2295	2296	2297	2298	2299	2300	2301	2302	2303	2304
	2305	2306	2307	2308	2309	2310	2311	2312	2313	2314	2315	2316	2317	2318	2319	2320	2321	2322
	2323	2324	2325	2326	2327	2328	2329	2330	2331	2332	2333	2334	2335	2336	2337	2338	2339	2340
	2341	2342	2343	2344	2345	2346	2347	2348	2349	2350	2351	2352	2353	2354	2355	2356	2357	2358
	2359	2360	2361	2362	2363	2364	2365	2366	2367	2368	2369	2370	2371	2372	2373	2374	2375	2376
	2377	2378	2379	2380	2381	2382	2383	2384	2385	2386	2387	2388	2389	2390	2391	2392	2393	2394
	2395	2396	2397	2398	2399	2400	2401	2402	2403	2404	2405	2406	2407	2408	2409	2410	2411	2412
	2413	2414	2415	2416	2417	2418	2419	2420	2421	2422	2423	2424	2425	2426	2427	2428	2429	2430
	2431	2432	2433	2434	2435	2436	2437	2438	2439	2440	2441	2442	2443	2444	2445	2446	2447	2448
	2449	2450	2451	2452	2453	2454	2455	2456	2457	2458	2459	2460	2461	2462	2463	2464	2465	2466
	2467	2468	2469	2470	2471	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	2504	2505	2506	2507	2508	2509	2510	2511	2512	2513	2514	2515	2516	2517	2518	2519	2520
	2521	2522	2523	2524	2525	2526	2527	2528	2529	2530	2531	2532	2533	2534	2535	2536	2537	2538
	2539	2540	2541	2542	2543	2544	2545	2546	2547	2548	2549	2550	2551	2552	2553	2554	2555	2556
	2557	2558	2559	2560	2561	2562	2563	2564	2565	2566	2567	2568	2569	2570	2571	2572	2573	2574
	2575	2576	2577	2578	2579	2580	2581	2582	2583	2584	2585	2586	2587	2588	2589	2590	2591	2592
	2593	2594	2595	2596	2597	2598	2599	2600	2601	2602	2603	2604	2605	2606	2607	2608	2609	2610
	2611	2612	2613	2614	2615	2616	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	2673	2674	2675	2676	2677	2678	2679	2680	2681	2682
	2683	2684	2685	2686	2687	2688	2689	2690	2691	2692	2693	2694	2695	2696	2697	2698	2699	2700
	2701	2702	2703	2704	2705	2706	2707	2708	2709	2710	2711	2712	2713	2714	2715	2716	2717	2718
	2719	2720	2721	2722	2723	2724	2725	2726	2727	2728	2729	2730	2731	2732	2733	2734	2735	2736
	2737	2738	2739	2740	2741	2742	2743	2744	2745	2746	2747	2748	2749	2750	2751	2752	2753	2754
	2755	2756	2757	2758	2759	2760	2761	2762	2763	2764	2765	2766	2767	2768	2769	2770	2771	2772
	2773	2774	2775	2776	2777	2778	2779	2780	2781	2782	2783	2784	2785	2786	2787	2788	2789	2790
	2791	2792	2793	2794	2795	2796	2797	2798	2799	2800	2801	2802	2803	2804	2805	2806	2807	2808
	2809	2810	2811	2812	2813	2814	2815	2816	2817	2818	2819	2820	2821	2822	2823	2824	2825	2826
	2827	2828	2829	2830	2831	2832	2833	2834	2835	2836	2837	2838	2839	2840	2841	2842	2843	2844
	2845	2846	2847	2848	2849	2850	2851	2852	2853	2854	2855	2856	2857	2858	2859	2860	2861	2862
	2863	2864	2865	2866	2867	2868	2869	2870	2871	2872	2873	2874	2875	2876	2877	2878	2879	2880
	2881	2882	2883	2884	2885	2886	2887	2888	2889	2890	2891	2892	2893	2894	2895	2896	2897	2898
	2899	2900	2901	2902	2903	2904	2905	2906	2907	2908	2909	2910	2911	29				

G12

MOV	2199	2204	2207	2211	2215	2217	2218	2219	2220	2221	2222	2223	2227	2228	2231
	2232	2233	2234	2239	2241	2242	2243	2245	2248	2249	2249	2250	2276	2278	2279
	2284	2287	2294	2296	2325	2335	2350	2360	2362	2374	2380	2401	2402	2403	2405
	2406	2407	2410	2411	2413	2414	2424	2425	2448	2449	2450	2452	2453	2454	2457
	2458	2461	2462	2464	2465	2467	2474	2476	2488	2489	2494	2508	2527	2528	2529
	2531	2532	2533	2536	2537	2538	2540	2541	2544	2546	2554	2557	2566	2568	2569
	2581	2582	2584	2585	2591	2593	2616	2617	2618	2620	2621	2622	2625	2626	2627
	2629	2630	2643	2645	2647	2648	2649	2667	2669	2672	2680	2700	2701	2702	2704
	2705	2706	2709	2710	2714	2715	2718	2750	2751	2752	2754	2755	2756	2759	2760
	2761	2763	2764	2765	2766	2769	2778	2787	2795	2798	2800	2817	2822	2845	2846
	2784	2789	2850	2851	2854	2855	2862	2865	2867	2868	2883	2884	2888	2906	2907
	2909	2910	2911	2912	2915	2916	2918	2922	2973	2974	2975	2977	2978	2979	2982
	2993	2984	2996	2987	2991	3001	3003	3005	3009	3011	3025	3031	3038	3053	3054
	3056	3091	3092	3093	3095	3096	3097	3100	3103	3104	3105	3106	3121	3167	3168
	3169	3171	3172	3173	3176	3183	3184	3187	3188	3189	3192	3200	3202	3204	3205
	3206	3207	3208	3209	3210	3212	3229	3230	3234	3235	3238	3257	3258	3297	3298
	3299	3301	3302	3303	3306	3307	3308	3309	3310	3311	3312	3315	3316	3317	3320
	3324	3325	3326	3335	3337	3339	3343	3354	3355	3356	3364	3365	3366	3370	3384
	3387	3391	3466	3467	3468	3470	3471	3472	3475	3480	3482	3483	3484	3488	3489
	3491	3492	3493	3497	3505	3506	3511	3512	3513	3514	3520	3522	3552	3553	3554
	3562	3611	3612	3613	3615	3616	3617	3622	3624	3626	3627	3628	3629	3632	3634
	3635	3654	3655	3657	3658	3672	3673	3679	3680	3681	3691	3696	3698	3700	3703
	3718	3719	3720	3721	3722	3724	3727	3736	3746	3748	3770	3771	3772	3774	3775
	3776	3780	3781	3782	3801	3807	3819	3825	3834	3840	3854	3861	3870	3875	3882
	3883	3891	3906	3911	3917	3925	3931	3937	3968	3976	3986	3994	4029	4034	4049
	4051	4081	4086	4091	4094	4095	4107	4111	4123	4151	4152	4156	4171	4226	4234
	4235	4236	4243	4267	4267	4268	4269	4270	4271	4292	4293	4294	4295	4296	4297
	4298	4303	4306	4333	4333	4333	4334	4335	4336	4338	4339	4366	4367	4368	4369
	4371	4410	4426	4433	4434	4435	4490	4519	4520	4523	4533	4542	4544	4555	4586
	4587	4588	4589	4589	4589	4626	4629	4638	4648	4649	4651	4652	4655	4656	4678
	4679	4680	4681	4681	4681	4684	4685	4686	4687	4694	4695	4696	4697	4698	4699
	4700	4701	4702	4703	4722	4723	4724	4685	4687	4694	4695	4696	4697	4698	4699
	4741	4742	4743	4744	4745	4749	4753	4758	4762	4766	4767	4768	4772	4773	4776
	4777	4778	4779	4780	4781	4782	4783	4804	4804	4834	4836	4842	4843	4844	4878
	4880	4881	4882	4883	4884	4885	4914	4918	4932	4933	4934	4935	4936	4937	4938
	4939	4940	4941	4942	4943	4944	4951	4952	4954	4955	4956	4959	4961	4962	4965
	4966	4978	4979	4980	4981	4982	4983	4987	4998	5002	5003	5004	5008	5035	5047
	5048	5056	5061	5062	5063	5064	5068	5114	5118	5131	5132	5133	5134	5143	5144
	5157	5159	5160	5161	5162	5163	5174	5175	5182	5191	5192	5203	5204	5205	5208
	5214	5220	5221	5222	5223	5224	5227	5228	5238	5239	5255	5256	5257	5258	5259
	5245	5246	5247	5248	5249	5250	5251	5252	5253	5254	5255	5256	5257	5258	5259
MOV8	2226	2227	2228	2229	2230	2231	2232	2233	2234	2235	2236	2237	2238	2239	2240
	2241	2242	2243	2244	2245	2246	2247	2248	2249	2250	2251	2252	2253	2254	2255
	2256	2257	2258	2259	2260	2261	2262	2263	2264	2265	2266	2267	2268	2269	2270
	2271	2272	2273	2274	2275	2276	2277	2278	2279	2280	2281	2282	2283	2284	2285
	2286	2287	2288	2289	2290	2291	2292	2293	2294	2295	2296	2297	2298	2299	2300
	2301	2302	2303	2304	2305	2306	2307	2308	2309	2310	2311	2312	2313	2314	2315
	2316	2317	2318	2319	2320	2321	2322	2323	2324	2325	2326	2327	2328	2329	2330
	2331	2332	2333	2334	2335	2336	2337	2338	2339	2340	2341	2342	2343	2344	2345
	2346	2347	2348	2349	2350	2351	2352	2353	2354	2355	2356	2357	2358	2359	2360
	2361	2362	2363	2364	2365	2366	2367	2368	2369	2370	2371	2372	2373	2374	2375
	2376	2377	2378	2379	2380	2381	2382	2383	2384	2385	2386	2387	2388	2389	2390
	2391	2392	2393	2394	2395	2396	2397	2398	2399	2400	2401	2402	2403	2404	2405
	2406	2407	2408	2409	2410	2411	2412	2413	2414	2415	2416	2417	2418	2419	2420
	2421	2422	2423	2424	2425	2426	2427	2428	2429	2430	2431	2432	2433	2434	2435
	2436	2437	2438	2439	2440	2441	2442	2443	2444	2445	2446	2447	2448	2449	2450
	2451	2452	2453	2454	2455	2456	2457	2458	2459	2460	2461	2462	2463	2464	2465
	2466	2467	2468	2469	2470	2471	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	2504	2505	2506	2507	2508	2509	2510
	2511	2512	2513	2514	2515	2516	2517	2518	2519	2520	2521	2522	2523	2524	2525
	2526	2527	2528	2529	2530	2531	2532	2533	2534	2535	2536	2537	2538	2539	2540
	2541	2542	2543	2544	2545	2546	2547	2548	2549	2550	2551	2552	2553	2554	2555
	2556	2557	2558	2559	2560	2561	2562	2563	2564	2565	2566	2567	2568	2569	2570
	2571	2572	2573	2574	2575	2576	2577	2578	2579	2580	2581	2582	2583	2584	2585
	2586	2587	2588	2589	2590	2591	2592	2593	2594	2595	2596	2597	2598	2599	2600
	2601	2602	2603	2604	2605	2606	2607	2608	2609	2610	2611	2612	2613	2614	2615
	2616	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	2673	2674	2675
	2676	2677	2678	2679	2680	2681	2682	2683	2684	2685	2686	2687	2688	2689	2690
	2691	2692	2693	2694	2695	2696	2697	2698	2699	2700	2701	2702	2703	2704	2705
	2706	2707	2708	2709	2710	2711	2712	2713	2714	2715	2716	2717	2718	2719	2720
	2721	2722	2723	2724	2725	2726	2727	2728	2729	2730	2731	2732	2733	2734	2735
	2736	2737	2738	2739	2740	2741	2742	2743	2744	2745	2746	2747	2748	2749	2750
	2751	2752	2753	2754	2755	2756	2757	2758	2759	2760	2761	2762	2763	2764	2765
	2766	2767	2768	2769	2770	2771	2772	2773	2774	2775	2776	2777	2778	2779	2780
	2781	2782	2783	2784	2785	2786	2787	2788	2789	2790	2791	2792	2793	2794	2795
	2796	2797	2798	2799	2800	2801	2802	2803	2804	2805	2806	2807	2808	2809	2810
	2811	2812	2813	2814	2815	2816	2817	2818	2819	2820	2821	2822	2823	2824	2825
	2826	2827	2828	2829	2830	2831	2832	2833	2834	2835	2836	2837	2838	2839	2840
	2841	2842	2843	2844	2845	2846	2847	2848	2849	2850	2851	2852	2853	2854	2855
	2856	2857	2858	2859	2860	2861	2862	2863	2864	2865	2866	2867	2868	2869	2870
	2871	2872	2873	2874	2875	2876	2877	2878	2879	2880	2881				

ROR	4105	5216	5394													
RTI	2240	4057	4158	4272	4340	4411	4423	4491	4524	4534	4590	4657	4698	4704	4805	
RTS	4128	4198	4372	4780	4798	4850	4869	4890	4908	4924	4945	4969	4992	5017	5043	
	5757	5071	5108	5123	5138	5151	5170	5194	5293	5335	5344	5356	5400	5430	5567	
	5526	5601	5641													
SBC	3518	3558	5090													
SDB	4745															
SUB	2594	3367	3500	3517	3557	3682	4035	4307	4769	4770	4916	5000	5026	5089	5096	
	5103	5261	5329	5392	5397	5399										
SWAB	3494	5085														
TRAP	4730	4807	4816	4817	4818	4819	4821	4823	4824	4825	4826	4827				
TST	2250	2395	2319	2339	2341	2355	2586	2661	2681	3177	3180	3368	3478	3515	3555	
	3572	3732	3856	3923	3952	4042	4049	4102	4155	4163	4185	4255	4312	4322	4370	
	4387	4368	4403	4446	4463	4498	4526	4551	4562	4585	4625	4737	4754	4766	4794	
	4919	4957	4963	4974	4984	5076	5078	5080	5200	5235	5277	5290	5306	5308	5319	
	5322	5339	5363	5408	5428	5436	5463	5487	5525	5598	5636	5637				
TSTB	2301	2326	3659	4147	4187	4314	4328	4404	4432	4458	4632	4733	4898	4949	4976	
	5395	5476	5479	5491	5596											
.ASCII	1221	1222	5676	5709	5714	5726	5783	5812	5829	5892	5983	5999	6034	6047	6121	
	6220	6244	6264	6304	6318	6339	6352	6391	6556	6580	6606	6627	6697	6713	6730	
	6895															
.ASCIIZ	1220	1223	3961	3967	3975	3992	4594	4595	4596	4597	4599	5645	5649	5651	5653	
	5656	5658	5660	5664	5665	5666	5670	5672	5685	5693	5698	5739	5745	5752	5758	
	5766	5770	5773	5778	5792	5797	5808	5818	5835	5845	5846	5850	5854	5856	5862	
	5870	5876	5884	5901	5909	5914	5918	5923	5928	5934	5939	5949	5954	5961	5966	
	5973	5992	6008	6014	6019	6024	6029	6042	6056	6061	6066	6073	6081	6088	6098	
	6103	6109	6113	6129	6134	6145	6157	6166	6171	6176	6181	6189	6196	6204	6213	
	6228	6236	6252	6255	6272	6276	6294	6294	6313	6326	6329	6334	6347	6359	6364	
	6370	6376	6384	6400	6404	6411	6419	6428	6438	6442	6445	6448	6452	6456	6464	
	6469	6474	6479	6494	6496	6505	6513	6522	6531	6540	6548	6563	6570	6577	6587	
	6574	6603	6614	6622	6635	6643	6650	6657	6668	6677	6686	6701	6705	6717	6721	
	6725	6733	6899													
.BLKB	4354	4593														
.BLKM	4345															
.BYTE	1189	1190	1195	1196	1204	1205	1213	1214	1215	1216	1289	1290	1291	1292	1293	
	1294	1295	1296	1304	1305	1306	1307	1308	1309	1310	1311	1625	1626	1627	1628	
	1632	1633	2299	2300	2353	2354	2377	2378	3789	3790	3804	3805	3822	3823	3837	
	3838	3878	3879	3914	3915	3934	3935	3970	3971	4004	4273	4274	4275	4276	4591	
	4592	5416	5417	5507	5508	6805	6806	6809	6810	6812	6813	6816	6817	6819	6820	
	6823	6824	6827	6828	6830	6831	6834	6835	6837	6838	6840	6841	6844	6845	6847	
	6848	6851	6852	6854	6855	6858	6859	6862	6863	6866	6867	6870	6871	6873	6874	
	6877	6878	6880	6881	6883	6884	6887	6888								
.DSABL	4508															
.EVAL	1	921	4350													
.END	6907															
.ENOC	926	938	940	941	961	965	969	971	989	1081	1095	1107	1118	1129	1183	
	1187	1189	1217	1218	1219	1220	1221	1225	1635	2191	2193	2197	2215	2216	2219	
	2221	2223	2224	2225	2227	2229	2245	2252	2258	2300	2301	2354	2355	2378	2379	
	2391	2392	2396	2397	2401	2402	2410	2411	2437	2438	2443	2444	2448	2449	2457	
	2458	2514	2515	2522	2523	2527	2528	2536	2537	2602	2603	2611	2612	2616	2617	
	2625	2626	2690	2691	2695	2696	2700	2701	2709	2710	2738	2739	2745	2746	2750	
	2751	2759	2760	2834	2835	2840	2841	2845	2846	2854	2855	2896	2897	2901	2902	
	2906	2907	2915	2916	2941	2942	2968	2969	2973	2974	2982	2983	3080	3081	3086	
	3087	3091	3092	3100	3101	3143	3144	3162	3163	3167	3168	3176	3177	3265	3266	
	3292	3293	3297	3298	3306	3307	3401	3402	3461	3462	3466	3467	3475	3596	3597	
	3606	3607	3611	3612	3620	3669	3673	3708	3709	3713	3714	3718	3719	3727	3759	

	3760	3765	3766	3770	3771	3779	3790	3791	3805	3806	3823	3824	3838	3839	3879
	3990	3915	3916	3935	3936	3952	3954	3956	3961	3967	3971	3972	3975	3979	3985
	3998	3999	3992	3994	3996	4002	4004	4005	4014	4017	4027	4034	4039	4040	4041
	4042	4053	4057	4058	4060	4133	4153	4204	4282	4350	4380	4389	4393	4424	4426
	4441	4472	4508	4512	4523	4535	4536	4544	4546	4549	4577	4594	4595	4601	4606
	4609	4613	4618	4620	4631	4632	4634	4636	4643	4645	4650	4651	4655	4658	4659
	4663	4709	4732	4739	4783	4787	4793	4796	4815	4816	4817	4818	4819	4820	4821
	4822	4823	4824	4825	4826	4827	4828	4896	5010	5036	5342	5417	5418	5434	5508
	5509														
.EQUIV	989	990	998	1013	1014	1043	1044	1045	1046	1047	1048	1049	1050	1051	1052
.EVEN	1071	1072	1073	1074	1075	1076	1077	1078	1079	1080					
.IF	3961	3967	3975	3992	4005	4601	6737	6891							
	922	938	939	940	941	961	964	967	969	987	1053	1081	1107	1118	1129
	1182	1186	1188	1217	1218	1219	1220	1224	1225	2191	2192	2196	2210	2215	2217
	2219	2221	2223	2224	2225	2227	2245	2249	2252	2299	2300	2353	2354	2377	2378
	2390	2392	2396	2400	2401	2410	2411	2436	2438	2443	2447	2448	2457	2458	2513
	2515	2522	2526	2527	2536	2537	2601	2603	2611	2615	2616	2625	2626	2639	2691
	2695	2699	2700	2709	2710	2737	2739	2745	2749	2750	2759	2760	2833	2835	2840
	2844	2845	2854	2855	2895	2897	2901	2905	2906	2915	2916	2940	2942	2968	2972
	2973	2982	2983	3079	3081	3086	3090	3091	3100	3101	3142	3144	3162	3166	3167
	3176	3177	3264	3266	3292	3296	3297	3306	3307	3400	3402	3461	3465	3466	3475
	3595	3597	3606	3610	3611	3620	3668	3672	3707	3709	3713	3717	3718	3727	3758
	3760	3765	3769	3770	3779	3789	3790	3804	3805	3822	3823	3837	3838	3878	3879
	3914	3915	3934	3935	3951	3952	3953	3954	3955	3956	3958	3960	3966	3970	3971
	3974	3984	3987	3989	3991	3994	3996	4002	4004	4005	4013	4016	4027	4030	4037
	4039	4040	4042	4046	4053	4057	4058	4059	4132	4153	4203	4281	4349	4351	4378
	4383	4389	4425	4426	4440	4464	4511	4512	4522	4535	4543	4545	4549	4550	4593
	4594	4601	4605	4608	4613	4618	4630	4632	4633	4634	4643	4645	4651	4652	4657
	4658	4659	4662	4708	4712	4731	4739	4786	4792	4796	4807	4816	4817	4818	4819
	4820	4821	4823	4824	4825	4826	4827	4828	4895	5009	5035	5341	5416	5417	5433
	5507	5508													
.IFF	938	940	941	965	969	971	997	1183	1186	1188	1217	1225	2193	2197	2215
	2300	2354	2378	2390	2391	2392	2397	2401	2402	2410	2436	2437	2438	2444	2448
	2449	2457	2513	2514	2515	2523	2527	2528	2536	2601	2602	2603	2612	2616	2617
	2625	2689	2690	2691	2696	2700	2701	2709	2737	2738	2739	2746	2750	2751	2759
	2833	2834	2835	2841	2845	2846	2854	2895	2896	2897	2302	2906	2907	2915	2941
	2941	2942	2969	2973	2974	2982	3079	3080	3081	3087	3091	3092	3100	3142	3143
	3144	3163	3167	3168	3176	3264	3265	3266	3293	3297	3298	3306	3400	3401	3402
	3462	3466	3467	3475	3595	3596	3597	3607	3611	3612	3620	3668	3672	3707	3708
	3709	3714	3718	3719	3727	3758	3759	3760	3766	3769	3771	3779	3790	3791	3805
	3806	3823	3824	3838	3839	3879	3890	3915	3916	3935	3936	3952	3955	3958	3971
	3984	3987	4034	4014	4016	4030	4053	4058	4060	4133	4204	4282	4350	4426	4441
	4451	4512	4515	4523	4535	4536	4545	4577	4593	4606	4631	4632	4634	4658	4663
	4709	4721	4771	4787	4793	4895	5009	5035	5341	5417	5418	5433	5508		
.IFT	3961	3967	3975	3992	4040	4514	4519	4642	4724	4732	4775	4782			
.IFTF	3961	3967	3975	3992	4039	4451	4512	4515	4640	4721	4726	4771			
.IIF	921	926	931	935	936	937	938	941	942	944	945	946			
	950	958	1224	2216	2219	2223	2224	2225	2227	2228	2297	2351	2375	3787	3802
	3920	3835	3876	3912	3932	3954	3969	3977	3979	3980	4004	4005	4017	4018	4019
	4010	4021	4026	4045	4053	4058	4108	4112	4200	4350	4356	4394	4395	4454	4585
	4534	4601	4609	4610	4611	4612	4613	4617	4641	4642	4655	4658	4659	4815	4816
	4817	4818	4819	4821	4823	4824	4825	4826	4827	5414	5505				
.IRP	1225	1316	2191	2390	2436	2513	2601	2689	2737	2833	2895	2940	3079	3142	3206
	3264	3400	3595	3707	3758	3958	4292	4332	4678	4698	4934	4940	5223		
.LIST	1	921	958	1095	1217	1218	2191	2229	2390	2410	2436	2457	2513	2536	2601
	2625	2689	2709	2737	2759	2833	2854	2895	2915	2940	2982	3079	3100	3142	3176

	3264	3306	3400	3475	3595	3620	3707	3727	3758	3779	3961	3967	3975	3979	3992
	3996	4053	4535	4613	4807	4815	4816	4817	4818	4819	4620	4821	4822	4823	4824
	4825	4826	4827	4828											
.MACRO	1	941	1179	1180	2191	2389	2435	2512	2600	2688	2736	2832	2894	2939	3078
.MCALL	3141	3263	3400	3595	3706	3757	4807								
.MLIST	921	1095	2229												
	921	958	1095	1217	1218	2191	2229	2390	2410	2436	2457	2513	2536	2601	
	2625	2689	2709	2737	2759	2833	2854	2895	2915	2940	2982	3079	3100	3142	3176
	3264	3306	3400	3475	3595	3620	3707	3727	3758	3779	3961	3967	3975	3979	3992
	3996	4053	4535	4613	4807	4815	4816	4817	4818	4819	4820	4821	4822	4823	4824
	4825	4826	4827	4828											
.PAGE	1180	1635	2190												
.P.M	1														
.REPT	958	1217	3027	5177	5210	5372									
.SETTL	931	952	962	973	985	1095	1180	1635	2194	2209	2249	2387	2390	2436	2513
	2601	2689	2737	2833	2995	2940	3079	3142	3264	3400	3595	3707	3758	3949	4009
	4011	4130	4201	4279	4347	4603	4660	4706	4784	4807	5643	5926			
.TITLE	921														
.WORD	958	959	960	970	1188	1191	1192	1193	1194	1197	1198	1199	1200	1201	1202
	1203	1206	1207	1208	1217	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	1256	1258	1259	1262	1263	1264	1265	1266	1270	1271
	1272	1276	1277	1278	1279	1280	1281	1282	1283	1284	1285	1315	1316	1317	1318
	1319	1320	1321	1322	1323	1324	1325	1326	1327	1328	1329	1330	1332	1333	1334
	1335	1336	1337	1338	1339	1340	1341	1342	1343	1344	1345	1346	1347	1351	1352
	1353	1354	1355	1356	1357	1358	1359	1360	1361	1362	1363	1364	1365	1366	1368
	1369	1370	1371	1372	1373	1374	1375	1376	1377	1378	1379	1380	1381	1382	1383
	1384	1386	1387	1388	1389	1390	1391	1392	1393	1394	1395	1396	1397	1398	1399
	1400	1402	1403	1404	1405	1406	1407	1408	1409	1410	1411	1412	1413	1414	1415
	1416	1417	1419	1420	1421	1422	1423	1424	1425	1426	1427	1429	1429	1430	1431
	1432	1433	1434	1436	1437	1438	1439	1440	1441	1442	1443	1444	1445	1446	1447
	1448	1449	1450	1451	1453	1454	1455	1456	1457	1458	1459	1460	1461	1462	1463
	1464	1465	1466	1467	1468	1470	1471	1472	1473	1474	1475	1476	1477	1478	1479
	1480	1481	1482	1483	1484	1485	1487	1488	1489	1490	1491	1492	1493	1494	1495
	1496	1497	1498	1499	1500	1501	1502	1504	1505	1506	1507	1508	1509	1510	1511
	1512	1513	1514	1515	1516	1517	1518	1519	1521	1522	1523	1524	1525	1526	1527
	1528	1529	1530	1531	1532	1533	1534	1535	1536	1538	1539	1540	1541	1542	1543
	1544	1545	1546	1547	1548	1549	1550	1551	1552	1553	1555	1556	1557	1558	1559
	1560	1561	1562	1563	1564	1565	1566	1567	1568	1569	1570	1572	1573	1574	1575
	1576	1577	1578	1579	1580	1581	1582	1583	1584	1585	1586	1587	1589	1590	1591
	1592	1593	1594	1595	1596	1597	1598	1599	1600	1601	1602	1603	1604	1606	1607
	1608	1609	1610	1611	1612	1613	1614	1615	1616	1617	1618	1619	1620	1621	1629
	1630	1631	3383	3984	3987	4003	4095	4090	4125	4197	4277	4351	4352	4353	4734
	4781	4782	4814	5295	5296	5297	5298	5299	5300	5301	5302	6741	6742	6745	6747
	6750	6753	6755	6757	6759	6761	6763	6764	6766	6769	6770	6772	6774	6776	6778
	6780	6783	6786	6789	6792	6793	6796	6798	6799	6800	6804	6808	6811	6815	6818
	6822	6826	6829	6833	6836	6839	6843	6846	6850	6853	6857	6861	6865	6869	6872
	6876	6879	6882	6886											

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

* ,NOW. SEQ/SOL/CRF/ML: TOC/PAGNUM=DZRPYB.SML, DZRPYB.CMB

K12

MD-11-DZRPY-B, RP11-E FUNCTIONAL LOGIC & R/W TEST MACY11 27(732) 02-NOV-76 14:50 PAGE 157
DZRPYB.CMB CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

RUN-TIME: 50 74 10 SECONDS
RUN-TIME RATIO: 472/135=3.4
CORE USED: 38K (75 PAGES)

