

RX11, RX01
RX02

RX02 FCTN/LGC
CZRXFBO

AH-E626B-MC
FICHE 1 OF 2

SEP 1982
COPYRIGHT © 79-82
MADE IN USA



The main body of the document is a microfiche card containing a grid of approximately 15 columns and 15 rows of frames. Each frame contains a small, high-contrast image of a document page, which appears to be a technical drawing or a data table. The images are very small and difficult to read, but they follow a consistent layout across the grid. The entire grid is set against a dark background.

RX11, RX01
RX02

RX02 FCTN/LGC
CZRXFBO

AH-E626B-MC
FICHE 2 OF 2

SEP 1982
COPYRIGHT © 79-82
MADE IN USA



.REM %

4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45

IDENTIFICATION

PRODUCT CODE:	AC-E625B-MC
PRODUCT NAME:	CZRFB0 RX02 FCTN/LGC
PRODUCT DATE:	29-MAR-82
MAINTAINER:	S.S.S.T.A.
AUTHOR:	L. S. PRUCHA

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

NO RESPONSIBLIITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1979,1982 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADE MARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83

TABLE OF CONTENTS

1.0	GENERAL INFORMATION
1.1	PROGRAM ABSTRACT
1.2	SYSTEM REQUIREMENTS
1.2.1	HARDWARE REQUIREMENTS
1.2.2	SOFTWARE REQUIREMENTS
1.3	RELATED DOCUMENTS AND STANDARDS
1.4	DIAGNOSTIC HIERARCHY PREREQUISITES
1.5	ASSUMPTIONS
1.6	MEMORY MAP
2.0	OPERATING INSTRUCTIONS
2.1	HARDWARE QUESTIONS
2.2	SOFTWARE QUESTIONS
3.0	ERROR INFORMATION
3.1	SYSTEM FATAL ERRORS
3.2	DEVICE FATAL ERRORS
3.3	HARD ERRORS
3.4	SOFT ERRORS
3.5	ERROR PRINTOUT FORMAT
4.0	PERFORMANCE AND PROGRESS REPORTS
5.0	DEVICE INFORMATION TABLES
5.1	DEVICE REGISTERS
5.2	DEVICE PROTOCOL
5.3	DEVICE HARDWARE CONFIGURATION
6.0	TEST SUMMARIES
7.0	LISTING INDEX
7.1	LISTING

85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

THIS PROGRAM CONTAINS A FUNCTION TEST OPTION AND A LOGIC TEST OPTION. A USER MAY SELECT TO RUN THE FUNCTION TEST ONLY, LOGIC TEST ONLY OR BOTH. THE DIAGNOSTIC WILL DEFAULT TO RUN THE LOGIC TEST ONLY. THE FUNCTION TEST WILL PERFORM A FUNCTIONAL EVALUATION OF THE DEVICE. IT WILL VERIFY THAT THE DRIVES CAN SEEK, THAT DATA CAN BE WRITTEN AND READ AND THAT DRIVE STATUS IS CORRECT. THE LOGIC TEST WILL ANALYZE DEVICE FAILURES, REPORT FAILING FIELD REPLACEABLE UNITS AND PROVIDE EXTENSIVE INFORMATION ON THE NATURE OF THE ERROR.

1.2 SYSTEM REQUIREMENTS

1.2.1 HARDWARE REQUIREMENTS

PDP-11/LSI-11 PROCESSOR WITH 16K OR MORE OF MEMORY
CONSOLE DEVICE (LA30, LA36, VT50, ETC.), LOAD MEDIA DEVICE.

1.2.2 SOFTWARE REQUIREMENTS

THIS DIAGNOSTIC IS DESIGNED TO RUN WITH THE DIAGNOSTIC SUPERVISOR AS DESCRIBED IN PARAGRAPH 2.0.

1.3 RELATED DOCUMENTS AND STANDARDS

XYDP+ USERS MANUAL

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

NONE

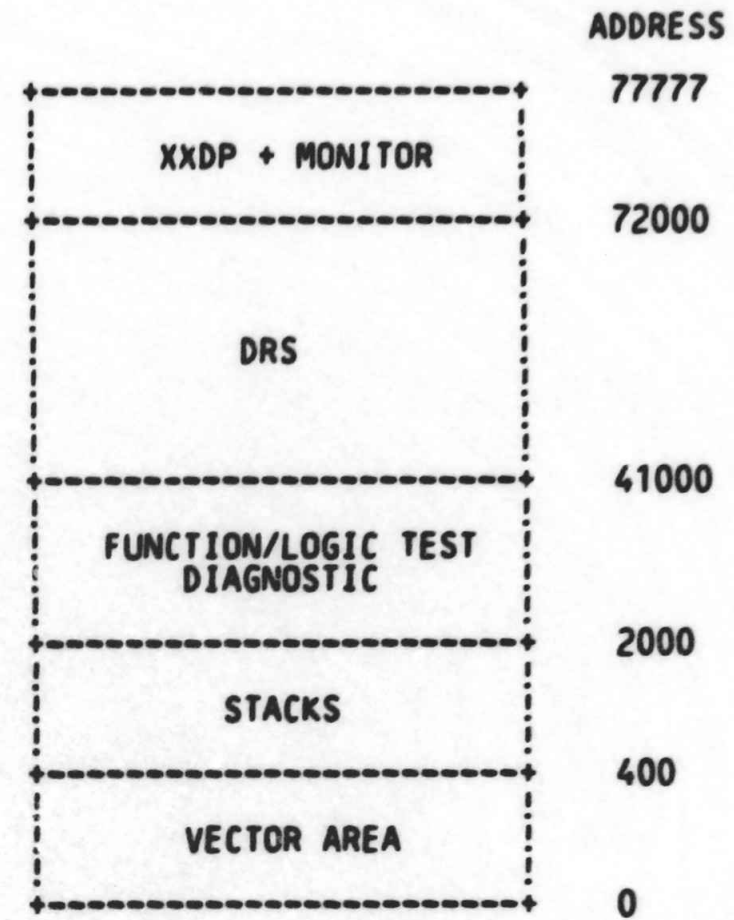
1.5 ASSUMPTIONS

THIS DIAGNOSTIC ASSUMES THAT ALL HARDWARE OTHER THAN THE RXV21/RX211 INTERFACE OR RX02 SUBSYSTEM BEING TESTED WORKS PROPERLY. FALSE ERRORS MAY BE REPORTED IF THE PROCESSOR, MEMORY, ETC., DOES NOT FUNCTION PROPERLY.

140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179

1.6 MEMORY MAP

MEMORY LAYOUT ON 16K MACHINE - XXDP ENVIRONMENT



IN A MACHINE WITH MORE MEMORY FREE SPACE WILL OCCUR BETWEEN THE DIAGNOSTIC AND THE DRS.

181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217

2.0 OPERATING INSTRUCTIONS

THIS IS A REV C SUPERVISOR DIAGNOSTIC: FOR OPERATING INSTRUCTIONS, PLEASE SEE CHAPTER 5 OF XXDP+ OPERATOR'S MANUAL. THEY ARE NO LONGER INCLUDED IN THE DIAGNOSTIC LISTING BECAUSE IT IS DESIRED THAT A CHANGE IN THOSE INSTRUCTIONS NOT REQUIRE A RE-ASSEMBLY OF ALL SUPERVISOR DIAGNOSTICS.

2.1 HARDWARE QUESTIONS

THE FOLLOWING SERIES OF QUESTIONS COMPRISE THE PARAMETERS NECESSARY TO IDENTIFY EACH FLOPPY DISK SUBSYSTEM.

RX ADDRESS -

THIS PARAMETER DEFINES THE BASE BUS ADDRESS FOR THE FLOPPY DISK SUBSYSTEM INTERFACE.

VECTOR ADDRESS -

THIS PARAMETER DEFINES THE INTERRUPT VECTOR ADDRESS FOR THE FLOPPY DISK SUBSYSTEM INTERFACE.

DRIVE # -

THIS PARAMETER DEFINES THE FLOPPY DISK SUBSYSTEM DRIVE NUMBER (0 - 1).

EXPANSION-TYPE -

THIS PARAMETER IS TO BE USED FOR FUTURE EXPANSION. TYPE A CARRIAGE RETURN.

BR-LEVEL -

THIS PARAMETER DEFINES THE BR-LEVEL OF THE FLOPPY DISK SUBSYSTEM INTERFACE. A BR LEVEL OF 0 -> 7 WILL BE ACCEPTED.

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
262
263
264
265
266
267
268

2.2

SOFTWARE QUESTIONS

THE FOLLOWING SERIES OF QUESTIONS ARE INTENDED TO PROVIDE SELECTION OF VARIOUS TEST OPTIONS.

TEST HELP -

IF ANSWER IS YES "Y" THEN A SHORT HELP DESCRIPTION ON USE OF THIS DIAGNOSTIC WILL BE TYPED.

LOGIC TEST MODE -

IF ANSWER IS YES "Y" THEN THE LOGIC TESTS WILL BE DONE. THESE TESTS PROVIDE EXTENSIVE TESTING OF THE FLOPPY DISK SUBSYSTEM LOGIC. FAILING FIELD REPLACEABLE UNITS WILL BE CALLED OUT AND EXTENSIVE ERROR INFORMATION WILL BE REPORTED. THE AMOUNT OF ERROR INFORMATION MAYBE SUPPRESSED WITH THE "DRS" "IXE" FLAG.

FUNCTION TEST MODE -

IF ANSWER IS YES "Y" THEN THE FUNCTION TESTS WILL BE DONE. THESE TESTS PROVIDE A QUICK VERIFICATION THAT THE FLOPPY DISK SUBSYSTEM IS FUNCTIONAL, ONLY VERY BASIC ERROR REPORTING IS DONE, MEDIA RELATED ERRORS ARE IGNORED.

DEVICE FATAL THRESHOLD LEVEL -

THE DEVICE FATAL THRESHOLD LEVEL (DFTL) IS INITIALLY SET=1. THIS THRESHOLD LEVEL EQUALS THE NUMBER OR HARD ERRORS THAT WILL CAUSE A DEVICE FATAL ERROR WHEN THE DRS "EVL" FLAG IS SET.

NON-EXISTENT MEMORY ADDRESS -

THIS ADDRESS IS USED BY THE DIAGNOSTIC TO TEST THE RX CAPABILITY TO DETECT NON EXISTENT MEMORY (VIA BUS TIME OUT). THIS IS ONLY TESTED DURING THE NON EXISTENT MEMORY TEST. THE STANDARD 160000 DIAGNOSTIC ADDRESS IS USED BY DEFAULT.

EXTENDED ADDRESS BITS -

THESE BITS ARE USED DURING THE NPR & NON EXISTENT MEMORY TESTS TO TEST THE RX EXTENDED MEMORY CAPABILITIES. BITS 13 & 12 ARE SET IN THE RXCS REGISTER CORRESPONDING TO BITS 1 & 0 SET BY THE USER.

TEST CONTROL FLAGS -

IF ANSWER IS YES "Y", THEN THE FOLLOWING QUESTION WILL BE ASKED.

PRINT ONLY 10 DATA ERRORS & CONTINUE

IF THIS QUESTION IS ANSWERED NO "N", THEN ALL ERRORS IN THE RX DATA BUFFER WILL BE PRINTED. A YES ANSWER "Y" WILL CAUSE ONLY THE FIRST 10 BYTES IN ERROR TO BE PRINTED.

270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311

3.0 ERROR INFORMATION

THIS PROGRAM HAS THREE TYPES OF ERROR CLASSIFICATIONS; SYSTEM FATAL, DEVICE FATAL, AND HARD ERRORS.

3.1 SYSTEM FATAL ERRORS

SYSTEM FATAL ERRORS ARE USED TO INDICATE THAT AN ERROR WAS DETECTED BY THE DIAGNOSTIC SUPERVISOR IN RELATION TO LOADING/ CONTROLLING THE DIAGNOSTIC PROCESS. WHEN A SYSTEM FATAL ERROR IS DETECTED THE UNIT IS USUALLY DROPPED.

THE CONTENT OF EACH ERROR IS SUCH THAT IT SHOULD BE SELF - EXPLANATORY. HOWEVER, THE MESSAGES UTILIZE SOME TERMS THAT ARE SPECIFIC TO THE FLOPPY DISK SUBSYSTEM, AND MAY REQUIRE SOME GETTING USE TO.

3.2 DEVICE FATAL ERRORS

DEVICE FATAL ERRORS ARE A RESULT OF:

1. REACHING A DEVICE FATAL THRESHOLD LEVEL ('DVTL'). THIS LEVEL IS INITIALLY SET=1, BUT MAY BE MODIFIED BY THE OPERATOR. AN 'DVTL' =1 WILL CAUSE 1 HARD ERROR TO BE CLASSIFIED A DEVICE FATAL ERROR.
2. AN ERROR THAT IS CONSIDERED FATAL TO THE DEVICE, BUT TESTING WILL CONTINUE.

3.3 HARD ERRORS

HARD ERRORS ARE A RESULT OF: A NON-RECOVERABLE ERROR

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
364

3.5 ERROR PRINTOUT FORMAT

EACH ERROR WILL BE PRINTED OUT USING THE STANDARD 'DRS' HEADER.

3.5.1 FUNCTION TESTS

THE SECOND LINE PRINTED OUT WILL GIVE THE TEST TITLE
THE THIRD LINE PRINTED OUT WILL IDENTIFY THE ERROR. IF IT
IS A CSR ERROR THE ACTUAL AND EXPECTED RESULTS WILL BE DISPLAYED.

EXAMPLE ERROR PRINTOUT:

CZRFB0 HRD ERR 00004 ON UNIT 01 TST 010 SUB 000 PC:003476
POSITIONING - FNC TST
CSR- ERROR
REG ACTUAL=005520
REG EXPECT=037566

3.5.2 LOGIC TESTS

THE SECOND AND THIRD LINES WILL BE PRINTED AS DESCRIBED FOR
THE FUNCTION TESTS.
DEPENDING ON THE TYPE OF ERROR ADDITIONAL ACTUAL AND EXPECTED
RESULTS WILL BE DISPLAYED. THEN THE TEST WILL CALL OUT WHICH ARE
THE MOST LIKELY FIELD REPLACEABLE UNITS 'FRU'S' THAT ARE
FAILING. ALL CURRENT DEVICE REGISTERS ARE THEN DISPLAYED,
INCLUDING A DATA BUFFER DUMP IF DATA WAS BAD.

EXAMPLE ERROR PRINTOUT:

CZRFB0 DEV FTL ERR 00019 ON UNIT 01 TST 024 SUB 000 PC:003476
WRD CNT INTEGRITY PRT:1 - LGC TST
WORD COUNT ERROR
REG ACTUAL=000020
REG EXPECT=000000

POSSIBLE FAILING 'FRU'S':
CONTROLLER - M7744
INTERFACE - M8256

UNIT#1 RXCSR=014440 RXESR=010040 CMD=000437 ->READ ERR CODE
ERROR CODE=230 ->WORD CNT OVF.
WORD CNT=020
CUR TRK DVO=76. CUR TRK DRV1= 0.
TARGET TRK =76. TARGET SEC =10. SOFT STAT=060 BAD TRK=15.

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

4.0 PERFORMANCE AND PROGRESS REPORTS

NONE

5.0 DEVICE INFORMATION

5.1 DEVICE REGISTERS

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	
RXCS:	ERR	INT	XM	XM	RX2		SID	DEN	TR	IE	DON	DRV	FUN	FUN	FUN	GO	! <FUNCTION>!
RXWC:	X	X	X	X	X	X	X	X	X								WORD COUNT
RXBA:	BUS ADDRESS																
RXES:	X	X	X	X	NXM	WC	SID	DRV	DRV	DEL	DSK	DEN	AC	INT	SID	CRC	
							OVF	#1	#1	RDY	DAT	DEN	ERR	LOW	DON	RDY	
RXTA:	X	X	X	X	X	X	X	X	X	0							TRACK ADDRESS
RXSA:	X	X	X	X	X	X	X	X	X	0	0	0					SECTOR ADDRESS
RXDB:	DATA BUFFER																

READ ERROR CODE REGISTERS - (SEE LABEL "XERUUT")

WORD	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	
#1	WORD COUNT								ERROR CODE								
#2	CURRENT TRACK DRV #1								CURRENT TRACK DRIVE #0								
#3	TARGET SECTOR								TARGET TRACK								
#4	BAD TRACK-ONLY VALID IF ERRCODE=150								UNT	DV1	HD	DVO	X	X	X	LCD	DEN
									SEL	DEN	LD	DEN					

5.2 DEVICE PROTOCOL

RX02 FUNCTIONAL PROCESS

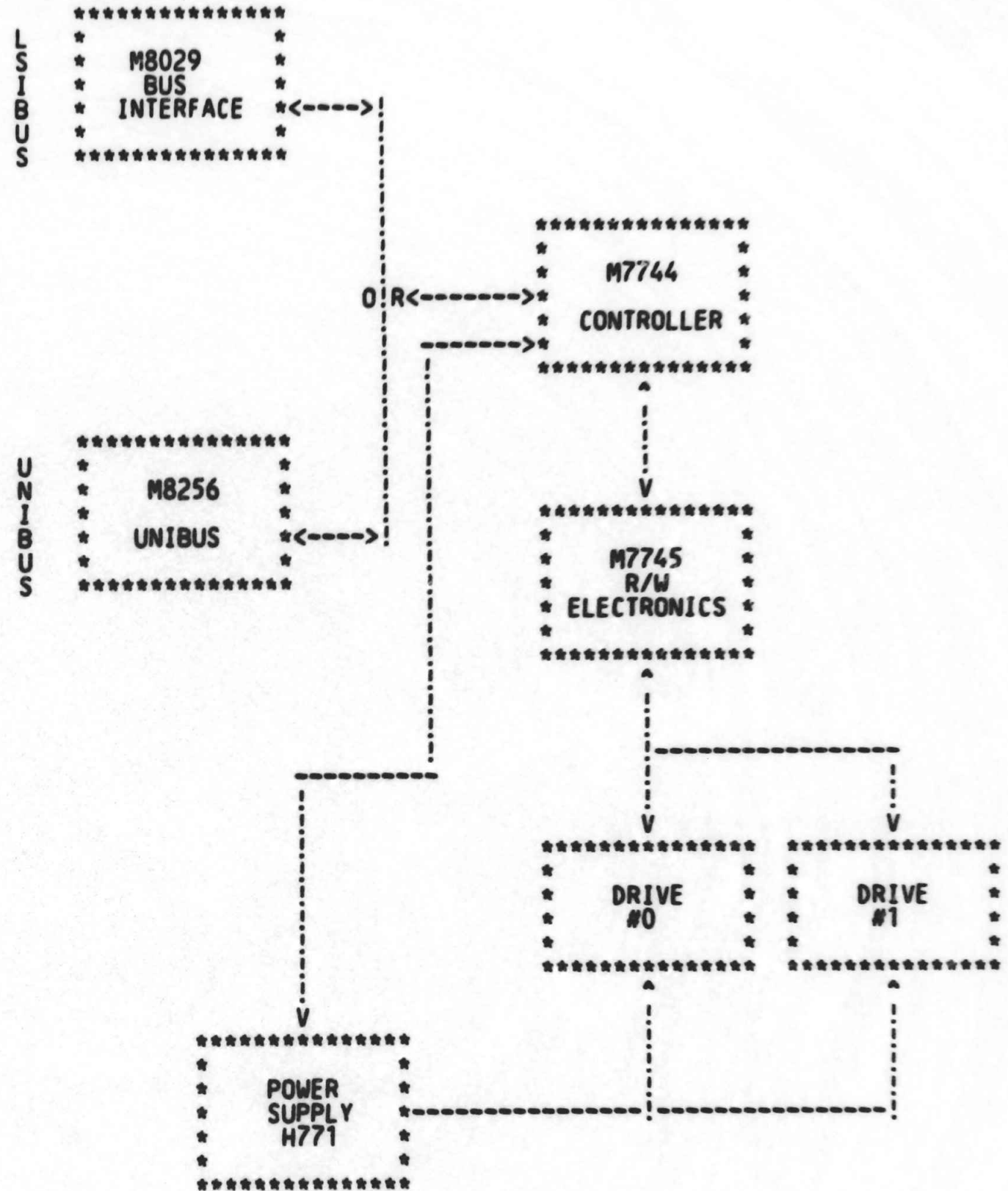
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459

FUNCTION CODE BIT # 3 2 1	FUNCTION	PROCEDURE (PROTOCOL)
0 0 0	FILL BUFFER	FUNCTION WORD --->TR--->WC--->TR--->BA--->D
0 0 1	EMPTY BUFFER	FUNCTION WORD --->TR--->WC--->TR--->BA--->D
0 1 0	WRITE SECTOR	FUNCTION WORD --->TR--->SA--->TR--->TA--->D
0 1 1	READ SECTOR	FUNCTION WORD --->TR--->SA--->TR--->TA--->D
1 0 0	SET DENSITY	FUNCTION WORD --->TR--->VW--->DONE
1 0 1	READ MAINT. STATUS	FUNCTION WORD --->DONE
1 1 0	WRITE SECTOR WITH DELETED DATA	FUNCTION WORD --->TR--->SA--->TR--->TA--->D
1 1 1	READ ERROR CODE	FUNCTION WORD --->TR--->BA--->DONE

TR = WAIT FOR TR BIT
 DONE = WAIT FOR DONE BIT
 BA = BUS ADDRESS (OUTPUT TO RX)
 VW = VERIFICATION WORD (OUTPUT TO RX)
 WC = WORD COUNT (OUTPUT TO RX)
 SA = SECTOR ADDRESS (OUTPUT TO RX)
 TA = TRACK ADDRESS (OUTPUT TO RX)

461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
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

5.3 DEVICE HARDWARE CONFIGURATION



515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568

6.0 TEST SUMMARIES

TEST 1 - INITIALIZE - FNC TST

TEST TO VERIFY THAT AN RX INITIALIZE WILL RETURN THE DEVICE TO A VALID STATE.

DESCRIPTION:

1. DO BUS INITIALIZE
2. IF RX ERR BIT IS SET REPORT ERROR
3. CALL PROGRAMMED INITIALIZE
4. IF RX ERR BIT IS SET REPORT ERROR

TEST 2 - READ ERROR CODE - FNC TST

TEST TO VERIFY THAT THE DEVICE WILL COMPLETE A READ ERROR CODE COMMAND WITHOUT ENCOUNTERING AN ERROR.

DESCRIPTION:

1. CALL PROGRAMMED INITIALIZE
2. IF RX ERR BIT IS SET REPORT ERROR
3. CALL READ ERROR CODE
4. IF RX ERR BIT IS SET REPORT ERROR

TEST 3 - FILL BUFFER - FNC TST

TEST TO VERIFY THE DEVICE BUFFER WILL FILL WITH NO RESULTING ERROR.

DESCRIPTION:

1. CALL FILL BUFFER
2. IF RX ERR BIT IS SET REPORT ERROR

TEST 4 - EMPTY BUFFER - FNC TST

TEST TO VERIFY THE DEVICE BUFFER WILL EMPTY WITHOUT ERRORS.

DESCRIPTION:

1. CALL EMPTY BUFFER
2. IF RX ERR BIT IS SET REPORT ERROR

TEST 5 - READ STATUS - FNC TST

TEST TO VERIFY THAT A DEVICE MAINTENANCE READ STATUS (RXES) COMMAND WILL EXECUTE WITHOUT ERROR.

DESCRIPTION:

1. CALL READ STATUS
2. IF RX ERR BIT IS SET REPORT ERROR

570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
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

TEST 6 - FILL & EMPTY BUFFER - FNC TST

TEST TO VERIFY THE DEVICE BUFFER DATA IS VALID AFTER A FILL/EMPTY BUFFER COMMAND SEQUENCE.

DESCRIPTION:

1. SETUP RANDOM DATA PATTERN
2. CALL FILL BUFFER
3. IF RX ERR BIT IS SET REPORT ERROR
4. CALL EMPTY BUFFER
5. IF RX ERR BIT IS SET REPORT ERROR
6. CALL DATA CHECK

TEST 7 - READ & WRITE SECTOR - FNC TST

TEST TO VERIFY THE DEVICE WILL READ AND WRITE IN BOTH DENSITIES WITHOUT AN ERROR.

DESCRIPTION:

1. SETUP TO DO TEST IN WRONG DENSITY
2. CALL WRITE SECTOR
3. IF RX ERR BIT IS NOT SET REPORT ERROR
4. CALL READ SECTOR
5. IF RX ERR BIT IS NOT SET REPORT ERROR
6. SETUP CORRECT DENSITY
7. CALL WRITE SECTOR
8. IF RX ERR BIT IS SET REPORT ERROR
9. CALL READ SECTOR
10. IF RX ERR BIT IS SET REPORT ERROR

TEST 8 - WRITE SECTOR DELETED DATA - FNC TST

TEST TO VERIFY THAT THE DEVICE WILL WRITE A DELETED DATA MARK ON THE DISKETTE WITHOUT ERROR.

DESCRIPTION:

1. SETUP TEST TO CORRECT DENSITY AND DELETED DATA MODE
2. CALL WRITE SECTOR DELETED DATA
3. IF RX ERR BIT IS SET REPORT ERROR
4. CALL READ SECTOR
5. IF RX ERR BIT IS SET REPORT ERROR
6. CLEAR DELETED DATA MODE
7. CALL WRITE SECTOR (CLEAR DELETED DATA MARK)
8. IF RX ERR BIT IS SET REPORT ERROR

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
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

TEST 9 - SET DENSITY - FNC TST

TEST TO VERIFY THE DEVICE WILL CHANGE DENSITIES WITHOUT INCURRING AN ERROR.

DESCRIPTION:

1. SETUP DENSITY = DISK DENSITY
2. CALL SET DENSITY
3. IF RX ERR BIT IS SET REPORT ERROR
4. CALL READ SECTOR
5. IF RX ERR BIT IS SET OR DENSITY NOT CORRECT REPORT ERROR
6. SETUP DENSITY = OPPOSITE DISK DENSITY
7. CALL SET DENSITY
8. IF RX ERR BIT IS SET REPORT ERROR
9. CALL READ SECTOR
10. IF RX ERR BIT IS SET OR DENSITY NOT CORRECT REPORT ERROR
11. SETUP DENSITY = DISK DENSITY
12. CALL SET DENSITY
13. IF RX ERR BIT IS SET REPORT ERROR

TEST 10 - POSITIONING - FNC TEST

TEST TO VERIFY THE DEVICE WILL CHANGE SECTORS AND TRACKS WITHOUT INCURRING AN ERROR.

DESCRIPTION:

1. SETUP RANDOM TRACK PATTERN AND DENSITY = DISK DENSITY
2. CALL GET A TRACK & SECTOR
3. CALL READ SECTOR
4. IF RX ERR BIT IS SET REPORT ERROR
5. DO 2. -> 4. UNTIL 76. TRACKS DONE

TEST 11 - CSR BITS - LGC TST

TEST TO VERIFY THAT THE READ/WRITE BITS OF THE CONTROL AND STATUS REGISTER CAN BE WRITTEN INTO AND READ AND OTHERWISE BEHAVE AS EXPECTED.

DESCRIPTION:

1. LOAD RX CSR WITH 1'S
2. CHECK & REPORT THAT ALL BITS THAT SHOULD SET, DO SET
3. LOAD RX CSR WITH 0'S
4. CHECK & REPORT THAT ALL BITS THAT SHOULD NOT BE SET, ARE NOT SET

TEST 12 - DBR BITS - LGC TST

TEST TO VERIFY THAT THE READ/WRITE BITS OF THE DATA BUFFER REGISTER CAN BE WRITTEN INTO AND READ AS EXPECTED.

DESCRIPTION:

1. WRITE RX DBR WITH ALL 1'S
2. CHECK & REPORT ALL BITS THAT SHOULD & SHOULD NOT BE SET

674
675
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710

3. WRITE RX DBR WITH ALL 0'S
4. CHECK & REPORT ALL BITS THAT SHOULD & SHOULD NOT BE SET

TEST 13 - CSR-DBR COMMON BYTE - LGC TST

TEST TO VERIFY THAT THE LOWER BYTE OF THE RXCS MAPS INTO THE RXDB AND THEREFORE CHECK WRITE ONLY BITS OF THE RXCS.

DESCRIPTION:

1. LOAD RX CSR LOW BYTE WITH ALL 1'S (EXCEPT BIT#0)
2. CHECK & REPORT IF RX DBR LOW BYTE NOT EQUAL TO ALL 1'S (EXCEPT BIT#0 & BIT#3)
3. LOAD RX CSR LOW BYTE WITH ALL 0'S
4. CHECK & REPORT IF RX DBR LOW BYTE NOT EQUAL TO ALL 0'S

TEST 14 - BUS INITIALIZE - LGC TST

TEST TO VERIFY THAT THE INTERFACE BOARD WILL COMPLETE A BUS INITIALIZE.

DESCRIPTION:

1. ISSUE BUS INITIALIZE
2. CHECK & REPORT IF ERROR BIT OR AC LOW BIT ARE SET OR IF DONE BIT IS NOT SET

TEST 15 - PROGRAMMED INITIALIZE - LGC TST

TEST TO VERIFY THAT THE INTERFACE BOARD WILL COMPLETE A PROGRAMMED INITIALIZE.

DESCRIPTION:

1. CALL PROGRAMMED INITIALIZE
2. CALL DEVICE STATE CHECK
3. CHECK & REPORT ERRORS

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

TEST 16 - POWER FAIL - LGC TST

TEST TO VERIFY THAT THE ACLOW CIRCUITS OPERATE AS EXPECTED.

DESCRIPTION:

1. IF MANNUAL INTERVENTION ALLOWED ASK OPERATOR TO POWER DOWN RX
2. IF POWERED DOWN, THEN CHECK & REPORT IF AC LOW BIT NOT SET
3. ASK OPERATOR TO POWER UP RX
4. IF POWERED UP, THEN INITIALIZE, CHECK & REPORT IF AC LOW BIT

TEST 17 - CONTROLLER-INTERFACE - LGC TST

TEST TO VERIFY THAT THE INTERFACE BOARD STATE SEQUENCER IS FUNCTIONAL.
ALSO TO VERIFY THE CONTROLLER-INTERFACE HANDSHAKE BY TRYING FUNCTIONS
WITH MINIMUM READ/WRITE BOARD INVOLVEMENT.

DESCIRPTION:

1. CALL READ ERROR CODE
2. IF ERROR, THEN REPORT ERROR
3. CALL FILL BUFFER
4. IF ERROR, THEN REPORT ERROR
5. CALL EMPTY BUFFER
6. IF ERROR, THEN REPORT ERROR
7. CALL READ STATUS
8. IF ERROR, THEN REPORT ERROR

741
742
743
744
745
746
747
748
749
750
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

TEST 18 - NPR - LGC TST

TEST TO VERIFY THAT THE NPR LOGIC WILL STORE A WORD IN MEMORY.

DESCRIPTION:

1. SETUP MEMORY LOCATION
2. CALL READ ERROR CODE (TO WRITE OVER LOCATION)
3. IF ERROR, THEN REPORT NPR ERROR
4. SETUP BUFFER AREAS BEGIN, END & END+1
5. CALL FILL BUFFER
6. IF ERROR, THEN REPORT ERROR
7. CALL EMPTY BUFFER
8. IF ERROR, THEN REPORT ERROR
9. CHECK BUFFER AREAS BEGIN & END SHOULD CHANGE & END+1 SHOULD NOT, REPORT AS NPR ERROR, IF CONDITIONS NOT MET

TEST 19 - NPR NON-EXISTENT MEM - LGC TST

TEST TO VERIFY THAT THE NPR NON-EXISTEND MEMORY LOGIC WILL TIME OUT WHEN GIVEN AN ILLEGAL ADDRESS.

DESCRIPTION:

1. SETUP BUS TRAPS AND NONEXISTANT MEMORY ADDRESS
2. CALL READ ERROR CODE
3. IF RX CSR ERROR BIT OR RX ESR NXM BIT NOT SET, THEN CALL ERRO
4. CALL INITIALIZE (CLEAR RX ERROR)
5. CLEAR BUS TRAP VECTOR

TEST 20 - INTERRUPT - LGC TST

TEST TO VERIFY THAT THE INTERRUPTS CAN BE SET AND THAT THE DEVICE RESPONDS AS EXPECTED.

DESCRIPTION:

1. SET PROCESSOR PRIORITY = 7 (NO INTERRUPTS)
2. SET RX INTERRUPT BIT & SETUP LOWER PRIORITY
3. CALL WATCH TO LOWER PROCESSOR PRIORITY & WAIT FOR INTERRUPT
4. CALL ERROR IF DID NOT INTERRUPT
5. CLEAR RX INTERRUPT BIT

TEST 21 - PRIORITY LVL - LGC TST

TEST TO VERIFY THAT THE DEVICE PRIORITY IS SET TO THE CORRECT LEVEL.

DESCRIPTION:

1. SETUP PROCESSOR PRIORITY = 7 (NO INTERRUPTS)
2. DO SET PROCESSOR PRIORITY
3. SET RX INTERRUPT BIT
4. IF INTERRUPT OCCURED, THEN CHECK LEVEL & REPORT IF PROCSSOR PRIORITY NOT LOWER THAN RX
5. IF INTERRUPT DID NOT OCCUR, THEN SETUP NEXT LOWER PROCESSOR PRIORITY & START AT 2. AGAIN

797

799
800
801
802
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

TEST 22 - INITIALIZE CONTROL - LGC TST

TEST TO VERIFY THAT THE CONTROLLER WILL COMPLETE INITIALIZE.

DESCRIPTION:

1. CALL PROGRAMMED INITIALIZE
2. IF ERROR, THEN REPORT ERROR

TEST 23 - DATA BUF INTEGRITY - LGC TST

TEST TO VERIFY ALL BITS OF DATA BUFFER, VARIOUS PATTERNS WILL BE USED.

DESCRIPTION:

1. SETUP RANDOM DATA PATTERN
2. CALL FILL BUFFER
3. IF ERROR, THEN REPORT ERROR
4. CALL EMPTY BUFFER
5. IF ERROR, THEN REPORT ERROR
6. CALL DATA CHECK
7. SETUP NEW DATA PATTERN
8. DO 2. -> 7. UNTIL ALL DATA PATTERNS DONE

TEST 24 - WRD CNT INTEGRITY - LGC TST

TEST TO VERIFY ALL BITS OF WORD COUNT REGISTER AND CHECK THAT EXCEEDING THE WORD COUNT FOR DISKETTE DENSITY WILL BE DETECTED.

DESCRIPTION:

1. SETUP BUFFER LENGTH = 128.
2. CALL FILL BUFFER
3. IF ERROR, THEN REPORT ERROR
4. CALL READ ERROR CODE
5. IF ERROR, THEN REPORT ERROR
6. IF RX WORD COUNT NOT = 0, THEN CALL ERROR
7. DECREMENT WORD COUNT TO RX, DO 2. -> 6. UNTIL WORD COUNT TO R IS = 0

TEST 25 - CONTROLLER-READ*WRITE ELECT - LGC TST

TEST TO VERIFY MINIMAL CONTROLLER BOARD-READ/WRITE ELECTRONICS BOARD INTERFACE VIA INITIALIZE OF A SELECTED DRIVE.

DESCRIPTION:

1. CALL PROGRAMMED INITIALIZE
2. IF ERROR, THEN REPORT ERROR
3. CALL READ ERROR CODE
4. IF ERROR, THEN REPORT ERROR

853
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

TEST 26 - READ SECTOR-PRT:1 - LGC TST

TEST TO VERIFY THAT THE CONTROLLER WILL COMPLETE A READ SECTOR IN BOTH DENSITIES AND RETURN A VALID ERROR CODE.

DESCRIPTION:

1. SETUP DENSITY = DISK DENSITY
2. CALL READ SECTOR
3. CALL READ ERROR CODE
4. IF ERROR, THEN REPORT ERROR
5. SETUP DENSITY = OPPOSITE DISK DENSITY
6. CALL READ SECTOR
7. CALL READ ERROR CODE
8. IF ERROR, THEN REPORT ERROR

TEST 27 - POSITIONING - LGC TST

TEST TO VERIFY THAT THE DRIVE WILL READ THE HEADERS ON ALL TRACKS OF THE DEIVE AS EXPECTED.

DESCRIPTION:

1. SETUP RANDOM TRACKS MODE
2. CALL GET A TRACK
3. CALL READ SECTOR
4. CALL READ ERROR CODE
5. IF TRACK OR OTHER ERROR, THEN REPORT ERROR
6. DO 2. -> 5. UNTIL 76. TRACKS DONE

TEST 28 - WRITE SECTOR-PRT:1 - LGC TST

TEST TO VERIFY THAT THE CONTROLLER WILL COMPLETE A WRITE SECTOR IN BOTH DENSITIES AND RETURN A VALID ERROR CODE.

DESCRIPTION:

1. SETUP DENSITY = DISK DENSITY
2. CALL WRITE SECTOR
3. IF ERROR, THEN REPORT ERROR
4. SETUP DENSITY = OPPOSITE DISK DENSITY
5. CALL WRITE SECTOR
6. IF NO DENSITY ERROR, THEN REPORT ERROR

899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955

TEST 29 - DELETED DATA WRITE PRT:1 - LGC TST

TEST TO VERIFY THAT THE DEVICE SET A DELETED DATA MARK ON THE DISKETTE HEADER AND RETRIEVE THIS DATA ON A KNOWN GOOD DISKETTE.

DESCRIPTION:

1. SETUP DENSITY = DISK DENSITY
2. SETUP DELETED DATA MODE
3. CALL WRITE SECTOR
4. IF ERROR, THEN REPORT ERROR
5. CALL READ SECTOR
6. IF RX CSR DELETED DATA BIT NOT SET, THEN REPORT ERROR
7. CLEAR DELETED DATA MODE
8. CALL WRITE SECTOR (CLEAR DELETED DATA MARK)

TEST 30 - SET DENSITY - LGC TST

TEST TO VERIFY THAT THE CONTROLLER WILL COMPLETE SET DENSITY IN BOTH DENSITIES. THE VALID WORD WILL ALSO BE CHECKED. ALSO TO VERIFY THAT THE DRIVE WILL READ IN BOTH DENSITIES, WITHOUT ERRORS.

DESCRIPTION:

1. GET & SAVE DISK DENSITY
2. SETUP DENSITY = SINGLE
3. CALL SET DENSITY
4. IF ERROR, THEN REPORT ERROR
5. SETUP INVALID KEY WORD = ASCII 'K'
6. CALL SET DENSITY
7. IF NO DENSITY ERROR, THEN REPORT ERROR
8. SETUP VALID KEY WORD = ASCII 'I'
9. SETUP DENSITY = DOUBLE
10. CALL SET DENSITY
11. IF ERROR, THEN REPORT ERROR
12. CHECK DISK DENSITY & REPORT IF NOT SET = DOUBLE
13. IF SAVED DISK DENSITY = DOUBLE, THEN SET DENSITY = SINGLE AND CALL SET DENSITY

TEST 31 - SECTOR ADR - LGC TST

TEST TO VERIFY THAT THE CONTROLLER WILL HANDLE ALL LEGAL AND ILLEGAL SECTOR ADDRESSES PROPERLY.

DESCRIPTION:

1. GET A SECTOR
2. CALL READ SECTOR
3. CALL READ ERROR CODE
4. IF SECTOR ADDRESS NOT = RX SECTOR ADDRESS OR OTHER ERROR, THEN REPORT ERROR
5. DO 1. -> 4. UNTIL ALL SECTORS DONE OR ERROR OCCURS
6. SETUP SECTOR = 0 (ILLEGAL SECTOR)
7. CALL READ SECTOR
8. CALL READ ERROR CODE
9. IF NO SECTOR ERROR OR IF OTHER ERROR, THEN REPORT ERROR

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
1001
1002
1003
1004
1005
1006

TEST 32 - TRACK ADR - LGC TST

TEST TO VERIFY THAT THE CONTROLLER WILL HANDLE ALL LEGAL AND ILLEGAL TRACK ADDRESSES PROPERLY.

DESCRIPTION:

1. GET A TRACK
2. CALL READ SECTOR
3. CALL READ ERROR CODE
4. IF TRACK ADDRESS NOT = RX TRACK ADDRESS, THEN CALL ERROR OR IF OTHER TRACK ERROR OCCURED, THEN CALL ERROR
5. DO 1. -> 4. UNTIL ALL TRACKS DONE OR FINI FLAG SET (COMMAND E
6. SETUP ILLEGAL TRACK
7. CALL READ SECTOR
8. CALL READ ERROR CODE
9. IF TRACK ADDRESS NOT = RX TRACK ADDRESS OR IF ERROR CODE NOT = 40 (TRACK > 76.), THEN CALL ERROR

TEST 33 - READ SECTOR-PRT:2 - LGC TST

TEST TO VERIFY THAT THE CONTROLLER WILL COMPLETE A READ SECTOR IN BOTH DENSITIES AND RETURN A VALID ERROR CODE.

DESCRIPTION:

1. SETUP DENSITY = DISK DENSITY
2. CALL READ SECTOR
3. CALL READ ERROR CODE
4. IF ERROR, THEN REPORT ERROR
5. SETUP DENSITY = OPPOSITE DISK DENSITY
6. CALL READ SECTOR
7. CALL READ ERROR CODE
8. IF NOT DENSITY ERROR, THEN REPORT ERROR

TEST 34 - WRITE SECTOR-PRT:2 - LGC TST

TEST TO VERIFY THAT THE CONTROLLER WILL COMPLETE A WRITE SECTOR IN BOTH DENSITIES AND RETURN A VALID ERROR CODE.

DESCRIPTION:

1. SETUP DENSITY = DISK DENSITY
2. CALL WRITE SECTOR
3. IF ERROR, THEN REPORT ERROR
4. SETUP DENSITY = OPPOSITE DISK DENSITY
5. CALL WRITE SECTOR
6. IF NOT DENSITY ERROR, THEN REPORT ERROR

1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063

TEST 35 - DELETED DATA WRITE PRT:2 - LGC TST

TEST TO VERIFY THAT THE DEVICE SET A DELETED DATA MARK ON THE DISKETTE HEADER AND RETRIEVE THIS DATA ON A KNOWN GOOD DISKETTE. THIS IS DONE IN OPPOSITE DENSITY OF PART: 1.

DESCRIPTION:

1. SETUP DENSITY = DISK DENSITY
2. SETUP DELETED DATA MODE
3. CALL WRITE SECTOR
4. CALL READ ERROR CODE
5. IF ERROR, THEN REPORT ERROR
6. CALL READ SECTOR
7. IF RX ESR DELETED DATA BIT NOT SET OR OTHER ERROR, THEN REPOR

TEST 36 - DISKETTE & DENSITY DATA CHECK - LGC TST

TEST TO VERIFY WITH A KNOWN GOOD DISKETTE THAT THE DEVICE WILL READ AND WRITE TO THE DISKETTE WITHOUT DATA ERRORS. BOTH DENSITIES WILL BE DONE.

DESCRIPTION:

1. SETUP DENSITY = DISK DENSITY
2. SETUP RANDOM DATA PATTERN
3. GET A TRACK & SECTOR
4. CALL FILL BUFFER
5. CALL WRITE SECTOR
6. SETUP TO CLEAR RX INTERNAL BUFFER
7. CALL FILL BUFFER
8. CALL READ SECTOR
9. CALL EMPTY BUFFER
10. CALL DATA CHECK
11. DO 3. -> 10. UNTIL AT LEAST ONE SECTOR OF EACH TRACK IS ACCES
12. SETUP DENSITY = OPPOSITE DISK DENSITY
13. CALL SET DENSITY
14. DO 3. -> 13. UNTIL BOTH DENSITIES DONE

1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1093
1094
1120
1122
1123
1124
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1145
1146
1147
1148
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171

002000
002000
002000
002122
002154
002162

7.0 LISTING INDEX

.NLIST SEQ,LD,BIN,CND

7.1 LISTING

%
.NLIST CND,MD,BEX,CND
.LIST SEQ,BIN
.TITLE PROGRAM HEADER AND TABLES
.SBTTL PROGRAM HEADER

.ENABL ABS,AMA
.=2000
.NLIST BEX,MD
BGNMOD

:++
: THE PROGRAM HEADER IS THE INTERFACE BETWEEN
: THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
:--

POINTER BGNSW,BGNSFT,BGNAU,BGNDU,ERRTBL,BGNSETUP

HEADER CZRFB0,0,0,170,0
DESCRIPT <RX02 FUNCTION-LOGIC TEST>
DEVTYP <RX02>

.SBTTL DISPATCH TABLE

:++
: THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
: IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
:--

DISPATCH 36

.REPT 0

.ENDR

1179
1180
1181
1182
1183
1184
1185 002274
1186 002276 177170
1187 002300 000264
1188 002302 000000
1189 002304 000000
1190 002306 000005
1196 002310
1197
1200
1201
1202
1203
1204
1205 002310
1206 002310 000000
1207 002312 177777
1208 002314 000004
1209 002316
1211
1212
1213
1214
1215
1216
1217 002316
1218 002320 000001
1219 002322 000000
1220 002324 000001
1221 002326 000000
1222 002330 000000
1223 002332 000020
1224 002334 000000
1225 002336 000114
1226 002340 000001
1227 002342 000032
1228 002344 160000
1229 002346 000000
1236 002350
1237
1238
1239
1240
1241
1242
1243
1244
1245 002350

.SBTTL DEFAULT HARDWARE P-TABLE

++
: THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
: THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE
: IS IDENTICAL TO THE STRUCTURE OF THE HARDWARE P-TABLES.
--

```

BGNHW      DFPTBL
.WORD      177170      :UNIBUS ADDRESS
.WORD      264        :VECTOR ADDRESS
.WORD      0          :DRIVE #
.WORD      0          :FUTURE EXPANSION
.WORD      5          :BR LEVEL #'S
ENDHW
    
```

.SBTTL LOAD DEVICE PROTECTION

++
: LOAD DEVICE PROTECTION TABLE - USED TO CHECK HARDWARE P-TABLE
: AGAINST LOAD DEVICE.
--

```

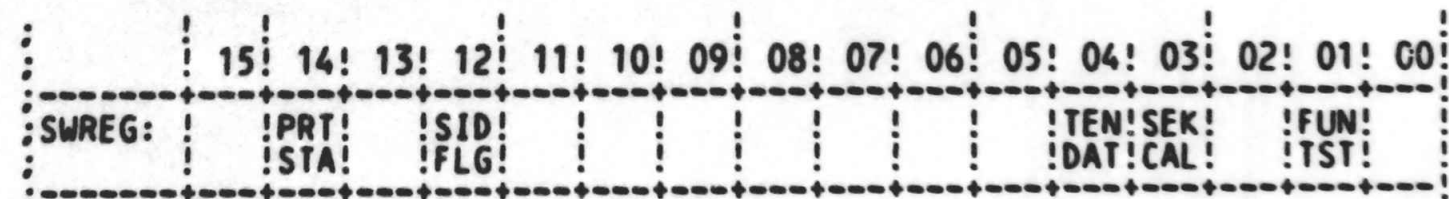
BGNPROT
.WORD      0          :P-TABLE OFFSET->CSR
.WORD      -1        :P-TABLE OFFSET->VECTOR-DON'T CARE
.WORD      4          :P-TABLE OFFSET->DRIVE
ENDPROT
    
```

.SBTTL SOFTWARE P-TABLE

++
: THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
: PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
--

```

          BGNSW      SFPTBL
DVTL:    .WORD      1          :HARD ERROR->DEVICE FATAL THRESHOLD LEVEL
          .WORD      0          :CONTROL WORD FOR SOFTWARE P-TABLES
TSTMOD:: .WORD      1          :TEST MODE
TSTPAT:: .WORD      0          :TEST PATTERN #
TRKSEQ:: .WORD      0          :TRACK SEQUENCE #
SWREG::  .WORD      20         :SOFTWARE SWITCH REG
OD::     .WORD      0          :OUTSIDE DIA. TRACK LIMIT
ID::     .WORD      114       :INSIDE DIA. TRACK LIMIT.
MINSFC:: .WORD      1          :MINIMUM SECTOR LIMIT
MAXSEC:: .WORD      32         :MAXIMUM SECTOR LIMIT
NXMADR:: .WORD      160000    :NON-EXISTENT MEMORY-ADR
XADBIT:: .WORD      0          :EXTENDED ADDRESS BITS
ENDSW
    
```



ENDMOD

1259
1260
1286
1296
1297
1298
1344
1362
1368
1397
1403
1415
1487
1493
1523
1529
1559 002350
1560
1561
1562
1563
1564
1565
1566 002350
(1)
(1)
(1)
(1) 100000
(1) 040000
(1) 020000
(1) 010000
(1) 004000
(1) 002000
(1) 001000
(1) 000400
(1) 000200
(1) 000100
(1) 000040
(1) 000020
(1) 000010
(1) 000004
(1) 000002
(1) 000001
(1)
(1) 001000
(1) 000400
(1) 000200
(1) 000100
(1) 000040
(1) 000020
(1) 000010
(1) 000004
(1) 000002
(1) 000001
(1)
(1)

.TITLE GLOBAL AREAS
.SBTTL GLOBAL EQUATES SECTION

:-< TEST MACROS >-
: THIS SECTION CONTAINS MACROS USED THROUGHOUT THE TESTS
:-----

BGNMOD

:+
: THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT
: ARE USED IN MORE THAN ONE TEST.
:--

EQUALS

:
: BIT DIFINITIONS

BIT15== 100000
BIT14== 40000
BIT13== 20000
BIT12== 10000
BIT11== 4000
BIT10== 2000
BIT09== 1000
BIT08== 400
BIT07== 200
BIT06== 100
BIT05== 40
BIT04== 20
BIT03== 10
BIT02== 4
BIT01== 2
BIT00== 1

:
BIT9== BIT09
BIT8== BIT08
BIT7== BIT07
BIT6== BIT06
BIT5== BIT05
BIT4== BIT04
BIT3== BIT03
BIT2== BIT02
BIT1== BIT01
BIT0== BIT00

:
: EVENT FLAG DEFINITIONS


```
(1) : EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
(1) :
(1) 000040 EF.START== 32. : START COMMAND WAS ISSUED
(1) 000037 EF.RESTART== 31. : RESTART COMMAND WAS ISSUED
(1) 000036 EF.CONTINUE== 30. : CONTINUE COMMAND WAS ISSUED
(1) 000035 EF.NEW== 29. : A NEW PASS HAS BEEN STARTED
(1) 000034 EF.PWR== 28. : A POWER-FAIL/POWER-UP OCCURRED
(1) :
(1) :
(1) : PRIORITY LEVEL DEFINITIONS
(1) :
(1) 000340 PRI07== 340
(1) 000300 PRI06== 300
(1) 000240 PRI05== 240
(1) 000200 PRI04== 200
(1) 000140 PRI03== 140
(1) 000100 PRI02== 100
(1) 000040 PRI01== 40
(1) 000000 PRI00== 0
(1) :
(1) : OPERATOR FLAG BITS
(1) :
(1) 000004 EVL== 4
(1) 000010 LOT== 10
(1) 000020 ADR== 20
(1) 000040 IDU== 40
(1) 000100 ISR== 100
(1) 000200 UAM== 200
(1) 000400 BOE== 400
(1) 001000 PNT== 1000
(1) 002000 PRI== 2000
(1) 004000 IXE== 4000
(1) 010000 IBE== 10000
(1) 020000 IER== 20000
(1) 040000 LOE== 40000
(1) 100000 HOE== 100000
```

```

1568                                     ;***** PROGRAM EQUIVALENTS *****
1569
1570         000010         DLDCMD      =          BIT3          ;DEL. DATA CMD BIT-----<CSR>
1571         100000         ERRBIT      =          BIT15         ;ERROR BIT-----<CSR>
1572         040000         RXINIT      =          BIT14         ;RXINIT BIT-----<CSR>
1573         004000         RX2BIT      =          BIT11         ;RX02 BIT-----<CSR>
1574         001000         SIDE1       =          BIT9          ;SIDE #1 BIT-----<CSR> & <CSR>
1575         000400         DENBIT      =          BIT8          ;DENSITY BIT-----<CSR>
1576         000200         TRBIT       =          BIT7          ;TR BIT-----<CSR>
1577         000040         DNBIT       =          BIT5          ;DONE BIT-----<CSR>
1578         000020         DRV1        =          BIT4          ;DRIVE 1-----<CSR>
1579         004000         NXMBIT      =          BIT11         ;NON-EXISTENT MEM-----<ESR>
1580         002000         WCOVRF      =          BIT10         ;WORD COUNT OVERFLOW----<ESR>
1581         000400         DRIVE1      =          BIT8          ;DRIVE #1 BIT-----<ESR>
1582         000200         DRVRDY      =          BIT7          ;DRIVE READY BIT-----<ESR>
1583         000100         DLDBIT      =          BIT6          ;DEL. DATA BIT-----<ESR>
1584         000040         DRVDEN      =          BIT5          ;DRIVE DENSITY-----<ESR>
1585         000020         DENERR      =          BIT4          ;DENSITY ERROR-----<ESR>
1586         000010         ACLOW       =          BIT3          ;AC LOW BIT-----<ESR>
1587         000004         INITDN      =          BIT2          ;INITIALIZE DONE BIT----<ESR>
1588         000002         SIDRDY      =          BIT1          ;SIDE READY BIT-----<ESR>
1589         000001         CRCERB      =          BIT0          ;CRC ERROR BIT-----<ESR>
1590         000004         BTRP4       =          4             ;BUS TRAP LOC #4 - TRAP HANDLER
1591         000006         BTRP6       =          6             ;BUS TRAP LOC #4 - PSW
1592         000001         LOGICT      =          BIT0          ;LOGIC TEST BIT-----<SWREG>
1593         000002         FUNCTT      =          BIT1          ;FUNCTION TEST BIT-----<SWREG>
1594         010000         SIDFLG      =          BIT12         ;SIDE FLAG SOFT-P TABLE-<SWREG>
1595         000400         ITK          =          BIT8          ;INITIALIZE TRACKS FLAG      <TKSCFG>
1596         001000         ISC          =          BIT9          ;INITIALIZE SECTORS FLAG     <TKSCFG>
1597         000001         STK          =          BIT0          ;SEQUENCE TRACKS FLAG       <TKSCFG>
1598         000002         SSC          =          BIT1          ;SEQUENCE SECTORS FLAG      <TKSCFG>
1599         000000         RTK          =          0             ;RANDOM TRACKS FLAG          <TKSCFG>
1600         000000         RSC          =          0             ;RANDOM SECTORS FLAG         <TKSCFG>
1601         000004         ILTK        =          BIT2          ;ILLEGAL TRACKS FLAG        <TKSCFG>
1602
1603                                     ;***** DEVICE COMMANDS *****
1604
1605         000000         FBCMD        =          0             ;FILL BUFFER CMD
1606         000002         EBCMD        =          2             ;EMPTY BUFFER CMD
1607         000004         WSCMD        =          4             ;WRITE SECTOR
1608         000006         RSCMD        =          6             ;READ SECTOR
1609         000010         SDCMD        =          10            ;SET DENSITY
1610         000012         STCMD        =          12            ;STATUS
1611         000014         WDDCMD       =          14            ;WRITE DELETED DATA CMD
1612         000016         RECCMD       =          16            ;READ ERROR CODE CMD

```



```

***** ERROR NUMBER EQUIVALENTS *****
1615
1616
1617      000002      WRTERR =      2.      :WRITE ERR      -HRD
1618      000003      RDERR  =      3.      :READ ERR       -HRD
1619      000004      CRCERR =      4.      :CRC ERR        -HRD
1620      000005      DATERR =      5.      :DATA ERR       -HRD
1621      000006      SEKERR =      6.      :SEEK ERR       -HRD
1622      000007      DLDERR =      7.      :DELETED DATA ERR -HRD
1623
1624
1625      000012      FILERR =     10.      :FILL BUFFER ERR -HRD
1626      000013      EMPERR =     11.      :EMPTY BUFFER ERR -HRD
1627      000014      INTNDN =     12.      :INTERRUPT, NO DONE ERR -HRD
1628      000015      DNNINT =     13.      :DONE, NO INTERRUPT ERR -HRD
1629      000016      ERRNST =     14.      :ERROR NOT SET ERR -HRD
1630      000017      ILLERC =     15.      :ILLEGAL ERROR CODE -HRD
1631      000020      DENDSK =     16.      :DENSITY OF DISK-NOT ERR -HRD
1632      000021      RECERR =     17.      :READ ERROR CODE ERR -HRD
1633
1634
1635      000023      WCERR  =     19.      :WORD COUNT ERROR -DVCFTL
1636      000024      SDRDYE =     20.      :SIDE READY      -DVCFTL
1637      000025      DVRDYE =     21.      :DRIVE READY     -DVCFTL
1638      000026      SIDWRG =     22.      :SIDE WRONG      -DVCFTL
1639      000027      DRVWRG =     23.      :DRIVE WRONG     -DVCFTL
1640      000030      DENERR =     24.      :DENSITY ERR     -DVCFTL
1641      000031      DENMIX =     25.      :DENSITY MIXED ON DISK ERR -DVCFTL
1642      000032      DLDTER =     26.      :DELETED DATA ERR -DVCFTL
1643      000033      CSRERR =     27.      :RXCSR-ERR      -DVCFTL
1644      000034      DBRERR =     28.      :RXESR-ERR      -DVCFTL
1645      000035      STDNER =     29.      :SET DENSITY ERR -DVCFTL
1646      000036      SDKYWD =     30.      :SET DENSITY KEYWORD (VARIFY) -DVCFTL
1647      000037      ACLOWD =     31.      :AC LOW          -DVCFTL
1648      000040      ALGO2E =     32.      :ALGO2 ERROR    -DVCFTL
1649      000041      TRKAER =     33.      :TRACK ADDRESS  -DVCFTL
1650      000042      SECAER =     34.      :SECTOR ADDRESS -DVCFTL
1651
1652
1653      000050      ACLOWF =     40.      :AC LOW FATAL ERR -SYSFTL
1654      000051      WCOVFE =     41.      :WORD COUNT OVERFLOW ERR -SYSFTL
1655      000052      NXMERR =     42.      :NON-EXISTENT MEMORY ERR -SYSFTL
1656      000053      NPRERR =     43.      :NPR LOGIC ERR  -SYSFTL
1657      000054      PRILEV =     44.      :PRIORITY LEVEL ERR -SYSFTL
1658      000055      DATABF =     45.      :DATA BUFFER INTEG ERR -SYSFTL
1659      000056      HDSFDG =     46.      :HARDWARE SELF DIAG ERR -SYSFTL
1660      000057      NOTRBT =     47.      :"TR" BIT TIME OUT ERR -SYSFTL
1661      000060      NODNBT =     48.      :"DONE" BIT TIBIT TIME OUT ERR -SYSFTL
1662      000061      NOITDB =     49.      :NO "INIT DONE" BIT ERR -SYSFTL
1663      000062      NOITDP =     50.      :NO PROG "INIT DONE" BIT ERR -SYSFTL
1664      000063      DNNOTR =     51.      :"DONE" BIT, NO "TR" BIT -SYSFTL
1665

```

```

1668
1669      000000
1670      000002
1671      000004
1672      000006
1673      000010
1674      000012
1675      000014
1676      000016
1677      000020
1678      000022
1679      000024
1680      000026
1681      000030
1682      000032
1683      000034
1684
1685
1686
1687      000001
1688      000002
1689      000004
1690      000010
1691      000020
1692      000040
1693      000100
1694      000200
1695      001000
1696      002000
1697      004000
1698      010000
1699      020000
1700      040000
1701      100000
1702
1703
1704
1705      000001
1706      000002
1707      000004
1708      000010
1709      000100
1710      000200
1711      000400
1712      010000
1713      040000
1714      100000
1715
1716
1717
1718      000020
1719      004000
1720      002000
1721

```

```

:***** FRU CALLOUT MESSAGE EQUIVALENTS *****
INTERF = 0 :INTERFACE=0
CONTRL = 2*1. :FRUM1 :CONTROLLER
RWELEC = 2*2. :FRUM2 :R-W ELECTRONICS
PHYDRV = 2*3. :FRUM3 :PHYSICAL DRIVE
CABLES = 2*4. :FRUM4 :CABLES
POWRSP = 2*5. :FRUM5 :POWER SUPPLY
DISKET = 2*6. :FRUM6 :DISKETTE
INTFSW = 2*7. :FRUM7 :INTERFACE SWITCHES
NPRJPR = 2*8. :FRUM8 :NPR JUMPER
CONTSW = 2*9. :FRUM9 :CONTROLLER SWITCHES
INTFCB = 2*10. :FRUM10 :INTERFACE CABLE
DOOROP = 2*11. :FRUM11 :DOOR OPEN
DISKSP = 2*12. :FRUM12 :DISK SPINNING-DRIVE BELT
MOTOR = 2*13. :FRUM13 :MOTOR AC POWER NOT ROTATING
NOPWR = 2*14. :FRUM14 :POWER CORD, BLOWN FUSE, DRIVE POWER
:CONNECTOR POWER SUPPLY FAULT.

:***** TEST FLAGS REGISTER EQUIVALENTS (FLAGST) *****
REGCK = BIT0 :REGISTER CHECK
DDCFLG = BIT1 :DOUBLE DENSITY CONTROL FLAG (DD=1)
DATCK = BIT2 :DATA CHECK
DLPDN = BIT3 :DO LOOP DONE
EMBUFF = BIT4 :EMPTY BUFFER-<USED BY DATA CHECK>
FUNTST = BIT5 :FUNCTION TEST FLAG
HDRPRT = BIT6 :ERROR CALL HEADER PRINT
RECFLG = BIT7 :READ ERROR CODE FLAG
TRKDON = BIT9 :TRACK DONE
SECDON = BIT10 :SECTOR DONE
NEGST = BIT11 :NEGATIVE TEST FLAG
ILLGAL = BIT12 :ILLEGAL FLAG
CKERR = BIT13 :CHECK ERROR WORDS FLAG
HRDERR = BIT14 :HARD ERROR
ERRFLG = BIT15 :ERROR

:***** PROGRAM/PRINT FLAGS REGISTER EQUIV (FLAGSP) *****
TKPRT = BIT0 :TRACKS PRINT
SCPRT = BIT1 :SECTORS PRINT
RGPRT = BIT2 :REGISTERS PRINT
PROPRT = BIT3 :PROTOCOL LEVEL PRINT
HDRPRT = BIT6 :HEADER PRINT
RECTST = BIT7 :ERROR CODE TEST (INVOKE ERROR CODE)
LSIFLG = BIT8 :LSI FLAG
FONZFG = BIT12 :FONZ FLAG
RESFLG = BIT14 :RESTART FLAG
STAFLG = BIT15 :START FLAG

:***** "SYS ERR" & "TYP ERR" REGISTER EQUIVALENTS *****
CMDERR = BIT4 :COMMAND ERROR
DVFERR = BIT11 :DEVICE FATAL ERROR
SYFERR = BIT10 :SYSTEM FATAL ERROR

```


1735
1736
1737
1738
1739
1740
1741
1742
1747
1748 002350 000000
1749 002352 000000
1750 002354 000000
1751 002356 000000
1752 002360 000000
1753 002362 000000
1754 002364 000000
1755 002366 000000
1756 002370 000000
1757 002372 000000
1758 002374 000000
1759 002376 000000
1760
1761 002400 000000
1762 002402 000000
1763 002404 000000
1764 002406 000000
1765 002410 000000
1766 002412 000000
1767 002414 000000
1768 002416 000000
1769 002420 000000
1770
1771 002422 000000
1772 002424 000000
1773 002426 000000
1774 002430 000000
1775 002432 000000
1776 002434 000000
1777 002436 000000
1778 002440 000000
1779
1780
1781
1782
1783
1784
1785
1786 002442 000
1787 002443 000
1788 002444 000
1789 002445 000
1790 002446 000
1791 002447 000
1792 002450 000
1793 002451 000
1794

.SBTTL GLOBAL DATA SECTION

++
: THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
: IN MORE THAN ONE TEST.

: STORAGE FOR DEVICE REGISTERS

```

-----
RXCS:  .WORD  0      ;UNIT BUS ADR-CSR          <UUT *>
RXDB:  .WORD  0      ;UNIT BUS ADR-DBR        <UUT *>
VECT:  .WORD  0      ;UNIT VECTOR             <UUT *>
RXPRI: .WORD  0      ;PRIORITY FOR DEVICE INTERRUPTS <UUT *>
EMPADR: .WORD  0      ;EMPTY BUFFER ADDRESS
FILADR: .WORD  0      ;FILL BUFFER ADDRESS
RECADR: .WORD  0      ;READ ERROR CODE ADDRESS
EXTADR: .WORD  0      ;EXTENDED ADDRESS (BITS: #12 & #13)
WDCNT:  .WORD  0      ;WORD COUNT
VARIFY: .WORD  0      ;VARIFY WORD
TRACK:  .WORD  0      ;TRACK ADR
SECTOR: .WORD  0      ;SECTOR ADR
-----
CMD:    .WORD  0      ;COMMAND WORD-TO DEVICE
DELDAT: .WORD  0      ;DELETED DATA FLAG & WORD <CMD>
INTERT: .WORD  0      ;INTERRUPT WORD           <CMD>
DRIVE:  .WORD  0      ;DRIVE WORD               <CMD*>
SIDE:   .WORD  0      ;SIDE WORD                <CMD*>
DENSITY: .WORD  0     ;DENSITY CONTROL WORD    <CMD>
DENSTA: .WORD  0     ;DENSITY STATUS WORD-DRIVE DENSITY
PRIORT:  .WORD  0     ;PRIORITY OF INTERRUPT HANDLER-WATCH DOG
DRVOFF: .WORD  0     ;DRIVE BYTE OFFSET
-----
ERRCMD: .WORD  0     ;ERROR COMMAND
LCMD:   .WORD  0     ;LAST COMMAND
LRXCSR: .WORD  0     ;LAST RX CSR STORAGE
LRXESR: .WORD  0     ;LAST RX ESR STORAGE
RXCSR:  .WORD  0     ;RX CSR STORAGE
RXESR:  .WORD  0     ;RX ESR STORAGE
REGEXP: .WORD  0     ;REGISTER EXPECTED
REGACT: .WORD  0     ;REGISTER ACTUAL
-----

```

* = INFO FROM HARDWARE P-TABLES

.SBTTL - READ ERROR CODE BUFFER

```

-----
XERUUT: .BYTE  0     ;ERROR CODE UUT
WC:     .BYTE  0     ;WORD COUNT UUT
CTKO:   .BYTE  0     ;CUR TRK DRV#0
CTK1:   .BYTE  0     ;CUR TRK DRV#1
TTRK:   .BYTE  0     ;TARGET TRACK
TSEC:   .BYTE  0     ;TARGET SECTOR
SFTSTS: .BYTE  0     ;MICRO CODE SOFT STATUS
BTRK:   .BYTE  0     ;BAD TRACK ADR (ONLY APPLIES IF ERR CODE = 150)
-----

```


1796
1797 002452 000000
1798 002454 000000
1799 002456 000000
1800 002460 000000
1801 002462 000000
1802 002464 000000
1803
1804 002466 000000
1805 002470 000000
1806 002472 000000
1807 002474 000012
1808
1809 002476 000000
1810 002500 000000
1811 002502 000004
1812 002504 000000
1813 002506 000000
1814 002510 000000
1815
1816 002512 000000
1817 002514 000
1818 002515 000
1819
1828 002516
(1) 002516 000000
(1) 002520 000000
(1) 002522 000000
(1) 002524 000000

```

-----
ABORT: .WORD 0 ;ABORT FLAG ; TEST
FIN: .WORD 0 ;FINI COMMAND FLAG ; ERROR
SYSERR: .WORD 0 ;SYSTEM ERROR ;
TYPERR: .WORD 0 ;TYPE ERROR ; &
REERN: .WORD 0 ;READ ERROR CODE-ERROR NUMBER ; STATU
NGTSE: .WORD 0 ;NEG TEST EXPECTED ERROR ; INFO
-----
TSTID: .WORD 0 ;TEST IDENT WORD ; TEST
TCMDCT: .WORD 0 ;TEST COMMAND COUNTER ;
PROTCT: .WORD 0 ;PROTOCOL COUNT ; INFO
DNWTMT: .WORD 12 ;DONE WAIT MULTIPLIER ;
-----
FLAGST: .WORD 0 ;SOFTWARE TEST FLAGS -> SEE BELOW ; TEST
FLAGSP: .WORD 0 ;SOFTWARE PROG/PRT FLAGS-->SEE BELOW ;
FLGDRS: .WORD 4 ;FLAGS FROM 'DRS' ; CONTR
TTEMP1: .WORD 0 ;TEST TEMP 1 ;
TSAVE1: .WORD 0 ;TEST SAVE 1 ; FLAGS
TKSCFG: .WORD 0 ;TRACK & SECTORS FLAGS --> SEE BELOW ;
-----
UNTPRT: .WORD 0 ;UNIT #-PRINT ; DEVIC
DRVPR: .BYTE 0 ;DRIVE #-PRINT ; PRINT
SIDPR: .BYTE 0 ;SIDE #-PRINT ;
-----
ERRTBL
ERRTYP:: .WORD 0
ERRNBR:: .WORD 0
ERRMSG:: .WORD 0
ERRBLK:: .WORD 0

```

1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853

```

***** SOFTWARE REGISTER DEFINITIONS *****
-----
      15! 14! 13! 12! 11! 10! 09! 08! 07! 06! 05! 04! 03! 02! 01! 00!
-----
TYPERR: BIT! NOT! NO! NO! ERR! ERR! EMP! ERR! -! MIS! UNK! CMD! DAT! SUM! SEK! CRC!
      SET! ITR! DON!      ERR!      ERR!
-----
SYSERR: UNR!      DEN!      DVF! SYF! -----! TO! RDY! RDY! -----! FUNCTION
      ERR!      ERR!      ERR! ERR! SID! DRV! ERR! ERR! ERR! FUN! INT! CAUSING
      ERROR
-----
FLAGST: ERR! HRD! CK! ILL! NEG! SEC! TRK! RTY! REC! HDR! FUN! EMB! DLP! DAT! DDC! REG!
      FLG! ERR! ERR! GAL! TST! DON! DON! ENB! FLG! PRT! TST! UFF! DN! CK! FLG! CK!
-----
FLAGSP: STA! RES!      LSI! REC! HDR!      PRO! RG! SC! TK!
      FLG! FLG!      FLG! TST! PRT!      PRT! PRT! PRT! PRT!
-----
TKSCFG:      I! I!      IL! S! S!
      SC! TK!      TK! SC! TK!
-----
:NOTE: RXXX IS REFERENCE FOR FURTHER EXPANSION

```

1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1886
1887
1888
1889
1890
1897
1898

.SBTTL GLOBAL TEXT SECTION

:++
: THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
: MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
: MORE THAN ONE TEST.
:--

:
: NAMES OF DEVICES SUPPORTED BY PROGRAM
:

:
: FORMAT STATEMENTS USED IN PRINT CALLS
:

1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918 002526
1919 002526 004737 002500
1920 002532
1921
1922 002534
1923 002534 004737 002570
1924 002540
1925
1926 002542
1927 002542 004737 002612
1928 002546
1929
1930 002550
1931 002566 000207
1932
1933 002570
1934 002610 000207
1935
1936 002612
1937 002634 000207
1938
1939 002636
1940 002662 000207
1941
1942 002664
1943 002712 000207
1944
1945 002714
1946 002732 000207
1947
1948 002734
1949 002754 000207
1950
1951 002756
1952 003000 000207
1953
1954 003002
1955 003026 000207
1956
1957 003030
1958 003056 000207
1959

.SBTTL GLOBAL ERROR REPORT SECTION

```

:++
: THE GLOBAL ERROR REPORT SECTION CONTAINS THE PRINTB AND PRINTX CALLS
: THAT ARE USED IN MORE THAN ONE TEST. IT ALSO INCLUDES THE ASCII MESSAGES
: THAT ARE USED BY THE PRINTB AND PRINTX CALLS..
:--

```

```

-----
:          BGNMSG  PRTB0
:          CALL    PRTB0S
:          ENDMSG
-----
:          BGNMSG  PRTB1
:          CALL    PRTB1S
:          ENDMSG
-----
:          BGNMSG  PRTB2
:          CALL    PRTB2S
:          ENDMSG
-----
: PRTB0S: PRINTB  R1
:          RETURN                                ;RETURN
-----
: PRTB1S: PRINTB  R1,R2
:          RETURN                                ;RETURN
-----
: PRTB2S: PRINTB  R1,R2,R3
:          RETURN                                ;RETURN
-----
: PRTB3S: PRINTB  R1,R2,R3,R4
:          RETURN                                ;RETURN
-----
: PRTB4S: PRINTB  R1,R2,R3,R4,R5
:          RETURN                                ;RETURN
-----
: PRTX0S: PRINTX  R1
:          RETURN
-----
: PRTX1S: PRINTX  R1,R2
:          RETURN
-----
: PRTX2S: PRINTX  R1,R2,R3
:          RETURN
-----
: PRTX3S: PRINTX  R1,R2,R3,R4
:          RETURN
-----
: PRTX4S: PRINTX  R1,R2,R3,R4,R5
:          RETURN
-----

```


1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006

```
.SBTTL - MOD U.ERR.ERR - ERROR
-----
BGNSUB      ERR
            IF ERR NBR NOT=0                [F]
            THEN-SET ERR SAVE = ERR NUMBER
            CALL ERROR NUMBER EVALUATION
            SETUP ERROR BLOCK CODE ADDRESS
            CALL PRINT ERROR TYPE
            IF PROGRAM FLAGS-PRT REGS ??? SET [J]
            THEN-IF ERRNBR=CSR ERROR        [I]
            :                               : THEN-CALL PRINT REGS
            :                               :   ENDIF
            :                               :   ENDIF
            IF COMMAND ERROR SET IN TYPERR  [B]
            : THEN-CALL PRINT COMMAND ERROR
            :                               :   ENDIF
            IF FUNCTION TEST NOT SET        [A]
            : THEN-IF PRINT FLAGS=REGS PRINT [E]
            :                               : THEN-CALL REGISTERS PRINT
            :                               :   ENDIF
            :                               :   IF PRINT FLAG=SECTOR PRINT [G]
            :                               :   : THEN-CALL SECTOR PRINT
            :                               :   :   ENDIF
            :                               :   IF PRINT FLAG=TRACK PRINT [C]
            :                               :   : THEN-CALL TRACKS PRINT
            :                               :   :   ENDIF
            :                               :   CALL PRINT FRU
            :                               :   CALL PRINT UNIT STATUS
            :                               :   ELSE-IF SWITCH REGISTER BIT #14 SET [D]
            :                               :   : THEN-CALL PRINT UNIT STATUS
            :                               :   :   ENDIF
            :                               :   ENDIF
            :                               :   IF ERR SAVE = ERR OLD [K]
            :                               :   : THEN - INCREMENT ERROR CTR
            :                               :   :   IF ERROR CTR = 10 ERRORS [L]
            :                               :   :   : THEN - SET ABORT = 20
            :                               :   :   :   ENDIF
            :                               :   :   ELSE - SET ERR OLD = ERR SAVE
            :                               :   :   :   CLEAR ERR SAVE
            :                               :   :   :   CLEAR ERR CTR
            :                               :   :   :   ENDIF
            :                               :   :   ENDIF
            :                               :   CALL CLEAR ERRORS
            :                               :   ENDIF
            :                               :   ENDIF
ENDSUB
-----
```


2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116

003344 000240
003346 022737 000047 002520
003354 101003
003356 005037 002516
003362 000416
003364 022737 000023 002520
003372 101004
003374 012737 000001 002516
003402 000406
003404 005737 002520
003410 001403
003412 012737 000002 002516
003420 022737 000002 002516
003426 001017
003430 032737 000004 002502
003436 001413
003440 005237 003472
003444 023737 002320 003472
003452 101005
003454 012737 000001 002516
003462 005037 003472
003466 000240
003470 000207
003472 000000

SBTTL - MOD U.SFT.ENV - ERROR NUMBER EVALUATION

```

-----
BGNSUB
IF ERR NBR > 39. [A]
: THEN-SET SYSTEM FATAL ERR->ERRTYP
: ELSE
: IF ERR NBR > 19. [B]
: : THEN-SET DEVICE FATAL ERR->ERRTYP
: : ELSE
: : IF ERR NBR > 0. [D]
: : : THEN-SET HARD ERR->ERRTYP
: : : ENDF
: : ENDF
: ENDF
ENDIF
IF ERRTYP=HARD ERROR [F]
: THEN-
: IF EVAL SET IN DRS FLAGS [G]
: : THEN-INCREMENT HARD ERR THRESHOLD LEVEL
: : IF HARD ERR THRESHOLD LEVEL=SET LEVEL [H]
: : : THEN-SET DEV FATAL ERR->ERRTYP
: : ENDF
: ENDF
ENDIF
ENDSUB
-----

```

```

-----
ERNBEV: NOP
IAENV: CMP #39.,ERRNBR ;IF ERR NBR > 39.
: BHI IBENV ;THEN
: CLR ERRTYP ;SET ERRTYP=SYS FTL
: BR IFENV ;BR TO IF 'F'
IBENV: CMP #19.,ERRNBR ;IF ERR NBR > 19.
: BHI IBENV ;THEN
: MOV #1,ERRTYP ;SET ERRTYP=DVC FTL
: BR IFENV ;BR TO IF 'F'
IDENV: TST ERRNBR ;IF ERR NBR > 0
: BEQ IFENV ;THEN
: MOV #2,ERRTYP ;SET ERRTYP=HARD ERR
IFENV: CMP #2,ERRTYP ;IF ERRTYP = HARD ERR
: BNE EFENV ;THEN
IGENV: BIT #BIT2,FLGDRS ;IF EVAL IN DRS FLAGS
: BEQ EFENV ;SET, THEN
: INC HETLCT ;INCREMENT HARD ERR THRESHOLD LEVEL CTR
IHENV: CMP DVTL,HETLCT ;IF DEVICE FTL THRES LVL=SFT LEV
: BHI EFENV ;THEN
: MOV #1,ERRTYP ;SET ERRTYP=DEV FTL ERR
: CLR HETLCT ;CLEAR HARD ERR THRES LVL CTR
EFENV: NOP
XERNBE: RETURN ;RETURN
-----
HETLCT: 0 ;HARD ERROR THRESHOLD LEVEL CTR
-----

```


2119
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214

003474 000240
003476
003500 000207

003502
003502 013701 002520
003506 006301
003510 000240
003512 016101 003534
003516 004737 002550
003522 013701 003534
003526 004737 002550
003532

```

.SBTTL - MOD U.PRT.PET - PRINT ERROR TYPE
-----
:BGNSUB
:CALL ERROR - REVC
:ENDSUB
-----
PRTY: NOP      ;
      ERROR    ;
      RETURN   ;RETURN
-----
.SBTTL - MOD U.ERR.IDT - GET & PRINT ERROR IDENTIFICATION MESSAGE
-----
: BGNMSG      ERIDENT
: LET R1=ERROR #
: DOUBLE R1 FOR ADDRESSING - MESSAGE OFFSET
: LET R1=ERR MSG TABLE ADD + MESSAGE OFFSET
: PRINT BASIC R1
: PUT MESSAGE TABLE ADDRESS IN R1
: PRINT BASIC R1
: ENDMSG
-----
: BGNMSG      ERIDNT
: MOV         ERRNBR,R1      ;GET ERR #
: ASL        R1              ;DOUBLE IT FOR ADDRESSING
: NOP
: MOV         ERMSTB(R1),R1  ;GET ERR MSG ADR FROM TABLE
: CALL       PRTBOS          ;CALL PRINT BASIC NO ARG
: MOV         ERMSTB,R1     ;GET REST OF ERR MSG FROM TABLE
: CALL       PRTBOS          ;CALL PRINT BASIC NO ARG
: ENDMSG
-----

```

			ERROR MESSAGE TABLE	ERROR DESCRIPTION	-CLASS
2217			:		
2218			ERMSTB:	.WORD ERMS0 :ERROR :	
2219	003534	003706		.WORD ERMS0 :ERR#0 :	
2220	003536	003706		.WORD ERMS2 :ERR#2 :WRITE	-HARD
2221	003540	003717		.WORD ERMS3 :ERR#3 :READ	-HARD
2222	003542	003731		.WORD ERMS4 :ERR#4 :CRC	-HARD
2223	003544	003742		.WORD ERMS5 :ERR#5 :DATA ERR	-HARD
2224	003546	003752		.WORD ERMS6 :ERR#6 :SEEK	-HARD
2225	003550	003763		.WORD ERMS7 :ERR#7 :DEL DATA	-HARD
2226	003552	003774		.WORD ERMS0 :ERR#8 :	-HARD
2227	003554	003706		.WORD ERMS0 :ERR#9 :	-HARD
2228	003556	003706		.WORD ERMS10 :ERR#10 :FILL BUFFER	-HARD
2229	003560	004015		.WORD ERMS11 :ERR#11 :EMPTY BUFFER	-HARD
2230	003562	004035		.WORD ERMS12 :ERR#12 :INTR-NO DONE	-HARD
2231	003564	004056		.WORD ERMS13 :ERR#13 :DONE-NO INTR	-HARD
2232	003566	004112		.WORD ERMS14 :ERR#14 :ERR-NOT SET	-HARD
2233	003570	004146		.WORD ERMS15 :ERR#15 :ILLEG ERR CODE	-HARD
2234	003572	004177		.WORD ERMS16 :ERR#16 :DISK DENSITY MIXED OR WRG	-HARD
2235	003574	004226		.WORD ERMS17 :ERR#17 :READ ERROR CODE-ERROR WRG	-HARD
2236	003576	004264		.WORD ERMS0 :ERR#18 :	
2237	003600	003706		.WORD ERMS19 :ERR#19 :WORD COUNT	
2238	003602	004314		.WORD ERMS20 :ERR#20 :SIDE NOT RDY	-DEVFTL
2239	003604	004333		.WORD ERMS21 :ERR#21 :DRIVE NOT RDY	-DEVFTL
2240	003606	004356		.WORD ERMS22 :ERR#22 :SIDE RESPONDING WRG	-DEVFTL
2241	003610	004402		.WORD ERMS23 :ERR#23 :DRIVE RESPONDING WRG	-DEVFTL
2242	003612	004434		.WORD ERMS24 :ERR#24 :DENSITY	-DEVFTL
2243	003614	004467		.WORD ERMS25 :ERR#25 :DENSITY DISK	-DEVFTL
2244	003616	004503		.WORD ERMS26 :ERR#26 :DEL DATA	-DEVFTL
2245	003620	004532		.WORD ERMS27 :ERR#27 :CSR	-DEVFTL
2246	003622	004553		.WORD ERMS28 :ERR#28 :DBR	-DEVFTL
2247	003624	004564		.WORD ERMS0 :ERR#29 :	-DEVFTL
2248	003626	003706		.WORD ERMS30 :ERR#30 :SET DENSITY KEYWORD	-DEVFTL
2249	003630	004625		.WORD ERMS31 :ERR#31 :AC LOW	-DEVFTL
2250	003632	004655		.WORD ERMS32 :ERR#32 :ALGO2	-DEVFTL
2251	003634	004670		.WORD ERMS33 :ERR#33 :TRACK ADDRESS	-DEVFTL
2252	003636	004711		.WORD ERMS34 :ERR#34 :SECTOR ADDRESS	-DEVFTL
2253	003640	004733		.WORD ERMS0 :ERR#35 :	
2254	003642	003706		.WORD ERMS0 :ERR#36 :	
2255	003644	003706		.WORD ERMS0 :ERR#37 :	
2256	003646	003706		.WORD ERMS0 :ERR#38 :	
2257	003650	003706		.WORD ERMS0 :ERR#39 :	
2258	003652	003706		.WORD ERMS40 :ERR#40 :AC LOW FATAL	-SYSFTL
2259	003654	004756		.WORD ERMS41 :ERR#41 :WORD COUNT OVERFLOW	-SYSFTL
2260	003656	004777		.WORD ERMS42 :ERR#42 :NON-EXISTENT MEM	-SYSFTL
2261	003660	005027		.WORD ERMS43 :ERR#43 :NON PROCESSOR REQUEST	-SYSFTL
2262	003662	005054		.WORD ERMS44 :ERR#44 :PRIORITY LEVEL	-SYSFTL
2263	003664	005102		.WORD ERMS45 :ERR#45 :DATA BUFFER INTEG	-SYSFTL
2264	003666	005125		.WORD ERMS46 :ERR#46 :HARDWARE SELF DIAG	-SYSFTL
2265	003670	005153		.WORD ERMS47 :ERR#47 :"TR" BIT TIME OUT	-SYSFTL
2266	003672	005202		.WORD ERMS48 :ERR#48 :"DONE" BIT TIME OUT	-SYSFTL
2267	003674	005230		.WORD ERMS49 :ERR#49 :NO BUS "INIT DONE"	-SYSFTL
2268	003676	005260		.WORD ERMS50 :ERR#50 :NO PROG "INIT DONE"	-SYSFTL
2269	003700	005307		.WORD ERMS51 :ERR#51 :"DONE" SET->WAITING FOR "TR" BIT	-SYSFTL
2270	003702	005337		.WORD ERMS0 :ERR#52 :	-SYSFTL
2271	003704	003706			
2272					

2275
2276
2277 003706 040445 042440 051122
2278
2279 003717 045 020101 053440
2280 003731 045 020101 051040
2281 003742 040445 020040 051103
2282 003752 040445 020040 040504
2283 003763 045 020101 051440
2284 003774 040445 020040 042504
2285
2286
2287 004015 045 020101 043040
2288 004035 045 020101 042440
2289 004056 040445 020040 047111
2290 004112 040445 020040 042042
2291 004146 040445 020040 051105
2292 004177 045 020101 044440
2293 004226 040445 020040 044504
2294 004264 040445 020040 042122
2295
2296 004314 040445 020040 047527
2297 004333 045 020101 051440
2298 004356 040445 020040 051104
2299 004402 040445 020040 051127
2300 004434 040445 020040 051127
2301 004467 045 020101 042040
2302 004503 045 020101 042040
2303 004532 040445 020040 042504
2304 004553 045 020101 041440
2305 004564 040445 020040 041104
2306 004575 045 020101 042040
2307 004625 045 020101 051440
2308 004655 045 020101 040440
2309 004670 040445 020040 046101
2310 004711 045 020101 052040
2311 004733 045 020101 051440
2312
2313
2314
2315
2316
2317 004756 040445 020040 041501
2318 004777 045 020101 053440
2319 005027 045 020101 047040
2320 005054 040445 020040 047516
2321 005102 040445 020040 051120
2322 005125 045 020101 042040
2323 005153 045 020101 044040
2324 005202 040445 020040 052042
2325 005230 040445 020040 042042
2326 005260 040445 020040 047516
2327 005307 045 020101 047040
2328 005337 045 020101 021040
2329
2330

.SBTTL - ERROR MESSAGES

ERMS0: .ASCIZ /%A ERROR/
:ERMS1: .ASCIZ /%A 1 ?/
ERMS2: .ASCIZ /%A WRITE/
ERMS3: .ASCIZ /%A READ/
ERMS4: .ASCIZ /%A CRC/
ERMS5: .ASCIZ /%A DATA/
ERMS6: .ASCIZ /%A SEEK/
ERMS7: .ASCIZ /%A DELETED DATA/
:ERMS8: .ASCIZ /%A 8 ?/
:ERMS9: .ASCIZ /%A 9 ?/
ERMS10: .ASCIZ /%A FILL BUFFER/
ERMS11: .ASCIZ /%A EMPTY BUFFER/
ERMS12: .ASCIZ /%A INTERRUPT-NO 'DONE' BIT/
ERMS13: .ASCIZ /%A 'DONE' BIT-NO INTERRUPT/
ERMS14: .ASCIZ /%A ERROR BIT NOT SET-ON/
ERMS15: .ASCIZ /%A ILLEGAL ERROR CODE/
ERMS16: .ASCIZ /%A DISK DENSITY MIXED OR WRG/
ERMS17: .ASCIZ /%A RD ERR CODE-ERR WRG/
:ERMS18: .ASCIZ /%A 18 ?/
ERMS19: .ASCIZ /%A WORD COUNT/
ERMS20: .ASCIZ /%A SIDE NOT READY/
ERMS21: .ASCIZ /%A DRIVE NOT READY/
ERMS22: .ASCIZ /%A WRONG SIDE RESPONDING/
ERMS23: .ASCIZ /%A WRONG DRIVE RESPONDING/
ERMS24: .ASCIZ /%A DENSITY/
ERMS25: .ASCIZ /%A DISK-MIXED DENSITY/
ERMS26: .ASCIZ /%A DELETED DATA/
ERMS27: .ASCIZ /%A CSR-/
ERMS28: .ASCIZ /%A DBR-/
ERMS29: .ASCIZ /%A DENSITY DID NOT SET/
ERMS30: .ASCIZ /%A SET DENSITY KEYWORD/
ERMS31: .ASCIZ /%A AC LOW/
ERMS32: .ASCIZ /%A ALGO2 FAILED/
ERMS33: .ASCIZ /%A TRACK ADDRESS/
ERMS34: .ASCIZ /%A SECTOR ADDRESS/
:ERMS35: .ASCIZ /%A 35 ?/
:ERMS36: .ASCIZ /%A 36 ?/
:ERMS37: .ASCIZ /%A 37 ?/
:ERMS38: .ASCIZ /%A 38 ?/
:ERMS39: .ASCIZ /%A 39 ?/
ERMS40: .ASCIZ /%A AC LOW FATAL/
ERMS41: .ASCIZ /%A WORD COUNT OVERFLOW/
ERMS42: .ASCIZ /%A NON-EXISTENT MEM/
ERMS43: .ASCIZ /%A NON-PROCESSOR REQ/
ERMS44: .ASCIZ /%A PRIORITY LEVEL/
ERMS45: .ASCIZ /%A DATA BUFFER INTEG/
ERMS46: .ASCIZ /%A HARDWARE SELF DIAG/
ERMS47: .ASCIZ /%A 'TR' BIT TIME OUT/
ERMS48: .ASCIZ /%A 'DONE' BIT TIME OUT/
ERMS49: .ASCIZ /%A NO BUS 'INIT DONE'/
ERMS50: .ASCIZ /%A NO PROG 'INIT DONE'/
ERMS51: .ASCIZ /%A 'DONE' SET->WAITING FOR 'TR' BIT/
:ERMS52: .ASCIZ /%A 52 ?/
:EVEN ;800. BYTES-->680.

2423 005612 000000
 2424 005614 006001
 2425 005616 006035
 2426 005620 006070
 2427 005622 006120
 2428 005624 006140
 2429 005626 006173
 2430 005630 006221
 2431 005632 006255
 2432 005634 006324
 2433 005636 006361
 2434 005640 006426
 2435 005642 006451
 2436 005644 006504
 2437 005646 006544
 2438
 2439
 2440
 2441
 2442 005650 047045 040445 020040
 2443 005713 045 030523 022461
 2444 005746 051445 030461 040445
 2445 006001 045 030523 022461
 2446 006035 045 030523 022461
 2447 006070 051445 030461 040445
 2448 006120 051445 030461 040445
 2449 006140 051445 030461 040445
 2450 006173 045 030523 022461
 2451 006221 045 030523 022461
 2452 006255 045 030523 022461
 2453 006324 051445 030461 040445
 2454 006361 045 030523 022461
 2455 006426 051445 030461 040445
 2456 006451 045 030523 022461
 2457 006504 051445 030461 040445
 2458 006544 051445 030461 040445
 2459
 2471

FRUTBM: .WORD 0
 .WORD FRUM1
 .WORD FRUM2
 .WORD FRUM3
 .WORD FRUM4
 .WORD FRUM5
 .WORD FRUM6
 .WORD FRUM7
 .WORD FRUM8
 .WORD FRUM9
 .WORD FRUM10
 .WORD FRUM11
 .WORD FRUM12
 .WORD FRUM13
 .WORD FRUM14

 .SBTTL - FRU MESSAGES

FRUM00: .ASCIZ /%N% POSSIBLE FAILING 'FRU'S': %N/
 FRUM0A: .ASCIZ /%S11% INTERFACE - M8029%N/
 FRUM0B: .ASCIZ /%S:1% INTERFACE - M8256%N/
 FRUM1: .ASCIZ /%S11% CONTROLLER - M7744%N/
 FRUM2: .ASCIZ /%S11% R-W ELECT - M7745%N/
 FRUM3: .ASCIZ /%S11% PHYSICAL DRIVE%N/
 FRUM4: .ASCIZ /%S11% CABLE%N/
 FRUM5: .ASCIZ /%S11% POWER SUPPLY-H771%N/
 FRUM6: .ASCIZ /%S11% BAD DISKETTE%N/
 FRUM7: .ASCIZ /%S11% INTERFACE SWITCHES%N/
 FRUM8: .ASCIZ /%S11% NPR JUMPER - PDP-11 BACKPLANE%N/
 FRUM9: .ASCIZ /%S11% CONTROLLER SWITCHES%N/
 FRUM10: .ASCIZ /%S11% INTERFACE->CONTROLLER CABLE%N/
 FRUM11: .ASCIZ /%S11% DOOR OPEN%N/
 FRUM12: .ASCIZ /%S11% BROKEN DRIVE BELT%N/
 FRUM13: .ASCIZ /%S11% DRIVE MOTOR - AC POWER%N/
 FRUM14: .ASCIZ /%S11% POWER CORD, BLOWN FUSE, DRIVE POWER, POWER SUPPLY %N/

.EVEN ;506. BYTES

2474
2475
2476 000000
2477 006640
2478 006640
 (2) 006640 000
 (2) 006641 002
 (2) 006642 024
 (2) 006643 377
2479
2480 006644
2481 006644
 (2) 006644 000
 (2) 006645 377
2482
2483 006646
2484 006646
 (2) 006646 002
 (2) 006647 000
 (2) 006650 377
2485
2486 006651
2487 006651
 (2) 006651 002
 (2) 006652 004
 (2) 006653 377
2488
2489 006654
2490 006654
 (2) 006654 002
 (2) 006655 024
 (2) 006656 377
2491
2492 006660

```

.SBTTL - FRU CALLOUT - PRESETUP FOR TESTS
-----
TN=0
INFCTL=TOFT0 ;INTERFACE & CONTROLLER
FRUTB 0,INTERF,CONTRL,INTFCB
TOFT0: .BYTE INTERF
        .BYTE CONTRL
        .BYTE INTFCB
        .BYTE -1
-----
INTONL=TOFT40 ;INTERFACE ONLY
FRUTB 40,INTERF
TOFT40: .BYTE INTERF
        .BYTE -1
-----
CTLINF=TOFT41 ;CONTROLLER & INTERFACE
FRUTB 41,CONTRL,INTERF
TOFT41: .BYTE CONTRL
        .BYTE INTERF
        .BYTE -1
-----
CTLRWE=TOFT42
FRUTB 42,CONTRL,RWELEC
TOFT42: .BYTE CONTRL
        .BYTE RWELEC
        .BYTE -1
-----
CTLONL=TOFT43
FRUTB 43,CONTRL,INTFCB
TOFT43: .BYTE CONTRL
        .BYTE INTFCB
        .BYTE -1
-----
.EVEN

```

ERROR CODE - FRU CALLOUT ADDRESS TABLE

2495		
2496		
2497	006660	000000
2498	006662	006736
2499	006664	006742
2500	006666	000000
2501	006670	006746
2502	006672	006751
2503	006674	000000
2504	006676	006755
2505	006700	000000
2506	006702	006763
2507	006704	006770
2508	006706	006776
2509	006710	000000
2510	006712	007002
2511	006714	007006
2512	006716	007012
2513	006720	007016
2514	006722	000000
2515	006724	007022
2516	006726	007025
2517	006730	007030
2518	006732	007035
2519	006734	177777

```

:-----:
:
:TOFTB: .WORD 0
:         .WORD TOFT1
:         .WORD TOFT2
:         .WORD 0
:         .WORD TOFT4
:         .WORD TOFT5
:         .WORD 0
:         .WORD TOFT7
:         .WORD 0
:         .WORD TOFT11
:         .WORD TOFT12
:         .WORD TOFT13
:         .WORD 0
:         .WORD TOFT15
:         .WORD TOFT16
:         .WORD TOFT17
:         .WORD TOFT20
:         .WORD 0
:         .WORD TOFT22
:         .WORD TOFT23
:         .WORD TOFT24
:         .WORD TOFT25
:         .WORD -1
:-----:

```

.SBTTL - FRU CALLOUT - PRESETUP FOR ERROR CODE

2520		
2521		
2522		
2523		
2524		000000
2525	006736	
(2)	006736	006
(2)	006737	002
(2)	006740	004
(2)	006741	377
2526	006742	
(2)	006742	006
(2)	006743	002
(2)	006744	004
(2)	006745	377
2527	006746	
(2)	006746	000
(2)	006747	002
(2)	006750	377
2528	006751	
(2)	006751	004
(2)	006752	006
(2)	006753	002
(2)	006754	377
2529	006755	
(2)	006755	014
(2)	006756	004
(2)	006757	006
(2)	006760	002
(2)	006761	000
(2)	006762	377

```

:-----:
:TN=0
:FRUTB 1,PHYDRV,CONTRL,RWELEC
:TOFT1: .BYTE PHYDRV
:         .BYTE CONTRL
:         .BYTE RWELEC
:         .BYTE -1
:FRUTB 2,PHYDRV,CONTRL,RWELEC
:TOFT2: .BYTE PHYDRV
:         .BYTE CONTRL
:         .BYTE RWELEC
:         .BYTE -1
:FRUTB 4,INTERF,CONTRL
:TOFT4: .BYTE INTERF
:         .BYTE CONTRL
:         .BYTE -1
:FRUTB 5,RWELEC,PHYDRV,CONTRL
:TOFT5: .BYTE RWELEC
:         .BYTE PHYDRV
:         .BYTE CONTRL
:         .BYTE -1
:FRUTB 7,DISKET,RWELEC,PHYDRV,CONTRL,INTERF
:TOFT7: .BYTE DISKET
:         .BYTE RWELEC
:         .BYTE PHYDRV
:         .BYTE CONTRL
:         .BYTE INTERF
:         .BYTE -1
:-----:

```

2530	006763	
(2)	006763	014
(2)	006764	004
(2)	006765	006
(2)	006766	002
(2)	006767	377
2531	006770	
(2)	006770	014
(2)	006771	004
(2)	006772	006
(2)	006773	002
(2)	006774	012
(2)	006775	377
2532	006776	
(2)	006776	014
(2)	006777	004
(2)	007000	002
(2)	007001	377
2533	007002	
(2)	007002	004
(2)	007003	006
(2)	007004	002
(2)	007005	377
2534	007006	
(2)	007006	004
(2)	007007	006
(2)	007010	002
(2)	007011	377
2535	007012	
(2)	007012	014
(2)	007013	004
(2)	007014	002
(2)	007015	377
2536	007016	
(2)	007016	014
(2)	007017	004
(2)	007020	002
(2)	007021	377
2537	007022	
(2)	007022	004
(2)	007023	002
(2)	007024	377
2538	007025	
(2)	007025	000
(2)	007026	002
(2)	007027	377
2539	007030	
(2)	007030	014
(2)	007031	002
(2)	007032	000
(2)	007033	004
(2)	007034	377
2540	007035	
(2)	007035	000
(2)	007036	002
(2)	007037	377

FRUTB	11,DISKET,RWELEC,PHYDRV,CONTRL
TOFT11:	.BYTE DISKET
	.BYTE RWELEC
	.BYTE PHYDRV
	.BYTE CONTRL
	.BYTE -1
FRUTB	12,DISKET,RWELEC,PHYDRV,CONTRL,POWRSP
TOFT12:	.BYTE DISKET
	.BYTE RWELEC
	.BYTE PHYDRV
	.BYTE CONTRL
	.BYTE POWRSP
	.BYTE -1
FRUTB	13,DISKET,RWELEC,CONTRL
TOFT13:	.BYTE DISKET
	.BYTE RWELEC
	.BYTE CONTRL
	.BYTE -1
FRUTB	15,RWELEC,PHYDRV,CONTRL
TOFT15:	.BYTE RWELEC
	.BYTE PHYDRV
	.BYTE CONTRL
	.BYTE -1
FRUTB	16,RWELEC,PHYDRV,CONTRL
TOFT16:	.BYTE RWELEC
	.BYTE PHYDRV
	.BYTE CONTRL
	.BYTE -1
FRUTB	17,DISKET,RWELEC,CONTRL
TOFT17:	.BYTE DISKET
	.BYTE RWELEC
	.BYTE CONTRL
	.BYTE -1
FRUTB	20,DISKET,RWELEC,CONTRL
TOFT20:	.BYTE DISKET
	.BYTE RWELEC
	.BYTE CONTRL
	.BYTE -1
FRUTB	22,RWELEC,CONTRL
TOFT22:	.BYTE RWELEC
	.BYTE CONTRL
	.BYTE -1
FRUTB	23,INTERF,CONTRL
TOFT23:	.BYTE INTERF
	.BYTE CONTRL
	.BYTE -1
FRUTB	24,DISKET,CONTRL,INTERF,RWELEC
TOFT24:	.BYTE DISKET
	.BYTE CONTRL
	.BYTE INTERF
	.BYTE RWELEC
	.BYTE -1
FRUTB	25,INTERF,CONTRL
TOFT25:	.BYTE INTERF
	.BYTE CONTRL
	.BYTE -1

GLOBAL AREAS
CZRFB.P11

MACY11 30(1046)
09-APR-82 15:14

12-APR-82
-

13:23 PAGE 37-2
FRU CALLOUT - PRESETUP FOR ERROR CODE

I 4

SEQ 0047

2541
2542

:-----
.EVEN

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

007040	000240		
007042	012701	007204	
007046	004737	002550	
007052	013702	002422	
007056	032702	040000	
007062	001405		
007064	012701	007452	
007070	012702	000012	
007074	000404		
007076	042702	177761	
007102	016201	007164	
007106	004737	002550	
007112	012701	007216	
007116	004737	002550	
007122	022702	000012	
007126	001415		
007130	032737	000010	002500
007136	001411		
007140	013702	002472	
007144	012701	007502	
007150	004737	002734	
007154	042737	000020	002460
007162	000207		

```

.SBTTL - MOD U.ERR.PCE - PRINT COMMAND ERROR
-----
BGNSUB
  SETUP & PRINT COMMAND FORMAT MESSAGE
  GET COMMAND
  IF INITIALIZE COMMAND
  : THEN-SETUP INIT COMMAND MSG
  : ELSE-CLEAR TOP BITS & GO BIT
  :   CLEAR TOP BITS & GO BIT
  :   GET COMMAND MSG ADDRESS FROM TABLE (INDEXED BY COMMAND)
  ENDIF
  CALL PRINT
  SETUP & PRINT END OF COMMAND ERROR
  IF PROTOCOL TYPE COMMAND
  : THEN-IF PRINT FLAGS=PRINT PROTOCOL SET
  :   THEN-SETUP & PRINT PROTOCOL ERR
  :   ENDIF
  ENDIF
ENDSUB
-----

PRTCDE: NOP          ;
        MOV          #CMFTMS,R1      ;SETUP COMMAND FORMAT MESSAGE
        CALL         PRTBOS         ;CALL PRINT BASIC-NO ARG
        MOV          ERRCMD,R2      ;GET COMMAND
IAPCE:  BIT          #BIT14,R2      ;IF INITIALIZE BIT
        BEQ          LAPCE          ;SET, THEN
        MOV          #CMDM8,R1      ;SET PROGRAMMED INIT MSG
        MOV          #12,R2         ;SET R2 TO SHOW COMMAND WITH NO PROTOCOL
        BR          EAPCE          ;BR TO END 'A'
LAPCE:  BIC          #177761,R2     ;CLEAR TOP BITS & GO BIT
        MOV          CMDMSG(R2),R1  ;GET COMMAND MSG ADR FROM TABLE
EAPCE:  CALL         PRTBOS         ;CALL PRINT BASIC-NO ARG
        MOV          #CMERMS,R1     ;SETUP "COMMAND ERR" MSG
        CALL         PRTBOS         ;CALL PRINT BASIC-NO ARG
IBPCE:  CMP          #12,R2         ;IF R2 CONTAINS PROTOCOL TYPE COMMAND
        BEQ          XPCE          ;THEN
ICPCE:  BIT          #PROPRT,FLAGSP ;IF PRINT PROTOCOL FLAG=FLAGSP
        BEQ          XPCE          ;SET, THEN
        MOV          PROTCT,R2      ;SETUP PRINT PROTOCOL CNT
        MOV          #CMDPE,R1      ;SETUP PRINT PROTOCOL ERR MSG
        CALL         PRTX1S         ;PRINT MSG
        BIC          #CMDERR,TYPERR ;CLEAR TYP ERR COMMAND ERROR
XPCE:   RETURN          ;RETURN
-----

```

```

2592
2593
2594 007164 007237
2595 007166 007255
2596 007170 007274
2597 007172 007313
2598 007174 007331
2599 007176 007347
2600 007200 007374
2601 007202 007430
2602 007204 047045 051445 022466
2603 007216 040445 041440 046517
2604 007237 045 043101 046111
2605 007255 045 042501 050115
2606 007274 040445 051127 052111
2607 007313 045 051101 040505
2608 007331 045 051501 052105
2609 007347 045 051101 040505
2610 007374 040445 051127 052111
2611 007430 040445 042522 042101
2612 007452 040445 051120 043517
2613 007502 047045 051445 022470
2614 007564
2615

```

.SBTTL - COMMAND ERROR MESSAGE TABLE

```

:-----:
CMDMSG: .WORD  CMDM0
        .WORD  CMDM1
        .WORD  CMDM2
        .WORD  CMDM3
        .WORD  CMDM4
        .WORD  CMDM5
        .WORD  CMDM6
        .WORD  CMDM7
CMFTMS: .ASCIZ  /%N%S6%A->/
CMERMS: .ASCIZ  /%A COMMAND ERROR/
CMDM0:  .ASCIZ  /%AFILL BUFFER/
CMDM1:  .ASCIZ  /%AEMPTY BUFFER/
CMDM2:  .ASCIZ  /%AWRITE SECTOR/
CMDM3:  .ASCIZ  /%AREAD SECTOR/
CMDM4:  .ASCIZ  /%ASET DENSITY/
CMDM5:  .ASCIZ  /%AREAD MAINT. STATUS/
CMDM6:  .ASCIZ  /%AWRITE SECTOR-DELETED DATA/
CMDM7:  .ASCIZ  /%AREAD ERROR CODE/
CMDM8:  .ASCIZ  /%APROGRAMMED INITIALIZE/
CMDPE:  .ASCIZ  /%N%S8%APROTOCOL FAILED-WAITING TO PASS WORD #%01/
        .EVEN
:-----:

```



```

2618
2619
2620
2621 007564 000240
2622 007566 012701 007620
2623 007572 013702 002440
2624 007576 013703 002436
2625 007602 004737 002612
2626 007606 005037 002440
2627 007612 005037 002436
2628 007616 000207
2629
2630 007620 047045 051445 022466
2631 007646 051445 022466 051101
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646 007674 000240
2647 007676 032737 000200 002476
2648 007704 001424
2649 007706 013702 002376
2650 007712 012701 010156
2651 007716 004737 002550
2652 007722 012701 007760
2653 007726 004737 002570
2654 007732 113702 002447
2655 007736 012701 010220
2656 007742 004737 002550
2657 007746 012701 007760
2658 007752 004737 002570
2659 007756 000207
2660
2661 007760 040445 051440 041505
2662 010002
2663

```

.SBTTL - MOD U.ERR.PRE - PRINT REGISTER ERROR

```

-----
PRTREG: NOP
MOV #PRTGMS,R1 ;SETUP REGISTER MESSAGE
MOV REGACT,R2 ;SETUP REG ACTUAL
MOV REGEXP,R3 ;SETUP REG EXPECTED
CALL PRTB2S ;CALL PRINT BASIC-2 ARG
CLR REGACT ;CLEAR OLD RESULTS
CLR REGEXP ;CLEAR OLD RESULTS
RETURN ;RETURN
-----

```

```

PRTGMS: .ASCII /%N%S6%AREG ACTUAL=%0%N/
.ASCIIZ /%S6%AREG EXPECT=%0%N/
.EVEN
-----

```

.SBTTL - MOD U.PRT.SCP - PRINT SECTORS

```

-----
BGNSUB
IF READ ERROR CODE FLAG SET
: THEN-SETUP PRINT EXPECTED SECTOR
: SETUP PRINT DEVICE SECTOR
: CALL PRINT
ENDIF
ENDSUB
-----

```

```

PRTSEC: NOP
IASCP: BIT #RECFLG,FLAGST ;IF READ ERROR CODE FLAG
BEQ XSCP ;SET, THEN
MOV SECTOR,R2 ;SETUP EXPECTED SECTOR
MOV #EXMS,R1 ;SETUP EXPECTED MSG
CALL PRTB0S ;CALL PRINT BASIC-0 ARG
MOV #ADSCMS,R1 ;SETUP SECTOR MSG
CALL PRTB1S ;CALL PRINT BASIC-1 ARG
MOVB TSEC,R2 ;SETUP DEVICE SECTOR
MOV #TGMS,R1 ;SETUP TARGET MSG
CALL PRTB0S ;CALL PRINT BASIC-0 ARG
MOV #ADSCMS,R1 ;SETUP SECTOR MSG
CALL PRTB1S ;CALL PRINT BASIC-1 ARG
XSCP: RETURN ;RETURN
-----

```

```

ADSCMS: .ASCIIZ /%A SECTOR=%D2%A./
.EVEN
-----

```

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

010002 004737 020600
010006 032737 000200 002476
010014 001445
010016 013702 002374
010022 012701 010156
010026 004737 002550
010032 012701 010257
010036 004737 002570
010042 005737 002406
010046 001403
010050 113702 002445
010054 000402
010056 113702 002444
010062 012701 010177
010066 004737 002550
010072 012701 010257
010076 004737 002570
010102 113702 002446
010106 012701 010220
010112 004737 002550
010116 012701 010257
010122 004737 002570
010126 000412
010130 013702 002374
010134 012701 010241
010140 004737 002550
010144 012701 010257
010150 004737 002570
010154 000207
010156 047045 051445 022466
010177 045 022516 033123
010220 047045 051445 022466
010241 045 022516 020101
010257 045 020101 051124
010300

```
.SBTTL - MOD U.PRT.TKP - PRINT TRACKS
-----
BGNSUB
CALL PRINT UNIT IDENT
IF READ ERROR CODE FLAG SET
: THEN-SETUP PRINT EXPECTED TRACK
: CALL PRINT 1 PARAMETER
: IF DRIVE #1 SELECTED
: : THEN-SETUP CURRENT TRACK DRV1-PRINT
: : ELSE-SETUP CURRENT TRACK DRV0-PRINT
: ENDIF
: CALL PRINT 1 PARAMETER
: SETUP PRINT DRIVE TARGET TRACK
: CALL PRINT 1 PARAMETER
: ELSE-SETUP PRINT ERROR ON TRACK
: CALL PRINT 1 PARAMETER
: ENDIF
ENDSUB
-----
PRTRK: CALL PRTDID ;CALL PRINT DRIVE IDENT
IATKP: BIT #RECFLG,FLAGST ;IF READ ERROR CODE FLAG
: BEQ LATKP ;FLAG SET, THEN
: MOV TRACK,R2 ;SETUP EXPECTED TRACK
: MOV #EXMS,R1 ;SETUP EXPECTED MSG
: CALL PRTBOS ;CALL PRINT BASIC-0 ARG
: MOV #ADTKMS,R1 ;SETUP PRINT TRACK
: CALL PRTB1S ;CALL PRINT BASIC-1 PAR.
IBTKP: TST DRIVE ;IF DRIVE #1
: BEQ LBTKP ;SELECTED, THEN
: MOVB CTK1,R2 ;SETUP CUR TRK D1-PRT
: BR EBTKP ;BR TO END 'B'
LBTKP: MOVB CTK0,R2 ;SETUP CUR TRK D0-PRT
EBTKP: MOV #CDMS,R1 ;SETUP DRIVE CURRENT MSG
: CALL PRTBOS ;CALL PRINT BASIC-0 ARG
: MOV #ADTKMS,R1 ;SETUP PRINT TRACK
: CALL PRTB1S ;CALL PRINT BASIC-1 PAR.
: MOVB TTRK,R2 ;SETUP TARGET TRACK
: MOV #TGMS,R1 ;SETUP TARGET MSG
: CALL PRTBOS ;CALL PRINT BASIC-0 ARG
: MOV #ADTKMS,R1 ;SETUP PRINT TRACK
: CALL PRTB1S ;CALL PRINT BASIC-1 PAR.
: JR XTKPRT ;BR TO EXIT
LATKP: MOV TRACK,R2 ;SETUP ERROR TRACK
: MOV #ERTKMS,R1 ;SETUP ERROR TRACK MSG
: CALL PRTBOS ;CALL PRINT BASIC-0 ARG
: MOV #ADTKMS,R1 ;SETUP PRINT TRACK
: CALL PRTB1S ;CALL PRINT BASIC-1 PAR.
XTKPRT: RETURN
-----
EXMS: .ASCIZ /%N%S6% EXPECTED/
CDMS: .ASCIZ /%N%S6% CUR DRV/
TGMS: .ASCIZ /%N%S6% TARGET/
ERTKMS: .ASCIZ /%N% ERROR ->/
ADTKMS: .ASCIZ /%A TRACK=%D2%. /
: .EVEN
-----
```

2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744

010300 000240
010302 005037 002442
010306 005037 002520
010312 005037 002516
010316 012737 003502 002524
010324 005037 002454
010330 005037 002452
010334 042737 100000 002476
010342 000207

.SBTTL - MOD U.ERR.CLE - CLEAR ERRORS

: BGNSUB
: CLEAR ERROR NUMBER
: CLEAR ERROR TYPE
: CLEAR ERROR BLOCK
: CLEAR FIN
: CLEAR ABORT
: ENDSUB

CLRERR: NOP ;
CLR XERUUT ; CLEAR READ ERR CODE WORD
CLR ERRNBR ; CLEAR ERROR NUMBER
CLR ERRTYP ; CLEAR ERROR TYPE
MOV #ERIDNT,ERRBLK ; CLEAR ERROR BLOCK
CLR FIN ; CLEAR FINI
CLR ABORT ; CLEAR ABORT
BIC #ERRFLG,FLAGST ; CLEAR FLAGST ERR FLAG
RETURN ; RETURN

2747
2748
2749
2750
2751
2752
2753
2813
2814
2815
2816
2823
2829
2836
2842
2849
2858
2866
2872
2873
2880
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908

010344 012700 000001
010350 063700 010432
010354 063700 010434
010360 042700 170000
010364 000241
010366 006100
010370 006100
010372 010037 010432
010376 005000
010400 013700 010434
010404 006000
010406 006000
010410 063700 010432
010414 042700 170000
010420 010037 010434
010424 010037 010436
010430 000207
010432 000000
010434 000000
010436 000000

.SBTTL GLOBAL SUBROUTINES SECTION

;++
: THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
: THAT ARE USED IN MORE THAN ONE TEST.
:--

.SBTTL - MOD U.1.0 - RANDOM GENERATOR

++
: FUNCTIONAL DESCRIPTION:
: SUBROUTINE TO GENERATE A RANDOM NUMBER
: INPUTS: NONE
: IMPLICIT INPUTS: NONE
: OUTPUTS: RANUM
: IMPLICIT OUTPUTS: NONE
: SUBORDINATE ROUTINES USED: NONE
: FUNCTIONAL SIDE EFFECTS: NONE
: CALLING SEQUENCE: SUB
:--

----- RANDOM GENERATOR -----

RANGEN: MOV #1,R0
ADD RAN1,R0
ADD RAN2,R0
BIC #170000,R0
CLC
ROL R0
ROL R0
MOV R0,RAN1
CLR R0
MOV RAN2,R0
ROR R0
ROR R0
ADD RAN1,R0
BIC #170000,R0
MOV R0,RAN2
MOV R0,RANUM
RTS PC

: RAN1: 0
: RAN2: 0
: RANUM: 0
:-----

2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953

010440	012737	040000	002400
010446	013777	002400	171674
010454	004737	012032	
010460	004737	011544	
010464	004737	012244	
010470	000207		
010472	012701	040000	
010476	010177	171646	
010502	004737	012032	
010506	000207		

```

.SBTTL - MOD U.DEV.INT - INITIALIZE DEVICE
:++
: FUNCTIONAL DESCRIPTION: SUBR TO SEND INITIALIZE TO DEVICE.-ERROR CK
: INPUTS: NONE
: IMPLICIT INPUTS: ERROR BIT
: OUTPUTS: DEVICE INITIALIZE
: IMPLICIT OUTPUTS:
: SUBORDINATE ROUTINES USED: COMMAND ERR CK, GET DEV. REGS, WAIT DONE
: FUNCTIONAL SIDE EFFECTS:
: CALLING SEQUENCE: SUBR
:--

```

```

-----
INITIAL: MOV #40000,CMD ;SET INT COMMAND
          MOV CMD,@RXCS ;INIT UNIT 0
          CALL AWDN ;GO AWAIT DONE
XINT:    CALL CDERCK ;CALL COMMAND ERROR CK
          CALL GETREG ;CALL GET DEV REGS
          RTS PC ;RETURN
-----

```

```

.SBTTL - MOD U.DEV.CLD - CLEAR DEVICE
:++
: FUNCTIONAL DESCRIPTION: SUBR TO SEND INIT TO DEVICE - NO ERROR CK
: INPUTS: NONE
: IMPLICIT INPUTS: NONE
: OUTPUTS: DEVICE INITIALIZE
: IMPLICIT OUTPUTS:
: SUBORDINATE ROUTINES USED: A WAIT 'DONE'
: FUNCTIONAL SIDE EFFECTS:
: CALLING SEQUENCE: SUBR
:--

```

```

-----
CLRDEV: MOV #40000,R1 ;SET INITIALIZE COMMAND
          MOV R1,@RXCS ;CLEAR DEVICE
          CALL AWDN ;AWAIT DONE
          RETURN ;RETURN
-----

```

2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989

010510	004737	011634	
010514	005737	002454	
010520	001035		
010522	012737	000001	011542
010530	004737	011502	
010534	053737	002366	002400
010542	013777	002400	171600
010550	004737	012110	
010554	005737	002454	
010560	001015		
010562	013777	002370	171562
010570	004737	012110	
010574	005737	002454	
010600	001005		
010602	013777	002362	171542
010610	004737	011610	
010614	004737	011544	
010620	004737	012244	
010624	000207		

```

.SBTTL - MOD U.DEV.FLB - FILL BUFFER
:++
: FUNCTIONAL DESCRIPTION: SIBR TO SEND FILL BUFFER COMMAND TO DEVICE.
: INPUTS: NONE
: IMPLICIT INPUTS: NONE
: OUTPUTS: FILL BUFFER TO RX
: IMPLICIT OUTPUTS:
: SUBORDINATE ROUTINES USED: SET COMMANDS, WAIT 'DONE', WAIT 'TR'
: FUNCTIONAL SIDE EFFECTS:
: CALLING SEQUENCE:
:--
:-----
FILBUF: CALL   DVDNCK           ;CALL DEVICE READY CHECK
          TST   FIN             ;IF FINI FLAG SET
          BNE   XFILBF         ;NOT SET, THEN
          MOV   #1,NCMD        ;SET FILL BUFFER COMMAND
          CALL  SETSCD         ;CALL SET SUBSYS COMMAND - MOD U.DEV.SSC
          BIS   EXTADR,CMD     ;SET EXT. ADR. BITS
          MOV   CMD,@RXCS     ;LOAD COMMAND
          CALL  AWTR           ;WAIT FOR 'TR'
IBFLB:  TST   FIN             ;IF FINI FLAG
          BNE   XFILBF         ;EQUALS ZERO THEN
          MOV   WDCNT,@RXDB    ;LOAD WORD COUNT FOR OUTPUT BUFFER
          CALL  AWTR           ;WAIT FOR 'TR'
ICFLB:  TST   FIN             ;IF FINI FLAG
          BNE   XFILBF         ;EQUALS ZERO THEN
          MOV   FILADR,@RXDB   ;LOAD BASE ADDRESS FOR OUTPUT BUFFER
          CALL  WAIT           ;WAIT FOR 'DONE' OR INTERRUPT
XFILBF: CALL  CDERCK          ;CALL COMMAND ERROR CHECK
          CALL  GETREG         ;CALL GET DEV REGS
          RTS   PC             ;RETURN
:-----

```


2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025

010626	004737	011634	
010632	005737	002454	
010636	001035		
010640	012737	000003	011542
010646	004737	011502	
010652	053737	002366	002400
010660	013777	002400	171462
010666	004737	012110	
010672	005737	002454	
010676	001015		
010700	013777	002370	171444
010706	004737	012110	
010712	005737	002454	
010716	001005		
010720	013777	002360	171424
010726	004737	011610	
010732	004737	011544	
010736	004737	012244	
010742	000207		

```

.SBTTL - MOD U.DEV.EMB - EMPTY BUFFER
:
:
: **
: FUNCTIONAL DESCRIPTION: SUBR TO SEND EMPTY BUFFER TO DEVICE.
: INPUTS: NONE
: IMPLICIT INPUTS: NONE
: OUTPUTS: EMPTY BUFFER TO RX
: IMPLICIT OUTPUTS:
: SUBORDINATE ROUTINES USED: SETUP COMMANDS, WAIT 'DONE', WAIT 'TR'
: FUNCTIONAL SIDE EFFECTS:
: CALLING SEQUENCE:
:
:--
:-----
EMPBUF: CALL DVDNCK          :CALL DEVICE READY CHECK
        TST  FIN             :IF FINI FLAG
        BNE  XEMPBF          :NOT SET, THEN
        MOV  #3,NCMD         :SET EMPTY BUFFER COMMAND
        CALL SETSCD          :CALL SET SUBSYS COMMAND
        BIS  EXTADR,CMD      :SET EXT. ADR. BITS
        MOV  CMD,@RXCS       :ELSE LOAD COMMAND
        CALL AWTR            :WAIT FOR 'TR' DO MOD U.TR
IBEMB:  TST  FIN             :IF FINI FLAG
        BNE  XEMPBF          :EQUALS ZERO
        MOV  WDCNT,@RXDB     :THEN LOAD WORD COUNT FOR INPUT BUFFER
        CALL AWTR            :WAIT FOR 'TR' DO MOD U.TR
ICEMB:  TST  FIN             :IF FINI FLAG
        BNE  XEMPBF          :EQUALS ZERO
        MOV  EMPADR,@RXDB    :THEN LOAD BASE ADDRESS FOR INPUT BUFFER
        CALL WAIT            :WAIT FOR 'DONE' OR INTERRUPT
XEMPBF: CALL CDERCK          :CALL COMMAND ERROR CHECK
        CALL GETREG          :CALL GET DEV REGS
        RTS  PC              :RETURN
:-----

```

3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061

010744	004737	011634	
010750	005737	002454	
010754	001037		
010756	012737	000005	011542
010764	053737	002402	011542
010772	004737	011502	
010776	004737	011462	
011002	013777	002400	171340
011010	004737	012110	
011014	005737	002454	
011020	001015		
011022	013777	002376	171322
011030	004737	012110	
011034	005737	002454	
011040	001005		
011042	013777	002374	171302
011050	004737	011610	
011054	004737	012244	
011060	000207		

```

.SBTTL - MOD U.DEV.WRT - WRITE SUBROUTINE
:
:
: *+
: FUNCTIONAL DESCRIPTION: SUBR TO SEND WRITE SECTOR TO DEVICE.
: INPUTS: NONE
: IMPLICIT INPUTS: DELETED DATA MODE
: OUTPUTS: WRITE SECTOR TO RX
: IMPLICIT OUTPUTS:
: SUBORDINATE ROUTINES USED: SETUP COMMANDS, WAIT 'DONE', WAIT 'TR'
: FUNCTIONAL SIDE EFFECTS:
: CALLING SEQUENCE:
:--
:-----
WRITE: CALL DVDNCK           :CALL DEVICE READY CHECK
      TST  FIN              :IF FINI FLAG
      BNE  XWRITE          :EQUALS ZERO THEN
      MOV  #5,NCMD         :SET TO WRITE SECTOR
      BIS  DELDAT,NCMD     :SETUP WRITE DELETED DATA, IF SET
      CALL SETSCD         :CALL SET SUBSYS COMMAND
      CALL SETDCD         :CALL SET DEVICE COMMAND - MOD U.DEV.CMD
WRITE1: MOV  CMD,@RXCS     :LOAD COMMAND
      CALL AWTR           :GO AWAIT TRANSFER READY 'TR'
IBWRT:  TST  FIN          :IF FINI FLAG
      BNE  XWRITE          :EQUALS ZERO THEN
      MOV  SECTOR,@RXDB   :LOAD SECTOR ADDRESS
      CALL AWTR           :GO AWAIT TRANSFER READY 'TR'
ICWRT:  TST  FIN          :IF FINI FLAG
      BNE  XWRITE          :EQUALS ZERO THEN
      MOV  TRACK,@RXDB   :LOAD TRACK ADDRESS
      CALL WAIT           :WAIT FOR INTERRUPT OR 'DONE'
XWRITE: CALL GETREG       :CALL GET DEV REGS
      RTS  PC              :RETURN
:-----

```

3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096

011062	004737	011634	
011066	005737	002454	
011072	001034		
011074	012737	000007	011542
011102	004737	011502	
011106	004737	011462	
011112	013777	002400	171230
011120	004737	012110	
011124	005737	002454	
011130	001015		
011132	013777	002376	171212
011140	004737	012110	
011144	005737	002454	
011150	001005		
011152	013777	002374	171172
011160	004737	011610	
011164	004737	012244	
011170	000207		

```

.SBTTL - MOD U.DEV.RED - READ SUBROUTINE
:++
: FUNCTIONAL DESCRIPTION: SUBR TO SEND READ SECTOR TO DEVICE.
: INPUTS: NONE
: IMPLICIT INPUTS: NONE
: OUTPUTS: READ SECTOR TO RX
: IMPLICIT OUTPUTS:
: SUBORDINATE ROUTINES USED: SETUP COMMANDS, WAIT 'DONE', WAIT 'TR'
: FUNCTIONAL SIDE EFFECTS:
: CALLING SEQUENCE:
:--
:-----
READ:  CALL  DVDNCK      :CALL DEVICE READY CHECK
      TST   FIN        :IF FINI FLAG
      BNE   XREAD      :EQUALS ZERO, THEN
      MOV   #7,NCMD    :SET READ COMMAND
      CALL  SETSCD     :CALL SET SUBSYS COMMAND
      CALL  SETDCD     :CALL SET DEVICE COMMAND - MOD U.DEV.SDC
READ1: MOV   CMD,@RXCS :LOAD COMMAND
      CALL  AWTR       :GO AWAIT TRANSFER READY
IBRED: TST   FIN        :IF FINI FLAG
      BNE   XREAD      :EQUALS ZERO, THEN
      MOV   SECTOR,@RXDB :LOAD SECTOR ADDRESS
      CALL  AWTR       :GO AWAIT TRANSFER READY
ICRED: TST   FIN        :IF FINI FLAG
      BNE   XREAD      :EQUALS ZERO, THEN
      MOV   TRACK,@RXDB :LOAD TRACK ADDRESS
      CALL  WAIT       :WAIT FOR INTERRUPT OR 'DONE'
XREAD: CALL  GETREG    :CALL GET DEV REGS
      RETURN          :RETURN
:-----

```


3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128

011172	004737	011634	
011176	005737	002454	
011202	001024		
011204	012737	000011	011542
011212	004737	011502	
011216	004737	011462	
011222	013777	002400	171120
011230	004737	012110	
011234	005737	002454	
011240	001005		
011242	013777	002372	171102
011250	004737	011610	
011254	004737	011544	
011260	004737	012244	
011264	000207		

```

.SBTTL - MOD U.DEV.SDN - SET DENSITY
:
:++
: FUNCTIONAL DESCRIPTION: SUBR TO SEND SET DENSITY COMMAND TO DEVICE.
: INPUTS: NONE
: IMPLICIT INPUTS: DENSITY
: OUTPUTS: SET DENSITY TO RX
: IMPLICIT OUTPUTS:
: SUBORDINATE ROUTINES USED: SETUP COMMANDS, WAIT 'DONE', WAIT 'TR'
: FUNCTIONAL SIDE EFFECTS:
: CALLING SEQUENCE:
:--
:-----
SETDN: CALL DVDNCK ;CALL DEVICE READY CHECK
      TST FIN ;IF FINI FLAG
      BNE XSETDN ;NOT SET, THEN
      MOV #11,NCMD ;SETUP DENSITY COMMAND
      CALL SETSCD ;CALL SET SUBSYS COMMAND
      CALL SETDCD ;CALL SET DEVICE COMMAND - MOD U.DEV.SDC
      MOV CMD,@RXCS ;SEND COMMAND
      CALL AWTR ;GO AWAIT 'TR'
IBSDN: TST FIN ;IF FINI FLAG IS
      BNE XSETDN ;ZERO
      MOV VARIFY,@RXDB ;SEND VARIFY WORD
      CALL WAIT ;WAIT FOR 'DONE' OR INTERRUPT
XSETDN: CALL CDERCK ;CALL COMMAND ERROR CHECK
        CALL GETREG ;CALL GET DEV REGS
        RTS PC ;RETURN
:-----

```

3131
3132
3133
3134
3135
3136
3137
3138
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3150
3151
3152
3153
3154
3155

011266	004737	011634	
011272	022737	000060	002520
011300	001416		
011302	012737	000013	011542
011310	004737	011502	
011314	004737	011462	
011320	013777	002400	171022
011326	004737	011610	
011332	004737	012244	
011336	000207		

.SBTTL - MOD U.DEV.RST - READ STATUS

```

:++
: FUNCTIONAL DESCRIPTION: SUBR TO SEND READ STATUS COMMAND TO DEVICE.
: INPUTS: NONE
: IMPLICIT INPUTS: NONE
: OUTPUTS: READ STATUS TO RX
: IMPLICIT OUTPUTS: NONE
: SUBORDINATE ROUTINES USED: SETUP COMMANDS, WAIT 'DONE', WAIT 'TR'
: FUNCTIONAL SIDE EFFECTS: NONE
: CALLING SEQUENCE:
:--

```

```

-----
RDSTAT: CALL   DVDNCK           ;CALL DEVICE READY CHECK
          CMP    #NODNBT,ERRNBR ;IF ERRNBR NOT SET=
          BEQ   XRDSTA          ;'NO DONE BIT', THEN
          MOV   #13,NCMD        ;SET READ STATUS
          CALL  SETSCD          ;CALL SET SUBSYS COMMAND
          CALL  SETDCD          ;CALL SET DEVICE COMMAND - MOD U.DEV.SDC
          MOV   CMD,@RXCS       ;SEND COMMAND
          CALL  WAIT            ;GO AWAIT 'DONE' OR INTERRUPT
          CALL  GETREG          ;CALL GET DEV REGS
XRDSTA: RETURN                  ;RETURN
-----

```


3194
3195
3196
3197
3198
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238

011462 000240
011464 053737 002406 002400
011472 053737 002410 002400
011500 000207

011502 000240
011504 013737 002400 002424
011512 013737 011542 002400
011520 005037 002472
011524 053737 002412 002400
011532 053737 002404 002400
011540 000207

011542 000000

.SBTTL - MOD U.DEV.CMD - SETUP DEVICE COMMAND
:++
: FUNCTIONAL DESCRIPTION: SUBR TO SETUP DEVICE COMMAND WORD - I.E.,
: SET DRIVE & SIDE BITS
: INPUTS: NONE
: IMPLICIT INPUTS: SIDE & DRIVE BITS, COMMAND
: OUTPUTS: COMMAND WORD FOR DEVICE
: IMPLICIT OUTPUTS: NONE
: SUBORDINATE ROUTINES USED: NONE
: FUNCTIONAL SIDE EFFECTS: NONE
: CALLING SEQUENCE: SUBR
:--

SETDCD: NOP ;
BIS DRIVE,CMD ;SETUP DRIVE BIT
BIS SIDE,CMD ;SETUP SIDE BIT
RETURN ;RETURN

.SBTTL - MOD U.DEV.SSC - SETUP SUBSYSTEM COMMANDS
:++
: FUNCTIONAL DESCRIPTION: SUBR TO SETUP SUBSYSTEM COMMANDS - I.E.
: SET BITS THAT ARE NOT DRIVE RELATED
: INPUTS: NEW COMMAND
: IMPLICIT INPUTS: COMMAND, DENSITY, INTERRUPT BIT
: OUTPUTS: COMMAND
: IMPLICIT OUTPUTS: LAST COMMAND, PROTOCOL CTR
: SUBORDINATE ROUTINES USED: NONE
: FUNCTIONAL SIDE EFFECTS: NONE
: CALLING SEQUENCE: SUBR
:--

SETSCD: NOP ;
MOV CMD,LCMD ;SAVE LAST COMMAND
MOV NCMD,CMD ;SETUP NEW COMMAND
CLR PROTCT ;CLEAR TEST COMMAND PROTOCOL COUNTER
BIS DENSTY,CMD ;SETUP DENSITY BIT
BIS INTERT,CMD ;SETUP INTERRUPT BIT
RETURN ;

NCMD: 0 ;NEW COMMAND

3241
3242
3243
3244
3245
3246
3247
3248
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3260
3261
3262
3263
3264
3265
3266
3267

011544 000240
011546 005737 002454
011552 001415
011554 005737 002520
011560 001412
011562 032737 004000 002476
011570 001006
011572 004737 003060
011576
011604
011606 000207

```

.SBTTL - MOD U.DEV.CEC - DEVICE COMMAND ERROR CHECK
-----
:++
: FUNCTIONAL DESCRIPTION: SUBR TO CHECK FOR DEVICE COMMAND FATAL ERRORS.
: INPUTS: NONE
: IMPLICIT INPUTS: FIN FLAG, FLAGS(NEG TEST), ERR NBR
: OUTPUTS: NONE
: IMPLICIT OUTPUTS: ERROR CONDITION
: SUBORDINATE ROUTINES USED: ERROR
: FUNCTIONAL SIDE EFFECTS: DROP UNIT & CLEAN UP
: CALLING SEQUENCE: SUBR
:--
:-----
:
CDERCK: NOP
:
: TST FIN :IF FINI FLAG
: BEQ XCEC :SET, THEN
: TST ERRNBR :IF ERROR NUMBER
: BEQ XCEC :NOT=0, THEN
: BIT #NEGTST,FLAGST :IF NEG TEST FLAG
: BNE XCEC :NOT SET, THEN
: CALL ERROR :CALL ERROR-MOD
: DODU UNIT :DROP UNIT
: DOCLN :DO CLEAN-UP
: XCEC: RETURN :RETURN
:-----

```

3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3310
3311
3312
3313
3314
3315
3316

011610	032737	000100	002400
011616	001403		
011620	004737	011662	
011624	000402		
011626	004737	012032	
011632	000207		
011634	005003		
011636	032777	000040	170504
011644	001005		
011646	005203		
011650	001372		
011652			
011654	004737	012032	
011660	000207		

```
.SBTTL - MOD U.DEV.WAT - WAIT SUBROUTINE
:++
: FUNCTIONAL DESCRIPTION: SUBR TO DETERMINE TO WAIT FOR 'DONE' OR INTERRUPTS
: INPUTS: DEVICE COMMAND
: IMPLICIT INPUTS: NONE
: OUTPUTS: NONE
: IMPLICIT OUTPUTS: NONE
: SUBORDINATE ROUTINES USED: WATCH & A WAIT DONE
: FUNCTIONAL SIDE EFFECTS: NONE
: CALLING SEQUENCE: SUBR
:--
```

```
-----
WAIT: BIT #100,CMD ;IF COMMAND-INTERRUPT BIT
      BEQ 1$ ;SET, THEN
      CALL WATCH ;CALL WATCH DOG WAIT FOR INTERRUPT
      BR XWAIT ;BR TO END
1$: CALL AWDN ;ELSE, CALL WAIT FOR DONE
XWAIT: RETURN ;RETURN
-----
```

```
.SBTTL - MOD U.DEV.DRC - DEVICE DONE CHECK
:++
: FUNCTIONAL DESCRIPTION: SUBR TO CK IF DEVICE IS READY TO ACCEPT A CMD
: INPUTS: NONE
: IMPLICIT INPUTS: DONE BIT
: OUTPUTS: NONE
: IMPLICIT OUTPUTS: NONE
: SUBORDINATE ROUTINES USED: WATCH & A WAIT DONE
: FUNCTIONAL SIDE EFFECTS: NONE
: CALLING SEQUENCE: SUBR
:--
```

```
-----
DVDNCK: CLR R3 ;CLEAR REC
1$: BIT #DNBIT,ARXCS ;IF DEVICE DONE
      BNE XDVRCK ;NOT SET
      INC R3 ;BUMP TIMEOUT COUNTER
      BNE 1$ ;IF TIME OUT, THEN
      BRESET ;EXECUTE BUS RESET
      CALL AWDN ;CALL A WAIT 'DONE'
XDVRCK: RETURN ;RETURN
-----
```


3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362

011662 005037 012030
011666
011674 013704 012024
011700 013703 012026
011704 005737 012030
011710 001413
011712 032777 000040 170430
011720 001035
011722 012737 000014 002520
011730 012737 010000 002460
011736 000426
011740
011742 005303
011744 001357
011746 005304
011750 001353
011752 032777 000040 170370
011760 001407
011762 012737 000015 002520
011770 052737 020000 002460
011776 000406
012000 052737 000020 002456
012006 012737 000060 002520
012014
012022 000207
012024 000010
012026 100000
012030 000000

```
.SBTTL - MOD U.DEV.WCH - WATCH DOG TIMER
:++
: FUNCTIONAL DESCRIPTION: SUBR TO WATCH DOG DEVICE 'DONE' & INTERRUPTS
: INPUTS: PROCESSOR LOW PRIORITY
: IMPLICIT INPUTS: DEVICE 'DONE' & INTERRUPTS
: OUTPUTS: DONE TIMEOUT ERROR, NO INTERRUPT ERROR
: IMPLICIT OUTPUTS: NONE
: SUBORDINATE ROUTINES USED: NONE
: FUNCTIONAL SIDE EFFECTS: NONE
: CALLING SEQUENCE: SUBR
:--
:-----
WATCH: CLR DNFLAG ;CLEAR DONE FLAG
        SETPRI PRIORT ;SET PROCESSOR PRI - ALLOW INTERRUPTS
        MOV DX,R4 ;SET DELAY MULT
BAUWCH: MOV DLY,R3 ;SET DELAY
IBUWCH: TST DNFLAG ;IF INTERRUPTS DONE FLAG
        BEQ LBUWCH ;IS SET, THEN
ICUWCH: BIT #DNBIT,@RXCS ;IF DONT BIT
        BNE XUWCH ;IS NOT SET, THEN
        MOV #INTNDN,ERRNBR ;SET ERROR #=NO DONE BIT
        MOV #BIT12,TYPERR ;SET INTERR, BUT NO DONE ERROR
        BR XUWCH ;BR TO MOD 'EXIT'
LBUWCH: BREAK
        DEC R3 ;DECREMENT DELAY COUNT
UDUWCH: BNE IBUWCH ;DO UNIT DELAY COUNT=0
        DEC R4 ;DECREMENT DELAY MULT
UAUWCH: BNE BAUWCH ;DO UNTIL DELAY MULT=0
IEUWCH: BIT #DNBIT,@RXCS ;IF DONE BIT IS
        BEQ LEUWCH ;SET, THEN
        MOV #DNNINT,ERRNBR ;SET ERR #=DONE, NO I:TR
        BIS #BIT13,TYPERR ;SET DONE, BUT NO INTERRUPT ERROR
        BR XUWCH ;BR TO MOD 'EXIT'
LEUWCH: BIS #BIT4,SYSERR ;SET NO DONE T.O. ERROR
        MOV #NODNBT,ERRNBR ;SET ERR #=NO DONE BIT
XUWCH: SETPRI #PRI07 ;SET PROCESSOR PRI=7 - NO INTERRUPTS
        RTS PC ;RETURN
:-----
DX: 10 ;DELAY MULT
DLY: 100000 ;DELAY
DNFLAG: 0 ;DONE FLAG
:MOD U.2.3.4 ---- END MODULE -----
```

3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3390
3391
3392
3393

012032 005004
012034 005003
012036 032777 000040 170304
012044 001020
012046
012050 005203
012052 001371
012054 005204
012056 023704 002474
012062 101364
012064 012737 000060 002520
012072 052737 000020 002456
012100 012737 000001 002454
012106 000207

.SBTTL - MOD U.DEV.WDN - AWAIT DONE BIT SUBROUTINE
:++
: FUNCTIONAL DESCRIPTION: SUBR TO WAIT FOR DEVICE 'DONE' BIT
: INPUTS: TIMEOUT PASS COUNTER
: IMPLICIT INPUTS: DEVICE 'DONE' BIT, (RXCSR), DONE WAIT MULTIPLIER
: OUTPUTS: 'DONE' BIT TIMEOUT ERROR
: IMPLICIT OUTPUTS: NONE
: SUBORDINATE ROUTINES USED: GET DEVICE REGISTERS
: FUNCTIONAL SIDE EFFECTS: NONE
: CALLING SEQUENCE: SUBR
:--

AWDN: CLR R4 ;RESET TIME OUT MULTIPLIER
1\$: CLR R3 ;PRESET TIME OUT COUNTER
2\$: BIT #DNBIT,@RXCS ;SEE IF DONE SET
BNE 3\$;IF SO: BR
BREAK ;TEMPORARY RETURN TO MONITOR
INC R3 ;BUMP TIME OUT COUNTER
BNE 2\$;IF NOT TIMED OUT: BR
INC R4 ;INCREMENT TIMEOUT MULTIPLIER
CMP DNWTMT,R4 ;IF ON 2ND
BHI 1\$;TIMEOUT PASS, THEN
MOV #NODNBT,ERRNBR ;SET ERR #=NO DONE BIT
BIS #BIT4,SYSERR ;SET NO DONE BIT ON SYSTEM ERROR
MOV #1,FIN ;EXIT THIS COMMAND
3\$: RTS PC ;EXIT

3396
3397
3398
3399
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
3434

012110 005237 002472
012114 005004
012116 005003
012120 032777 000040 170222
012126 001013
012130 032777 000200 170212
012136 001041
012140
012142 005203
012144 001365
012146 005204
012150 022704 000004
012154 101360
012156 012737 000001 002454
012164 052737 000020 002460
012172 013737 002400 002422
012200 012737 000057 002520
012206 052737 000200 002456
012214 032777 000040 170126
012222 001004
012224 052737 000020 002456
012232 000403
012234 012737 000063 002520
012242 000207

.SBTTL - MOD U.DEV.WTR - AWAIT TRANSFER READY SUBROUTINE
:++
: FUNCTIONAL DESCRIPTION: SUBR TO WAIT FOR DEVICE "TR" BIT
: INPUTS: NONE
: IMPLICIT INPUTS: DEVICE "TR", "DONE" & CSR, ESR
: OUTPUTS: "TR" TIMEOUT ERROR, NO DONE BIT, PROTOCOL COUNTER
: IMPLICIT OUTPUTS: NONE
: SUBORDINATE ROUTINES USED: GET DEVICE REGISTERS
: FUNCTIONAL SIDE EFFECTS: NONE
: CALLING SEQUENCE: SUBR
:--

AWTR: INC PROTCT ;INCREMENT TEST PROTOCOL COUNTER
CLR R4 ;PRESET TIMEOUT MULTIPLIER
1\$: CLR R3 ;PRESET TIME OUT COUNTER
2\$: BIT #DNBIT,ARXCS ;IF DONE BIT
BNE 3\$;NOT SET, THEN
BIT #TRBIT,ARXCS ;SEE IF TRANSFER READY SET
BNE 5\$;IF SO: BR
BREAK ;TEMPORARY RETURN TO MONITOR
INC R3 ;BUMP TIME OUT COUNTER
BNE 2\$;IF NOT TIMED OUT: BR
INC R4 ;INCREMENT TIMEOUT MULTIPLIER
CMP #4,R4 ;IF ON 2ND
BHI 1\$;TIMEOUT PASS, THEN
3\$: MOV #1,FIN ;EXIT THIS COMMAND
BIS #CMDERR,TYPERR ;**** ERROR ON COMMAND ****
MOV CMD,ERRCMD ;SETUP ERROR COMMAND
MOV #NOTRBT,ERRNBR ;SET ERR #-NO "TR" BIT
BIS #TRBIT,SYSERR ;SET SYS ERR=NO "TR" BIT
BIT #DNBIT,ARXCS ;IF DONE BIT
BNE 4\$;NOT SET, THEN
BIS #BIT4,SYSERR ;SET NO DONE BIT EITHER
BR 5\$;BR TO EXIT
4\$: MOV #DNNOTR,ERRNBR ;SET ERR #-"DONE" NO "TR"
5\$: RTS PC ;RETURN

3437
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472
3473
3474
3475
3476
3477

012244	013737	002432	002426
012252	013737	002434	002430
012260	017737	170064	002432
012266	017737	170060	002434
012274	000207		
012276	012737	000001	012030
012304	000002		

```
.SBTTL - MOD U.DEV.REG - GET DEVICE REGISTERS  
:++  
: FUNCTIONAL DESCRIPTION: SUBROUTINE TO GET RX02 CSR & ESR  
: INPUTS: NONE  
: IMPLICIT INPUTS: DEVICE CSR & ESR  
: OUTPUTS: DEVICE CSR & ESR  
: IMPLICIT OUTPUTS: OLD CSR & ESR  
: SUBORDINATE ROUTINES USED: NONE  
: FUNCTIONAL SIDE EFFECTS: NONE  
: CALLING SEQUENCE: SUBR  
:--
```

```
-----  
GETREG: MOV     RXCSR,LRXCSR    ;SAVE LAST CSR  
         MOV     RXESR,LRXESR  ;SAVE LAST ESR  
         MOV     @RXCS,RXCSR   ;GET RXCSR FOR PRINT  
         MOV     @RXDB,RXESR   ;GET RXESR FOR PRINT  
         RETURN                ;RETURN  
-----
```

```
.SBTTL - MOD U.DEV.ITR - INTERRUPT HANDLER  
:++  
: FUNCTIONAL DESCRIPTION: ;DEVICE INTERRUPT HANDLER  
: INPUTS: NONE  
: IMPLICIT INPUTS: DEVICE 'DONE' BIT & INTERRUPT BIT  
: OUTPUTS: DONE FLAG  
: IMPLICIT OUTPUTS: NONE  
: SUBORDINATE ROUTINES USED: NONE  
: FUNCTIONAL SIDE EFFECTS: NONE  
: CALLING SEQUENCE: DEVICE INTERRUPT  
:--
```

```
-----  
INTRHD: MOV     #1,DNFLAG     ;SET DONE FLAG  
         RTI                    ;RETURN FROM INTERRUPT  
-----
```

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

012306 042737 000377 012372
012314 005037 012654
012320 005737 012660
012324 001003
012326 012737 000007 012660
012334 013704 012660
012340 005304
012342 006304
012344 150437 012372
012350 012704 036224
012354 013705 002370
012360 006305
012362 062705 036222
012366 162705 000004
012372 000777
012374 000137 012430
012400 000137 012446
012404 000137 012456
012410 000137 012524
012414 000137 012532
012420 000137 012556
012424 000137 012566
012430 005037 012656
012434 004737 012614
012440 005705
012442 001463
012444 000773
012446 112737 000377 012656
012454 000767
012456 112737 000376 012656
012464 000261
012466 012702 000000
012472 103001
012474 005202
012476 004737 012614
012502 005705
012504 001442
012506 000241
012510 005702
012512 001401

.SBTTL - MOD U.SFT.DPT - SET DATA PATTERN

PAT #	SUBROUTINE	DATA PATTERN
0	RANDAT	NO PATTERN SPECIFIED (FORCE RANDOM DATA)
1	DATA0	ALL ZEROS
2	DATA1	ALL ONES
3	FLOAT0	FLOATING ZERO
4	FLOAT1	FLOATING ONE
5	PAT125	ALTERNATING BITS
6	PAT333	ALTERNATING PAIRS OF BITS
7	RANDAT	RANDOM

```

STDATP: BIC    #377,#BRONPT ;CLEAR BRANCH OFFSET
        CLR    SUM          ;SET UP FOR ACCUMULATION OF CHECK SUM
        TST    PAT          ;IF NO PATTERN SPECIFIED FORCE PATTERN 7
        BNE    1$
        MOV    #7,PAT
1$:     MOV    PAT,R4        ;GET PATTERN BITS
        DEC    R4           ;ADJUST FOR CORRECT OFFSET
        ASL    R4
        BISB   R4,#BRONPT  ;INSERT OFFSET
        MOV    #DATPAT+2,R4 ;SET UP ADDRESS OF FIRST BYTE
        MOV    WDCNT,R5     ;SETUP WORD COUNT
        ASL    R5           ;DOUBLE WORD COUNT FOR ADR
        ADD    #DATPAT,R5   ;ADD DATA PATTERN ADR
        SUB    #4,R5        ;ADJ. FOR CHECKSUM
BRONPT: BR    .            ;BRANCH BY OFFSET SELECTED
        JMP    DATA0      ;000 DATA BYTE
        JMP    DATA1      ;377 DATA BYTE
        JMP    FLOAT0      ;FLOAT A 0 THROUGH ALL 1'S
        JMP    FLOAT1      ;FLOAT A 1 THROUGH ALL 0'S
        JMP    PAT125      ;125/052 DATA WORD
        JMP    PAT333      ;314/063 DATA WORD
        JMP    RANDAT      ;RANDOM DATA BYTE

DATA0:  CLR    DATBYT
PG:     JSR    PC,LOAD      ;GO LOAD THE DATA BUFFER
        TST    R5           ;IF R5
        BEQ    END131      ;NOT =0 ,THEN
        BR    PG

DATA1:  MOVB   #377,DATBYT
        BR    PG

FLOAT0: MOVB   #376,DATBYT ;SET UP A ONES FIELD
XPG:    SEC
1$:     MOV    #0,R2        ;SET THE C BIT TO ROTATE THROUGH THE DATA
        BCC   2$           ;CLR R2 (CAN'T USE "CLR" AS IT CLEARS "C" BIT)
        INC   R2           ;BR IF THE "C" BIT IS CLEARED
        INC   R2           ;SET R2 IF NOT
2$:     JSR    PC,LOAD      ;GO LOAD THE DATA BUFFER
        TST    R5           ;IF R5
        BEQ    END131      ;NOT ZERO THEN
        CLC
        TST    R2          ;IS R2 NONZERO
        BEQ    3$
3$:

```

```

3536 012514 000261          SEC          ;YES, SET THE 'C' BIT
3537 012516 106137 012656 3$:  ROLB      DATBYT
3538 012522 000761          BR        1$
-----
3540 012524 005037 012656  FLOAT1: CLR      DATBYT
3541 012530 000755          BR        XPG
-----
3543 012532 112737 000125 012656 PAT125: MOVB     #125,DATBYT
3544 012540 004737 012614  XXPG:  JSR      PC,LOAD
3545 012544 005705          TST      R5          ;IF R5
3546 012546 001421          BEQ      END131     ;NOT ZERO THEN
3547 012550 105137 012656          COMB     DATBYT
3548 012554 000771          BR        XXPG
-----
3550 012556 112737 000333 012656 PAT333: MOVB     #333,DATBYT
3551 012564 000765          BR        XXPG
-----
3553 012566 004737 010344          RANDAT: JSR     PC,RANGEN ;GET RANDOM NUMBER
3554 012572 113737 010436 012656 MOVB     RANUM,DATBYT
3555 012600 004737 012614          JSR     PC,LOAD
3556 012604 005705          TST     R5          ;IF R5
3557 012606 001401          BEQ     END131     ;NOT ZERO THEN
3558 012610 000766          BR     RANDAT
-----
3560 012612 000207  END131: RTS     PC          ;RETURN.
-----
3565 012614 063737 012656 012654 LOAD:  ADD     DATBYT,SUM ;ACCUMULATE THE PATTERN CHECK SUM
3566 012622 113724 012656          MOVB    DATBYT,(R4)+ ;LOAD THE DATA BUFFER
3567 012626 020504          CMP     R5,R4      ;HAVE 124 BYTES BEEN GENERATED
3568 012630 001401          BEQ     1$        ;IF YES, RETURN
3569 012632 000407          BR     ENLDL      ;IF NO, RETURN TO PATTERN GENERATOR
3570 012634 113724 012654          1$:  MOVB    SUM,(R4)+ ;PUT CHECKSUM INTO TABLE
3571 012640 005137 012654          COM     SUM        ;COMPLIMENT CHECKSUM
3572 012644 113714 012654          MOVE   SUM,(R4)    ;PUT COMP CHECK SUM INTO TABLE
3573 012650 005005          CLR     R5        ;CLEAR TEMP #5 - FLAG DONE MODULE
3574 012652 000207  ENLDL: RTS     PC          ;RETURN
-----
3576 012654 000000  SUM:    0
3577 012656 000000  DATBYT: 0
3578 012660 000000  PAT:    0
3579          ;MOD 1.3.1 ----- END MODULE -----

```



```

3582
3583
3584
3585 012662 000240
3586 012664 032737 000400 002510
3587 012672 001423
3588 012674 042737 000400 002510
3589 012702 013737 002336 013100
3590 012710 163737 002334 013100
3591 012716 005237 013100
3592 012722 013737 002334 002374
3593 012730 005337 002374
3594 012734 042737 001000 002476
3595 012742 032737 000004 002510
3596 012750 001416
3597 012752 012737 000115 002374
3598 012760 052737 001000 002476
3599 012766 000443
3600 012770 032737 000001 002510
3601 012776 001403
3602 013000 005237 002374
3603 013004 000426
3604 013006 004737 010344
3605 013012 042737 177600 010436
3606 013020 123737 010436 002336
3607 013026 003401
3608 013030 000766
3609 013032 123737 010436 002334
3610 013040 002001
3611 013042 000761
3612 013044 123737 010436 002374
3613 013052 001755
3614 013054 013737 010436 002374
3615 013062 005337 013100
3616 013066 001003
3617 013070 052737 001000 002476
3618 013076 000207
3619
3620 013100 000000
3621 013102 000000
3622
    
```

```

.SBTTL - MOD U.SFT.GTK - GET TRACK
-----
GETTRK: NOP
IAGTK: BIT #ITK,TKSCFG ;IF INITIALIZE TRK IS
      BEQ ICGTK ;SET, THEN
      BIC #ITK,TKSCFG ;RESET INITIALIZE TRK FLG
      MOV ID,TRKCNT ;GET INSIDE TRACK
      SUB OD,TRKCNT ;GET OUTSIDE TRACK
      INC TRKCNT ;INCREMENT # OF TRACKS
      MOV OD,TRACK ;SET TRACK=O.D.
      DEC TRACK ;DECREMENT TRACK
      BIC #TRKDON,FLAGST ;CLEAR TRACK DONE FLAG
ICGTK: BIT #ILTK,TKSCFG ;IF TK/SC FLAGS=ILLEGAL TRACK
      BEQ LBGTK ;BIT SET, THEN
      MOV #77,TRACK ;SET TRACK=77=ILLEGAL TRACK
      BIS #TRKDON,FLAGST ;SET TRACK DONE FLAG
      BR XGTK ;BR TO EXIT
IBGTK: BIT #STK,TKSCFG ;IF TK & SE FLAG=SEQ TRK FLAG
      BEQ LBGTK ;SET, THEN
      INC TRACK ;INCREMENT TRACK
      BR EBGTK ;BR TO END 'B'
LBGTK: CALL RANGEN ;GET A RANDOM NUMBER
      BIC #177600,RANUM ;CLEAR ALL BUT LOW 7 BITS
IDCOMP: CMPB RANUM,ID ;IF RANUM LARGER THAN ID ADDRESS
      BLE ODCOMP ;THEN
      BR LBGTK ;BR TO GET ANOTHER RANDOM NUMBER
ODCOMP: CMPB RANUM,OD ;IF RANUM SMALLER THAN OD ADDRESS
      BGE PRESCK ;THEN
      BR LBGTK ;BR TO GET ANOTHER RANDOM NUMBER
PRESCK: CMPB RANUM,TRACK ;IF RANUM EQUALS PRESENT TRACK
      BEQ LBGTK ;GET ANOTHER RANDOM NUMBER
      MOV RANUM,TRACK ;RANUM OK PUT IT IN TARGET TRACK
EBGTK: DEC TRKCNT ;IF TOTAL # OF TRACKS
      BNE XGTK ;DONE, THEN
      BIS #TRKDON,FLAGST ;THEN SET TRACK DONE FLAG
XGTK: RTS PC
-----
TRKCNT: .WORD 0 ;DRV TRK TABLE LOCATOR
INITTK: .WORD 0 ;INITIALIZE TRK FLAG
-----
    
```

```

3625
3626
3627
3628 013104 000240
3629 013106 032737 001000 002510
3630 013114 001411
3631 013116 042737 001000 002510
3632 013124 012737 000001 002376
3633 013132 042737 002000 002476
3634 013140 105737 002510
3635 013144 001416
3636 013146 062737 000001 002376
3637 013154 022737 000033 002376
3638 013162 101030
3639 013164 012737 000001 002376
3640 013172 052737 002000 002476
3641 013200 000421
3642 013202 004737 010344
3643 013206 042737 177740 010436
3644 013214 123727 010436 000033
3645 013222 103367
3646 013224 105737 010436
3647 013230 001002
3648 013232 105237 010436
3649 013236 113737 010436 002376
3650 013244 000207
3651

```

```

.SBTTL - MOD U.SFT.GSC - GET SECTOR
-----
GETSEC: NOP
IAGSC: BIT #ISC,TKSCFG ;IF TK/SC FLAGS=INIT SECTORS BIT
      BEQ IBGSC ;SET, THEN
      BIC #ISC,TKSCFG ;CLEAR THE FLAG
      MOV #1,SECTOR ;SET SECTOR=1
      BIC #SECDON,FLAGST ;CLEAR FLAGST-SECTOR DONE FLAG
IBGSC: TSTB TKSCFG ;IF SEQUENCE SECTOR
      BEQ BCGSC ;SET, THEN
      ADD #1,SECTOR ;BUMP SECTOR ADDRESS
      CMP #33,SECTOR ;IF SECTORS
      BHI XGSC ;DONE, THEN
      MOV #1,SECTOR ;SET SECTOR=1
      BIS #SECDON,FLAGST ;SET FLAGST-SECTOR DONE FLAG
      BR XGSC ;BR EXIT
BCGSC: CALL RANGEN ;BGN DO 'C'-CALL RANDOM NO. GENERATOR
      BIC #177740,RANUM ;CLEAR TOP BITS RANDOM NUM.
UCGSC: CMPB RANUM,#27. ;DUNTIL RANUM < 27.
      BHS BCGSC
IDGSC: TSTB RANUM ;IF RANDOM NO.
      BNE EDGSC ;EQUALS ZERO, THEN
      INCB RANUM ;SET RANUM = 1
EDGSC: MOVB RANUM,SECTOR ;SET SECTOR ADR = RANDOM NO.
XGSC: RTS PC
-----

```

```

3654
3655
3656
3657 013246 005037 013520
3658 013252 052737 000100 002500
3659 013260 013737 002370 013514
3660 013266 006337 013514
3661 013272 005037 013516
3662 013276 013705 013516
3663 013302 116501 036222
3664 013306 116502 036622
3665 013312 120102
3666 013314 001465
3667 013316 005237 013520
3668 013322 023727 013520 000012
3669 013330 103404
3670 013332 032737 000020 002332
3671 013340 001053
3672 013342 110137 013522
3673 013346 110237 013524
3674 013352 032737 000100 002500
3675 013360 001431
3676 013362 042737 000100 002500
3677 013370 012737 000005 002520
3678 013376 004737 003060
3679 013402 032737 000020 002476
3680 013410 001011
3681 013412 012701 013526
3682 013416 013702 002374
3683 013422 013703 002376
3684 013426 004737 002612
3685 013432 000400
3686 013434 012701 013561
3687 013440 004737 002550
3688 013444 012701 013615
3689 013450 013702 013516
3690 013454 013703 013522
3691 013460 013704 013524
3692 013464 004737 002636
3693 013470 005237 013516
3694 013474 005337 013514
3695 013500 005737 013514
3696 013504 003274
3697 013506 004737 013642
3698 013512 000207
3699
3700 013514 000000
3701 013516 000000
3702 013520 000000
3703 013522 000000
3704 013524 000000
3705
3706 013526 047045 040445 052040
3707 013561 045 022516 020101
3708 013615 045 022516 031523
3709

```

```

.SBTTL - MOD U.SFT.DCK - DATA CHECK
-----
DATAACK: CLR DAERCT :CLEAR DATA ERR COUNT
          BIS #HDRPRT,FLAGSP :SET PRINT HEADER FLAG
          MOV WDCNT,BYTCNT :SAVE WORD COUNT
          ASL BYTCNT :DOUBLE IT SO BYTE COUNT
          CLR BYTNUM :CLEAR BYTE NUMBER
BADCK: MOV BYTNUM,R5 :SETUP BYTE NUMBER FOR AUTO INDEX
        MOV DATPAT(R5),R1 :SET TEMP#1=DATA SOURCE BYTE
        MOV DATBUF(R5),R2 :SET TEMP#2=DATA BUFFER BYTE
IBDCK: CMP R1,R2 :IF SOURCE BYTE & BUFFER BYTE
        BEQ EBDCK :NOT EQUAL
        INC DAERCT :INCREMENT DATA ERR COUNT
IEDCK: CMP DAERCT,#10. :IF OVER 10 DATA ERRORS
        BLO TFDCK :THEN
IFDCK: BIT #20,SWREG :IF PRINT ONLY 10 DATA ERROR FLAG
        BNE EBDCK :IS NOT SET, THEN
TFDCK: MOV R1,DATASB :GET DATA SHOULD BE->PRINT
        MOV R2,DATAWS :GET DATA WAS->PRINT
IMDCK: BIT #HDRPRT,FLAGSP :IF PRINT HEADER
        BEQ EMDCK :OK, THEN
        BIC #HDRPRT,FLAGSP :CLEAR PRINT HEADER
        MOV #DATER,ERRNBR :SETUP ERR NBR= DATA ERR
        CALL ERROR :CALL ERROR
INDCK: BIT #EMBUFF,FLAGST :IF EMPTY BUFFER BIT
        BNE ENDCK :NOT SET, THEN
        MOV #DMSG1B,R1 :SETUP MSG FORMAT
        MOV TRACK,R2 :SETUP TRACK # PRT
        MOV SECTOR,R3 :SETUP SECTOR # PRT
        CALL PRTB2S :CALL PRINT BASIC-2 ARG
        BR ENDCK :BR TO END 'N'
ENDCK: MOV #DMSG1,R1 :SETUP MSG FORMAT
        CALL PRTB0S :CALL PRINT BASIC-0 ARG
EMDCK: MOV #DMSG2,R1 :SETUP MSG FORMAT
        MOV BYTNUM,R2 :SETUP BYTE #
        MOV DATASB,R3 :SETUP DATA SHOULD BE
        MOV DATAWS,R4 :SETUP DATA WAS
        CALL PRTB3S :CALL PRINT BASIC-3 ARG
EBDCK: INC BYTNUM :INCREMENT BYTE #
        DEC BYTCNT :DECREMENT BYTE COUNT
UADCK: TST BYTCNT :DOUNTIL BYTE COUNT
        BGT BADCK :EQUALS 0
ENDDCK: CALL CLRDAT :CALL CLEAR DATA BUFFER
        RTS PC :RETURN
-----
BYTCNT: 0 :BYTE COUNT
BYTNUM: 0 :BYTE NUMBER
DAERCT: 0 :DATA ERR COUNT
DATASB: 0 :DATA SHOULD BE
DATAWS: 0 :DATA WAS
-----
DMSG1B: .ASCIZ /%N% TRK#%D3%. SEC#%D2%. /
DMSG1: .ASCIZ /%N% BYTE#%S2%AGOOD%S6%ABAD/
DMSG2: .ASCIZ /%N%S3%D3%S2%B8%S2%B8/
.EVEN

```


GLOBAL AREAS
CZRFB.P11

MACY11 30(1046)
09-APR-82 15:14

12-APR-82

13:23 PAGE 62-1

- MOD U.SFT.DCK - DATA CHECK

SEQ 0074

3710

;------

3713
3714
3715
3716
3717
3718
3719
3720
3721
3722
3723

013642 012705 036622
013646 012704 000200
013652 005025
013654 005304
013656 005704
013660 001374
013662 000207

```

.SBTTL - MOD U.SFT.CDB - CLEAR DATA BUFFER
-----
CLRDAT: MOV      #DATBUF,R5      ;GET BEGIN OF DATA BUFFER
          MOV      #128.,R4      ;SET WORD LENGTH OF TABLE
BACDB:   CLR      (R5)+          ;CLEAR WORD IN DATA BUFFER TABLE
          DEC      R4            ;DECREMENT WORD COUNT
          TST      R4            ;DO UNTIL
UACDB:   BNE      BACDB         ;ALL TABLE WORDS ZEROED
          RETURN                ;RETURN
-----

```

3726
3727
3728
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781

013664 000240
013666 005037 014670
013672 005737 002454
013676 001160
013700 004737 014276
013704 013701 002466
013710 005721
013712 100401
013714 000775
013716 013702 002470
013722 006302
013724 060201
013726 011103
013730 012337 014260
013734 012337 014262
013740 012337 014264
013744 011337 014266
013750 013701 002432
013754 042701 172027
013760 043701 014262
013764 043737 014262 014270
013772 053737 014270 014260

SBTTL - MOD U.SFT.RCR - REGISTER CHECK & REPORT

```
-----
BGNSUB
IF FINI FLAG NOT SET
THEN-
: GET TEST TABLE ADDRESS
: DOUNTIL TEST TABLE ENTRY=-1
: ADVANCE TEST TABLE ADDRESS
: ENDDO
: ADVANCE TEST TABLE ADDRESS
: GET COMMAND COUNTER
: DOUBLE COMMAND COUNTER
: ADDRESS OF REG TABLE THIS CMD=CMD COUNTER + TEST TABLE ADR
: GET ADDRESS OF REG TABLE THIS COMMAND
: SET RXCSR COMPARE WORD=COMPARE WORD FROM TABLE
: SET RXCSR MASK WORD=DON'T CARE BITS FROM REG TABLE
: SET RXESR COMPARE WORD=COMPARE WORD FROM TABLE
: SET RXESR MASK WORD=DON'T CARE BITS FROM REG TABLE
: SETUP CSR REG CK
: IF RXCSR NOT=CSRCMP
: THEN-CALL CK BITS
: ENDF
: SETUP ESR REG CK
: IF ESR NOT=ESRCMP
: THEN-CALL CK BITS
: ENDF
: ENDF
: GET REGISTER ERR TABLE PTR
: TERMINATE ERROR TABLE
: ENDF
: NOP
ENDSUB
-----
```

```
REGSCK: NOP ; CLEAR REG ERROR TABLE PTR
: CLR RGETPT ; IF FINI FLAG
IARCR: TST FIN ; NOT SET, THEN
: BNE EARCR ; CALL SETUP REGS CHECK
: CALL SURGCK ; GET TEST TABLE ADDRESS
: MOV TSTID,R1 ; DO UNTIL TEST TABLE ENTRY
UBRCR: TST (R1)+ ; EQUALS -1, ADVANCE TEST TABLE ADRS
: BMI EBRCR ; END DO 'B'
: BR UBRCR ; GET TEST COMMAND CTR
EBRCR: MOV TCMDCR,R2 ; DOUBLE COMMAND CTR
: ASL R2 ; CAL ADRS OF REG TABLE FOR THIS CMD
: ADD R2,R1 ; GET ADRS FROM TEST TABLE
: MOV (R1),R3 ; SET RXCSR COMPARE WORD=TABLE CSR CMP
: MOV (R3)+,CSRCMP ; SET RXCSR MASK WORD=TABLE CSR MSK
: MOV (R3)+,CSRMSK ; SET RXESR COMPARE WORD=TABLE ESR CMP
: MOV (R3)+,ESRCMP ; SET RXESR MASK WORD=TABLE ESR MSK
: MOV (R3),ESRMSK
CSRCHK: MOV RXCSR,R1 ; GET RXCS
: BIC #172027,R1 ; CLEAR OFF WRITE ONLY BIT-CK DRV SELECT BIT ****
: BIC CSRMSK,R1 ; MASK OFF BITS DON'T CARE ABOUT
: BIC CSRMSK,CSRSET ; MASK OFF CSRSET BITS DON'T CARE
: BIS CSRSET,CSRCMP ; SET CSR COMMAND VARIABLE BITS
```



```

3837
3838
3839 014276 000240
3840 014300 005037 014272
3841 014304 032737 040000 002400
3842 014312 001406
3843 014314 042737 001000 002400
3844 014322 042737 000400 002400
3845 014330 013705 002400
3846 014334 042705 177761
3847 014340 022705 000016
3848 014344 001015
3849 014346 032737 000200 002500
3850 014354 001011
3851 014356 013737 002424 002400
3852 014364 013737 002426 002432
3853 014372 013737 002430 002434
3854 014400 013705 002400
3855 014404 010537 014270
3856 014410 042737 176277 014270
3857 014416 032705 001000
3858 014422 001403
3859 014424 052737 001000 014272
3860 014432 032705 000020
3861 014436 001403
3862 014440 052737 000400 014272
3863 014446 032705 000400
3864 014452 001403
3865 014454 052737 000040 014272
3866 014462 042705 177761
3867 014466 005737 002402
3868 014472 001411
3869 014474 022705 000006
3870 014500 001403
3871 014502 022705 000014
3872 014506 001003
3873 014510 052737 000100 014272
3874 014516 000240
3875 014520 000207
3876

.SBTTL - MOD U.SFT.SRC - SETUP REGISTER CHECK
-----
SURGCK: NOP
IGSRC: BIT #RXINIT,CMD
      BEQ EGSRC
      BIC #SIDE1,CMD
      BIC #DRIVE1,CMD
EGSRC: MOV CMD,R5
      BIC #177761,R5
IASRC: CMP #16,R5
      BNE EASRC
IFSRC: BIT #RECTST,FLAGSP
      BNE EASPC
      MOV LCMDCMD
      MOV LRXCSCR,RXCSCR
      MOV LRXESR,RXESR
EASRC: MOV CMD,R5
      MOV R5,CSRSET
      BIC #176277,CSRSET
IBSRC: BIT #SIDE1,R5
      BEQ ICSRC
      BIS #SIDE1,ESRSET
ICSRC: BIT #DRV1,R5
      BEQ IDSRC
      BIS #DRIVE1,ESRSET
IDSRC: BIT #DENBIT,R5
      BEQ EDSRC
      BIS #DRVDEN,ESRSET
EDSRC: BIC #177761,R5
IESRC: TST DELDAT
      BEQ EESRC
      CMP #RSCMD,R5
      BEQ 1$
      CMP #WDDCMD,R5
      BNE EESRC
1$: BIS #DLDBIT,ESRSET
EESRC: NOP
XSRC: RETURN
-----
;
;CLEAR ESR SET
;IF CMD WAS RX INITIALIZE
;THEN
;CLEAR SIDE #1 SELECT BIT
;CLEAR DRIVE #1 SELECT BIT
;GET COMMAND
;CLEAR ALL BUT COMMAND
;IF COMMAND = READ ERROR CODE
;THEN
;IF FLAGSP NOT=REC TEST
;THEN
;SET COMMAND=LAST COMMAND
;GET LAST RXCSR
;GET LAST RXESR
;GET COMMAND
;SETUP CRS SET
;SAVE ONLY: SIDE,DENS,INTR ENA,(DRV SEL CK) BITS
;IF SIDE #1 SELECTED
;THEN
;SETUP ESR SET -> SIDE1
;IF DRIVE #1 SELECTED
;THEN
;SETUP ESRSET -> DRIVE1
;IF DOUBLE DENSITY SELECTED
;THEN
;SETUP ESR SET = DOUBLE DENSITY
;CLEAR ALL BUT COMMAND
;IF DELETED DATA MODE
;AND
;COMMAND=READ SECTOR
;OR
;COMMAND-WRITE DELETED DATA SECTOR
;THEN
;SETUP ESR SET ->DELETED DATA BIT
;RETURN

```



```

3879
3880
3881
3882 014522 013702 014700
3883 014526 013701 014702
3884 014532 040201
3885 014534 005102
3886 014536 053702 014702
3887 014542 005102
3888 014544 050201
3889 014546 005737 014676
3890 014552 001407
3891 014554 005337 014676
3892 014560 000241
3893 014562 006001
3894 014564 005737 014676
3895 014570 001371
3896 014572 005037 014672
3897 014576 032701 000001
3898 014602 001417
3899 014604 013702 014672
3900 014610 006302
3901 014612 063702 014704
3902 014616 011203
3903 014620 005703
3904 014622 001407
3905 014624 013704 014670
3906 014630 006304
3907 014632 010364 014706
3908 014636 005237 014670
3909 014642 005237 014672
3910 014646 000241
3911 014650 006001
3912 014652 023737 014674 014672
3913 014660 101346
3914 014662 005037 014672
3915 014666 000207
3916
3917 014670 000000
3918 014672 000000
3919 014674 000000
3920 014676 000000
3921 014700 000000
3922 014702 000000
3923 014704 000000
3924

```

.SBTTL - MOD U.SFT.BTK - BITS SET/NOT SET CHECK

```

-----
CKBITS: MOV    CMPWRD,R2    ;GET COMPARE WORD
          MOV    BADWRD,R1  ;GET BAD WORD
          BIC    R2,R1      ;SET R1=BITS THAT SHOULDN'T BE SET
          COM    R2         ;COMPLIMENT COMPARE WORD
          BIS    BADWRD,R2  ;SET BAD BITS
          COM    R2         ;SET R2=BITS THAT SHOULD BE SET
          BIS    R2,R1      ;SET R1=ALL BITS THAT SHOULD OR SHOULDN'T BE SET
          TST    BITOFF     ;IF BIT OFFSET
          BEQ    2$         ;NOT=0, THEN
1$:      DEC    BITOFF     ;
          CLC                    ;CLEAR CARRY
          ROR    R1         ;
          TST    BITOFF     ;IF BIT OFFSET
          BNE    1$         ;EQUALS 0, THEN
2$:      CLR    BITCNT     ;CLEAR BIT COUNTER
3$:      BIT    #1,R1      ;IF LSB
          BEQ    4$         ;NOT=0, THEN
          MOV    BITCNT,R2  ;GET BIT COUNTER
          ASL    R2         ;DOUBLE IT FOR ADDRESSING
          ADD    RTBADR,R2  ;ADD REG TABLE ADR
          MOV    (R2),R3    ;GET ERR# THIS BIT ERROR FROM TABLE
          TST    R3         ;IF ERR #
          BEQ    4$         ;NOT=0, THEN
          MOV    RGETPT,R4  ;SET UP REG ERR TABLE POINTER
          ASL    R4         ;DOUBLE IT FOR ADDRESSING
          MOV    R3,RGETB(R4);SET THIS ERR# IN TABLE OF REG ERRORS
          INC    RGETPT    ;ADVANCE TABLE POINTER TO NEXT LOCATION
4$:      INC    BITCNT     ;INCREMENT BIT COUNTER
          CLC                    ;CLEAR CARRY
          ROR    R1         ;SHIFT NEXT BIT FOR TEST
          CMP    BITLIM,BITCNT ;IF ALL BITS SPECIFIED
          BHI    3$         ;DONE, THEN
          CLR    BITCNT     ;RESET BIT COUNT
XCRBIT:  RETURN          ;RETURN
-----

```

```

-----
RGETPT: 0 ;REG ERROR TABLE POINTER
BITCNT: 0 ;BIT COUNTER
BITLIM: 0 ;BIT REGISTER LIMIT
BITOFF: 0 ;BIT REGISTER OFFSET
CMPWRD: 0 ;COMPARE WORD
BADWRD: 0 ;BAD WORD
RTBADR: 0 ;REGISTER ERROR TABLE ADDRESS
-----

```


3927
3928
3929 014706 000000
3930 014710 177777
3931 014712 177777
3932 014714 177777
3933 014716 177777
3934 014720 177777
3935 014722 177777
3936 014724 177777
3937 014726 177777
3938 014730 177777
3939 014732 177777
3940 014734 177777
3941 014736 177777
3942 014740 177777
3943 014742 177777
3944 014744 177777

REGISTER ERROR #'S - TABLE

```

RGERTB: .WORD 0
        .WORD -1
        .WORD -1
        .WORD -1
        .WORD -1
        .WORD -1
        .WORD -1
        .WORD -1
        .WORD -1
        .WORD -1
        .WORD -1
        .WORD -1
        .WORD -1
        .WORD -1
        .WORD -1
        .WORD -1
    
```

3945
3946
3947

TABLE - ESR ERROR #'S

3948
3949 014746 000004
3950 014750 000024
3951 014752 000062
3952 014754 004050
3953 014756 004030
3954 014760 004020
3955 014762 000032
3956 014764 000025
3957 014766 000027
3958 014770 000026
3959 014772 004051
3960 014774 004052

```

ESERTB: .WORD CRCERR          ;BIT #00 - CRC ERR
        .WORD SDRDYE          ;BIT #01 - SIDE 1 RDY
        .WORD NOITDP          ;BIT #02 - INIT DONE
        .WORD ACLOW!NEGST     ;BIT #03 - AC LOW
        .WORD DENERR!NEGST    ;BIT #04 - DEN ERR
        .WORD DENDSK!NEGST    ;BIT #05 - DRV DEN-->NOT ERROR
        .WORD DLDTER          ;BIT #06 - DEL DATA
        .WORD DVRDYE          ;BIT #07 - DRV RDY
        .WORD DRVWRG          ;BIT #08 - UNIT SEL
        .WORD SIDWRG          ;BIT #09 - HEAD SEL
        .WORD WCOVFE!NEGST    ;BIT #10 - WC OVFL
        .WORD NXMERR!NEGST    ;BIT #11 - NXM
    
```

3961
3962
3963

TABLE - CSR ERROR #'S

3964
3965 014776 000033
3966 015000 000033
3967 015002 000033
3968 015004 000033
3969 015006 000033
3970 015010 000033
3971 015012 000033
3972 015014 000033
3973 015016 000033
3974 015020 000033
3975 015022 000033
3976 015024 000033
3977

```

CSERTB: .WORD CSRERR          ;BIT #04 - UNIT SEL      - R/W
        .WORD CSRERR          ;BIT #05 - "DONE"      - R
        .WORD CSRERR          ;BIT #06 - INTER ENB   - R/W
        .WORD CSRERR          ;BIT #07 - "TR"       - R
        .WORD CSRERR          ;BIT #08 - DENSITY     - R/W
        .WORD CSRERR          ;BIT #09 - HEAD SEL    - R/W
        .WORD CSRERR          ;BIT #10 -              -
        .WORD CSRERR          ;BIT #11 - RX02       - R
        .WORD CSRERR          ;BIT #12 -              - W
        .WORD CSRERR          ;BIT #13 -              - W
        .WORD CSRERR          ;BIT #14 -              - W
        .WORD CSRERR          ;BIT #15 - ERR BIT         - R
    
```

```

3980
3981
3982
3983      000000
3984      015026      004040
(2)      015026      004040
(2)      015030      000000
(2)      015032      000000
(2)      015034      177777
3985      015036      004040
(2)      015036      004040
(2)      015040      000000
(2)      015042      000000
(2)      015044      000000
3986      015046      004040
(2)      015046      004040
(2)      015050      000000
(2)      015052      000004
(2)      015054      177773
3987      015056      004040
(2)      015056      004040
(2)      015060      000000
(2)      015062      000004
(2)      015064      001640
3988      015066      004040
(2)      015066      004040
(2)      015070      000000
(2)      015072      000200
(2)      015074      000060
3989      015076      004040
(2)      015076      004040
(2)      015100      000000
(2)      015102      000000
(2)      015104      001440
3990
3991      015026
3992      015036
3993      015046
3994      015056
3995      015066
3996      015076
3997

```

```

.SBTTL - PRESETUP REGISTER TABLES
-----
TN=0
REGTB  1,04040,0,0,-1      :RXCS ONLY
      TORT1: .WORD 04040      :RXCSR SHOULD BE
      .WORD 0      :RXCSR DONT CARE
      .WORD 0      :RXESR SHOULD BE
      .WORD -1      :RXESR DONT CARE

REGTB  2,04040,0,0,0      :RXCS & RXES/ALL
      TORT2: .WORD 04040      :RXCSR SHOULD BE
      .WORD 0      :RXCSR DONT CARE
      .WORD 0      :RXESR SHOULD BE
      .WORD 0      :RXESR DONT CARE

REGTB  3,04040,0,4,177773 :RXCS & RXES INITIALIZE CK
      TORT3: .WORD 04040      :RXCSR SHOULD BE
      .WORD 0      :RXCSR DONT CARE
      .WORD 4      :RXESR SHOULD BE
      .WORD 177773      :RXESR DONT CARE

REGTB  4,04040,0,4,1640   :RXCS & RXES INITIALIZE ALL CK
      TORT4: .WORD 04040      :RXCSR SHOULD BE
      .WORD 0      :RXCSR DONT CARE
      .WORD 4      :RXESR SHOULD BE
      .WORD 1640      :RXESR DONT CARE

REGTB  5,04040,0,200,60   :RXCS & RXES READ STATUS CK
      TORT5: .WORD 04040      :RXCSR SHOULD BE
      .WORD 0      :RXCSR DONT CARE
      .WORD 200      :RXESR SHOULD BE
      .WORD 60      :RXESR DONT CARE

REGTB  6,04040,0,0,1440   :RXCS & RXES NO DISK OPERATION
      TORT6: .WORD 04040      :RXCSR SHOULD BE
      .WORD 0      :RXCSR DONT CARE
      .WORD 0      :RXESR SHOULD BE
      .WORD 1440      :RXESR DONT CARE
-----
CSONLY = TORT1      :RXCS ONLY
CSESAL = TORT2      :RXCS & RXES ALL
CEINIT = TORT3      :RXCS & RXES INITIALIZE CK
CSESIT = TORT4      :RXCS & RXES INITIALIZE ALL
CSESRS = TORT5      :RXCS & RXES READ STATUS CK
CSESND = TORT6      :RXCS & RXES NO DISK OPERATION
-----

```

```

4000
4001
4002
4003 015106 005002
4004 015110 105737 002442
4005 015114 001422
4006 015116 122737 000260 002442
4007 015124 101003
4008 015126 012702 000017
4009 015132 000407
4010 015134 052737 100000 002476
4011 015142 004737 017106
4012 015146 016102 015164
4013 015152 010237 002462
4014 015156 010237 020164
4015 015162 000207
4016
4017
4018
4019
4020
4021 015164 000000
4022 015166 000006
4023 015170 000006
4024 015172 000017
4025 015174 004041
4026 015176 000006
4027 015200 000017
4028 015202 004003
4029 015204 000017
4030 015206 000003
4031 015210 000003
4032 015212 000003
4033 015214 000017
4034 015216 000006
4035 015220 000003
4036 015222 000003
4037 015224 000004
4038 015226 000017
4039 015230 000056
4040 015232 004051
4041 015234 004030
4042 015236 004036
4043

```

```

.SBTTL - MOD U.SET.GEN - GET ERROR CODE-ERR #
-----
GTECEN: CLR R2 ;CLEAR TEMP REG #2
IAGEN: TSTB XERUUT ;IF X ERR CODE UUT
      BEQ XGTECN ;NOT=0, THEN
IBGEN: CMPB #260,XERUUT ;IF ERR CODE UUT
      BHI LBGEN ;EXCEEDS 260, THEN
      MOV #ILLERC,R2 ;SET ERR CODE #
      BR EBGEN ;BR TO END 'B'
LBGEN: BIS #ERRFLG,FLAGST ;SET FLAGS-ERR FLAG
      CALL GTECOF ;CALL GET ERROR CODE OFFSET
      MOV ECERNR(R1),R2 ;GET ERROR CODE ERR # FROM TABLE
EBGEN: MOV R2,RECERN ;READ ERR CODE ERR #
      MOV R2,ECERNB ;PASS ERR CODE ERR # TO 'ERRCHK' MOD
XGTECN: RETURN ;RETURN
-----

```

```

: ERROR CODE ERROR # TABLE
-----
ECERNR: .WORD ; 00->NO ERROR -
        .WORD SEKERR ; 10->NO HOME DRVO -SEEK
        .WORD SEKERR ; 20->NO HOME DRV1 -SEEK
        .WORD ILLERC ; 30-> -- -
        .WORD TRKAER!NEGTST ; 40->ACC TK > 76 -TRACK ERR
        .WORD SEKERR ; 50->HOME BEFORE TRK -SEEK
        .WORD ILLERC ; 60-> -- -
        .WORD RDERR!NEGTST ; 70->NO SEC-52 TRIES -READ
        .WORD ILLERC ; 100-> -- -
        .WORD RDERR ; 110->NO STEP CLOCK -READ
        .WORD RDERR ; 120->NO PREAMBLE -READ
        .WORD RDERR ; 130->PREAMBLE-NO I.D. -READ
        .WORD ILLERC ; 140-> -- -
        .WORD SEKERR ; 150->GD TRK NOT=TRK -SEEK
        .WORD RDERR ; 160->TOO MY TRIES IDAM -READ
        .WORD RDERR ; 170->DATA AM NOT FND -READ
        .WORD CRCERR ; 200->CRC -CRC
        .WORD ILLERC ; 210-> -- -
        .WORD HDSFDG ; 220->SELF DIAG -SELF DIAG
        .WORD WCOVFE!NEGTST ; 230->WRD COUNT OVF -WRD CTOV
        .WORD DENERR!NEGTST ; 240->DENSITY ERR -DEN ERR
        .WORD SDKYWD!NEGTST ; 250->WRG KEYWD-S.D. -WRG KEY
-----

```


4046
4047
4048
4049 015240 012701 015446
4050 015244 013702 002512
4051 015250 013703 002432
4052 015254 013704 002434
4053 015260 013705 002400
4054 015264 004737 002664
4055 015270 005737 002400
4056 015274 001417
4057 015276 032737 040000 002400
4058 015304 001403
4059 015306 012701 007452
4060 015312 000406
4061 015314 013705 002400
4062 015320 042705 177761
4063 015324 016501 007164
4064 015330 004737 002550
4065 015334 032737 000200 002476
4066 015342 001435
4067 015344 004737 015744
4068 015350 004737 017124
4069 015354 012701 015531
4070 015360 113702 002443
4071 015364 113703 002444
4072 015370 113704 002445
4073 015374 004737 003002
4074 015400 012701 015627
4075 015404 113702 002446
4076 015410 113703 002447
4077 015414 113704 002450
4078 015420 113705 002451
4079 015424 004737 003030
4080 015430 042737 000200 002476
4081 015436 005037 015444
4082 015442 000207
4083
4084 015444 000000
4085
4086 015446 047045 040445 020040
4087 015531 045 022516 020101
4088 015627 045 022516 020101
4089
4090

.SBTTL - MOD U.PRT.STA - PRINT UNIT STATUS

```

-----
PRTSTA: MOV #IDENT1,R1 ;SETUP FORMAT MSG
MOV UNTPRT,R2 ;SETUP UNIT PRT
MOV RXCSR,R3 ;SETUP RXCSR
MOV RXESR,R4 ;SETUP RXESR
MOV CMD,R5 ;SETUP COMMAND
CALL PRTB4S ;CALL PRINT BASIC 4-PARM.
IBSTA: TST CMD ;IF CMD
BEQ IASTA ;NOT = 0, THEN
ICSTA: BIT #BIT14,CMD ;IF PROG INIT
BEQ LCSTA ;THEN
MOV #CMDMB,R1 ;SETUP PROG INIT MSG
BR ECSTA ;BR TO END 'C'
LCSTA: MOV CMD,R5 ;GET COMMAND
BIC #177761,R5 ;CLR ALL BUT CMD
MOV CMDMSG(R5),R1 ;GET CMD MSG
ECSTA: CALL PRTB0S ;CALL PRINT BASIC 0 - PAR
IASTA: BIT #RECFLG,FLAGST ;IF ERR CODE FLAG
BEQ XPTSTA ;SET, THEN
CALL PRTECD ;CALL PRINT ERROR CODE
CALL CLRRGS ;CALL CLEAR REGISTER
MOV #XER2,R1 ;SETUP FORMAT MSG
MOVB WC,R2 ;SETUP WORD COUNT
MOVB CTK0,R3 ;SETUP CTK0
MOVB CTK1,R4 ;SETUP CTK1
CALL PRTX3S ;CALL PRINT-EXT 3 PARAMETERS
MOV #XER3,R1 ;SETUP FORMAT MSG
MOVB TTRK,R2 ;SETUP TTRK
MOVB TSEC,R3 ;SETUP TSEC
MOVB SFTSTS,R4 ;SETUP SFTSTS
MOVB BTRK,R5 ;SETUP BTRK
CALL PRTX4S ;CALL PRINT-EXT 4 PAR
XPTSTA: BIC #RECFLG,FLAGST ;CLEAR ERROR CODE FLAG
CLR ERRREG ;CLEAR ERROR REGISTER
RTS PC ;RETURN
-----
ERRREG: 0
-----
IDENT1: .ASCIZ /%N% UNIT=%01% RXCSR=%0% RXESR=%0% CMD=%0% ->/
XER2: .ASCIZ /%N% WORD CNT=%03% CUR TRK DV0=%D2%. CUR TRK DV1=%D2%./
XER3: .ASCIZ /%N% TARGET TRK =%D2%. TARGET SEC =%D2%. SOFT STAT=%03% BAD
.EVEN
-----

```

4093
4094
4095
4096 015744 012701 016014
4097 015750 113702 002442
4098 015754 042702 177400
4099 015760 004737 002734
4100 015764 105737 002442
4101 015770 001410
4102 015772 004737 017106
4103 015776 016101 016042
4104 016002 004737 002714
4105 016006 105037 002442
4106 016012 000207
4107
4108 016014 047045 040445 020040
4109
4110
4111
4112 016044 016136
4113 016046 016175
4114 016050 016116
4115 016052 016234
4116 016054 016274
4117 016056 016116
4118 016060 016337
4119 016062 016116
4120 016064 016407
4121 016066 016451
4122 016070 016477
4123 016072 016116
4124 016074 016567
4125 016076 016640
4126 016100 016670
4127 016102 016716
4128 016104 016116
4129 016106 016753
4130 016110 017013
4131 016112 017033
4132 016114 017052
4133

.SBTTL - MOD U.PRT.EC - PRINT UNIT ERROR CODE

```

PRTECD: MOV      #XER1,R1      ;SETUP FORMAT MSG
          MOVB    XERUUT,R2     ;GET ERROR CODE
          BIC     #177400,R2    ;CLEAR TOP R2
          CALL    PRTX1S        ;CALL PRINT EXTENDED-1 ARG
          TSTB    XERUUT        ;IF ERROR
          BEQ     ENDXER        ;NOT=0, THEN
          CALL    GTECOF        ;CALL GET ERROR CODE OFFSET
          MOV     ECTAB-2(R1),R1 ;SET ADR OF ERROR MSG FOR PRINT
          CALL    PRTX0S        ;CALL PRINT EXTENDED-NO ARG
          CLRB    XERUUT        ;CLEAR ERROR CODE
ENDXER: RTS     PC              ;RETURN

```

```

XER1: .ASCIZ  /%N%  ERR CODE=%03%  ->/
      .EVEN

```

```

ECTAB: .WORD  EC1
        .WORD  EC2
        .WORD  EC0
        .WORD  EC4
        .WORD  EC5
        .WORD  EC0
        .WORD  EC7
        .WORD  EC0
        .WORD  EC11
        .WORD  EC12
        .WORD  EC13
        .WORD  EC0
        .WORD  EC15
        .WORD  EC16
        .WORD  EC17
        .WORD  EC20
        .WORD  EC0
        .WORD  EC22
        .WORD  EC23
        .WORD  EC24
        .WORD  EC25

```

4136
4137
4138 016116 040445 044440 046114
4139 016136 040445 047516 044040
4140 016175 045 047101 020117
4141
4142 016234 040445 051124 042511
4143 016274 040445 047510 042515
4144
4145 016337 045 032501 020062
4146
4147 016407 045 047101 020117
4148 016451 045 050101 042522
4149 016477 045 050101 042522
4150 016550 040445 046111 020114
4151 016567 045 043501 047517
4152 016640 040445 042111 046501
4153 016670 040445 047516 042040
4154 016716 040445 051103 020103
4155
4156 016753 045 051101 053455
4157 017013 045 053501 051117
4158 017033 045 042101 047105
4159 017052 040445 042523 020124
4160
4161
4162
4163
4164
4165 017106 013701 002442
4166 017112 006201
4167 017114 006201
4168 017116 042701 177700
4169 017122 000207
4170
4171
4172
4173
4174
4175 017124 005001
4176 017126 005002
4177 017130 005003
4178 017132 005004
4179 017134 005005
4180 017136 000207
4181

.SBTTL - UNIT ERROR CODE MESSAGES

```

:-----:
ECO: .ASCIZ /%A ILL ERR CODE/
EC1: .ASCIZ /%ANO HOME ON INITIALIZE DRV 0./
EC2: .ASCIZ /%ANO HOME ON INITIALIZE DRV 1./
:EC3: .ASCIZ /%A ILL ERR CDE./
EC4: .ASCIZ /%A TRIED TO ACCESS A TRACK > 76./
EC5: .ASCIZ /%A HOME FOUND BEFORE DESIRED TRACK./
:EC6: .ASCIZ /%A ILL ERR CDE./
EC7: .ASCIZ /%A 52 HEADERS PASSED & SECTOR NOT FOUND./
:EC10: .ASCIZ /%A ILL ERR CDE./
EC11: .ASCIZ /%ANO STEPCLK SEEN IN 40 MICROSEC./
EC12: .ASCIZ /%A PREAMBLE NOT FOUND./
EC13: .ASCIZ /%A PREAMBLE FOUND BUT NO ID MARK IN TIME./
EC14: .ASCIZ /%A ILL ERR CDE./
EC15: .ASCIZ /%A GOOD HEADER TRACK ADR NOT=SELECTED TRK/
EC16: .ASCIZ /%A IDAM->TOO MANY TRIES./
EC17: .ASCIZ /%ANO DATA AM IN TIME./
EC20: .ASCIZ /%A CRC ERR ON READING SECTOR./
:EC21: .ASCIZ /%A ILL ERR CDE./
EC22: .ASCIZ /%A R-W ELECT. FAILED MAINT. TST./
EC23: .ASCIZ /%A WORD CNT OVF./
EC24: .ASCIZ /%A DENSITY ERR./
EC25: .ASCIZ /%A SET DENSITY WRG KEY WORD./
:-----:

```

.EVEN

.SBTTL - MOD U.SFT.GEO - GET ERROR CODE OFFSET

```

:-----:
GTECOF: MOV XERUUT,R1 ;SAVE EXTENDED ERROR CODE IN TEMP #1
ASR R1 ;FORMAT E.C.
ASR R1 ;FORMAT E.C. FOR ADR
BIC #177700,R1 ;CLR TOP BYTE
RETURN ;RETURN
:-----:

```

.SBTTL - MOD U.SFT.CRS - CLEAR REGISTERS

```

:-----:
CLARGS: CLP R1
CLR R2
CLR R3
CLR R4
CLR R5
RETURN ;RETURN
:-----:

```


4184
4185
4186
4187
4188
4189
4190
4191
4192
4193
4194
4195
4196
4197
4198
4199
4200
4201
4202
4203
4204
4205
4206
4207
4208
4209
4210
4211
4212
4213
4214
4215
4216
4217
4218
4219
4220

017140 013701 002350
017144 032711 100000
017150 001423
017152 032721 000040
017156 001420
017160 032711 000010
017164 001415
017166 012737 000050 002520
017174 004737 003060
017200 012701 017222
017204 004737 002550
017210
017216
017220 000207
017222 047045 040445 026440

.SBTTL - MOD U.SFT.DSC - DEVICE STATE CHECK

```

-----
BGNSUB
: IF RXCS ERROR BIT SET
:   THEN
:     IF RXCS DONE BIT SET
:       THEN
:         IF RXES ACLOW BIT SET
:           THEN-SETUP ERROR
:             SETUP MSG->'NO PWR, CABLED BACK, RX01 STRAP, PDP-8'
:               CALL ERROR
:                 SETUP DROP UNIT
:                   DO DROP UNIT
:             ENDIF
:         ENDIF
:     ENDIF
:   ENDIF
ENDSUB
-----

```

```

DVSTCK: MOV     RXCS,R1      ;SET R1=RXCS ADDRESS
IADSC:  BIT     #ERRBIT,(R1) ;IF RXCS REG=ERR BIT
        BEQ     EADSC       ;SET, THEN
IBDSC:  BIT     #DNBIT,(R1)+ ;IF RXCS REG=DONE BIT
        BEQ     EADSC       ;SET, THEN
ICDSC:  BIT     #ACLOW,(R1)  ;IF RXES REG=AC LOW BIT
        BEQ     EADSC       ;SET, THEN
        MOV     #ACLOWF,ERRNBR ;SET ERR NBR=AC LOW FATAL ERROR
        CALL    ERROR        ;CALL ERROR
        MOV     #STATER,R1   ;SET MSG->'NO PWR, CABLE BACK...ETC.'
        CALL    PRTBOS       ;CALL PRINT BASIC-NO ARG
        DODU    UNIT         ;DROP UNIT
        DOCLN   ;DO CLEAN
EADSC:  RETURN              ;RETURN
-----

```

```

: STATER: .ASCIZ  /%NZA ->NO PWR, CABLED BACKWARDS, STRAPPED RX01, PDP-8/
: .EVEN
:-----

```

4222
4223
4224
4225
4226
4227
4228
4229
4230
4231
4232
4233
4234
4235
4236
4237
4238
4239
4240
4241
4242
4243

017310	000240		
017312	004737	011266	
017316	032777	000200	163026
017324	001010		
017326	052737	000040	002456
017334	012737	000025	002520
017342	004737	003060	
017346	000207		

```
.SBTTL - MOD U.SFT.DRC - DEVICE READY CHECK
-----
: BGNSUB
:   CALL READ STATUS
:   IF RXES DRV RDY NOT SET [A]
:   : THEN-SET SYS ERR-DRV RDY ERR
:   :   SETUP ERR # DRV RDY ERR
:   :   CALL ERR
:   ENDIF
: ENDSUB
-----
DVRVCK: NOP ;
: CALL RDSTAT ;CALL READ STATUS
IADRC: BIT #DRV RDY, @RXDB ;IF RXDB-DRIVE RDY
: BNE EADRC ;NOT SET, THEN
: BIS #BITS, SYSERR ;SET SYS ERR=DRV RDY ERR
: MOV #DVRDYE, ERRNBR ;SET ERR NBR=DRV RDY ERROR
: CALL ERROR ;CALL ERROR
EADRC: RETURN ;BR TO EXIT
-----
```

4246
4247
4248
4249
4250
4251
4252
4253
4254
4255
4256
4257
4258
4259
4260
4261
4262
4263
4264
4265
4266
4267
4268
4269
4270
4271
4272
4273
4274
4275
4276
4277
4278
4279
4280
4281
4282
4283
4284
4285
4286
4287
4288
4289
4290
4291
4292
4293
4294
4295
4296
4297
4298

017350 004737 017310
017354 032737 000040 002456
017362 001054
017364 005037 002374
017370 012737 000012 002376
017376 004737 011062
017402 005737 002454
017406 001042
017410 032777 000040 162734
017416 001404
017420 012737 000400 002414
017426 000402
017430 005037 002414
017434 012737 000114 002374
017442 004737 011062
017446 017701 162700
017452 042701 177737
017456 006301
017460 006301
017462 006301
017464 020137 002414
017470 001411
017472 012737 000020 002520
017500 004737 003060
017504
017512
017514 000207

SBTTL - MOD U.SFT.DDC - DEVICE DENSITY CK

```

-----
BGNSUB
CALL DEVICE READY CK
IF SYS ERR=DEVICE READY ERR NOT SET
THEN-SET TRACK=0, SECTOR=10
CALL READ SECTOR
IF FINI NOT SET [A]
THEN
IF RXES DRIVE DENSITY=DOUBLE DEN [B]
THEN-SET DENSITY STATUS=DOUBLE DENSITY
ELSE-SET DENSITY STATUS=SINGLE DENSITY
ENDIF
SET TRACK=76, SECTOR=10
CALL READ SECTOR
IF RXES DRIVE DENSITY NOT=DENSITY STATUS [C]
THEN-
SETUP ERROR # & ERROR MSG=>'DISKETTE-MIXED DENSITY'
CALL ERROR
DO DROP UNIT
ENDIF
ENDIF
ENDSUB
-----

```

```

-----
DENCHK: CALL DVRYCK ;CALL DEVICE READY CK
IDDC: BIT #BIT5,SYSERR ;IF SYS ERR=DEVICE RDY ERR
BNE EADDC ;NOT SET, THEN
CLR TRACK ;SET TRACK=0
MOV #10.,SECTOR ;SET SECTOR=10
CALL READ ;CALL READ SECTOR
IADDC: TST FIN ;IF FINI
BNE EADDC ;NOT SET, THEN
IBDDC: BIT #DRV DEN,@RXDB ;IF DRIVE DEN=DOUBLE DEN BIT
BEQ LBDDC ;SET, THEN
MOV #DENBIT,DENSTA ;SET DENSITY STATUS=DOUBLE DEN
BR EBDDC ;BR TO END 'B'
LBDDC: CLR DENSTA ;SET DENSITY STATUS=SINGLE DEN
EBDDC: MOV #76.,TRACK ;SET TRACK=76.
CALL READ ;CALL READ SECTOR
MOV @RXDB,R1 ;GET RXES
BIC #*CDRV DEN,R1 ;CLEAR ALL BUT DRIVE DENSITY
ASL R1 ; ADV DRIVE DENSITY
ASL R1 ; SO EQUAL TO
ASL R1 ; DENSITY STATUS
ICDDC: CMP R1,DENSTA ;IF RXES DRIVE DENSITY & DENSITY STATUS
BEQ EADDC ;NOT=, THEN
MOV #DENDSK,ERRNBR ;SET ERR NBR=DISK DENSITY ERROR
CALL ERROR ;CALL ERROR-MOD
DODU UNIT ;DROP UNIT
DOCLN ;DO CLEAN
EADDC: RETURN ;RETURN
-----

```


4301
4302
4303
4304
4305
4306
4307
4308
4309
4310
4311
4312
4313
4314
4315
4316
4317
4318
4319
4320
4321
4322
4323
4324
4325
4326
4327
4328
4329
4330
4331
4332
4333
4334
4335
4336

.SBTTL - MOD U.SFT.TKE - TRACK ERROR CHECK

```

-----
BGNSUB
: IF LAST COMMAND=READ OR WRITE SECTOR [A]
: THEN-IF FLAG=READ ERROR CODE BIT SET [B]
:   THEN-IF DRIVE #0 SELECTED [C]
:     THEN-IF CURRENT TRK DRV #0 NOT=TRACK [D]
:       THEN-
:         IF FLAGS=NEG TST NOT SET [E]
:           THEN-SETUP ERROR #
:             SET PRINT TRACKS-PRINT FLAGS
:             CALL ERROR REPORT
:           ENDIF
:         ELSE-IF CURRENT TRK DRV #1 NOT=TRACK [F]
:           THEN-
:             IF FLAGS=NEG TST NOT SET [G]
:               THEN-SETUP ERROR
:                 SET PRINT TRACKS-PRINT FLAGS
:                 CALL ERROR REPORT
:             ENDIF
:           ENDIF
:         ENDIF
:       ELSE-IF ERROR ON COMMAND [H]
:         THEN-
:           IF FLAGS=NEG TEST NOT SET [I]
:             THEN-SETUP ERR #
:               SET PRINT TRACKS-PRINT FLAGS
:               CALL ERR REPORT
:             ENDIF
:           ENDIF
:         ENDIF
:       ENDIF
:     ENDIF
:   ENDIF
: ENDIF
: NOP
ENDSUB
-----

```

GLOBAL AREAS
CZRFB.P11MACY11 30(1046)
09-APR-82 15:1412-APR-82 13:23 PAGE 76
- MOD U.SFT.TKE - TRACK ERROR CHECK

SEQ 0090

4339	017516	000240			TKERCK: NOP		
4340	017520	022737	000017	002424	IATKE: CMP	#17,LCMD	:IF LAST COMMAND
4341	017526	001471				EATKE	: WAS
4342	017530	032737	000004	002400		BIT	: #4,CMD
4343	017536	001465				BEQ	: READ OR WRITE
4344	017540	032737	000200	002476	IBTKE: BIT	#RECFLG,FLAGST	:THEN
4345	017546	001442				BEQ	:IF FLAGS=READ ERROR CODE BIT
4346	017550	005737	002406		ICTKE: TST	DRIVE	:SET, THEN
4347	017554	001016				BNE	:IF DRIVE# 0
4348	017556	123737	002444	002374	IDTKE: CMPB	CTKO,TRACK	:SELECTED, THEN
4349	017564	001452				BEQ	:IF CURRENT TRACK DRIVE 0 & TRACK
4350	017566	032737	004000	002476	IETKE: BIT	#NEGST,FLAGST	:NOT=, THEN
4351	017574	001046				BNE	:IF FLAGS=NEG TEST BIT
4352	017576	012737	000041	002520		MOV	:NOT SET, THEN
4353	017604	004737	003060			CALL	:SET ERR NBR=TRACK ADDRESS ERROR
4354	017610	000440				BR	:CALL ERROR
4355	017612	123737	002445	002374	IFTKE: CMPB	CTK1,TRACK	:BR TO END 'A'
4356	017620	001434				BEQ	:IF CURRENT TRACK DRIVE 1 & TRACK
4357	017622	032737	004000	002476	IGTKE: BIT	#NEGST,FLAGST	:NOT=, THEN
4358	017630	001030				BNE	:IF FLAGS=NE TEST BIT
4359	017632	012737	000041	002520		MOV	:NOT SET, THEN
4360	017640	052737	000001	002500		BIS	:SET ERR NBR=TRACK ADDRESS ERROR
4361	017646	004737	003060			CALL	:SET PRINT TRACKS FLAG-PROGRAM FLAGS
4362	017652	000417				BR	:CALL ERROR
4363	017654	005737	002432		IHTKE: TST	RXCSR	:BR TO END 'A'
4364	017660	100014				BPL	:IF ERROR ON COMMAND (READ OR WRITE)
4365	017662	032737	004000	002476	IITKE: BIT	#NEGST,FLAGST	:SET, THEN
4366	017670	001010				BNE	:IF FLAGS=NEG TEST BIT
4367	017672	012737	000041	002520		MOV	:SET, THEN
4368	017700	052737	000001	002500		BIS	:SET ERR NBR=TRACK ADDRESS ERROR
4369	017706	004737	003060			CALL	:SET PRINT TRACKS FLAG
4370	017712	000240			EATKE: NOP		:CALL ERROR
4371	017714	042737	000001	002500		BIC	:CFAR PRINT TRACKS FLAG
4372	017722	000207			XTKECK: RETURN		:RETURN
4373							

4376
4377
4378
4379
4380
4381
4382
4383
4384
4385
4386
4387
4388
4389
4390
4391
4392
4393
4394
4395
4396
4397
4398
4399
4400
4401
4402
4403
4404
4405
4406
4407
4408

```

.SBTTL - MOD U.SFT.ECK - ERROR CHECK
-----
: BGNSUB
: IF REG CHECK SET [A]
: : THEN-CALL REGISTER CHECK
: ENDIF
: IF READ ERROR CODE SET [B]
: : THEN-IF FLAGSP=READ ERROR CODE TEST NOT SET [N]
: : : THEN-CALL READ ERROR CODE CHECK
: : : CALL ERROR NEG TEST CK
: : : ENDIF
: ENDIF
: IF ERROR FLAG SET [C]
: : THEN
: : : IF ERR NUMBER NOT SET=SYSFTL ERROR [D]
: : : : THEN-CLEAR REG ERR #
: : : : DOWHILE REG ERR # TABLE ENTRY NOT=-1 [E]
: : : : : SET TEMP R2=REG ERR # TABLE ENTRY
: : : : : IF TEMP REG #2 > REG ERR # [I]
: : : : : THEN-SET REG ERR #=TEMP REG
: : : : : ENDIF
: : : : ENDDO
: : : : IF REG ERR # > ERR CODE ERR # [M]
: : : : : THEN-SET ERR NUMBER=REG ERR #
: : : : : ELSE-SET ERR NUMBER=ERR CODE ERR #
: : : : : ENDIF
: : : : ENDIF
: : : CLEAR REG ERR #
: : : CLEAR ERR CODE ERR #
: : : CALL ERROR
: ENDIF
: ENDSUB
-----

```


4411	017724	0C0240			ERRCHK: NOP		
4412	017726	032737	000001	002476	IAECK: BIT	#REGCK,FLAGST	:IF FLAGS=REG CK BIT
4413	017734	001402			BEQ	IBECK	:SET, THEN
4414	017736	004737	013664		CALL	REGSCK	:CALL REGISTER CHECK
4415	017742	032737	000200	002476	IBECK: BIT	#RECFLG,FLAGST	:IF FLAGS=READ ERROR CODE BIT
4416	017750	001420			BEQ	ICECK	:SET, THEN
4417	017752	032737	0C0200	002500	INECK: BIT	#RECTST,FLAGSP	:IF FLAGSP=READ ERROR CODE TEST
4418	017760	001014			BNE	ICECK	:NOT SET, THEN
4419	017762	032737	100000	002432	IOECK: BIT	#ERRBIT,RXCSR	:IF RXCSR ERR BIT
4420	017770	001410			BEQ	ICECK	:SET, THEN
4421	017772	004737	015106		CALL	GTECEN	:CALL GET READ ERROR CODE ERR #
4422	017776	013702	020164		MOV	ECERNB,R2	:PASS ERROR CODE ERR # TO 'NEG TEST CK' MOD
4423	020002	004737	020170		CALL	ERNTCK	:CALL ERROR NEG TEST CHECK
4424	020006	010237	020164		MOV	R2,ECERNB	:SAVE REC ERR
4425	020012	032737	100000	002476	ICECK: BIT	#ERRFLG,FLAGST	:IF FLAGS=ERROR FLAG
4426	020020	001460			BEQ	XERRCK	:SET, THEN
4427	020022	022737	000047	002520	IDECK: CMP	#39.,ERRNBR	:IF ERR NUMBER NOT=SYS FTL ERR
4428	020030	103434			BLO	EDECK	:THEN
4429	020032	005037	020166		CLR	RGERNB	:CLEAR REGISTER ERROR #
4430	020036	005001			CLR	R1	:CLEAR REGISTER ERROR TABLE PTR
4431	020040	005761	014706		WEECK: TST	RGERTB(R1)	:DOWHILE REG ERR TABLE ENTRY
4432	020044	100413			BMI	IMECK	:NOT=-1, THEN
4433	020046	016102	014706		MOV	RGERTB(R1),R2	:PASS REG ERR # TABLE ENTRY TO 'NEG TEST CK' MOD
4434	020052	004737	020170		CALL	ERNTCK	:CALL ERROR NEG TEST CHECK
4435	020056	020237	020166		IEECK: CMP	R2,RGERNB	:IF TEMP R2 > REG ERR NBR
4436	020062	1C3402			BLO	EIECK	:THEN
4437	020064	010237	020166		MOV	R2,RGERNB	:SET REG ERR NUMBER=R2
4438	020070	005721			EIECK: TST	(R1)+	:INCREMENT INDEX
4439	020072	000762			EEEECK: BR	WEECK	:BR TO DOWHILE 'E'
4440	020074	023737	020166	020164	IMECK: CMP	RGERNB,ECERNB	:IF REG ERR# > ERR CODE ERR#
4441	020102	103404			BLO	LMECK	:THEN
4442	020104	013737	020166	002520	MOV	RGERNB,ERRNBR	:SET ERR NUMBER=REG ERR #
4443	020112	000403			BR	EDECK	:BR TO END 'D'
4444	020114	013737	020164	002520	LMECK: MOV	ECERNB,ERRNBR	:SET ERR NUMBER=ERR CODE ERR#
4445	020122	000240			EDECK: NOP		
4446	020124	032737	020000	002332	IPECK: BIT	#BIT13,SWREG	:IF SW REG BIT #13
4447	020132	001402			BEQ	EPECK	:SET, THEN
4448	020134	004737	020240		CALL	TSTDBG	**
4449	020140	005037	020166		EPECK: CLR	RGERNB	:CLEAR REG ERR #
4450	020144	005037	020164		CLR	ECERNB	:CLEAR ERR CODE ERR #
4451	020150	004737	003060		CALL	ERROR	:CALL ERROR
4452	020154	042737	000200	002500	BIC	#RECFLG,FLAGSP	:CLEAR RD ERR CODE FLG
4453	020162	000207			XERRCK: RETURN		:RETURN
4454							
4455	020164	000000			ECERNB: 0		:ERR CODE ERR #
4456	020166	000000			RGERNB: 0		:REG ERR #
4457							

4460
4461
4462
4463
4464
4465
4466
4467
4468
4469
4470
4471
4472
4473
4474
4475
4476
4477
4478
4479
4480
4481
4482
4483
4484
4485
4486
4487
4488
4489
4490
4491

020170 000240
020172 032702 004000
020176 001417
020200 042702 004000
020204 032737 004000 002476
020212 001411
020214 023702 002464
020220 001002
020222 005002
020224 000404
020226 022702 000020
020232 001001
020234 005002
020236 000207

.SBTTL - MOD U.SFT.ENC - ERROR NEG TEST CHECK

```

-----
: BGNSUB
: IF TEMP REG #2=NEG TEST FLAG SET [A]
: THEN-CLEAR NEG TEST FLAG FROM ERR #
: IF FLAGS=NEG TEST FLAG SET [B]
: THEN-IF NEG TEST ERR #=SET NEG TEST ERR [C]
: THEN-CLEAR THE ERROR
: ELSE-IF REG #2=DISK ERROR [D]
: THEN-CLEAR-NOT ERROR
: ENDIF
: ENDIF
: ENDIF
: ENDSUB
-----

```

```

ERNTCK: NOP
IAENC: BIT #NEGTST,R2 ;IF TEMP REG=NEG TEST FLAG
      BEQ XENTCK ;SET, THEN
      BIC #NEGTST,R2 ;CLEAR NEG TEST FLAG
IBENC: BIT #NEGTST,FLAGST ;IF FLAGS=NEG TEST BIT
      BEQ XENTCK ;SET, THEN
ICENC: CMP NGTSER,R2 ;IF NEG TEST ERR # & SET NEG TEST ERR
      BNE IDENC ;ARE EQUAL, THEN
      CLR R2 ;OK, CLEAR THE ERROR !!
      BR XENTCK ;BR TO IF 'I'
IDENC: CMP #DENDSK,R2 ;IF DISK DEN
      BNE XENTCK ;ERROR, THEN
      CLR R2 ;CLEAR-NOT ERROR<-----
XENTCK: RETURN ;RETURN
-----

```

```

4494
4495
4496
4497 020240 013702 002476
4498 020244 013703 002500
4499 020250 013704 002522
4500 020254 012701 020306
4501 020260 004737 002636
4502 020264 012701 020364
4503 020270 013702 020166
4504 020274 013703 020164
4505 020300 004737 002612
4506 020304 000207
4507
4508 020306 047045 040445 037055
4509 020364 040445 051040 043505
4510
4511
4512
4513
4514
4515
4516 020430 000240
4517 020432 005737 002412
4518 020436 001406
4519 020440 042737 000002 002476
4520 020446 005037 002412
4521 020452 000406
4522 020454 012737 000400 002412
4523 020462 052737 000002 002476
4524 020470 000207
4525
4526
4527
4528
4529
4530 020472 013737 002414 002412
4531 020500 005737 002414
4532 020504 001407
4533 020506 052737 000002 002476
4534 020514 012737 000200 002370
4535 020522 000406
4536 020524 042737 000002 002476
4537 020532 012737 000100 002370
4538 020540 000207
4539

```

```

.SBTTL - MOD U.SFT.DBG - TEST STATUS
-----
TSTDBG: MOV    FLAGST,R2
        MOV    FLAGSP,R3
        MOV    ERRMSG,R4
        MOV    #TSDGMS,R1
        CALL   PRTB3S
        MOV    #TSDGM1,R1
        MOV    RGERNB,R2
        MOV    ECERNB,R3
        CALL   PRTB2S
        RETURN                                ;RETURN
-----
TSDGMS: .ASCIZ  /%N%->FLAGST=%0% FLAGSP=%0% ERRMSG ADR=%0%N/
TSDGM1: .ASCIZ  /%A REG ERR #%0% ERR CODE ERR #%0%N/
        .EVEN
-----
.SBTTL - MOD U.SFT.CDC - COMPLIMENT DENSITY CONTROL
-----
CDENC:  NOP
IACDC:  TST    DENSTY                ;IF CONTROL DENSITY
        BEQ    LACDC                ;EQUALS DOUBLE, THEN
        BIC    #DDCFLG,FLAGST      ;CLEAR DOUBLE DENSITY CONTROL FLAG
        CLR    DENSTY                ;SET CONTROL DENSITY=SINGLE
        BR     XCDENC                ;BR TO 'X'
LACDC:  MOV    #DENBIT,DENSTY        ;SET CONTROL DENSITY=DOUBLE
        BIS    #DDCFLG,FLAGST      ;SET DOUBLE DENSITY CONTROL FLAG
XCDENC: RETURN                        ;RETURN
-----
.SBTTL - MOD U.SFT.SDC - SETUP DENSITY CONTROL
-----
SDENC:  MOV    DENSTA,DENSTY        ;SET DENSTY CONTROL=DENSITY STATUS
IASDC:  TST    DENSTA                ;IF DENSITY STATUS SET TO
        BEQ    LASDC                ;DOUBLE DENSITY, THEN
        BIS    #DDCFLG,FLAGST      ;SET DOUBLE DENSITY CONTROL FLAG
        MOV    #128.,WDCNT          ;SET WORD COUNT=128
        BR     XSDC                  ;BR TO EXIT
LASDC:  BIC    #DDCFLG,FLAGST        ;CLEAR DOUBLE DENSITY CONTROL FLAG
        MOV    #64.,WDCNT           ;SET WORD COUNT=64
XSDC:   RETURN                        ;RETURN
-----

```


4542
4543
4544
4545
4546
4547
4548
4549
4550
4551 020542 013702 002512
4552 020546 012701 020560
4553 020552 004737 002570
4554 020556 000207
4555
4556 020560 047045 040445 052440
4557 020600
4558
4559
4560
4561
4562
4563
4564
4565
4566
4567
4568
4569
4570
4571
4572
4573
4574 020600 013702 002514
4575 020604 012701 020657
4576 020610 004737 002570
4577 020614 032737 010000 002332
4578 020622 001406
4579 020624 013702 002515
4580 020630 012701 020642
4581 020634 004737 002570
4582 020640 000207
4583
4584 020642 040445 051440 042111
4585 020657 045 022516 020101
4586 020700
4587

```

.SBTTL - MOD U.PRT.UNT - PRINT UNIT IDENT
-----
:BGNSUB
:   GET UNIT #
:   GET UNIT MSG
:   CALL PRINT-1 ARG
:ENDSUB
-----
      MOV     UNTPRT,R2      :GET UNIT #
      MOV     #PTUTMS,R1    :GET UNIT MSG
      CALL    PRTB1S        :CALL PRINT BASIC-1 ARG
      RETURN                    :RETURN
-----
PTUTMS: .ASCIZ  /%N% UNIT #%D2/
        .EVEN
-----
.SBTTL - MOD U.PRT.DID - PRINT DRIVE IDENT
-----
:BGNSUB
:   GET DRIVE #
:   GET SIDE #
:   IF DOUBLE SIDED DEVICE
:   :   THEN-SETUP PRINT IDENT DOUBLE SIDED DEVICE
:   :   CALL PRINT BASIC-2 PAR.
:   :   ELSE-SETUP PRINT IDENT SINGLE SIDED DEVICE
:   :   CALL PRINT BASIC-1 PAR.
:   ENDIF
:ENDSUB
-----
PRTDID: MOV     DRVPRT,R2      :SETUP R2=DRV #
      MOV     #IDSSMS,R1    :SETUP PRINT IDENT SINGLE SIDED DEVICE
      CALL    PRTB1S        :CALL PRINT BASIC-1 PAR.
IADID:  BIT     #SIDFLG,SWREG :IF DOUBLE SIDED DEVICE
      BEQ     XPTDID        :FLAG SET, THEN
      MOV     SIDPRT,R2     :SETUP R3=SID #
      MOV     #IDDSMS,R1    :SETUP PRINT IDENT DOUBLE SIDED DEVICE
      CALL    PRTB1S        :CALL PRINT BASIC-2 PAR.
XPTDID: RETURN                    :RETURN
-----
IDDSMS: .ASCIZ  /%A SIDE #%01/
IDSSMS: .ASCIZ  /%N% DRIVE #%01/
        .EVEN
-----

```

4590
4591
4592
4593
4594
4595
4596
4597
4598
4599 020700 000240
4600 020702 004737 021122
4601 020706 012737 000040 002476
4602 020714 017737 161546 002522
4603 020722 052737 000001 002476
4604 020730 004737 021014
4605 020734 000207
4606
4607
4608
4609
4610
4611
4612
4613
4614
4615
4616
4617
4618
4619
4620 020736 000240
4621 020740 004737 021122
4622 020744 042737 000040 002476
4623 020752 017737 161510 002522
4624 020760 004737 020772
4625 020764 004737 021014
4626 020770 000207
4627
4628
4629
4630
4631 020772 000240
4632 020774 013701 002466
4633 021000 005721
4634 021002 012137 002476
4635 021006 111137 002500
4636 021012 000207
4637

.SBTTL - MOD U.TST.FTS - FUNCTION TEST SETUP

```

-----
: BGNSUB
:   SET FUNCTION TEST BIT-FLAGS
:   SETUP TEST IDENT MSG IN 'ERRMSG'
:   SET FLAGS REGISTER CHECK
:   NOP
: ENDSUB
-----
: FTSTUP: NOP
:   CALL CLRCR           ;CALL CLEAR CTRS & REGS
:   MOV  #FUNTST,FLAGST ;SET FUNCTION TEST BIT-FLAGS
:   MOV  @TSTID,ERRMSG  ;SETUP TEST IDENT MSG
:   BIS  #REGCK,FLAGST  ;SET FLAGS-REGISTER CHECK
:   CALL SUDVCD         ;CALL SETUP DEVICE COMMANDS
:   RETURN              ;RETURN
-----

```

.SBTTL - MOD U.TST.LTS - LOGIC TEST SETUP

```

-----
: BGNSUB
:   CLEAR FUNCTION TEST BIT-FLAGS
:   SETUP TEST IDENT MSG IN 'ERRMSG'
:   GET TEST TABLE ADDRESS
:   INCREMENT TO NEXT ADDRESS
:   SET ANY FLAGS FROM THAT ADDRESS
:   SET FLAGS REGISTER CHECK
:   NOP
: ENDSUB
-----
: LTSTUP: NOP
:   CALL CLRCR           ;CALL CLEAR CTRS & REGS
:   BIC  #FUNTST,FLAGST ;CLEAR FUNCTION TEST BIT-FLAGS
:   MOV  @TSTID,ERRMSG  ;SETUP TEST IDENT MSG
:   CALL SUTSFG         ;CALL SETUP TEST FLAGS
:   CALL SUDVCD         ;CALL SETUP DEVICE COMMANDS
:   RETURN              ;RETURN
-----

```

.SBTTL - MOD U.TST.SFG - SETUP TEST FLAGS

```

-----
: SUTSFG: NOP
:   MOV  TSTID,R1       ;GET TEST TABLE ADDRESS
:   TST  (R1)+          ;INC TEST TABLE ADDRESS
:   MOV  (R1)+,FLAGST   ;SET TEST FLAGS FROM TABLE
:   MOVB (R1),FLAGSP    ;SET PRINT FLAGS FROM TABLE
:   RETURN              ;RETURN
-----

```

```

4640
4641
4642
4643 021014 012737 036622 002360
4644 021022 012737 000111 002372
4645 021030 012737 036222 002362
4646 021036 032737 000002 002476
4647 021044 001407
4648 021046 012737 000400 002412
4649 021054 012737 000200 002370
4650 021062 000405
4651 021064 005037 002412
4652 021070 012737 000100 002370
4653 021076 012737 002442 002364
4654 021104 012737 000001 002374
4655 021112 012737 000001 002376
4656 021120 000207

```

.SBTTL - MOD U.SFT.SDC - SETUP DEVICE COMMANDS

```

-----
SUDVCD: MOV #DATBUF,EMPADR ;SETUP EMPTY BUFFER ADDRESS
MOV #'I,VARIFY ;SETUP SET DENSITY KEYWORD='I'
MOV #DATPAT,FILADR ;SETUP FILL BUFFER ADDRESS
1$: BIT #DDCFLG,FLAGST ;IF DOUBLE DENSITY FLAGS
BEQ 2$ ;SET, THEN
MOV #DENBIT,DENSTY ;SET DEVICE DENSITY=DOUBLE
MOV #128.,WDCNT ;SET WORD COUNT=DOUBLE DEN SIZE
BR 3$ ;BR
2$: CLR DENSTY ;SET DEVICE DENSITY=SINGLE
MOV #64.,WDCNT ;SET WORD COUNT=SINGLE DEN SIZE
3$: MOV #XERUUT,RECADR ;SET READ ERROR CODE ADR=NORMAL ADR
MOV #1,TRACK ;SETUP TRACK=1
MOV #1,SECTOR ;SETUP SECTOR=1
RETURN ;RETURN
-----

```

.SBTTL - MOD U.TST.CCR - CLEAR TEST CTRS & ERROR REGS

```

-----
: BGNSUB
: CLEAR ANY ERRORS FROM PREVIOUS TESTS
: ENDSUB
-----

```

```

4665
4666 021122 005037 002400
4667 021126 005037 002454
4668 021132 005037 002460
4669 021136 005037 002470
4670 021142 005037 002442
4671 021146 005037 002510
4672 021152 005037 002402
4673 021156 005037 002504
4674 021162 000240
4675 021164 000240
4676 021166 000240
4677 021170 000240
4678 021172 000207

```

```

-----
CLRCR: CLR CMD ;CLEAR COMMAND WORD
CLR FIN ;CLEAR COMMAND FINI FLAG
CLR TYPERR ;CLEAR TYPE ERROR
CLR TCMDCT ;CLEAR TEST COMMAND CTR
CLR XERUUT ;CLEAR READ ERR CODE WORD
CLR TKSCFG ;CLEAR TRK & SEC FLAGS
CLR DELDAT ;CLEAR DELETED DATA MODE
CLR TTEMP1 ;CLEAR TEST TEMP #1
NOP
NOP
NOP
NOP
RETURN ;RETURN
-----

```

.SBTTL - MOD U.TST.T76 - SET TRACK=76

```

-----
STTK76: MOV #76.,TRACK ;SET TRACK=76.
MOV #10.,SECTOR ;SET SECTOR=10.
RETURN ;RETURN
-----

```

```

4688
4689 021212
4690

```

ENDMOD

4703
4704
4732
4733 021212
4734
4735
4736
4737
4738
4739
4740 021212
4746
4747 021212
4748
4759

.TITLE MISCELLANEOUS SECTIONS
.SBTTL REPORT CODING SECTION

BGNMOD

::++
: THE REPORT CODING SECTION CONTAINS THE
: 'PRINTS' CALLS THAT GENERATE STATISTICAL REPORTS.
:--

BGNRPT

ENDRPT

.EVEN

4761
 4762
 4763
 4764
 4765
 4766
 4767
 4768 021214
 4769
 4770 021214 000240
 4775 021216
 4777 021224
 4778 021232
 4779 021234
 4780 021242
 4781 021244 042737 140000 002500
 4782 021252
 4783 021260
 4784 021262
 4785 021264
 4786 021266 052737 000400 002500
 4787 021274 022737 004177 002120
 4788 021302 101007
 4789 021304 052737 010000 002500
 4790 021312 000403
 4791 021314 042737 000400 002500
 4792 021322 052737 100000 002500
 4793 021330 000414
 4794 021332
 4795 021340
 4796 021342 052737 040000 002500
 4797 021350 000404
 4798 021352
 4799 021360
 4800 021362 012737 177777 021526
 4801 021370 062737 000001 021526
 4802 021376 023737 002012 021526
 4803 021404 001426
 4804 021406
 4805 021420
 4806 021422 004737 021742
 4807 021426 004737 021572
 4808 021432
 4809 021460 000414
 4810 021462
 4811 021502 012737 000001 002452
 4812 021510
 4813 021512 000240
 4820 021514 013737 021526 002512
 4831 021522
 4832
 4833 021524 000000
 4834 021526 177777
 4835
 4836 021530 047045 040445 052123
 4837 021572

```

.SBTTL INITIALIZE SECTION

:++
: THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
: AT THE BEGINNING OF EACH PASS.
:--

      BGNINIT

INIT:  NOP
      RFLAGS  FLGDRS
      REDEF   #EF.CONTINUE      ;IF CONTINUE
      BCOMPLETE XINIT          ;NOT SET, THEN
      REDEF   #EF.PWR           ;IF POWER FAIL
      BCOMPLETE XINIT          ;NOT SET, THEN
      BIC     #RESFLG!STAFLG,FLAGSP ;CLEAR RESTART & START FLAGS
START: REDEF   #EF.START        ;IF START FLAG
      BNCOMPLETE RESTAR       ;SET, THEN
STARTO: READBUS                ;IF BUS IS 'LSI-BUS'
      BNCOMPLETE UN1          ;THEN
      BIS     #LSIFLG,FLAGSP   ;SET LSI FLAG-PROGRAM FLAGS
      CMP     #4177,LSHIMEM    ;IF HI MEMORY (417776=HI LIMIT 124k)
      BHI     START1           ;IS 124K OR HIGHER, THEN
      BIS     #FONZFG,FLAGSP   ;SET LSI 11/23 FLAG
      BR      START1           ;BR TO "START1"
UN1:   BIC     #LSIFLG,FLAGSP   ;CLEAR LSI FLAG-PROGRAM FLAGS
START1: BIS     #STAFLG,FLAGSP  ;SET START FLAG
      BR      SETUP            ;BR TO "SET UP"
RESTAR: REDEF   #EF.RESTART    ;IF RESTART FLAG
      BNCOMPLETE NEW          ;SET, THEN
      BIS     #RESFLG,FLAGSP   ;SET RESTART FLAG
      BR      SETUP            ;BR TO "SETUP"
NEW:   REDEF   #EF.NEW         ;IF NEW PASS FLAG
      BNCOMPLETE NEXT        ;THEN
SETUP: MOV     #-1,UNIT        ;SETUP TO START GETING UNITS OVER
NEXT:  ADD     #1,UNIT         ;BUMP UNIT TO NEXT UNIT
      CMP     LSUNIT,UNIT      ;IF 'DRS' UNIT CNT & DIAG UNIT
      BEQ     INTER           ;NOT EXCEEDED, THEN
      GPHARD  UNIT,PLOC        ;GET NEXT UNIT
      BNCOMPLETE NEXT        ;IF FOUND A UNIT, THEN
      CALL    INTTBL           ;CALL INITIALIZE TABLES
      CALL    UNPKHP           ;UNPACK HARDWARE P-TABLES
      SETVEC  VECT,#INTRHD,#PRI07
      BR      XINIT
INTER: PRINTF  #INTER1        ;PRINT "TOO MANY UNITS"
      MOV     #1,ABORT        ;SET ABORT FLAG
      DOCLN
XINIT: NOP
      MOV     UNIT,UNTPRT     ;SET USER # = LOGICAL UNIT #
      ENDINIT

-----
PLOC:  .WORD  0              ;P-TABLE LOCATION
UNIT:  .WORD -1              ;LOGICAL UNIT# UNDER TEST
-----
INTER1: .ASCIZ  /%NZASTART OVER -> TOO MANY UNITS/
      .EVEN
  
```

4838

:-----

4850
 4851
 4852
 4853 021572 013701 021524
 4854 021576 012137 002350
 4855 021602 013737 002350 002352
 4856 021610 062737 000002 002352
 4857 021616 012137 002354
 4858 021622 005721
 4859 021624 001007
 4860 021626 005037 002406
 4861 021632 005037 002420
 4862 021636 105037 002514
 4863 021642 000411
 4864 021644 012737 000020 002406
 4865 021652 012737 000001 002420
 4866 021660 112737 000001 002514
 4867 021666 005721
 4868 021670 001005
 4869 021672 005037 002410
 4870 021676 105037 002515
 4871 021702 000406
 4872 021704 012737 001000 002410
 4873 021712 112737 000001 002515
 4874 021720 011102
 4875 021722 116237 021732 002356
 4876 021730 000207
 4877
 4878 021732 000
 4879 021733 040
 4880 021734 100
 4881 021735 140
 4882 021736 200
 4883 021737 240
 4884 021740 300
 4885 021741 340
 4886
 4887
 4888
 4889
 4890
 4891 021742 000240
 4892 021744 012701 002452
 4893 021750 012702 000010
 4894 021754 005021
 4895 021756 005302
 4896 021760 001375
 4897 021762 000207
 4898

.SBTTL - MOD I.1 - UNPACK HARDWARE P-TABLES

```

UNPKHP: MOV PLOC,R1 ;SAVE P-TABLE LOCATION
          MOV (R1)+,RXCS ;LOAD UNIT BUS ADR-CSR
          MOV RXCS,RXDB ;LOAD UNIT BUS ADR-DBR
          ADD #2,RXDB ;SET UNIT BUS ADR-DBR
          MOV (R1)+,VECT ;LOAD UNIT VECTOR
IAI1: TST (R1)+ ;IF DRIVE #0
      BNE LAI1 ;THEN
      CLR DRIVE ;SETUP TO SELECT DRIVE #0
      CLR DRVOFF ;SETUP DRIVE BYTE OFFSET DRVO
      CLRB DRVPRT ;SET PRINT DRV #=0
      BR IB1 ;BR TO IF 'B'
LAI1: MOV #DRV1,DRIVE ;SETUP TO SELECT DRIVE #1
      MOV #1,DRVOFF ;SETUP DRIVE BYTE OFFSET DRV1
      MOV #1,DRVPRT ;SET PRINT DRV #=1
IBI1: TST (R1)+ ;IF SIDE #0 SELECTED
      BNE LBI1 ;THEN
      CLR SIDE ;SETUP TO SELECT SIDE #0
      CLRB SIDPRT ;SET PRINT SID #=0
      BR EBI1 ;BR TO END 'B'
LBI1: MOV #SIDE1,SIDE ;SETUP TO SELECT SIDE #1
      MOV #1,SIDPRT ;SET PRINT SID #=1
EBI1: MOV (R1),R2 ;GET DEVICE PRIORITY
      MOV PRITAB(R2),RXPRI ;SETUP PROPER DEVICE PRIORITY
      RETURN ;RETURN
  
```

```

PRITAB: .BYTE PRI00 ;PRIORITY 0
        .BYTE PRI01 ;PRIORITY 1
        .BYTE PRI02 ;PRIORITY 2
        .BYTE PRI03 ;PRIORITY 3
        .BYTE PRI04 ;PRIORITY 4
        .BYTE PRI05 ;PRIORITY 5
        .BYTE PRI06 ;PRIORITY 6
        .BYTE PRI07 ;PRIORITY 7
  
```

.SBTTL - MOD I.2 - INITIALIZE TABLES

```

INTTBL: NOP ;
          MOV #ABORT,R1 ;GET ADDRES SOF TABLE TO CLEAR
          MOV #10,R2 ;SET TABLE LENGTH
1$: CLR (R1)+ ;CLEAR LOCATOIN
      DEC R2 ;DECREMENT TABLE COUNT
      BNE 1$ ;IF DONE, THEN
      RETURN ;RETURN
  
```

4900
4901
4902
4903
4904
4905
4906
4907
4908
4909
4916
4927
4928
4929
4930
4931
4932
4933
4934
4935
4941
4946
4947
4948
4949
4950
4951
4962
4963
4964
4967
4968
4969
4970
4971
4972
4974
4975
4976
4977
4978
4979
4980
4981
4982
4983
4989
4990
4991
5002
5003
5004

021764
021764
021772
021774

021776
021776 010002
022000 012701 022012
022004 004737 002570
022010

022012 047045 040445 042040

022054

022054
022054 004737 022066
022060

022062
022062 000240
022064

022066

.SBTTL CLEANUP CODING SECTION

++
: THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
: AFTER THE HARDWARE TESTS HAVE BEEN PERFORMED.
--

BGNCLN
CLRVEC VECT ;CLEAR VECTOR
BRESET ;BUS RESET
ENDCLN
.EVEN

.SBTTL DROP UNIT SECTION

++
: THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE TO NO
: LONGER BE TESTED.

BGNDU
MOV R0,R2 ;GET LOGICAL UNIT #
MOV #DUMSG1,R1 ;SET DROP MSG
CALL PRTB1S ;CALL PRINTB 1 ARG
ENDDU

DUMSG1: .ASCIZ /%N% DROP UNIT#%D1% FROM TEST%N/

.EVEN

.SBTTL AUTO DROP UNIT SECTION

BGNAUTO
CALL ADRTST ;CALL ADDRESSING TST
ENDAUTO

.SBTTL ADD UNIT SECTION

++
: THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
: TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
: TO THE TEST CYCLE.

BGNAU
NOP
ENDAU

.EVEN
ENDMOD

5007
5018
5054 022066
5055
5056
5057
5058
5059
5060
5061
5062
5063
5064
5065
5066
5067
5068
5069
5070
5071
5072
5073
5074
5075
5076
5077
5078
5079 022066 000240
5080 022070 005037 002452
5081 022074
5082 022122 017701 160222
5083 022126
5084 022134 005737 002452
5085 022140 001413
5086 022142 012701 022212
5087 022146 013702 002350
5088 022152
5089 022162
5090 022170 000207
5091
5092 022172 042101 051104 051505
5093 022212 040445 041040 051525
5094 022246 040445 044440 052116
5095 022324
5096
5097

.TITLE HARDWARE TESTS

BGNMOD

.SBTTL TEST 0 - ADDRESSING TEST

++
TEST TO ASSURE THAT THE DEVICE WILL RESPOND WITHOUT A BUS TRAP.

BGNSUB

SETUP TEST
SETUP BUS TRAPS
READ RXCSR
RESET BUS TRAPS
IF TRAP
: THEN-SET SYSTEM FATAL FLAG
: CALL FUNCTION TEST ERROR
: REPORT BUS TRAP ON RXCSR
ENDIF
READ RXDBR
IF TRAP
: THEN-SET SYSTEM FATAL FLAG
: CALL FUNCTION TEST ERROR
: REPORT BUS TRAP ON RXDBR
ENDIF
RESET BUS TRAPS

ENDSUB

ADRTST: NOP
CLR ABORT ;CLEAR ABORT FLAG
SETVEC #BTRP4,#TRAP,#PRI07
MOV @RXCS,R1 ;READ RXCSR
CLRVEC #BTRP4
TST ABORT ;IF ABORT FLAG
BEQ 1\$;SET, THEN
MOV #TRPMS1,R1 ;SET TRAP MESSAGE
MOV RXCS,R2 ;SET TRAP ADDRESS
ERRSF 60,TOMSG,PRTB1
DODU UNIT
1\$: RETURN ;RETURN

TOMSG: .ASCIZ /ADDRESSING TEST/
TRPMS1: .ASCII /%A BUS TRAP AT ADDRESS:%06XN/
.ASCIZ /%A INTERFACE BAD OR NOT SET TO ABOVE ADDRESS/
.EVEN

5099
5100
5101
5102
5103
5104
5105
5106
5107
5108
5109
5110
5111
5112
5113
5114
5115
5116
5117
5118
5119
5120
5121
5122
5124
(2)
(2)
(2)
5125
5126
5127
5128
5129
5130
5131
5132
5133
5134
5135
5136
5137
5138
5139
5140
5141
5142
5143
5144
(2)
(2)
(2)
(2)
5145
5146
5147
5148

022324 005237 002452
022330 000002

000000
000000
000001
022332 000414
022334 020040 047111 052111
022364

022364
022372 032737 000002 002324
022400 001417
022402 004737 020700
022406
022410 004737 011610
022414 004737 012244
022420 004737 017724

```
.SBTTL - MOD U.SFT.TRP - BUS TRAP HANDLER
:++
: FUNCTIONAL DESCRIPTION: SUBR TO HANDLE DEVICE BUS TRAP
: INPUTS: NONE
: IMPLICIT INPUTS: BUS TRAP
: OUTPUTS: ABORT FLAG
: IMPLICIT OUTPUTS: NONE
: SUBORDINATE ROUTINES USED: NONE
: FUNCTIONAL SIDE EFFECTS: NONE
: CALLING SEQUENCE: INTERRUPT
:--

-----
TRAP:   INC      ABORT          ;SET ABORT FLAG
        RTI          ;RETURN FROM TRAP INTERRUPT
-----

:
: TEST SETUP DEFINITIONS
:
FRUS1=0
TN=0
FUNCT=1
.SBTTL TEST 1 - INITIALIZE - FNC TST
BR      BGNT1      ;BR TO BGN TST
T1MSG: .ASCIZ / INITIALIZE - FNC TST/
        .EVEN

:++
: TEST TO VERIFY THAT AN RX INITIALIZE WILL RETURN THE DEVICE TO A VALID
: STATE.
:--
BGNTST
: IF FUNCTION TEST
: THEN-SETUP TEST I.D.
: CALL FUNCTION TEST SETUP
: BUS INITIALIZE
: CALL ERROR CHECK
: CALL DEVICE STATE CHECK
: INCREMENT COMMAND PTR
: PROGRAM INITIALIZE RX
: CALL ERROR CHECK
: ENDF
: ENDTST
:--

BGNT1: TSETUP
IAT1:  MOV      #T1TBL,TSTID      ;SETUP TEST ID TBL-TEST# 1
        BIT      #FUNCT,TSTMOD    ;IF TEST MODE=FUNCTION TEST
        BEQ      XT1             ;BIT SET, THEN
        CALL     FTSTUP          ;CALL FUNCTION TEST SETUP
        BRESET
        CALL     WAIT
        CALL     GETREG          ;CALL GET REGS
        CALL     ERRCHK         ;CALL ERROR CHECK
```

5149 022424 004737 017140
 5150 022430 004737 010440
 5151 022434 004737 017724
 5152 022440
 5153 022444
 (1) 015026
 5154 022444
 (2) 022444 022334
 (2) 022446 177777
 (3) 022450
 (3) 022450 015026
 (3) 022452 177777
 5155 022454

XT1:

CALL DVSTCK
 CALL INTIAL
 CALL ERRCHK
 EXIT TST
 REGTBL CSONLY

;CALL DEVICE CK
 ;CALL PROG INITIALIZE
 ;CALL ERROR CHECK

REGS1=CSONLY

TTBL

T1TBL: .WORD T1MSG
 .WORD -1
 T1RTB: .WORD REGS1
 .WORD -1

ENDTST

5158
 (2) 022456 000416
 (2) 022460 020040 042522 042101
 (2)
 5159
 5160
 5161
 5162
 5163
 5164
 5165
 5166
 5167
 5168
 5169
 5170
 5171
 5172
 5173
 5174
 5175
 5176
 5177 022514
 (2) 022514 012737 022604 002466
 (2) 022522 032737 000002 002324
 (2) 022530 001423
 (2) 022532 004737 020700
 5178 022536 004737 010440
 5179 022542 004737 017724
 5180 022546 012737 002442 002364
 5181 022554 052737 000200 002500
 5182 022562 004737 011340
 5183 022566 004737 017724
 5184 022572 042737 000200 002500
 5185 022600
 5186 022604
 (1) 015026
 5187 022604
 (2) 022604 022460
 (2) 022606 177777
 (3) 022610
 (3) 022610 015026
 (3) 022612 177777
 5188 022614

.SBTTL TEST 2 - READ ERROR CODE - FNC TST
 BR BGNT2 ;BR TO BGN TST
 T2MSG: .ASCIZ / READ ERROR CODE - FNC TST/
 .EVEN

++
 : TEST TO VERIFY THAT THE DEVICE WILL COMPLETE A READ ERROR CODE COMMAND
 : WITHOUT ENCOUNTERING AN ERROR.

 : BGNTST
 : IF FUNCTION TEST
 : THEN-SETUP TEST IDENT
 : CALL FUNCTION TEST SETUP
 : PROGRAM INITIALIZE RX
 : CALL ERROR CHECK
 : SETUP ERROR CODE ADDRESS
 : CALL READ ERROR CODE
 : CALL ERROR CHECK
 : ENDIF
 : ENDTST

TSETUP
 BGNT2: MOV #T2TBL,TSTID ;SETUP TEST ID TBL-TEST# 2
 IAT2: BIT #FUNCTT,TSTMOD ;IF TEST MODE=FUNCTION TEST
 BEQ XT2 ;BIT SET, THEN
 CALL FTSTUP ;CALL FUNCTION TEST SETUP
 CALL INTIAL ;CALL PROGRAM INITIALIZE
 CALL ERRCHK ;CALL ERROR CHECK
 MOV #XERUUT,RECADR ;SETUP READ ERROR CODE ADDRESS
 BIS #RECTST,FLAGSP ;SET READ ERROR CODE TEST=FLAGSP
 CALL RDERCD ;CALL READ ERROR LJDE
 CALL ERRCHK ;CALL ERROR CHECK
 BIC #RECTST,FLAGSP ;CLEAR READ ERROR CODE TEST=FLAGSP
 XT2: EXIT TST
 REGTBL CSONLY
 REGS1=CSONLY
 TTBL
 T2TBL: .WORD T2MSG
 .WORD -1
 T2RTB: .WORD REGS1
 .WORD -1
 ENDTST

5191
 (2) 022616 000414
 (2) 022620 020040 044506 046114
 (2)
 5192
 5193
 5194
 5195
 5196
 5197
 5198
 5199
 5200
 5201
 5202
 5203
 5204
 5205
 5206
 5207 022650
 (2) 022650 012737 022720 002466
 (2) 022656 032737 000002 002324
 (2) 022664 001413
 (2) 022666 004737 020700
 5208 022672 052737 000002 002476
 5209 022700 004737 021014
 5210 022704 004737 010510
 5211 022710 004737 017724
 5212 022714
 5213 022720
 (1) 015026
 5214 022720
 (2) 022720 022620
 (2) 022722 177777
 (3) 022724
 (3) 022724 015026
 (3) 022726 177777
 5215 022730

.SBTTL TEST 3 - FILL BUFFER - FNC TST
 BR BGNT3 ;BR TO BGN TST
 T3MSG: .ASCIZ / FILL BUFFER - FNC TST/
 .EVEN

++
 TEST TO VERIFY THE DEVICE BUFFER WILL FILL WITH NO RESULTING ERROR.

 BGNTST
 IF FUNCTION TEST
 : THEN-SETUP TEST IDENT
 : SETUP DENSITY CONTROL
 : CALL SETUP DEVICE COMMANDS
 : CALL FILL BUFFER
 : NOP
 : ENDIF
 ENDTST

TSETUP
 BGNT3: MOV #T3TBL,TSTID ;SETUP TEST ID TBL-TEST# 3
 IAT3: BIT #FUNCT,TSTMOD ;IF TEST MODE=FUNCTION TEST
 BEQ XT3 ;BIT SET, THEN
 CALL FTSTUP ;CALL FUNCTION TEST SETUP
 BIS #DDCFLG,FLAGST ;SET DOUBLE DENSITY CONTROL FLAG COMMANDS
 CALL SUDVCD ;CALL SETUP DEVICE COMMANDS
 CALL FILBUF
 CALL ERRCHK ;CALL ERROR CHECK
 XT3: EXIT TST
 REGTBL CSONLY
 REGS1=CSONLY
 TTBL
 T3TBL: .WORD T3MSG
 .WORD -1
 T3RTB: .WORD REGS1
 .WORD -1
 ENDTST

5218
(2) 022732 000415
(2) 022734 020040 046505 052120
(2) 022766

.SBTTL TEST 4 - EMPTY BUFFER - FNC TST
BR BGNT4 ;BR TO BGN TST
T4MSG: .ASCIZ / EMPTY BUFFER - FNC TST/
.EVEN

5219
5220
5221
5222
5223
5224
5225
5226
5227
5228
5229
5230
5231
5232
5233
5234

:+
: TEST TO VERIFY THE DEVICE BUFFER WILL EMPTY WITHOUT ERRORS.

: BGNTST
: IF FUNCTION TEST
: THEN-SETUP TEST IDENT
: SETUP DENSITY CONTROL
: CALL SETUP DEVICE COMMANDS
: CALL EMPTY BUFFER
: CALL ERROR CHECK
: NOP
: ENDIF
: ENDTST

5235 022766
(2) 022766 012737 023036 002466
(2) 022774 032737 000002 002324
(2) 023002 001413
(2) 023004 004737 020700
5236 023010 052737 000002 002476
5237 023016 004737 021014
5238 023022 004737 010626
5239 023026 004737 017724
5240 023032
5241 023036
(1) 015026
5242 023036
(2) 023036 022734
(2) 023040 177777
(3) 023042
(3) 023042 015026
(3) 023044 177777
5243 023046

TSETUP
BGNT4: MOV #T4TBL,TSTID ;SETUP TEST ID TBL-TEST# 4
IAT4: BIT #FUNCT,TSTMOD ;IF TEST MODE=FUNCTION TEST
BEQ XT4 ;BIT SET, THEN
CALL FTSTUP ;CALL FUNCTION TEST SETUP
BIS #DDCFLG,FLAGST ;SET DOUBLE DENSITY CONTROL FLAG
CALL SUDVCD ;CALL SETUP DEVICE COMMANDS
CALL EMPBUF ;CALL EMPTY BUFFER
CALL ERRCHK ;CALL ERROR CHECK
XT4: EXIT TST
REGTBL CSONLY
REGS1=CSONLY
TTBL
T4TBL: .WORD T4MSG
.WORD -1
T4RTB: .WORD REGS1
.WORD -1
ENDTST

5246
(2) 023050 000414
(2) 023052 020040 042522 042101
(2)

.SBTTL TEST 5 - READ STATUS - FNC TST
BR BGNT5 ;BR TO BGN TST
T5MSG: .ASCIZ / READ STATUS - FNC TST/
.EVEN

5247
5248
5249
5250
5251
5252
5253
5254
5255
5256
5257
5258
5259
5260
5261
5262
5263

..**
: TEST TO VERIFY THAT A DEVICE MAINTENANCE READ STATUS (RXES) COMMAND
: WILL EXECUTE WITHOUT ERROR.

: BGNTST
: IF FUNCTION TEST
: THEN-SETUP TEST IDENT
: SETUP DENSITY CONTROL=SINGLE
: CALL SETUP DEVICE COMMANDS
: CALL READ MAINT STATUS
: CALL ERROR CHECK
: NOP
: ENDIF
: ENDTST

5264 023102
(2) 023102 012737 023154 002466
(2) 023110 032737 000002 002324
(2) 023116 001414
(2) 023120 004737 020700
5265 023124 042737 000002 002476
5266 023132 000240
5267 023134 004737 021014
5268 023140 004737 011266
5269 023144 004737 017724
5270 023150
5271 023154
(1) 015026
5272 023154
(2) 023154 023052
(2) 023156 177777
(3) 023160
(3) 023160 015026
(3) 023162 177777
5273 023164

TSETUP
BGNT5: MOV #TSTBL,TSTID ;SETUP TEST ID TBL-TEST# 5
IAT5: BIT #FUNCT,TSTMOD ;IF TEST MODE=FUNCTION TEST
BEQ XT5 ;BIT SET, THEN
CALL FTSTUP ;CALL FUNCTION TEST SETUP
BIC #DDCFLG,FLAGST ;CLEAR DOUBLE DENSITY CONTROL FLAG
NOP
CALL SUDVCD ;CALL SETUP DEVICE COMMANDS
CALL RDSTAT ;CALL READ MAINT STATUS
CALL ERRCHK ;CALL ERROR CHECK
XT5: EXIT TST
REGTBL CSONLY
REGS1=CSONLY
TTBL
T5TBL: .WORD T5MSG
.WORD -1
T5RTB: .WORD REGS1
.WORD -1
ENDTST

5276
(2) 023166 000420
(2) 023170 020040 044506 046114
(2)

.SBTTL TEST 6 - FILL & EMPTY BUFFER - FNC TST
BR BGNT6 ;BR TO BGN TST
T6MSG: .ASCIZ / FILL & EMPTY BUFFER - FNC TST/
.EVEN

5277
5278
5279
5280
5281
5282
5283
5284
5285
5286
5287
5288
5289
5290
5291
5292
5293
5294
5295
5296
5297
5298

..**
: TEST TO VERIFY THE DEVICE BUFFER DATA IS VALID AFTER A FILL/EMPTY
: BUFFER COMMAND SEQUENCE.

: BGNTST
: IF FUNCTION TEST
: THEN-SETUP TEST IDENT
: SETUP DENSITY CONTROL=DOUBLE
: CALL SETUP DEVICE COMMANDS
: SET DATA PATTERN=RANDOM
: CALL DATA PATTERN SETUP
: CALL FILL BUFFER
: CALL ERROR CHECK
: CALL EMPTY BUFFER
: CALL ERROR CHECK
: SET EMPTY BUFFER FLAG
: CALL DATA CHECK
: ENDIF
: ENDTST

5299 023230
(2) 023230 012737 023340 002466
(2) 023236 032737 000002 002324
(2) 023244 001433
(2) 023246 004737 020700
5300 023252 052737 000002 002476
5301 023260 004737 021014
5302 023264 005037 012660
5303 023270 004737 012306
5304 023274 004737 010510
5305 023300 004737 017724
5306 023304 004737 010626
5307 023310 004737 017724
5308 023314 052737 000020 002476
5309 023322 004737 013246
5310 023326 042737 000020 002476
5311 023334
5312 023340
(1) 015026
5313 023340
(2) 023340 023170
(2) 023342 177777
(3) 023344
(3) 023344 015026
(3) 023346 177777
5314 023350

TSETUP
BGNT6: MOV #T6TBL,TSTID ;SETUP TEST ID TBL-TEST# 6
IAT6: BIT #FUNCT,TSTMOD ;IF TEST MODE=FUNCTION TEST
BEQ XT6 ;BIT SET, THEN
CALL FTSTUP ;CALL FUNCTION TEST SETUP
BIS #DDCFLG,FLAGST ;SET DOUBLE DENSITY CONTROL FLAG
CALL SUDVCD ;CALL SETUP DEVICE COMMANDS
CLR PAT ;SET DATA PATTERN=RANDOM
CALL STDATP ;CALL SET DATA PATTERN
CALL FILBUF ;CALL FILL BUFFER
CALL ERRCHK ;CALL ERROR CHECK
CALL EMPBUF ;CALL EMPTY BUFFER
CALL ERRCHK ;CALL ERROR CHECK
BIS #EMBUFF,FLAGST ;SET EMPTY BUFFER FLAG
CALL DATAK ;CALL DATA CHECK
BIC #EMBUFF,FLAGST ;CLEAR EMPTY BUFFER FLAG
XT6: EXIT TST
REGTBL CSONLY
REGS1=CSONLY
TTBL
T6TBL: .WORD T6MSG
.WORD -1
T6RTB: .WORD REGS1
.WORD -1
ENDTST

5317
(2) 023352 000420
(2) 023354 020040 042522 042101
(2)
5318
5319
5320
5321
5322
5323
5324
5325
5326
5327
5328
5329
5330
5331
5332
5333
5334
5335
5336
5337
5338
5339
5340
5341
5342
5343
5344
5345
5346
5347
5348

.SBTTL TEST 7 - READ & WRITE SECTOR - FNC TST
BR BGNT7 :BR TO BGN TST
T7MSG: .ASCIZ / READ & WRITE SECTOR - FNC TST/
.EVEN

..**
: TEST TO VERIFY THE DEVICE WILL READ AND WRITE IN BOTH DENSITIES WITHOUT
: AN ERROR.

: BGNTST
: IF FUNCTION TEST
: THEN-SETUP TEST IDENT
: CALL DENSITY CHECK
: SETUP TRACK=0, SECTOR=10
: CLEAR ENDDO FLAG
: BGND0
: : SET DENSITY CONTROL WORD=OPPOSITE DENSITY STATUS
: : SET NEGATIVE TEST FLAG
: : SETUP EXPECTED DEN ERR
: : CALL WRITE SECTOR
: : CALL ERROR CK
: : CALL READ SECTOR
: : CALL ERROR CK
: : SET DENSITY CONTROL WORD=DOUBLE DEN
: : CALL WRITE SECTOR
: : CALL ERROR CK
: : CALL READ SECTOR
: : CALL ERROR CK
: : CALL SET TRACK=76, SECTOR=10
: : COMP ENDDO FLAG
: DUNTIL ENDDO FLAG=0
: ENDIF
: ENDTST

```
5351 023414          TSETUP
(2) 023414 012737 023602 002466      BGNT7: MOV #T7TBL,TSTID ;SETUP TEST ID TBL-TEST# 7
(2) 023422 032737 000002 002324      IAT7: BIT #FUNCTT,TSTMOD ;IF TEST MODE=FUNCTION TEST
(2) 023430 001462          BEQ XT7 ;BIT SET, THEN
(2) 023432 004737 020700          CALL FTSTUP ;CALL FUNCTION TEST SETUP
5352 023436 004737 017350          CALL DENCHK ;CALL DENSITY CHECK
5353 023442 005037 002374          CLR TRACK ;SET TRACK=0
5354 023446 012737 000012 002376      MOV #10.,SECTOR ;SET SECTOR=10.
5355 023454 005037 002504          CLR TTEMP1 ;CLEAR ENDDO FLAG
5356 023460 000240          BBT7: NOP ;
5357 023462 004737 020472          CALL SDENC ;CALL SET DENSITY CONTROL=DENSITY STATUS
5358 023466 042737 004000 002476      BIC #NEGTST,FLAGST ;CLEAR NEG TEST FLAG
5359 023474 004737 010744          CALL WRITE ;CALL WRITE SECTOR
5360 023500 004737 017724          CALL ERRCHK ;CALL ERROR CHECK
5361 023504 004737 011062          CALL READ ;CALL READ SECTOR
5362 023510 004737 017724          CALL ERRCHK ;CALL ERROR CHECK
5363 023514 004737 020430          CALL CDENC ;CALL COMPLIMENT DENSITY CONTROL
5364 023520 052737 004000 002476      BIS #NEGTST,FLAGST ;SET NEG TEST FLAG
5365 023526 012737 000030 002464      MOV #DENERR,NGTSE ;SETUP EXPECTED NEG TEST ERR=DEN ERR
5366 023534 004737 010744          CALL WRITE ;CALL WRITE SECTOR
5367 023540 004737 017724          CALL ERRCHK ;CALL ERROR CHECK
5368 023544 004737 011062          CALL READ ;CALL READ SECTOR
5369 023550 004737 017724          CALL ERRCHK ;CALL ERROR CHECK
5370 023554 005137 002504          COM TTEMP1 ;COMPLIMENT ENDDO FLAG
5371 023560 004737 020430          CALL CDENC ;CALL COMPLIMENT DENSITY CONTROL
5372 023564 004737 021174          CALL STTK76 ;CALL SET TRACK=76.
5373 023570 005737 002504          UBT7: TST TTEMP1 ;DUNTIL ENDDO FLAG
5374 023574 001331          BNE BBT7 ;EQUALS 0
5375 023576          XT7: EXIT TST
5376 023602          REGTBL CSESAL
(1) 015036          REGS1=CSESAL
5377 023602          TTBL
(2) 023602 023354          T7TBL: .WORD T7MSG
(2) 023604 177777          .WORD -1
(3) 023606          T7RTB:
(3) 023606 015036          .WORD REGS1
(3) 023610 177777          .WORD -1
5378 023612          ENDTST
```


5381
 (2) 023614 000423
 (2) 023616 020040 051127 052111
 (2)
 5382
 5383
 5384
 5385
 5386
 5387
 5388
 5389
 5390
 5391
 5392
 5393
 5394
 5395
 5396
 5397
 5398
 5399
 5400
 5401
 5402
 5403
 5404 023664
 (2) 023664 012737 023772 002466
 (2) 023672 032737 000002 002324
 (2) 023700 001432
 (2) 023702 004737 020700
 5405 023706 004737 021174
 5406 023712 004737 017350
 5407 023716 004737 020472 002402
 5408 023722 012737 000010
 5409 023730 004737 010744
 5410 023734 004737 017724
 5411 023740 004737 011062
 5412 023744 004737 017724
 5413 023750 005037 002402
 5414 023754 004737 010744
 5415 023760 004737 017724
 5416 023764 000240
 5417 023766
 5418 023772
 (1) 015036
 5419 023772
 (2) 023772 023616
 (2) 023774 177777
 (3) 023776
 (3) 023776 015036
 (3) 024000 177777
 5420 024002

.SBTTL TEST 8 - WRITE SECTOR DELETED DATA - FNC TST
 BR BGNT8 ;BR TO BGN TST
 T8MSG: .ASCIZ / WRITE SECTOR DELETED DATA - FNC TST/
 .EVEN

++
 TEST TO VERIFY THAT THE DEVICE WILL WRITE A DELETED DATA MARK ON THE
 DISKETTE WITHOUT ERROR.

```

-----
BGNTST
  IF FUNCTION TEST
  THEN-SETUP TEST IDENT
  SET TRACK=76, SECTOR=10
  CALL DENSITY CHECK
  SET DELETED DATA FLAG
  SET DENSITY CONTROL WORD=DISK DENSITY
  CALL WRITE SECTOR
  CALL ERROR CK
  CALL READ SECTOR SECTOR
  CALL ERROR CK
  CLEAR DELETED DATA FLAG
  CALL WRITE SECTOR
  CALL ERROR CK
  ENDIF
ENDTST
-----

```

```

TSETUP
BGNT8: MOV #T8TBL,TSTID ;SETUP TEST ID TBL-TEST# 8
IAT8: BIT #FUNCT,TSTMOD ;IF TEST MODE=FUNCTION TEST
      BEQ XT8 ;BIT SET, THEN
      CALL FTSTU ;CALL FUNCTION TEST SETUP
      CALL STTK76 ;CALL SET TRACK=76.
      CALL DENCHK ;CALL DENSITY CHECK
      CALL SDENC ;CALL SET DENSITY CONTROL=DENSITY STATUS
      MOV #DLDCMD,DEL DAT ;SETUP DELETED DATA COMMAND MODE
      CALL WRITE ;CALL WRITE SECTOR
      CALL ERRCHK ;CALL ERROR CHECK
      CALL READ ;CALL READ SECTOR
      CALL ERRCHK ;CALL ERROR CHECK
      CLR DEL DAT ;CLEAR DELETED DATA COMMAND MODE
      CALL WRITE ;CALL WRITE SECTOR
      CALL ERRCHK ;CALL ERROR CHECK
      NOP ;
XT8: EXIT TST
      REGTBL CSESAL ;REGS1=CSESAL
      TTBL
T8TBL: .WORD T8MSG
      .WORD -1
T8RTB: .WORD REGS1
      .WORD -1
ENDTST

```

5423
(2) 024004 000414
(2) 024006 020040 042523 020124
(2)
5424
5425
5426
5427
5428
5429
5430
5431
5432
5433
5434
5435
5436
5437
5438
5439
5440
5441
5442
5443
5444
5445
5446
5447
5448
5449
5450
5451
5452
5453
5454
5455

.SBTTL TEST 9 - SET DENSITY - FNC TST
BR BGNT9 ;BR TO BGN TST
T9MSG: .ASCIZ / SET DENSITY - FNC TST/
.EVEN

++
: TEST TO VERIFY THE DEVICE WILL CHANGE DENSITIES WITHOUT INCURRING AN
: ERROR.

: BGNTST
: IF FUNCTION TEST
: THEN-SETUP TEST IDENT
: CALL DENSITY CHECK
: SET DENSITY CONTROL WORD=DISK DENSITY
: CALL SET DENSITY
: CALL ERROR CK
: CALL SET TRACK=76, SECTOR=10
: CALL READ SECTOR
: CALL ERROR CK
: SET TRACK=0
: CALL READ SECTOR
: CALL ERROR CK
: CALL COMPLIMENT DENSITY CONTROL
: CALL SET DENSITY
: CALL ERROR CK
: CALL READ SECTOR
: CALL ERROR CK
: CALL SET TRACK=76., SECTOR=10.
: CALL READ SECTOR
: CALL ERROR CK
: SET DENSITY CONTROL WORD=DISK DENSITY
: CALL SET DENSITY
: CALL ERROR CK
: ENDIF
: ENDTST

```
5458 024036 012737 024210 002466          TSETUP
(2) 024036 032737 000002 002324      BGNT9: MOV      #T9TBL,TSTID      ;SETUP TEST ID TBL-TEST# 9
(2) 024052 001454                                IAT9:  BIT      #FUNCTT,TSTMOD    ;IF TEST MODE=FUNCTION TEST
(2) 024054 004737 020700                        BEQ      XT9                      ;BIT SET, THEN
5459 024060 004737 017350                        CALL    FTSTUP                    ;CALL FUNCTION TEST SETUP
5460 024064 004737 020472                        CALL    DENCHK                    ;CALL DENSITY CHECK
5461 024070 004737 011172                        CALL    SDENC                     ;CALL SET DENSITY CONTROL=DENSITY STATUS
5462 024074 004737 017724                        CALL    SETDN                     ;CALL SET DENSITY
5463 024100 004737 021174                        CALL    ERRCHK                    ;CALL ERROR CHECK
5464 024104 004737 011062                        CALL    STTK76                   ;CALL SET TRACK=76.
5465 024110 004737 017724                        CALL    READ                      ;CALL READ SECTOR
5466 024114 005037 002374                        CALL    ERRCHK                    ;CALL ERROR CHECK
5467 024120 004737 011062                        CLR     TRACK                     ;SET TRACK=0
5468 024124 004737 017724                        CALL    READ                      ;CALL READ SECTOR
5469 024130 004737 020430                        CALL    ERRCHK                    ;CALL ERROR CHECK
5470 024134 004737 011172                        CALL    CDENC                    ;CALL COMPLIMENT DENSITY CONTROL
5471 024140 004737 017724                        CALL    SETDN                     ;CALL SET DENSITY
5472 024144 004737 011062                        CALL    ERRCHK                    ;CALL ERROR CHECK
5473 024150 004737 017724                        CALL    READ                      ;CALL READ SECTOR
5474 024154 004737 021174                        CALL    ERRCHK                    ;CALL ERROR CHECK
5475 024160 004737 011062                        CALL    STTK76                   ;CALL SET TRACK=76.
5476 024164 004737 017724                        CALL    READ                      ;CALL READ SECTOR
5477 024170 004737 020472                        CALL    ERRCHK                    ;CALL ERROR CHECK
5478 024174 004737 011172                        CALL    SDENC                     ;CALL SET DENSITY CONTROL=DENSITY STATUS
5479 024200 004737 017724                        CALL    SETDN                     ;CALL SET DENSITY
5480 024204                                XT9:  CALL    ERRCHK              ;CALL ERROR CHECK
5481 024210                                REGTBL CSESAL
(1) 015036                                REGS1=CSESAL
5482 024210                                TTBL
(2) 024210 024006                                T9TBL: .WORD  T9MSG
(2) 024212 177777                                .WORD  -1
(3) 024214                                T9RTB:
(3) 024214 015036                                .WORD  REGS1
(3) 024216 177777                                .WORD  -1
5483 024220                                ENDTST
```


5486
(2) 024222 000414
(2) 024224 050040 051517 052111
(2) 024254

.SBTTL TEST 10 - POSITIONING - FNC TST
BR BGNT10 :BR TO BGN TST
T10MSG: .ASCIZ / POSITIONING - FNC TST/
.EVEN

5487
5488
5489
5490
5491
5492
5493
5494
5495
5496
5497
5498
5499
5500
5501
5502
5503
5504
5505
5506
5507
5508

:+
: TEST TO VERIFY THE DEVICE WILL CHANGE SECTORS AND TRACKS WITHOUT
: INCURRING AN ERROR.

BGNTST
: IF FUNCTION TEST
: THEN-SETUP TEST IDENT
: SET TRACK PATTERN=RANDOM
: CALL DENSITY CHECK
: SET DENSITY CONTROL WORD=DRV DENSITY
: BGND0
: : CALL GET A TRACK
: : CALL GET A SECTOR
: : CALL READ SECTOR
: : CALL ERROR CK
: DOUNTIL TRACKS DONE FLAG SET
: NOP
: ENDIF
ENDTST

5509 024254
(2) 024254 012737 024352 002466
(2) 024262 032737 000002 002324
(2) 024270 001426
(2) 024272 004737 020700
5510 024276 012737 000400 002510
5511 024304 004737 017350
5512 024310 004737 020472
5513 024314 004737 012662
5514 024320 004737 013104
5515 024324 004737 011062
5516 024330 004737 017724
5517 024334 032737 001000 002476
5518 024342 001764
5519 024344 000240
5520 024346
5521 024352
(1) 015036
5522 024352
(2) 024352 024224
(2) 024354 177777
(3) 024356
(3) 024356 015036
(3) 024360 177777
5523 024362

TSETUP
BGNT10: MOV #T10TBL,TSTID ;SETUP TEST ID TBL-TEST# 10
IAT10: BIT #FUNCTT,TSTMOD ;IF TEST MODE=FUNCTION TEST
BEQ XT10 ;BIT SET, THEN
CALL FTSTUP ;CALL FUNCTION TEST SETUP
MOV #ITK!RTK,TKSCFG ;SET TRK/SEC FLAGS-->TRACK=INIT & RANDOM
CALL DENCHK ;CALL DENSITY CHECK
CALL SDENC ;CALL SET DENSITY CONTROL=DENSITY STATUS
BBT10: CALL GETTRK ;CALL GET TRACK
CALL GETSEC ;CALL GET SECTOR
CALL READ ;CALL READ SECTOR
CALL ERRCHK ;CALL ERROR CHECK
UBT10: BIT #TRKDON,FLAGST ;DOUNTIL FLAGS->TRACK DONE FLAG
BEQ BBT10 ;SET
NOP ;
XT10: EXIT TST
REGTBL CSESAL
REGS1=CSESAL
TTBL
T10TBL: .WORD T10MSG
.WORD -1
T10RTB: .WORD REGS1
.WORD -1
ENDTST

5527
(2) 024364 000412
(2) 024366 041440 051123 041040
(2)
5528
5529
5530
5531
5532
5533
5534
5535
5536
5537
5538
5539
5540
5541
5542
5543
5544
5545
5546
5547
5548
5549
5550
5551
5552
5553
5554
5555
5556
5557

.SBTTL TEST 11 - CSR BITS - LGC TST
BR BGNT11 ;BR TO BGN TST
T11MSG: .ASCIZ / CSR BITS - LGC TST/
.EVEN

:+
TEST TO VERIFY THAT THE READ/WRITE BITS OF THE CONTROL AND STATUS REGISTER CAN BE WRITTEN INTO AND READ AND OTHERWISE BEHAVE AS EXPECTED.

```
-----  
BGNTST  
: IF LOGIC TEST  
: THEN-SETUP TEST TEST IDENT  
: CALL RX LEGAL STATE CHECK  
: WRITE RXCSR-ALL 1'S (EXCEPT BITS #14 & #1)  
: SETUP EXPECTED REGISTER RESULTS  
: IF RXCSR DOES NOT=037566  
: THEN-SETUP ACTUAL REGISTER RESULTS  
: SETUP ERRNBR=CSR ERROR  
: CALL ERROR  
: ENDIF  
: WRITE RXCSR-ALL 0'S  
: IF RXCSR DOES NOT=004040  
: THEN-SETUP ACTUAL REGISTER RESULTS  
: SETUP ERRNBR=CSR ERROR  
: CALL ERROR  
: ENDIF  
: NOP  
: ENDIF  
ENDTST  
-----  
BOARD CALLOUT:  
1. INTERFACE  
-----  
--
```

5560	024412					TSETUP		
(2)	024412	012737	024602	002466	BGNT11:	MOV	#T11TBL,TSTID	:SETUP TEST ID TBL-TEST# 11
(2)	024420	032737	000001	002324	IAT11:	BIT	#LOGICT,TSTMOD	:IF TEST MODE=LOGIC TEST
(2)	024426	001463				BEQ	XT11	:BIT SET, THEN
(2)	024430	004737	020736			CALL	LTSTUP	:CALL LOGIC TEST SETUP
5561	024434					BRESET		:BUS RESET
5562	024436	004737	011610			CALL	WAIT	:WAIT FOR DONE
5563	024442	004737	017140			CALL	DVSTCK	:CALL DEVICE STATE CHECK
5564	024446	012777	137776	155674		MOV	#137776,@RXCS	:WRITE RXCSR=ALL 1'S (EXCEPT BIT#14 & #1)
5565	024454	032737	000400	002500	IDT11:	BIT	#LSIFLG,FLAGSP	:IF LSI FLG - FLAGSP
5566	024462	001404				BEQ	LDT11	:SET, THEN
5567	024464	012737	005560	002436		MOV	#5560,REGEXP	:SETUP EXPECTED REG RESULTS = 5560
5568	024472	000403				BR	IBT11	:BR TO IF 'B'
5569	024474	012737	037566	002436	LDT11:	MOV	#037566,REGEXP	:SETUP EXPECTED REGISTER RESULTS=037566
5570	024502	023777	002436	155640	IBT11:	CMP	REGEXP,@RXCS	:IF RXCSR NOT=EXPECTED REGISTER
5571	024510	001410				BEQ	EBT11	:THEN
5572	024512	017737	155632	002440		MOV	@RXCS,REGACT	:SETUP ACTUAL REGISTER
5573	024520	012737	000033	002520		MOV	#CSRERR,ERRNBR	:SET ERRNBR=CSRERR
5574	024526	004737	003060			CALL	ERROR	:CALL ERROR
5575	024532	012737	004040	002436	EBT11:	MOV	#4040,REGEXP	:SETUP EXPECTED REGISTER RESULTS=4040
5576	024540	012777	000000	155602		MOV	#0,@RXCS	:WRITE RXCSR=ALL 0'S
5577	024546	023777	002436	155574	ICT11:	CMP	REGEXP,@RXCS	:IF RXCSR NOT=EXPECTED REGISTER
5578	024554	001410				BEQ	XT11	:THEN
5579	024556	017737	155566	002440		MOV	@RXCS,REGACT	:SETUP ACTUAL REGISTER
5580	024564	012737	000033	002520		MOV	#CSRERR,ERRNBR	:SET ERRNBR=CSR ERR
5581	024572	004737	003060			CALL	ERROR	:CALL ERROR
5582	024576				XT11:	EXIT	TST	
5583	024602					REGTBL		
5584	024602					TBL	0,RGPRT	
(2)	024602	024366					T11TBL:	.WORD T11MSG
(2)	024604	000000						.WORD 0
(2)	024606	000004						.WORD RGPRT
(2)	024610	177777						.WORD -1
(3)	024612						T11RTB:	.WORD -1
(3)	024612	177777						.WORD -1
5585	024614				FRUTBL	INTONL	T11FTB:	.WORD INTONL
(2)	024614							.WORD -1
(2)	024614	006644						
(2)	024616	177777						
5586	024620					ENDTST		

5589
(2) 024622 000412
(2) 024624 042040 051102 041040
(2)
5590
5591
5592
5593
5594
5595
5596
5597
5598
5599
5600
5601
5602
5603
5604
5605
5606
5607
5608
5609
5610
5611
5612
5613
5614
5615
5616
5617
5618
5619

.SBTTL TEST 12 - DBR BITS - LGC TST
BR BGNT12 ;BR TO BGN TST
T12MSG: .ASCIZ / DBR BITS - LGC TST/
.EVEN

++
: TEST TO VERIFY THAT THE READ/WRITE BITS OF THE DATA BUFFER REGISTER
: CAN BE WRITTEN INTO AND READ AS EXPECTED.

```
-----  
: BGNTST  
: IF LOGIC TEST  
: THEN-SETUP TEST IDENT  
: WRITE RXDBR-ALL 1'S  
: SETUP EXPECTED REGISTER RESULTS  
: IF RXDBR NOT=173767  
: THEN-SETUP ACTUAL REGISTER RESULTS  
: SETUP ERR NBR=DBR ERR  
: CALL ERROR  
: ENDIF  
: WRITE RXDBR-ALL 0'S  
: SETUP EXPECTED REGISTER RESULTS  
: IF RXDBR NOT=000000  
: THEN-SETUP ACTUAL REGISTER RESULTS  
: SET ERRNBR=DBR ERR  
: CALL ERROR  
: ENDIF  
: NOP  
: ENDIF  
: ENDTST  
-----  
: BOARD CALLOUT:  
: 1. INTERFACE  
-----  
:--
```

5622 024650
(2) 024650 012737 025012 002466
(2) 024656 032737 000001 002324
(2) 024664 001450
(2) 024666 004737 020736
5623 024672
5624 024674 004737 011610
5625 024700 012777 177777 155444
5626 024706 012737 173767 002436
5627 024714 023777 002436 155430
5628 024722 001410
5629 024724 017737 155422 002440
5630 024732 012737 000034 002520
5631 024740 004737 003060
5632 024744 005037 002436
5633 024750 012777 000000 155374
5634 024756 023777 002436 155366
5635 024764 001410
5636 024766 017737 155360 002440
5637 024774 012737 000034 002520
5638 025002 004737 003060
5639 025006
5640 025012
(2) 025012 024624
(2) 025014 000000
(2) 025016 000004
(2) 025020 177777
(3) 025022
(3) 025022 177777
5641 025024
(2) 025024
(2) 025024 006644
(2) 025026 177777
5642 025030

TSETUP
BGNT12: MOV #T12TBL,TSTID ;SETUP TEST ID TBL-TEST# 12
IAT12: BIT #LOGICT,TSTMOD ;IF TEST MODE=LOGIC TEST
BEQ XT12 ;BIT SET, THEN
CALL LTSTUP ;CALL LOGIC TEST SETUP
BRESET ;BUS RESET
CALL WAIT ;WAIT FOR DONE
MOV #-1,@RXDB ;WRITE RXDBR-ALL 1'S
MOV #173767,REGEXP ;SETUP EXPECTED REGISTER RESULTS=173767
IBT12: CMP REGEXP,@RXDB ;IF RXDBR NOT=EXPECTED REGISTER
BEQ EBT12 ;THEN
MOV @RXDB,REGACT ;SETUP ACTUAL REGISTER RESULTS
MOV #DBRERR,ERRNBR ;SET ERRNBR=DBR ERR
CALL ERROR ;CALL ERROR
EBT12: CLR REGEXP ;SETUP EXPECTED REGISTER RESULTS=0'S
MOV #0,@RXDB ;WRITE RXDBR=ALL 0'S
ICT12: CMP REGEXP,@RXDB ;IF RXDBR NOT=EXPECTED REGISTER
BEQ XT12 ;THEN
MOV @RXDB,REGACT ;SETUP ACTUAL REGISTER RESULTS
MOV #DBRERR,ERRNBR ;SET ERRNBR=DBR ERR
CALL ERROR ;CALL ERROR
XT12: EXIT
TST
TTBL 0, RGPRT
T12TBL: .WORD T12MSG
.WORD 0
.WORD RGPRT
.WORD -1
T12RTB: .WORD -1
FRUTBL INTONL
T12FTB: .WORD INTONL
.WORD -1
ENDTST

5645
(2) 025032 000420
(2) 025034 041440 051123 042055
(2) 025074
5646
5647
5648
5649
5650
5651
5652
5653
5654
5655
5656
5657
5658
5659
5660
5661
5662
5663
5664
5665
5666
5667
5668
5669
5670
5671
5672
5673
5674
5675

.SBTTL TEST 13 - CSR-DBR COMMON BYTE - LGC TST
BR BGNT13 ;BR TO BGN TST
T13MSG: .ASCIZ / CSR-DBR COMMON BYTE - LGC TST/
.EVEN

++
: TEST TO VERIFY THAT THE LOWER BYTE OF THE RXCS MAPS INTO THE RXDB AND
: THEREFORE CHECK WRITE ONLY BITS OF THE RXCS.

BGNTST
: IF LOGIC TEST
: THEN-SETUP TEST IDENT
: WRITE RXCSR LOW BYTE-ALL 1'S (EXCEPT BIT #1)
: SETUP EXPECTED REGISTER RESULTS
: IF RXDBR LOW BYTE NOT=376
: THEN-SETUP ACTUAL REGISTER RESULTS
: SETUP ERR NBR=CSR ERR
: CALL ERROR
: ENDIF
: WRITE RXCSR LOW BYTE-ALL 0'S
: SETUP EXPECTED REGISTER RESULTS
: IF RXDBR LOW BYTE NOT=000
: THEN-SETUP ACTUAL REGISTER RESULTS
: SETUP ERR NBR=CSR ERR
: CALL ERROR
: ENDIF
ENDIF
ENDTST

: BOARD CALLOUT:
: 1. INTERFACE

--

5678	025074					TSETUP		
(2)	025074	012737	025242	002466	BGNT13:	MOV	#T13TBL,TSTID	:SETUP TEST ID TBL-TEST# 13
(2)	025102	032737	000001	002324	IAT13:	BIT	#LOGICT,TSTMOD	:IF TEST MODE=LOGIC TEST
(2)	025110	001452				BEQ	XT13	:BIT SET, THEN
(2)	025112	004737	020736			CALL	LTSTUP	:CALL LOGIC TEST SETUP
5679	025116					BRESET		:BUS RESET
5680	025120	004737	011610			CALL	WAIT	:WAIT FOR DONE
5681	025124	012777	000376	155216		MOV	#376,@RXCS	:WRITE RXCSR LOW BYTE-ALL IF (EXCEPT BIT #1)
5682	025132	012737	000366	002436		MOV	#366,@GEXP	:SETUP EXPECTED REGISTER RESULTS=366
5683	025140	123777	002436	155204	IBT13:	CMPB	REGEXP,@RXDB	:IF RXDBR LOW BYTE NOT=376
5684	025146	001413				BEQ	EBT13	:THEN
5685	025150	117737	155176	002440		MOVB	@RXDB,REGACT	:SETUP ACTUAL REGISTER RESULTS
5686	025156	042737	177400	002440		BIC	#177400,REGACT	:CLEAR TOP BYTE
5687	025164	012737	000033	002520		MOV	#CSRERR,ERRNBR	:SET ERRNBR=CSR ERR
5688	025172	004737	003060			CALL	ERROR	:CALL ERROR
5689	025176	005037	002436		EBT13:	CLR	REGEXP	:SETUP EXPECTED REGISTER RESULTS=0'S
5690	025202	112777	000000	155140		MOVB	#0,@RXCS	:WRITE RXDBR=ALL 0'S
5691	025210	123777	002436	155134	ICT13:	CMPB	REGEXP,@RXDB	:IF RXDBR NOT=EXPECTED RESULTS
5692	025216	001407				BEQ	XT13	
5693	025220	005037	002440			CLR	REGACT	:SETUP ACTUAL REGISTER RESULTS
5694	025224	012737	000033	002520		MOV	#CSRERR,ERRNBR	:SETUP ERRNBR=CSR ERR
5695	025232	004737	003060			CALL	ERROR	:CALL ERROR
5696								
5697	025236				XT13:	EXIT	TST	
5698								
5699	025242					TTBL	0,RGPRT	
(2)	025242	025034						T13TBL: .WORD T13MSG
(2)	025244	000000						.WORD 0
(2)	025246	000004						.WORD RGPRT
(2)	025250	177777						.WORD -1
(3)	025252							T13RTB: .WORD -1
(3)	025252	177777						
5700	025254					FRUTBL	INTONL	
(2)	025254							T13FTB: .WORD INTONL
(2)	025254	006644						.WORD -1
(2)	025256	177777						
5701	025260					ENDTST		

5704
(2) 025262 000415
(2) 025264 041040 051525 044440
(2)
5705
5706
5707
5708
5709
5710
5711
5712
5713
5714
5715
5716
5717
5718
5719
5720
5721
5722
5723
5724
5725
5726
5727
5728

.SBTTL TEST 14 - BUS INITIALIZE - LGC TST
BR BGNT14 ;BR TO BGN TST
T14MSG: .ASCIZ / BUS INITIALIZE - LGC TST/
.EVEN

```

:++
: TEST TO VERIFY THAT THE INTERFACE BOARD WILL COMPLETE A BUS INITIALIZE.
:-----
: BGNTST
:   IF LOGIC TEST
:   : THEN-SETUP TEST IDENT
:   :   ISSUE BUS INITIALIZE
:   :   IF RXCSR ERROR BIT SET
:   :   : THEN-IF RXESR AC LOW BIT SET
:   :   :   THEN-SETUP ERROR
:   :   :   CALL ERROR
:   :   ENDF
:   : ENDF
:   : NOP
: ENDF
: ENDTST
:-----
: BOARD CALLOUT:
: 1. INTERFACE
: 2. CONTROLLER
:-----
:--

```

5731									
5732	025316					TSETUP			
(2)	025316	012737	025462	002466	BGNT14:	MOV	#T14TBL,TSTID	;SETUP TEST ID TBL-TEST# 14	
(2)	025324	032737	000001	002324	IAT14:	BIT	#LOGICT,TSTMOD	;IF TEST MODE=LOGIC TEST	
(2)	025332	G01451				BEG	XT14	;BIT SET, THEN	
(2)	025334	004737	020736			CALL	LTSTUP	;CALL LOGIC TEST SETUP	
5733	025340					BRESET			
5734	025342	032777	100000	155000	IBT14:	BIT	#ERRBIT,@RXCS	;IF RXCSR ERROR BIT	
5735	025350	001442				BEG	XT14	;SET, THEN	
5736	025352	032777	000004	154772	ICT14:	BIT	#INITDN,@RXDB	;IF RXESR=INIT DONE	
5737	025360	001406				BEG	IDT14	;SET, THEN	
5738	025362	012737	000061	002520		MOV	#NOITDB,ERRNBR	;SET ERR NBR=NO INIT DONE-BUS	
5739	025370	004737	003060			CALL	ERROR	;CALL ERROR	
5740	025374	000430				BR	XT14	;BR TO EXIT	
5741	025376	032777	000030	154746	IDT14:	BIT	#DENERR,@RXDB	;IF RXESR=DENSITY ERR	
5742	025404	001406				BEG	IET14	;SET, THEN	
5743	025406	012737	000020	002520		MOV	#DENDSK,ERRNBR	;SET ERR NBR=DISK DEN ERR	
5744	025414	004737	003060			CALL	ERROR	;CALL ERROR	
5745	025420	000416				BR	XT14	;BR TO EXIT	
5746	025422	032777	000010	154722	IET14:	BIT	#ACLOW,@RXDB	;IF RXESR NOT=INITIALIZE DONE BIT	
5747	025430	001006				BNE	LET14	;SET, THEN	
5748	025432	012737	000050	002520		MOV	#ACLOWF,ERRNBR	;SET ERR NBR=NO INIT DONE-BIT	
5749	025440	004737	003060			CALL	ERROR	;CALL ERROR	
5750	025444	000404				BR	XT14	;BR TO EXIT	
5751	025446	004737	011340		LET14:	CALL	RDERCD	;CALL READ ERROR CODE	
5752	025452	004737	017724			CALL	ERRCHK	;CALL ERROR CHECK	
5753	025456				XT14:	EXIT	TST		
5754	025462					REGTBL			
5755	025462					TTBL	REGCK,0		
(2)	025462	025264					T14TBL:	.WORD	T14MSG
(2)	025464	000001						.WORD	REGCK
(2)	025466	000000						.WORD	0
(2)	025470	177777						.WORD	-1
(3)	025472						T14RTB:		
(3)	025472	177777						.WORD	-1
5756	025474				FRUTBL	INFCTL	T14FTB:		
(2)	025474							.WORD	INFCTL
(2)	025474	006640						.WORD	-1
(2)	025476	177777							
5757	025500				ENDTST				

5760
 (2) 025502 000421
 (2) 025504 050040 047522 051107
 (2) 025546
 5761
 5762
 5763
 5764
 5765
 5766
 5767
 5768
 5769
 5770
 5771
 5772
 5773
 5774
 5775
 5776
 5777
 5778
 5779
 5780
 5781
 5782 025546
 (2) 025546 012737 025610 002466
 (2) 025554 032737 000001 002324
 (2) 025562 001410
 (2) 025564 004737 020736
 5783 025570 004737 010440
 5784 025574 004737 017140
 5785 025600 004737 017724
 5786 025604
 5787
 5788 025610
 (1) 015056
 5789 025610
 (2) 025610 025504
 (2) 025612 000001
 (2) 025614 000000
 (2) 025616 177777
 (3) 025620
 (3) 025620 015056
 (3) 025622 177777
 5790 025624
 (2) 025624
 (2) 025624 006640
 (2) 025626 177777
 5791 025630

.SBTTL TEST 15 - PROGRAMMED INITIALIZE - LGC TST
 BR BGNT15 ;BR TO BGN TST
 T15MSG: .ASCIZ / PROGRAMMED INITIALIZE - LGC TST/
 .EVEN

..
 : TEST TO VERIFY THAT THE INTERFACE BOARD WILL COMPLETE A PROGRAMMED
 : INITIALIZE.

 : BGNTST
 : IF LOGIC TEST
 : THEN-SETUP TEST IDENT
 : CALL PROGRAMMED INITIALIZE
 : CALL DEVICE STATE CK
 : CALL ERROR CHECK
 : NOP
 : ENDIF
 : ENDTST

BOARD CALLOUT:
 1. INTERFACE
 2. CONTROLLER

```

TSETUP
BGNT15: MOV #T15TBL,TSTID ;SETUP TEST ID TBL-TEST# 15
IAT15: BIT #LOGICT,TSTMOD ;IF TEST MODE=LOGIC TEST
      BEQ XT15 ;BIT SET, THEN
      CALL LTSTUP ;CALL LOGIC TEST SETUP
      CALL INTIAL ;CALL PROG INITIALIZE
      CALL DVSTCK ;CALL DEVICE STATE CK
      CALL ERRCHK ;CALL ERROR CHECK
XT15: EXIT TST

REGTBL CSESIT REGS1=CSESIT
TTBL REGCK,0
T15TBL: .WORD T15MSG
        .WORD REGCK
        .WORD 0
        .WORD -1
T15RTB: .WORD REGS1
        .WORD -1
FRUTBL INFCTL
T15FTB: .WORD INFCTL
        .WORD -1
ENDTST
  
```

5794
(2) 025632 000413
(2) 025634 050040 053517 051105
(2)

.SBTTL TEST 16 - POWER FAIL - LGC TST
BR BGNT16 :BR TO BGN TST
T16MSG: .ASCIZ / POWER FAIL - LGC TST/
.EVEN

5795
5796
5797
5798
5799
5800
5801
5802
5803
5804
5805
5806
5807
5808
5809
5810
5811
5812
5813
5814
5815
5816
5817
5818
5819
5820
5821
5822
5823
5824

```

:++
: TEST TO VERIFY THAT THE ACLOW CIRCUITS OPERATE AS EXPECTED.
-----
: BGNTST
:   IF LOGIC TEST [A]
:   : THEN-SETUP TEST IDENT
:   :   IF MANUAL INTERVENTION ALLOWED [B]
:   :   : THEN-ASK OPERATOR TO POWER DOWN RX02 ONLY
:   :   :   IF OPERATION COMPLETE [C]
:   :   :   THEN-CALL PROGRAMMED INITIALIZE
:   :   :   : SETUP EXPECTED ERROR=AC LOW
:   :   :   : SET NEG TEST FLAG=TEST FLAGS
:   :   :   : CALL ERROR CHECK
:   :   :   : ASK OPERATOR TO POWER UP RX02
:   :   :   : IF OPERATION COMPLETE [D]
:   :   :   :   THEN-CLEAR OUT EXPECTED ERROR
:   :   :   :   : CLEAR NEG TEST FLAG=TEST FLAGS
:   :   :   :   : CALL INITIAL
:   :   :   :   : CALL ERROR CHECK
:   :   :   :   ENDIF
:   :   :   ENDIF
:   :   ENDIF
:   ENDIF
: ENDTST
-----
: BOARD CALLOUT:
: 1. INTERFACE
-----

```

```
5827 025662          TSETUP
(2) 025662 012737 026102 002466  BGNT16: MOV      #T16TBL,TSTID ;SETUP TEST ID TBL-TEST# 16
(2) 025670 032737 000001 002324  IAT16:  BIT      #LOGICT,TSTMOD ;IF TEST MODE=LOGIC TEST
(2) 025676 001477          ;BEQ      XT16 ;BIT SET, THEN
(2) 025700 004737 020736  ;CALL    LTSTUP ;CALL LOGIC TEST SETUP
5828 025704 005037 002504  ;CLR     TTEMP1 ;SET TEMP1=0
5829 025710          IBT16:  MANUAL          ;IF MANUAL INTERVENTION
5830 025712          BNCOMPLETE XT16 ;ALLOWED, THEN
5831 025714          PRINTF #PWRMS,UNIT ;PRINT MSG
5832 025740          GMANIL PWDNRY,TTEMP1,1,YES
5833 025754 032737 000001 002504  ICT16:  BIT      #1,TTEMP1 ;IF RX02 IS
5834 025762 001445          ;BEQ      XT16 ;POWERED DOWN, THEN
5835 025764 004737 010440          ;CALL    INTIAL ;CALL INITIALIZE
5836 025770 012737 000050 002464  ;MOV     #ACLOWF,NGTSER ;SETUP EXPECTED ERROR=AC LOW
5837 025776 052737 004000 002476  ;BIS     #NEGTST,FLAGST ;SET NEG TEST FLAG=TEST FLAGS
5838 026004 004737 017724          ;CALL    ERRCHK ;CALL ERROR CHECK
5839 026010          PRINTF #PWRMS,UNIT ;PRINT MSG
5840 026034          GMANIL PWUPRY,TTEMP1,1,YES
5841 026050 032737 000002 002504  IDT16:  BIT      #2,TTEMP1 ;IF RX02 IS
5842 026056 001407          ;BEQ      XT16 ;POWERED UP, THEN
5843 026060 042737 004000 002476  ;BIC     #NEGTST,FLAGST ;CLEAR NEG TEST FLAG=TEST FLAGS
5844 026066 004737 010440          ;CALL    INTIAL ;CALL INITIALIZE
5845 026072 004737 017724          ;CALL    ERRCHK ;
5846 026076          XT16:  EXIT     TST
5847 026102          REGTBL CSESIT
(1) 015056          REGS1=CSESIT
5848 026102          TTBL   REGCK,0
(2) 026102 025634          T16TBL: .WORD   T16MSG
(2) 026104 000001          ;.WORD   REGCK
(2) 026106 000000          ;.WORD   0
(2) 026110 177777          ;.WORD   -1
(3) 026112          T16RTB:
(3) 026112 015056          ;.WORD   REGS1
(3) 026114 177777          ;.WORD   -1
5849 026116          FRUTBL INTONL
(2) 026116          T16FTB:
(2) 026116 006644          ;.WORD   INTONL
(2) 026120 177777          ;.WORD   -1
5850
5851 026122 047045 040445 044440  PWRMS:  .ASCIZ  /%N% IS FLOPPY SYSTEM CONTAINING UNIT #%02/
5852 026175 040 050040 053517  PWDNRY: .ASCIZ  / POWERED DOWN/
5853 026214 020040 047520 042527  PWUPRY: .ASCIZ  / POWERED UP/
5854 026232          .EVEN
5855
5856 026232          ENDTST
```


5859
(2) 026234 000420
(2) 026236 041440 047117 051124
(2)
5860
5861
5862
5863
5864
5865
5866
5867
5868
5869
5870
5871
5872
5873
5874
5875
5876
5877
5878
5879
5880
5881
5882
5883
5884
5885

.SBTTL TEST 17 - CONTROLLER-INTERFACE - LGC TST
BR BGNT17 ;BR TO BGN TST
T17MSG: .ASCIZ / CONTROLLER-INTERFACE - LGC TST/
.EVEN

..**
: TEST TO VERIFY THAT THE INTERFACE BOARD STATE SEQUENCER IS FUNCTIONAL.
: ALSO TO VERIFY THE CONTROLLER-INTERFACE HANDSHAKE BY TRYING FUNCTIONS
: WITH MINIMUM READ/WRITE BOARD INVOLVEMENT.

: BGNTST
: IF LOGIC TEST
: THEN-SETUP TEST IDENT
: SET PROTOCOL CHECK (TEST SETUP SETS VIS TEST TABLE)
: CALL READ ERROR CODE
: CALL ERROR CHECK
: CALL FILL BUFFER
: CALL ERROR CHECK
: CALL EMPTY BUFFER
: CALL ERROR CHECK
: CALL READ MAINT STATUS
: CALL ERROR CHECK
: ENDIF
: ENDTST

: BOARD CALLOUT:
: 1. CONTROLLER
: 2. INTERFACE

: --

```
5888 026276          TSETUP
(2) 026276 012737 026404 002466      BGNT17: MOV      #T17TBL,TSTID      ;SETUP TEST ID TBL-TEST# 17
(2) 026304 032737 000001 002324      IAT17:  BIT      #LOGICT,TSTMOD    ;IF TEST MODE=LOGIC TEST
(2) 026312 001432          BEQ      XT17              ;BIT SET, THEN
(2) 026314 004737 020736          CALL     LTSTUP           ;CALL LOGIC TEST SETUP
5889 026320 052737 000200 002500      BIS      #RECTST,FLAGSP     ;SET READ ERROR CODE TEST=FLAGSP
5890 026326 004737 011340          CALL     RDERCD          ;CALL READ ERROR CODE
5891 026332 004737 017724          CALL     ERRCHK          ;CALL ERROR CHECK
5892 026336 042737 000200 002500      BIC      #RECTST,FLAGSP     ;CLEAR READ ERROR CODE TEST=FLAGSP
5893 026344 004737 010510          CALL     FILBUF          ;CALL FILL BUFFER
5894 026350 004737 017724          CALL     ERRCHK          ;CALL ERROR HECK
5895 026354 004737 010626          CALL     EMPBUF          ;CALL EMPTY BUFFER
5896 026360 004737 017724          CALL     ERRCHK          ;CALL ERROR CHECK
5897 026364 005237 002470          INC      TCMDCCT         ;INCREMENT TST COMMAND CTR      *****
5898 026370 004737 011266          CALL     RDSTAT          ;CALL READ MAINTENANCE STATUS
5899 026374 004737 017724          CALL     ERRCHK          ;CALL ERROR CHECK
5900 026400          XT17:  EXIT      TST
5901
5902 026404          REGTBL  CSESND,CSESRS      REGS1=CSESND
(1)          015076          REGS2=CSESRS
(1)          015066
5903 026404          TTBL    REGCK,PROPRT
(2) 026404 026236          T17TBL: .WORD    T17MSG
(2) 026406 000001          .WORD    REGCK
(2) 026410 000010          .WORD    PROPRT
(2) 026412 177777          .WORD    -1
(3) 026414          T17RTB:
(3) 026414 015076          .WORD    REGS1
(3) 026416 015066          .WORD    REGS2
(3) 026420 177777          .WORD    -1
5904 026422          FRUTBL  CTLINF
(2) 026422          T17FTB:
(2) 026422 006646          .WORD    CTLINF
(2) 026424 177777          .WORD    -1
5905 026426          ENDTST
```

5908
(2) 026430 000410
(2) 026432 047040 051120 026440
(2) 026452
5909
5910
5911
5912
5913
5914
5915
5916
5917
5918
5919
5920
5921
5922
5923
5924
5925
5926
5927
5928
5929
5930
5931
5932
5933
5934
5935
5936
5937
5938
5939
5940
5941
5942
5943
5944
5945
5946

.SBTTL TEST 18 - NPR - LGC TST
BR BGNT18 ;BR TO BGN TST
T18MSG: .ASCIZ / NPR - LGC TST/
.EVEN

++
: TEST TO VERIFY THAT THE NPR LOGIC WILL STORE A WORD IN MEMORY.

: BGNST
: IF LOGIC TEST
: THEN-SETUP TEST IDENT
: SET ERROR CODE STORAGE=1'S
: CALL READ ERROR CODE
: IF ERROR CODE STORAGE=1'S
: THEN-CALL ERROR
: ENDIF
: SET WORD COUNT=128.
: SET DATA PATTERN=0'S
: CALL SET DATA PATTERN
: DATA BUFFER AREA=1'S (BEGIN, END & END+1)
: SET DENSITY CONTROL=DOUBLE DENSITY
: CALL FILL BUFFER
: CALL LOGIC TEST ERROR CK
: CALL EMPTY BUFFER
: CALL ERROR CK
: IF BEGIN DATA BUFFER AREA NOT=0'S
: THEN-SETUP NPR ERROR
: CALL ERROR
: ENDIF
: IF END DATA BUFFER AREA NOT=0'S
: THEN-CALL NPR ERROR
: CALL ERROR
: ENDIF
: IF END+1 DATA BUFFER NOT=1'S
: THEN-CALL NPR ERROR
: CALL ERROR
: ENDIF
: ENDTST

--

5949	026452					TSETUP		
(2)	026452	012737	026752	002466	BGNT18:	MOV	#T18TBL,TSTID	:SETUP TEST ID TBL-TEST# 18
(2)	026460	032737	000001	002324	IAT18:	BIT	#LOGICT,TSTMOD	:IF TEST MODE=LOGIC TEST
(2)	026466	001527				BEQ	XT18	:BIT SET, THEN
(2)	026470	004737	020736			CALL	LTSTUP	:CALL LOGIC TEST SETUP
5950	026474	012737	177777	002442		MOV	#-1,XERUUT	:SET READ ERROR CODE STORAGE=1'S
5951	026502	004737	011340			CALL	RDERCD	:CALL READ ERROR CODE
5952	026506	022737	177777	002442	IBT18:	CMP	#-1,XERUUT	:IF READ ERROR CODE STORAGE=1'S
5953	026514	001005				BNE	EBT18	:THEN
5954	026516	012737	000053	002520		MOV	#NPRERR,ERRNBR	:SET ERR NUMBER=NPR ERROR
5955	026524	004737	003060			CALL	ERROR	:CALL ERROR
5956	026530	042737	000200	002476	EBT18:	BIC	#RECF LG,FLAGST	:CLEAR RED ERR COD FLG = FLAGS TST
5957	026536	012737	000200	002370		MOV	#128,WDCNT	:SET DEVICE WORD COUNT=128
5958	026544	012737	000001	012660		MOV	#1,PAT	:SET DATA PAT=ALL ZEROS
5959	026552	004737	012306			CALL	STDATP	:CALL SET DATA PATTERN
5960	026556	012702	177777			MOV	#-1,R2	:SET R2=ALL 1'S
5961	026562	013737	002370	002504		MOV	WDCNT,TTEMP1	:SET TEMP1=WORD COUNT
5962	026570	006337	002504			ASL	TTEMP1	:DOUBLE IT FOR ADDRESSING WORDS IN MEM
5963	026574	162737	000004	002504		SUB	#4,TTEMP1	:ADJUST TO END OF BUFFER
5964	026602	013701	002504			MOV	TTEMP1,R1	:SET R1=TEMP1
5965	026606	010237	036622			MOV	R2,DATBUF	:SET DATA BUFFER BEGIN=1'S
5966	026612	110261	036622			MOVB	R2,DATBUF(R1)	:SET DATA BUFFER END=1'S
5967	026616	005201				INC	R1	:BUMP INDEX
5968	026620	110261	036622			MOVB	R2,DATBUF(R1)	:SET DATA BUFFER END +1=1'S
5969	026624	012737	000400	002412		MOV	#DENBIT,DENSTY	:SET DENSITY CONTROL=DOUBLE DENSITY
5970	026632	004737	010510			CALL	FILBUF	:CALL FILL BUFFER
5971	026636	004737	017724			CALL	ERRCHK	:CALL ERROR CHECK
5972	026642	004737	010626			CALL	EMPBUF	:CALL EMPTY BUFFER
5973	026646	004737	017724			CALL	ERRCHK	:CALL ERROR CHECK
5974	026652	005737	036622		ICT18:	TST	DATBUF	:IF DATA BUFFER BEGIN
5975	026656	001406				BEQ	ECT18	:NOT=0, THEN
5976	026660	012737	000053	002520		MOV	#NPRERR,ERRNBR	:SETUP ERRNBR=NPR ERR
5977	026666	004737	003060			CALL	ERROR	:CALL ERROR
5978	026672	000425				BR	XT18	:BR TO EXIT
5979	026674	013701	002504		ECT18:	MOV	TTEMP1,R1	:SET R1=TEMP1
5980	026700	105761	036622		IDT18:	TSTB	DATBUF(R1)	:IF DATA BUFFER END
5981	026704	001406				BEQ	EDT18	:NOT=0, THEN
5982	026706	012737	000053	002520		MOV	#NPRERR,ERRNBR	:SETUP ERRNBR=NPR ERR
5983	026714	004737	003060			CALL	ERROR	:CALL EPROR
5984	026720	000412				BR	XT18	:BR TO EXIT
5985	026722	005201			EDT18:	INC	R1	:BUMP INDEX
5986	026724	126127	036622	177777		CMPB	DATBUF(R1),#-1	:IF DATA BUFFER END +1
5987	026732	001405				BEQ	XT18	:NOT=1'S, THEN
5988	026734	012737	000053	002520		MOV	#NPRERR,ERRNBR	:SETUP ERRNBR=NPR ERR
5989	026742	004737	003060			CALL	ERROR	:CALL ERROR
5990	026746				XT18:	EXIT	TST	
5991	026752					REGTBL	CSESND	
(1)		015076						REGS1=CSESND
5992	026752				TTBL	REGCK,0		
(2)	026752	026432					T18TBL:	.WORD T18MSG
(2)	026754	000001						.WORD REGCK
(2)	026756	000000						.WORD 0
(2)	026760	177777						.WORD -1
(3)	026762						T18RTB:	
(3)	026762	015076						.WORD REGS1
(3)	026764	177777						.WORD -1

5993 026766
(2) 026766
(2) 026766 006640
(2) 026770 177777
5994 026772

FRUTBL INFCTL

T18FTB:

.WORD INFCTL
.WORD -1

ENDTST

5997
 5998
 5999
 6000
 6001
 6002
 6003
 6004
 6005
 6006
 6007
 6008
 6009
 6010
 6011
 6012
 6013
 6014
 6015
 6016
 6017
 6018
 6019
 6020
 6021
 6022
 6023
 6024
 6025
 6026
 6027
 6028
 6029
 6030
 6031
 6032
 6033
 6034
 6035
 6036
 6037
 6038
 6039
 6040
 6041

026774	000240		
026776	022737	000002	027100
027004	103014		
027006	005737	027100	
027012	001003		
027014	005037	027102	
027020	000403		
027022	012737	000002	027102
027030	005237	027100	
027034	000420		
027036	005237	027100	
027042	006337	027102	
027046	022737	040000	027102
027054	101407		
027056	005037	027100	
027062	005037	027102	
027066	052737	000010	002476
027074	000240		
027076	000207		
027100	000000		
027102	000000		

```

.SBTTL - MOD U.SFT.NAT - ADDRESS NPR ADDRESS TEST
-----
BGNSUB      NAT
:          NOP
:          IF CTR < 2
:            THEN-IF CTR=0
:              : THEN-CLEAR ADR
:              : ELSE-ADR=ADR+2
:            ENDIF
:          ELSE-INCREMENT COUNTER
:              DOUBLE ADR (ADR=2XADR)
:              IF ADR > 40000
:                THEN-SET DONE IN FLAGS
:                CLEAR CTR
:                CLEAR ADR
:                SET DO LOOP DONE FLAG
:          ENDIF
:        ENDIF
:      ENDSUB
-----
NAT:      NOP
IANAT:    CMP      #2,NATCTR      :IF CTR
:          BHS     LANAT          :LESS THAN 2, THEN
IBNAT:    TST     NATCTR         :IF CTR
:          BNE     LBNAT          :EQUALS 0, THEN
:          CLR     NATADR         :CLEAR ADRS
:          BR      EBNAT          :BR TO END 'B'
LBNAT:    MOV     #2,NATADR       :ELSE, SET ADR=2
EBNAT:    INC     NATCTR         :INCREMENT COUNTER
:          BR      EANAT          :BR TO END 'A'
LANAT:    INC     NATCTR         :INCREMENT COUNTER
:          ASL     NATADR         :DOUBLE ADDRESS
ICNAT:    CMP     #40000,NATADR  :IF ADDRESS
:          BLOS    ECNAT          :GREATER THAN 40000, THEN
:          CLR     NATCTR         :CLEAR COUNTER
:          CLR     NATADR         :CLEAR ADDRESS
:          BIS     #DLPDN,FLAGST  :SET DO LOOP DONE FLAG
ECNAT:    NOP
EANAT:    RETURN                :RETURN
-----
NATCTR:  0                      :COUNTER
NATADR:  0                      :ADDRESS
-----
    
```


6044
(2) 027104 000420
(2) 027106 047040 051120 047040
(2)
6045
6046
6047
6048
6049
6050
6051
6052
6053
6054
6055
6056
6057
6058
6059
6060
6061
6062
6063
6064
6065
6066
6067
6068
6069

.SBTTL TEST 19 - NPR NON-EXISTENT MEM - LGC TST
BR BGNT19 ;BR TO BGN TST
T19MSG: .ASCIZ / NPR NON-EXISTENT MEM - LGC TST/
.EVEN

:+
: TEST TO VERIFY THAT THE NPR NON-EXISTEND MEMORY LOGIC WILL TIME OUT
: WHEN GIVEN AN ILLEGAL ADDRESS.

BGNTST
: IF LOGIC TEST
: THEN-SETUP TEST IDENT
: IF NOT FONZ WITH 124K
: THEN-SETUP BUS TRAPS
: SETUP NON EXISTENT ADDRESS
: CALL READ ERROR CODE
: IF RXCSR ERROR BIT OR RXESR NON-EXISTENT MEMORY BIT NOT
: THEN-CALL LOGIC TEST ERROR
: ENDIF
: CLEAR ERROR SET BY TRAP
: CLEAR BUS TRAP VECTOR
: ENDIF
ENDIF
ENDTST

: BOARD CALLOUT:
: 1. INTERFACE

:-

6072	027146					TSETUP			
(2)	027146	012737	027316	002466		BGNT19: MOV	#T19TBL,TSTID	:	SETUP TEST ID TBL-TEST# 19
(2)	027154	032737	000001	002324		IAT19: BIT	#LOGICT,TSTMOD	:	IF TEST MODE=LOGIC TEST
(2)	027162	001447				BEQ	XT19	:	BIT SET, THEN
(2)	027164	004737	020736			CALL	LTSTUP	:	CALL LOGIC TEST SETUP
6073	027170	032737	010000	002500		IBT19: BIT	#FONZFG,FLAGSP	:	IF FONZ 124K FLAG
6074	027176	001041				BNE	XT19	:	NOT SET, THEN
6075	027200	005037	002452			CLR	ABORT	:	CLEAR ABORT FLAG
6076	027204					SETVEC	#BTRP4,#TRAP,#PRI07	:	
6077	027232	013737	002344	002364		MOV	NXMADR,RECADR	:	SETUP NON EXISTENT MEMORY ADR
6078	027240	004737	011340			CALL	RDERCD	:	CALL READ ERROR CODE
6079	027244	012737	000052	002464		MOV	#NXMERR,NGTSER	:	SETUP EXPECTED NEG TEST ERR=NXM ERR
6080	027252	042737	000200	002476		BIC	#RECFLG,FLAGST	:	CLEAR READ ERR CODE FLAG (SU ERR CODE NOT EVALU
6081	027260	004737	017724			CALL	ERRCHK	:	CALL ERROR CHECK
6082	027264	005037	002452			CLR	ABORT	:	FLAG
6083	027270					CLRVEC	#BTRP4	:	
6084	027276	004737	010440			CALL	INTIAL	:	CALL PROG INITIALIZE
6085	027302					XT19: EXIT	TST	:	
6086	027306					REGTBL		:	
6087	027306					REGTB	1,404,400,NXMBIT,173777	:	CHECK ERR BIT & NXM ERR SET
(2)	027306	004040				T19RT1: .WORD	4040	:	RXCSR SHOULD BE
(2)	027310	000400				.WORD	400	:	RXCSR DONT CARE
(2)	027312	004000				.WORD	NXMBIT	:	RXESR SHOULD BE
(2)	027314	173777				.WORD	173777	:	RXESR DONT CARE
6088	027316					TTBL	REGCK!NEGTST,0	:	
(2)	027316	027106				T19TBL: .WORD	T19MSG	:	
(2)	027320	004001				.WORD	REGCK!NEGTST	:	
(2)	027322	000000				.WORD	0	:	
(2)	027324	177777				.WORD	-1	:	
(3)	027326					T19RTB: .WORD		:	
(3)	027326	027306				.WORD	T19RT1	:	
(3)	027330	177777				.WORD	-1	:	
6089	027332					FRUTBL	INTONL	:	
(2)	027332					T19FTB: .WORD		:	
(2)	027332	006644				.WORD	INTONL	:	
(2)	027334	177777				.WORD	-1	:	
6090	027336					ENDTST		:	

6093
 (2) 027340 000413
 (2) 027342 044440 052116 051105
 (2) 027370
 6094
 6095
 6096
 6097
 6098
 6099
 6100
 6101
 6102
 6103
 6104
 6105
 6106
 6107
 6108
 6109
 6110
 6111
 6112
 6113
 6114
 6115 027370
 (2) 027370 012737 027462 002466
 (2) 027376 032737 000001 002324
 (2) 027404 001424
 (2) 027406 004737 020736
 6116 027412 005037 012024
 6117 027416
 6118 027424 052777 000100 152716
 6119 027432 004737 011662
 6120 027436 004737 003060
 6121 027442 042777 000100 152700
 6122 027450 012737 000010 012024
 6123 027456
 6124 027462
 6125 027462
 (2) 027462 027342
 (2) 027464 000000
 (2) 027466 000000
 (2) 027470 177777
 (3) 027472
 (3) 027472 177777
 6126 027474
 (2) 027474
 (2) 027474 006644
 (2) 027476 177777
 6127 027500

.SB.TL TEST 20 - INTERRUPT - LGC TST
 BR BGNT20 ;BR TO BGN TST
 T20MSG: .ASCIZ / INTERRUPT - LGC TST/
 .EVEN

 TEST TO VERIFY THAT THE INTERRUPTS CAN BE SET AND THAT THE DEVICE
 RESPONDS AS EXPECTED.

 BGNTST
 IF LOGIC TEST
 THEN-SETUP TEST IDENT
 SET PROCESSOR PRIORITY-> -> NO INTERRUPTS
 CAUSE RX INTERRUPT (WHEN PRIORITY LOWERED)
 CALL WATCH DOG TO LOWER PRIORITY & WAIT FOR INTERRUPT
 CALL ERROR
 CLEAR RX INTERRUPT BIT
 NOP
 ENDIF
 ENDTST

BOARD CALLOUT:
 1. INTERFACE

TSETUP
 BGNT20: MOV #T20TBL,TSTID ;SETUP TEST ID TBL-TEST# 20
 IAT20: BIT #LOGICT,TSTMOD ;IF TEST MODE=LOGIC TEST
 BEQ XT20 ;BIT SET, THEN
 CALL LTSTUP ;CALL LOGIC TEST SETUP
 CLR DX ;SET WATCH DOG MULTIPLIER=0
 SETPRI #PRI07 ;SET PROCESSOR PRI=NO INTERRUPTS
 BIS #100,ARXCS ;CAUSE RX TO INTERRUPT-WHEN PRI LOWERED
 CALL WATCH ;CALL WATCH DOG-LOWER PRI & WAIT FOR INTERRUPT
 CALL ERROR ;CALL ERROR
 BIC #100,ARXCS ;CLEAR RX INTERRUPT BIT
 MOV #10,DX ;RESET WATCH DOG MULTIPLIER
 XT20: EXIT TST
 REGTBL
 TTBL 0,0

T20TBL: .WORD T20MSG
 .WORD 0
 .WORD 0
 .WORD -1
 T20RTB: .WORD -1
 T20FTB: .WORD INTONL
 .WORD -1

FRUTBL INTONL

ENDTST

6130
(2) 027502 000414
(2) 027504 050040 044522 051117
(2)
6131
6132
6133
6134
6135
6136
6137
6138
6139
6140
6141
6142
6143
6144
6145
6146
6147
6148
6149
6150
6151
6152
6153
6154
6155
6156
6157
6158

.SBTTL TEST 21 - PRIORITY LVL - LGC TST
BR BGNT21 ;BR TO BGN TST
T21MSG: .ASCIZ / PRIORITY LVL - LGC TST/
.EVEN

```
:+  
: TEST TO VERIFY THAT THE DEVICE PRIORITY IS SET TO THE CORRECT LEVEL.  
:-----  
: BGNST  
: IF LOGIC TEST  
: THEN-SETUP TEST IDENT  
: LOWER WATCH DOG TIMEOUT MULTIPLIER  
: SETUP PROCESSOR PRIORITY=7 (PRESET VALUE)  
: BGND0  
: SET PROCESSOR PRIORITY (PRESET VALUE)  
: SET DEVICE INTERRUPT BIT TO ENABLE INTERRUPT  
: IF INTERRUPT OR ERROR OCCURRED  
: THEN-SET DO LOOP DONE BIT -> FLAGS  
: ELSE-LOWER SETUP PROCESSOR PRIORITY  
: CLEAR DEVICE INTERRUPT BIT  
: ENDIF  
: DUNTIL DO LOOP DONE BIT SET, PROCESSOR PRI=0 OR NO DONE BIT ESR  
: IF SETUP PROCESSOR PRI NOT=DEVICE PRIORITY  
: THEN-CALL LOGIC TEST ERROR  
: ENDIF  
: ENDIF  
: ENDTST  
:-----  
: BOARD CALLOUT:  
: 1. INTERFACE  
:-----  
:--
```

6161	027534					TSETUP			
(2)	027534	012737	030106	002466		BGNT21: MOV	#T21TBL,TSTID	:	SETUP TEST ID TBL-TEST# 21
(2)	027542	032737	000001	002324		IAT21: BIT	#LOGICT,TSTM0D	:	IF TEST MODE=LOGIC TEST
(2)	027550	001511					BEQ	XT21	:BIT SET, THEN
(2)	027552	004737	020736				CALL	LTSTUP	:CALL LOGIC TEST SETUP
6162	027556	004737	010440				CALL	INTIAL	:CALL PROG INITIALIZE
6163	027562	012737	000001	012024			MOV	#1,DX	:SET WATCH DOG MULTIPLIER=1
6164	027570						SETPRI	#PRI07	:SET PROCESSOR PRI=07 (NO INTERRUPTS)
6165	027576	005005					CLR	R5	:SET PRIORITY TABLE INDEX
6166	027600	005037	002520			BBT21: CLR	ERRNBR	:	CLEAR ERROR NUMBER INDICATOR
6167	027604	116537	021741	002416			MOVB	PRI TAB+7(R5),PRI	ORT ;SET PROCESSOR PRIORITY TO NEW LEVEL
6168	027612	013737	002416	002504			MOV	PRIORT,TTEMP1	:SETUP TEMP1 = PROCESSOR PRI
6169	027620	062737	000040	002504			ADD	#40,TTEMP1	:NOW SETUP FOR COMPARE, I.E. ONE PRI LVL HIGHER
6170	027626	052777	000100	152514			BIS	#100,ARXCS	:SET RX INTERRUPT BIT,AS PROCESSOR PRI LOWERED,
6171	027634	004737	011662				CALL	WATCH	:CALL WATCH DOG TO WAIT FOR INTERRUPT
6172	027640	022737	000015	002520		ICT21: CMP	#DNNINT,ERRNBR	:	IF INTERRUPT OR ERROR
6173	027646	001404					BEQ	ECT21	:OCCURRED, THEN
6174	027650	052737	000010	002476			BIS	#DLPDN,FLAGST	:SET DO LOOP DONE FLAG
6175	027656	000404					BR	UBT21	:BR TO DOUNTIL 'B'
6176	027660	005305				ECT21: DEC	R5	:	SET INDEX TO NEXT LOWER PROCESSOR PRI
6177	027662	042777	000100	152460		EDT21: BIC	#100,ARXCS	:	CLEAR DEVICE INTERRUPT BIT
6178	027670	020527	177770			UBT21: CMP	R5,#-8.	:	DO UNTIL PROCESSOR PRI TABLE ALL DONE
6179	027674	001404					BEQ	IET21	:OR
6180	027676	032737	000010	002476			BIT	#DLPDN,FLAGST	:DOUNTIL FLAGST DO LOOP DONE FLAG
6181	027704	001735					BEQ	BBT21	:SET
6182	027706	005737	002520			IET21: TST	ERRNBR	:	IF INTERRUPT OCCURRED
6183	027712	001026					BNE	LET21	:THEN
6184	027714	032737	000400	002500		IFT21: BIT	#LSIFLG,FLAGSP	:	IF FLAGSP=LSI FLAG
6185	027722	001024					BNE	XT21	:NOT SET, THEN
6186	027724	023737	002504	002356		IGT21: CMP	TTEMP1,RXPRI	:	IF SETUP PROCESSOR PRIORITY & RX PRIORITY
6187	027732	001420					BEQ	XT21	:DONT MATCH
6188	027734	012737	000054	002520			MOV	#PRILEV,ERRNBR	:SETUP ERR NBR=PRI LEV ERR
6189	027742	004737	003060				CALL	ERROR	:CALL ERROR
6190	027746	013703	002416				MOV	PRIORT,R3	:SETUP INTERRUPT PRI LEV FOR PRT
6191	027752	013702	002356				MOV	RXPRI,R2	:SETUP RX PRI LEV FOR PRINT
6192	027756	012701	030006				MOV	#PRIMSG,R1	:SETUP PRI LEV MSG
6193	027762	004737	002756				CALL	PRTX2S	:PRINT MSG
6194	027766	000402					BR	XT21	:BR TO TEXT EXIT
6195	027770	004737	003060			LET21: CALL	ERROR	:	CALL ERROR
6196	027774	012737	000010	012024		XT21: MOV	#10,DX	:	RESET WATCHDOG MULTIPLIER
6197	030002						EXIT	TST	
6198	030006	047045	051445	022466		PRIMSG: .ASCIZ	/XN%S6XA RX SET AT PRI LEV=%03XN%S6XA INTERRUPTED AT PRI LEV=%03		
6199							.EVEN		
6200	030106						REGTBL		
6201	030106						TTBL	0,0	
(2)	030106	027504						T21TBL: .WORD	T21MSG
(2)	030110	000000						.WORD	0
(2)	030112	000000						.WORD	0
(2)	030114	177777						.WORD	-1
(3)	030116							T21RTB: .WORD	-1
(3)	030116	177777						.WORD	-1
6202	030120					FRUTBL	INTONL	T21FTB: .WORD	INTONL
(2)	030120							.WORD	-1
(2)	030120	006644							
(2)	030122	177777							
6203	030124					ENDTST			

6206
(2) 030126 000417
(2) 030130 044440 044516 044524
(2)
6207
6208
6209
6210
6211
6212
6213
6214
6215
6216
6217
6218
6219
6220
6221
6222
6223
6224 030166
(2) 030166 012737 030224 002466
(2) 030174 032737 000001 002324
(2) 030202 001406
(2) 030204 004737 020736
6225 030210 004737 010440
6226 030214 004737 017724
6227 030220
6228 030224
(1) 015046
6229 030224
(2) 030224 030130
(2) 030226 000001
(2) 030230 000004
(2) 030232 177777
(3) 030234
(3) 030234 015046
(3) 030236 177777
6230 030240
(2) 030240
(2) 030240 006646
(2) 030242 177777
6231 030244

.SBTTL TEST 22 - INITIALIZE CONTROL - LGC TST
BR BGNT22 ;BR TO BGN TST
T22MSG: .ASCIZ / INITIALIZE CONTROL - LGC TST/
.EVEN

: TEST TO VERIFY THAT THE CONTROLLER WILL COMPLETE INITIALIZE.

: BGNTST
: IF LOGIC TEST
: THEN-SETUP TEST IDENT
: ISSUE DEVICE PROGRAMMED INITIALIZE
: CALL ERRCHK
: ENDIF
: ENDTST

BOARD CALLOUT:
1. CONTROLLER
2. INTERFACE

: TSETUP
BGNT22: MOV #T22TBL,TSTID ;SETUP TEST ID TBL-TEST# 22
IAT22: BIT #LOGICT,TSTMOD ;IF TEST MODE=LOGIC TEST
BEQ XT22 ;BIT SET, THEN
CALL LTSTUP ;CALL LOGIC TEST SETUP
CALL INTIAL ;CALL INITIALIZE
CALL ERRCHK ;CALL ERROR CHECK
XT22: EXIT TST
REGTBL CEINIT
REGS1=CEINIT
TTBL REGCK,RGPRT
T22TBL: .WORD T22MSG
 .WORD REGCK
 .WORD RGPRT
 .WORD -1
T22RTB: .WORD REGS1
 .WORD -1
FRUTBL CTLINF
T22FTB: .WORD CTLINF
 .WORD -1
ENDTST

6234
(2) 030246 000417
(2) 030250 042040 052101 020101
(2)
6235
6236
6237
6238
6239
6240
6241
6242
6243
6244
6245
6246
6247
6248
6249
6250
6251
6252
6253
6254
6255
6256
6257
6258
6259
6260
6261
6262
6263
6264
6265
6266
6267
6268

.SBTTL TEST 23 - DATA BUF INTEGRITY - LGC TST
BR BGNT23 ;BR TO BGN TST
T23MSG: .ASCIZ / DATA BUF INTEGRITY - LGC TST/
.EVEN

++
: TEST TO VERIFY ALL BITS OF DATA BUFFER, VARIOUS PATTERNS WILL BE USED.

: BGNTST
: IF LOGIC TEST
: THEN-SETUP TEST IDENT
: SETUP RANDOM DATA PATTERN
: NOP
: BGND0
: CALL DATA PATTERN SETUP
: CALL FILL BUFFER
: IF NO ERROR (ESCAPE TEST)
: THEN-CALL EMPTY BUFFER
: IF NO ERROR (ESCAPE TEST)
: THEN-SET EMPTY BUFFER FLAG
: CALL DATA CHECK
: ADVANCE PATTERN COUNT
: GET NEW PATTERN #
: IF FOUR PATTERNS DONE
: THEN-SET DO LOOP DONE
: ENDF
: ENDF
: DOUNTIL DO LOOP DONE FLAG SET
: NOP
: ENDF
: ENDTST

BOARD CALLOUT:
1. CONTROLLER
2. INTERFACE

--

6271	030306					TSETUP			
(2)	030306	012737	030430	002466	BGNT23:	MOV	#T23TBL,TSTID	:	SETUP TEST ID TBL-TEST# 23
(2)	030314	032737	000001	002324	IAT23:	BIT	#LOGICT,TSTMOD	:	IF TEST MODE=LOGIC TEST
(2)	030322	001440				BEQ	XT23	:	BIT SET, THEN
(2)	030324	004737	020736			CALL	LTSTUP	:	CALL LOGIC TEST SETUP
6272	030330	012737	000001	012660		MOV	#1,PAT	:	SET DATA PATTERN = 0'S
6273	030336	004737	012306		BBT23:	CALL	STDATP	:	CALL SET DATA PATTERN
6274	030342	004737	010510			CALL	FILBUF	:	CALL FILL BUFFER
6275	030346					ESCAPE	TST	:	IF NO ERROR, THEN
6276	030352	004737	010626			CALL	EMPBUF	:	CALL EMPTY BUFFER
6277	030356					ESCAPE	TST	:	IF NO ERROR, THEN
6278	030362	004737	013246			CALL	DATAK	:	CALL DATA CHECK
6279	030366	005237	012660			INC	PAT	:	ADVANCE TO NEXT DATA PATTERN
6280	030372	022737	000010	012660	ICT23:	CMP	#8,PAT	:	IF ALL DATA PATTERNS
6281	030400	001003				BNE	UBT23	:	DONE, THEN
6282	030402	052737	000010	002476		BIS	#DLPDN,FLAGST	:	SET FLAGST=DO_LOOP_DONE_FLAG
6283	030410	032737	000010	002476	UBT23:	BIT	#DLPDN,FLAGST	:	DUNTIL FLAGST-DO_LOOP_DONE_FLAG
6284	030416	001747				BEQ	BBT23	:	IS SET
6285	030420	005037	012660			CLR	PAT	:	RESET DATA PATTERN
6286	030424				XT23:	EXIT	TST		
6287	030430					REGTBL			
6288	030430					TTBL	EMBUFF,0		
(2)	030430	030250					T23TBL:	.WORD	T23MSG
(2)	030432	000020						.WORD	EMBUFF
(2)	030434	000000						.WORD	0
(2)	030436	177777						.WORD	-1
(3)	030440						T23RTB:		
(3)	030440	177777						.WORD	-1
6289	030442					FRUTBL	CTLINF		
(2)	030442							T23FTB:	
(2)	030442	006646						.WORD	CTLINF
(2)	030444	177777						.WORD	-1
6290	030446					ENDTST			

6293
(2) 030450 000417
(2) 030452 053440 042122 041440
(2) 030510

.SBTTL TEST 24 - WRD CNT INTEGRITY - LGC TST
BR BGNT24 ;BR TO BGN TST
T24MSG: .ASCIZ / WRD CNT INTEGRITY - LGC TST/
.EVEN

6294
6295
6296
6297
6298
6299
6300
6301
6302
6303
6304
6305
6306
6307
6308
6309
6310
6311
6312
6313
6314
6315
6316
6317
6318
6319
6320
6321
6322
6323
6324
6325
6326
6327

..**
: TEST TO VERIFY ALL BITS OF WORD COUNT REGISTER AND CHECK THAT EXCEEDING
: THE WORD COUNT FOR DISKETTE DENSITY WILL BE DETECTED.

```
-----  
: BGNTST  
: IF LOGIC TEST  
: THEN-SETUP TEST IDENT  
: SET DENSITY CONTROL=DOUBLE  
: SET BUFFER LENGTH=128.  
: BGND0  
: CALL FILL BUFFER  
: IF NO ERROR (ESCAPE TEST)  
: THEN-CALL READ ERROR CODE  
: IF NO ERROR (ESCAPE TEST)  
: THEN-IF WORD COUNTS NOT EQUAL  
: THEN-SETUP WORD COUNT ERROR  
: CALL ERROR  
: ELSE-UPDATE WORD COUNT  
: IF WORD COUNT=0  
: THEN-SET DO LOOP DONE FLAG  
: ENDIF  
: ENDIF  
: ENDIF  
: DOUNTIL DO LOOP DONE FLAG SET  
: NOP  
: ENDIF  
: ENDTST
```

: BOARD CALLOUT:
: 1. CONTROLLER

:--

6330	030510					TSETUP			
(2)	030510	012737	030670	002466		BGNT24: MOV	#T24TBL,TSTID	:	SETUP TEST ID TBL-TEST# 24
(2)	030516	032737	000001	002324		IAT24: BIT	#LOGICT,TSTMOD	:	IF TEST MODE=LOGIC TEST
(2)	030524	001457				BEQ	XT24	:	BIT SET, THEN
(2)	030526	004737	020736			CALL	LTSTUP	:	CALL LOGIC TEST SETUP
6331	030532	012737	000400	002412		MOV	#DENBIT,DENSTY	:	SET DENSITY CONTROL=DOUBLE
6332	030540	012737	000200	002370		MOV	#128.,WDCNT	:	SET WORD COUNT=128.
6333	030546	004737	010510			BBT24: CALL	FILBUF	:	CALL FILL BUFFER
6334	030552					ESCAPE	TST	:	IF NO ERROR THEN
6335	030556	004737	011340			CALL	RDERCD	:	CALL READ ERROR CODE
6336	030562					ESCAPE	TST	:	IF NO ERROR THEN
6337	030566	105737	002443			ICT24: TSTB	WC	:	IF WORD COUNT
6338	030572	001420				BEQ	LCT24	:	NOT EQUAL 0, THEN
6339	030574	012737	000023	002520		MOV	#WCERR,ERRNBR	:	SETUP ERR NBR=WORD COUNT ERROR
6340	030602	005037	002440			CLR	REGACT	:	CLEAR REG ACTUAL
6341	030606	113737	002443	002440		MOVB	WC,REGACT	:	SETUP WORD COUNT ACTUAL
6342	030614	005037	002436			CLR	REGEXP	:	SETUP WORD COUNT EXPECTED
6343	030620	004737	003060			CALL	ERROR	:	CALL ERROR
6344	030624	052737	000010	002476		BIS	#DLPDN,FLAGST	:	SET FLAGST=DO LOOP DONE FLAG
6345	030632	000410				BR	UBT24	:	BR TO DOUNTIL 'B'
6346	030634	005337	002370			LCT24: DEC	WDCNT	:	DECREMENT WORD COUNT
6347	030640	005737	002370			IDT24: TST	WDCNT	:	IF WORD COUNT
6348	030644	001003				BNE	UBT24	:	EQUALS ZERO, THEN
6349	030646	052737	000010	002476		BIS	#DLPDN,FLAGST	:	SET FLAGST=DO LOOP_DONE_FLAG
6350	030654	032737	000010	002476		UBT24: BIT	#DLPDN,FLAGST	:	DOUNTIL FLAGST=DO_LOOP_DONE_FLAG
6351	030662	001731				BEQ	BBT24	:	SET
6352	030664					XT24: EXIT	TST		
6353	030670					REGTBL			
6354	030670					TTBL	0, RGPRT		
(2)	030670	030452					T24TBL: .WORD	T24MSG	
(2)	030672	000000					.WORD	0	
(2)	030674	000004					.WORD	RGPRT	
(2)	030676	177777					.WORD	-1	
(3)	030700						T24RTB: .WORD	-1	
(3)	030700	177777					.WORD	-1	
6355	030702					FRUTBL	CTLINF		
(2)	030702						T24FTB: .WORD	CTLINF	
(2)	030702	006646					.WORD	-1	
(2)	030704	177777					.WORD	-1	
6356	030706					ENDTST			

6359
(2) 030710 000424
(2) 030712 041440 047117 051124
(2) 030762
6360
6361
6362
6363
6364
6365
6366
6367
6368
6369
6370
6371
6372
6373
6374
6375
6376
6377
6378
6379
6380
6381
6382
6383

.SBTTL TEST 25 - CONTROLLER-READ*WRITE ELECT - LGC TST
BR BGNT25 ;BR TO BGN TST
T25MSG: .ASCIZ / CONTROLLER-READ*WRITE ELECT - LGC TST/
.EVEN

++
: TEST TO VERIFY MINIMAL CONTROLLER BOARD-READ/WRITE ELECTRONICS BOARD
: INTERFACE VIA INITIALIZE OF A SELECTED DRIVE.

: BGNTST
: IF LOGIC TEST
: THEN-SETUP TEST IDENT
: NOP
: ISSUE PROGRAMMED INITIALIZE
: CALL ERROR CK
: CALL READ ERROR CODE
: IF NO ERROR (ESCAPE TEST)
: THEN-CALL ERROR CK
: ENDIF
: NOP
: ENDIF
: ENDTST

: BOARD CALLOUT:
: 1. CONTROLLER
: 2. R/W ELECTRONICS
:-----
:--

```
6386 030762 012737 031054 002466      TSETUP
(2) 030762 032737 000001 002324      BGNT25: MOV #T25TBL,TSTID ;SETUP TEST ID TBL-TEST# 25
(2) 030770 001424 020736      IAT25: BIT #LOGICT,TSTMOD ;IF TEST MODE=LOGIC TEST
(2) 031000 004737 010440      BEQ XT25 ;BIT SET, THEN
6387 031004 004737 017724      CALL LTSTUP ;CALL LOGIC TEST SETUP
6388 031010 004737 017724      CALL INTIAL ;CALL INITIALIZE
6389 031014 005237 002470      CALL ERRCHK ;CALL ERROR CHECK
6390 031020 052737 000200 002500      INC TCMDCT ;INCREMENT TST CMD CTR *****
6391 031026 004737 011340      BIS #RECTST,FLAGSP ;SET READ ERROR CODE TEST=FLAGSP
6392 031032 004737 017724      CALL RDERCD ;CALL READ ERROR CODE
6393 031036 042737 000200 002500      ESCAPE TST ;IF NO ERROR
6394 031042 004737 017724      CALL ERRCHK ;CALL ERROR CHECK
6395 031050 042737 000200 002500      BIC #RECTST,FLAGSP ;CLEAR READ ERROR CODE TEST=FLAGSP
6396 031054 015046      XT25: EXIT TST
(1) 015076      REGTBL CEINIT,CSESND
(1) 015076      REGS1=CEINIT
6397 031054 030712      TTBL REGCK,0      REGS2=CSESND
(2) 031056 000001      T25TBL: .WORD T25MSG
(2) 031060 000000      .WORD REGCK
(2) 031062 177777      .WORD 0
(3) 031064 015046      T25RTB: .WORD -1
(3) 031066 015076      .WORD REGS1
(3) 031070 177777      .WORD REGS2
6398 031072 006651      FRUTBL CTLRWE      .WORD -1
(2) 031072 177777      T25FTB: .WORD CTLRWE
(2) 031074 177777      .WORD -1
6399 031076      ENDTST
```


6402
(2) 031100 000417
(2) 031102 051040 040505 020104
(2) 031140
6403
6404
6405
6406
6407
6408
6409
6410
6411
6412
6413
6414
6415
6416
6417
6418
6419
6420
6421
6422
6423
6424
6425
6426
6427
6428
6429
6430
6431
6432

.SBTTL TEST 26 - READ SECTOR-PRT:1 - LGC TST
BR BGNT26 :BR TO BGN TST
T26MSG: .ASCIZ / READ SECTOR-PRT:1 - LGC TST/
.EVEN

++
TEST TO VERIFY THAT THE CONTROLLER WILL COMPLETE A READ SECTOR IN
BOTH DENSITIES AND RETURN A VALID ERROR CODE.

BGNTST
: IF LOGIC TEST
: THEN-SETUP TEST IDENT
: CALL DEVICE DENSITY CK
: SET DENSITY CONTROL=DISK DEN
: CALL READ SECTOR
: CALL READ ERROR CODE
: IF NO COMMAND ERRORS
: THEN-CALL ERROR CK
: CALL COMPLIMENT DENSITY
: CALL READ SECTOR
: CALL READ ERROR CODE
: IF NO COMMAND ERRORS
: THEN-CALL ERROR CK
: ENDIF
: NOP
: ENDIF
ENDIF
ENDTST

BOARD CALLOUT:
1. CONTROLLER
2. R/W ELECTRONICS

--

6435	031140					TSETUP			
(2)	031140	012737	031256	002466	BGNT26:	MOV	#T26TBL,TSTID	:	SETUP TEST ID TBL-TEST# 26
(2)	031146	032737	000001	002324	IAT26:	BIT	#LOGICT,TSTMOD	:	IF TEST MODE=LOGIC TEST
(2)	031154	001436				BEQ	XT26	:	BIT SET, THEN
(2)	031156	004737	020736			CALL	LTSTUP	:	CALL LOGIC TEST SETUP
6436	031162	004737	017350			CALL	DENCHK	:	CALL DENSITY CHECK
6437	031166	004737	020472			CALL	SDENC	:	CALL SET DENSITY CONTROL=DENSITY STATUS
6438	031172	004737	011062			CALL	READ	:	CALL READ SECTOR
6439	031176	004737	011340			CALL	RDERCD	:	CALL READ ERROR CODE
6440	031202					ESCAPE	TST	:	IF NO COMMAND ERRORS, THEN
6441	031206	004737	017724			CALL	ERRCHK	:	CALL ERROR CHECK
6442	031212	004737	020430			CALL	CDENC	:	CALL COMPLIMENT DENSITY CONTROL
6443	031216	052737	004000	002476		BIS	#NEGTST,FLAGST	:	SET FLAGST=NEG TEST FLAG
6444	031224	012737	000030	002464		MOV	#DENERR,NGTSE	:	SETUP NEGTEST SET ERROR=DEN ERROR
6445	031232	004737	011062			CALL	READ	:	CALL READ SECTOR
6446	031236	004737	011340			CALL	RDERCD	:	CALL READ ERROR CODE
6447	031242					ESCAPE	TST	:	IF NO COMMAND ERRORS, THEN
6448	031246	004737	017724			CALL	ERRCHK	:	CALL ERROR CHECK
6449	031252				XT26:	EXIT	TST		
6450	031256					REGTBL	CSESAL		
(1)		015036						REGS1=CSESAL	
6451	031256					TTBL	REGCK,0		
(2)	031256	001102						T26TBL:	.WORD T26MSG
(2)	031260	000001							.WORD REGCK
(2)	031262	000000							.WORD 0
(2)	031264	177777							.WORD -1
(3)	031266							T26RTB:	
(3)	031266	015036							.WORD REGS1
(3)	031270	177777							.WORD -1
6452	031272					FRUTBL	CTLRWE	T26FTB:	
(2)	031272								.WORD CTLRWE
(2)	031272	006651							.WORD -1
(2)	031274	177777							
6453	031276					ENDTST			

6456
(2) 031300 000414
(2) 031302 050040 051517 052111
(2) 031332
6457
6458
6459
6460
6461
6462
6463
6464
6465
6466
6467
6468
6469
6470
6471
6472
6473
6474
6475
6476
6477
6478
6479
6480
6481
6482
6483
6484
6485
6486

.SBTTL TEST 27 - POSITIONING - LGC TST
BR BGNT27 ;BR TO BGN TST
T27MSG: .ASCIZ / POSITIONING - LGC TST/
.EVEN

..**
: TEST TO VERIFY THAT THE DRIVE WILL READ THE HEADERS ON ALL TRACKS OF
: THE DEIVE AS EXPECTED.

: BGNTST
: IF LOGIC TEST
: THEN-SETUP TEST IDENT
: SET TRACK INIT FLAG
: SET SECTOR=10
: BGND0
: : CALL GET TRACK
: : CALL READ ERROR CODE
: : CALL READ SECTOR
: : IF NO COMMAND ERRORS (ESCAPE TST)
: : THEN-CALL ERROR CHECK
: : : CALL TRACKS ERROR CK
: : : CLEAR TRACK INIT FLAG
: : : NOP
: : ENDF
: DOUNTIL TRACKS DONE, ABORT FLAG SET, OR TRACK ERRORS=10
: NOP
: ENDF
: ENDTST

: BOARD CALLOUT:
: 1. CONTROLLER
: 2. R/W ELECTRONICS
:-----
:--


```
6489 031332          TSETUP
(2) 031332 012737 031446 002466      BGNT27: MOV #T27TBL,TSTID ;SETUP TEST ID TBL-TEST# 27
(2) 031340 032737 000001 002324      IAT27: BIT #LOGICT,TSTMOD ;IF TEST MODE=LOGIC TEST
(2) 031346 001435          BEQ XT27 ;BIT SET, THEN
(2) 031350 004737 020736          CALL LTSTUP ;CALL LOGIC TEST SETUP
6490 031354 012737 000400 002510      MOV #ITK!RTK,TKSCFG ;SET TRK/SEC FLAG-->TRACK=INIT & RANDOM
6491 031362 004737 017350          CALL DENCHK ;CALL DENSITY CHECK
6492 031366 004737 020472          CALL SDENC ;CALL SET DENSITY CONTROL=DENSITY STATUS
6493 031372 012737 000010 002376      MOV #10,SECTOR ;SET SECTOR=10
6494 031400 004737 012662      BBT27: CALL GETTRK ;CALL GET TRACK
6495 031404 004737 011062          CALL READ ;CALL READ SECTOR
6496 031410 004737 011340          CALL RDERCD ;CALL READ ERROR CODE
6497 031414          ESCAPE TST ;IF NO COMMAND ERRORS, THEN
6498 031420 004737 017516          CALL TKERCK ;CALL TRACK ERROR CHECK
6499 031424 042737 000400 002510      BIC #ITK,TKSCFG ;CLEAR INT TRK FLAG
6500 031432 032737 001000 002476      UBT27: BIT #TRKDON,FLAGST ;DO UNTIL FLAGST-TRACK DONE FLAG
6501 031440 001757          BEQ BBT27 ;SET,
6502 031442          XT27: EXIT TST
6503 031446          REGTBL CSESAL
(1) 031446 015036          TTBL REGCK,0 REGS1=CSESAL
6504 031446 031302          T27TBL: .WORD T27MSG
(2) 031446 000001          .WORD REGCK
(2) 031450 000000          .WORD 0
(2) 031452 177777          .WORD -1
(2) 031454 177777          T27RTB: .WORD REGS1
(3) 031456 015036          .WORD -1
(3) 031456 177777          FRUTBL CTLRWE T27FTB: .WORD CTLRWE
6505 031462          (2) 031462 006651          .WORD -1
(2) 031462 177777          (2) 031464 177777
6506 031466          ENDTST
```

6509
(2) 031470 000417
(2) 031472 053440 044522 042524
(2)
6510
6511
6512
6513
6514
6515
6516
6517
6518
6519
6520
6521
6522
6523
6524
6525
6526
6527
6528
6529
6530
6531
6532
6533
6534
6535
6536
6537
6538

.SBTTL TEST 28 - WRITE SECTOR-PRT:1 - LGC TST
BR BGNT28 :BR TO BGN TST
T28MSG: .ASCIZ / WRITE SECTOR-PRT:1 - LGC TST/
.EVEN

++
: TEST TO VERIFY THAT THE CONTROLLER WILL COMPLETE A WRITE SECTOR IN
: BOTH DENSITIES AND RETURN A VALID ERROR CODE.

: BGNST
: IF LOGIC TEST
: THEN-SETUP TEST IDENT
: CALL DEVICE DENSITY CHECK
: SET DENSITY CONTROL=DISK DEN
: CALL WRITE SECTOR
: IF NO COMMAND ERROR (ESCAPE TEST)
: THEN-CALL ERROR CHECK
: CALL COMPLIMENT DENSITY CONTROL
: CALL WRITE SECTOR
: IF NO COMMAND ERROR (ESCAPE TEST)
: THEN-CALL ERROR CHECK
: ENDIF
: NOP
: ENDIF
: NOP
: ENDIF
: ENDTST

: BOARD CALLOUT:
: 1. CONTROLLER
: 2. R/W ELECTRONICS
:-----
:--

6541	031530					TSETUP			
(2)	031530	012737	031642	002466		BGNT28: MOV	#T28TBL,TSTID	:	SETUP TEST ID TBL-TEST# 28
(2)	031536	032737	000001	002324		IAT28: BIT	#LOGICT,TSTMOD	:	IF TEST MODE=LOGIC TEST
(2)	031544	001434				BEQ	XT28	:	BIT SET, THEN
(2)	031546	004737	020736			CALL	LTSTUP	:	CALL LOGIC TEST SETUP
6542	031552	004737	017350			CALL	DENCHK	:	CALL DENSITY CHECK
6543	031556	004737	020472			CALL	SDENC	:	CALL SET DENSITY CONTROL=DENSITY STATUS
6544	031562	004737	010744			CALL	WRITE	:	CALL WRITE SECTOR
6545	031566					ESCAPE	TST	:	IF NO COMMAND ERROR, THEN
6546	031572	004737	017724			CALL	ERRCHK	:	CALL ERROR CHECK
6547	031576	004737	020430			CALL	CDENC	:	CALL COMPLIMENT DENSITY CONTROL
6548	031602	052737	004000	002476		BIS	#NEGST,FLAGST	:	SET FLAGST-NEG TEST FLAG
6549	031610	012737	000030	002464		MOV	#DENERR,NGTSER	:	SETUP NEG TEST ERR ERR=DENSITY ERR
6550	031616	004737	010744			CALL	WRITE	:	CALL WRITE
6551	031622	004737	011340			CALL	RDERCD	:	CALL READ ERROR CODE
6552	031626					ESCAPE	TST	:	IF NO COMMAND ERROR
6553	031632	004737	017724			CALL	ERRCHK	:	CALL ERROR CHECK
6554	031636					XT28: EXIT	TST		
6555	031642					REGTBL	CSESAL		
(1)		015036							REGS1=CSESAL
6556	031642					TTBL	REGCK,0		
(2)	031642	031472						T28TBL:	.WORD T28MSG
(2)	031644	000001							.WORD REGCK
(2)	031646	000000							.WORD 0
(2)	031650	177777							.WORD -1
(3)	031652							T28RTB:	
(3)	031652	015036							.WORD REGS1
(3)	031654	177777							.WORD -1
6557	031656					FRUTBL	CTLRWE		
(2)	031656							T28FTB:	
(2)	031656	006651							.WORD CTLRWE
(2)	031660	177777							.WORD -1
6558	031662					ENDTST			

6561
(2) 031664 000422
(2) 031666 042040 046105 052105
(2)
6562
6563
6564
6565
6566
6567
6568
6569
6570
6571
6572
6573
6574
6575
6576
6577
6578
6579
6580
6581
6582
6583
6584
6585
6586
6587
6588
6589
6590

.SBTTL TEST 29 - DELETED DATA WRITE PRT:1 - LGC TST
BR BGNT29 :BR TO BGN TST
T29MSG: .ASCIZ / DELETED DATA WRITE PRT:1 - LGC TST/
.EVEN

..**
: TEST TO VERIFY THAT THE DEVICE SET A DELETED DATA MARK ON THE DISKETTE
: HEADER AND RETRIEVE THIS DATA ON A KNOWN GOOD DISKETTE.

: BGNTST
: IF LOGIC TEST
: THEN-SETUP TEST IDENT
: CALL DEVICE DENSITY CK
: SET DENSITY CONTROL=DENSITY STATUS
: SET DELETED DATA FLAG (BIT#3-CMD)
: CALL WRITE SECTOR
: IF NO COMMAND ERROR (ESCAPE TEST)
: THEN-CALL READ SECTOR
: IF NO COMMAND ERROR (ESCAPE TEST)
: THEN-IF RXESR-DELETED DATA BIT NOT SET
: THEN-SET ERROR NUMBER=DELETED DATA ERR
: CALL ERROR
: ENDF
: ENDF
: ENDF
: ENDF
: ENDTST

: BOARD CALLOUT:
: 1. CONTROLLER
: 2. R/W ELECTRONICS

:--

```
6593 031732 012737 032056 002466      TSETUP
(2) 031732 012737 032056 002466      BGNT29: MOV #T29TB!,TSTID ;SETUP TEST ID TBL-TEST# 29
(2) 031740 032737 000001 002324      IAT29: BIT #LOGICT,TSTMOD ;IF TEST MODE=LOGIC TEST
(2) 031746 001441 ;BIT SET, THEN
(2) 031750 004737 020736      CALL LTSTUP ;CALL LOGIC TEST SETUP
6594 031754 004737 017350      CALL DENCHK ;CALL DEVICE DENSITY CHECK
6595 031760 004737 020472      CALL SDENC ;CALL SET DENSITY CONTROL=DENSITY STATUS
6596 031764 012737 000010 002402      MOV #DLDCMD,DELDT ;SET DELETED DATA FLAG
6597 031772 004737 010744      CALL WRITE ;CALL WRITE SECTOR
6598 031776      IBT29: ESCAPE TST ;IF NO COMMAND ERROR, THEN
6599 032002 004737 017724      CALL ERRCHK ;CALL ERROR CHECK
6600 032006 004737 011062      CALL READ ;CALL READ SECTOR
6601 032012      ICT29: ESCAPE TST ;IF NO COMMAND ERROR, THEN
6602 032016 032777 000100 150326      IDT29: BIT #DLDBIT,@RXDB ;IF RXESR-DELETED DATA BIT
6603 032024 001006      BNE LCT29 ;NOT SET, THEN
6604 032026 012737 000032 002520      MOV #DLDTERR,ERRNBR ;SETUP ERROR NUMBER=DELETED DATA ERROR
6605 032034 004737 003060      CALL ERROR ;CALL ERROR
6606 032040 000404      BR XT29 ;EXIT TST
6607 032042 005037 002402      LCT29: CLR DELDAT ;CLEAR DEL DATA MODE
6608 032046 004737 010744      CALL WRITE ;CALL WRITE SECTOR - CLR DATA FIELD
6609 032052
6610 032056      XT29: EXIT TST
(1) 015036      REGTBL CSESAL ;REGS1=CSESAL
6611 032056      TTBL REGCK,0
(2) 032056 031666      T29TBL: .WORD T29MSG
(2) 032060 000001      .WORD REGCK
(2) 032062 000000      .WORD 0
(2) 032064 177777      .WORD -1
(3) 032066      T29RTB:
(3) 032066 015036      .WORD REGS1
(3) 032070 177777      .WORD -1
6612 032072      FRUTBL CTLRWE
(2) 032072      T29FTB:
(2) 032072 006651      .WORD CTLRWE
(2) 032074 177777      .WORD -1
6613 032076      ENDTST
```


6665	032132					TSETUP			
(2)	032132	012737	032356	002466		BGNT30: MOV	#T30TBL,TSTID	:	SETUP TEST ID TBL-TEST# 30
(2)	032140	032737	000001	002324		IAT30: BIT	#LOGICT,TSTMOD	:	IF TEST MODE=LOGIC TEST
(2)	032146	001501					BEQ	XT30	:BIT SET, THEN
(2)	032150	004737	020736				CALL	LTSTUP	:CALL LOGIC TEST SETUP
6666	032154	004737	017350				CALL	DENCHK	:CALL DENSITY CHECK
6667	032160	013737	002414	002506			MOV	DENSTA,TSAVE1	:SAVE DEVICE DENSITY
6668	032166	005037	002412				CLR	DENSTY	:SET DENSITY CONTROL=SINGLE DENSITY
6669	032172	004737	011172				CALL	SETDN	:CALL SET DENSITY
6670	032176						ESCAPE	TST	:IF NO COMMAND ERROR, THEN
6671	032202	004737	017724				CALL	ERRCHK	:CALL ERROR CHECK
6672	032206	012737	000113	002372			MOV	#'K,VARIFY	:SET VALIDITY WORD=ASCII 'K'
6673	032214	004737	011172				CALL	SETDN	:CALL SET DENSITY
6674	032220	004737	011340				CALL	RDERCD	:CALL READ ERROR CODE
6675	032224						ESCAPE	TST	:IF NO COMMAND ERROR
6676	032230	052737	004000	002476			BIS	#NEGST,FLAGST	:SET FLAGST-NEG TEST FLAG
6677	032236	012737	000036	002464			MOV	#SDKYWD,NGTSER	:SETUP EXPECTED ERROR=SET DEN KEYWORD ERROR
6678	032244	004737	017724				CALL	ERRCHK	:CALL ERROR CHECK
6679	032250	012737	000400	002412			MOV	#DENBIT,DENSTY	:SET DENSITY CONTROL=DOUBLE DENSITY
6680	032256	012737	000111	002372			MOV	#'I,VARIFY	:SET VALIDITY WORD=ASCII 'I'
6681	032264	042737	004000	002476			BIC	#NEGST,FLAGST	:CLEAR FLAGST-NEG TEST FLAG
6682	032272	004737	011172				CALL	SETDN	:CALL SET DENSITY
6683	032276						ESCAPE	TST	:IF NO COMMAND ERROR, THEN
6684	032302	004737	017724				CALL	ERRCHK	:CALL ERROR CHECK
6685	032306	004737	017350				CALL	DENCHK	:CALL DENSITY CHECK
6686	032312	023737	002414	002412			CMP	DENSTA,DENSTY	:IF DENSITY DID
6687	032320	001405					BEQ	IBT30	:NOT SET, THEN
6688	032322	012737	000035	002520			MOV	#STDNER,ERRNBR	:SET ERROR NBR=DENSITY DIDN'T SET ERROR
6689	032330	004737	003060				CALL	ERROR	:CALL ERROR
6690	032334	005737	002506			IBT30:	TST	TSAVE1	:IF SAVED DENSITY
6691	032340	001404					BEQ	XT30	:EQUALS DOUBLE DEN, THEN
6692	032342	005037	002412				CLR	DENSTY	:SET DENSITY CONTROL=SINGLE DEN
6693	032346	004737	011172				CALL	SETDN	:CALL SET DENSITY
6694	032352					XT30:	EXIT	TST	
6695	032356						REGTBL	CSESAL	
(1)		015036							REGS1=CSESAL
6696	032356						TTBL	REGCK,0	
(2)	032356	032102							T30TBL: .WORD T30MSG
(2)	032360	000001							.WORD REGCK
(2)	032362	000000							.WORD 0
(2)	032364	177777							.WORD -1
(3)	032366								T30RTB: .WORD REGS1
(3)	032366	015036							.WORD -1
(3)	032370	177777							
6697	032372					FRUTBL	CTLRWE		T30FTB: .WORD CTLRWE
(2)	032372								.WORD -1
(2)	032372	006651							
(2)	032374	177777							
6698	032376					ENDTST			

6701
(2) 032400 000413
(2) 032402 051440 041505 047524
(2)
6702
6703
6704
6705
6706
6707
6708
6709
6710
6711
6712
6713
6714
6715
6716
6717
6718
6719
6720
6721
6722
6723
6724
6725
6726
6727
6728
6729
6730
6731
6732
6733
6734
6735
6736
6737
6738
6739
6740
6741
6742
6743
6744
6745
6746
6747

.SBTTL TEST 31 - SECTOR ADR - LGC TST
BR BGNT31 ;BR TO BGN TST
T31MSG: .ASCIZ / SECTOR ADR - LGC TST/
.EVEN

++
TEST TO VERIFY THAT THE CONTROLLER WILL HANDLE ALL LEGAL AND ILLEGAL
SECTOR ADDRESSES PROPERLY.

BGNTST
IF LOGIC TEST
THEN-SETUP TEST IDENT
SET TRACK ADR=0
SET SECTOR LEGAL FLAG
SET SECTOR INIT
BGND0
CALL GET SECTOR ADR
CALL READ SECTOR
CALL READ ERROR CODE
IF FINI FLAG NOT SET
THEN-
IF SECTOR ADR NOT=TARGET SECTOR ADR
THEN-SETUP TO PRINT ERROR
CALL ERROR
ELSE-CALL ERROR CK
ENDIF
ENDIF
DOUNTIL SECTORS DONE FLAG SET OR ABORT FLAG SET
CLEAR SECTORS DONE FLAG
SET DONE TIME OUT MULTIPLIER=100
SET NEG TEST FLAG
BGND0
CALL READ SECTOR
CALL READ ERROR CODE
IF FINI FLAG NOT SET
THEN-IF SECTOR ADR NOT=TARGET SECTOR ADR
THEN-SET ERR=SECTOR ADR ERROR
CALL ERROR
ELSE-CALL ERROR CHECK
ENDIF
ENDIF
DOUNTIL SECTORS DONE FLAG SET OR FINI FLAG SET
NOP
ENDIF
ENDTST

BOARD CALLOUT:
1. CONTROLLER

--

6750	032430					TSETUP		
(2)	032430	012737	032730	002466	BGNT31:	MOV	#T31TBL,TSTID	:SETUP TEST ID TBL-TEST# 31
(2)	032436	032737	000001	002324	IAT31:	BIT	#LOGICT,TSTMOD	:IF TEST MODE=LOGIC TEST
(2)	032444	001527				BEQ	XT31	:BIT SET, THEN
(2)	032446	004737	020736			CALL	LTSTUP	:CALL LOGIC TEST SETUP
6751	032452	004737	010440			CALL	INITIAL	:CALL INITAILIZE
6752	032456	004737	017350			CALL	DENCHK	:CALL DENSITY CHECK
6753	032462	004737	020472			CALL	SDENC	:CALL SET DENSITY CONTROL=DENSITY STATUS
6754	032466	012737	001002	002510		MOV	#ISC!SSC,TKSCFG	:SETUP SECTOR FLAGS=INITIALIZE & SEQUENCE
6755	032474	004737	013104		BBT31:	CALL	GETSEC	:CALL GET SECTOR
6756	032500	004737	011062			CALL	READ	:CALL READ SECTOR
6757	032504	004737	011340			CALL	RDERCD	:CALL READ ERROR CODE
6758	032510	005737	002454		ICT31:	TST	FIN	:IF FINI FLAG
6759	032514	001024				BNE	UBT31	:NOT SET, THEN
6760	032516	123737	002376	002447	IDT31:	CMPB	SECTOR,TSEC	:IF SECTOR ADR & DEVICE TARGET SECTOR
6761	032524	001416				BEQ	LDT31	:NOT =, THEN
6762	032526	012737	000042	002520		MOV	#SECAER,ERRNBR	:SETUP ERR NBR=SECTOR ADDRESS ERROR
6763	032534	052737	000002	002500		BIS	#SCPRT,FLAGSP	:SET FLAGSP=-PRINT SECTOR ADDRESS FLAG
6764	032542	004737	003060			CALL	ERROR	:CALL ERROR
6765	032546	042737	000002	002500		BIC	#SCPRT,FLAGSP	:CLEAR FLAGSP-PRINT SECTOR ADDRESS FLAG
6766	032554	004737	010440			CALL	INITIAL	:CALL INITAILIZE
6767	032560	000402				BR	UBT31	:BR TO DOUNTIL 'B'
6768	032562	004737	017724		LDT31:	CALL	ERRCHK	:CALL ERROR CHECK
6769	032566	005737	002452		UBT31:	TST	ABORT	:DOUNTIL ABORT FLAG
6770	032572	001004				BNE	EBT31	:SET OR
6771	032574	032737	002000	002476		BIT	#SECDON,FLAGST	:FLAGST-SECTOR DONE FLAG
6772	032602	001734				BEQ	BBT31	:SET
6773	032604	042737	002000	002476	EBT31:	BIC	#SECDON,FLAGST	:CLEAR FLAGST-SECTOR DONE FLAG
6774	032612	052737	004000	002476		BIS	#NEGTST,FLAGST	:SET FLAGST-NEG TEST FLAG
6775	032620	012737	000003	002464		MOV	#RDERR,NGTSER	:SETUP EXPECTED ERROR=READ ERROR (SECTOR NOT FOU
6776	032626	012737	000100	002474		MOV	#100,DNWTMT	:SET DONE WAIT MULTIPLIER SO NO TIME OUT
6777	032634	012737	000000	002376		MOV	#0,SECTOR	:SET SECTOR ADR=0
6778	032642	004737	011062		BET31:	CALL	READ	:CALL READ SECTOR
6779	032646	004737	011340			CALL	RDERCD	:CALL READ ERROR CODE
6780	032652	005737	002454		IFT31:	TST	FIN	:IF FINI FLAG
6781	032656	001017				BNE	EFT31	:NOT SET, THEN
6782	032660	123737	002376	002447	IGT31:	CMPB	SECTOR,TSEC	:IF SECTOR ADR AND TARGET SECTOR
6783	032666	001411				BEQ	LGT31	:NOT EQUAL, THEN
6784	032670	052737	000002	002500		BIS	#SCPRT,FLAGSP	:SET FLAGSP=-PRINT SECTOR ADDRESS FLAG
6785	032676	012737	000042	002520		MOV	#SECAER,ERRNBR	:SETUP ERR NBR=SECTOR ADDRESS ERROR
6786	032704	004737	003060			CALL	ERROR	:CALL ERROR
6787	032710	000402				BR	EFT31	:CALL TO END 'G'
6788	032712	004737	017724		LGT31:	CALL	ERRCHK	:CALL ERROR CHECK
6789	032716	012737	000012	002474	EFT31:	MOV	#12,DNWTMT	:RESET DONE WAIT MUTIPLIER TO NORMAL
6790	032724				XT31:	EXIT	TST	
6791	032730					REGTBL	CSESAL	
(1)		015036						REGS1=CSESAL
6792	032730					TTBL	REGCK,0	
(2)	032730	032402						T31TBL: .WORD T31MSG
(2)	032732	000001						.WORD REGCK
(2)	032734	000000						.WORD 0
(2)	032736	177777						.WORD -1
(3)	032740							T31RTB: .WORD REGS1
(3)	032740	015036						.WORD -1
(3)	032742	177777						
6793	032744					FRUTBL	CTLRWE	

(2) 032744
(2) 032744 006651
(2) 032746 177777
6794 032750

T31FTB:
.WORD CTLRWE
.WORD -1

ENDTST

6797
(2) 032752 000413
(2) 032754 052040 040522 045503
(2) 033002

.SBTTL TEST 32 - TRACK ADR - LGC TST
BR BGNT32 :BR TO BGN TST
T32MSG: .ASCIZ / TRACK ADR - LGC TST/
.EVEN

6798
6799
6800
6801
6802
6803
6804
6805
6806
6807
6808
6809
6810
6811
6812
6813
6814
6815
6816
6817
6818
6819
6820
6821
6822
6823
6824
6825
6826
6827
6828
6829
6830
6831
6832
6833
6834
6835
6836
6837
6838
6839
6840
6841
6842
6843
6844
6845
6846

++
TEST TO VERIFY THAT THE CONTROLLER WILL HANDLE ALL LEGAL AND ILLEGAL
TRACK ADDRESSES PROPERLY.

```
-----  
BGNTST  
: IF LOGIC TEST  
: THEN-SETUP TEST IDENT  
: SET TRACK INIT & SEQUENCE FLAGS  
: BGND0  
: : BGND0  
: : CALL GET TRACK ADR  
: : CLEAR TRACK INIT FLAG  
: : CALL READ SECTOR  
: : CALL READ ERROR CODE  
: : IF FINI FLAG NOT SET  
: : THEN-  
: : IF TRACK ADR NOT=TARGET TRACK ADR  
: : THEN-CALL LOGIC TEST ERROR  
: : ENDIF  
: : IF ILLEGAL FLAG NOT SET  
: : THEN-IF TRACK ADR NOT=UNIT TRACK ADR  
: : THEN-SETUP TRACK ADR ERR & CALL ERROR  
: : ELSE-IF ERROR CODE=40  
: : THEN-CALL LOGIC TEST ERROR  
: : ENDIF  
: : ELSE-IF TRACK ADR=UNIT TRACK ADR  
: : THEN-SETUP TRACK ADR ERR & CALL ERROR  
: : ELSE-IF ERROR CODE NOT=40  
: : THEN-CALL LOGIC TEST ERROR  
: : ENDIF  
: : ENDIF  
: : ENDIF  
: : DOUNTIL TRACKS DONE FLAG SET OR ABORT FLAG SET  
: : SET TRACK INIT FLAG  
: : IF TRACKS LEGAL FLAG SET  
: : THEN-SET TRACKS ILLEGAL FLAG  
: : ELSE-SET TRACKS LEGAL FLAG  
: : ENDIF  
: : DOUNTIL TRACKS LEGAL FLAG SET  
: ENDIF  
ENDTST  
-----
```

BOARD CALLOUT:
1. CONTROLLER

--

6849	033002			
(2)	033002	012737	033336	002466
(2)	033010	032737	000001	002324
(2)	033016	001545		
(2)	033020	004737	020736	
6850	033024	004737	017350	
6851	033030	004737	020472	
6852	033034	012737	000401	002510
6853	033042	000240		
6854	033044	004737	012662	
6855	033050	042737	000401	002510
6856	033056	004737	011062	
6857	033062	004737	011340	
6858	033066	005737	002454	
6859	033072	001062		
6860	033074	123737	002374	002446
6861	033102	001405		
6862	033104	012737	000041	002520
6863	033112	004737	003060	
6864	033116	013705	002420	
6865	033122	032737	010000	002476
6866	033130	001024		
6867	033132	123765	002374	002444
6868	033140	001406		
6869	033142	012737	000041	002520
6870	033150	004737	003060	
6871	033154	000431		
6872	033156	122737	000040	002442
6873	033164	001025		
6874	033166	012737	000021	002520
6875	033174	004737	003060	
6876	033200	000417		
6877	033202	123765	002374	002444
6878	033210	001004		
6879	033212	012737	000041	002520
6880	033220	000407		
6881	033222	122737	000040	002442
6882	033230	001403		
6883	033232	012737	000021	002520
6884	033240	005737	002454	
6885	033244	001004		
6886	033246	032737	001000	002476
6887	033254	001673		
6888	033256	042737	001000	002476
6889	033264	005037	002510	
6890	033270	052737	000004	002510
6891	033276	032737	010000	002476
6892	033304	001004		
6893	033306	052737	010000	002476
6894	033314	000403		
6895	033316	042737	010000	002476
6896	033324	032737	010000	002476
6897	033332			
6898	033336			
6899	033336			
(2)	033336	032754		

```

TSETUP
BGNT32: MOV #T32TBL,TSTID ;SETUP TEST ID TBL-TEST# 32
IAT32: BIT #LOGICT,TSTMOD ;IF TEST MODE=LOGIC TEST
      BEQ XT32 ;BIT SET, THEN
      CALL LTSTUP ;CALL LOGIC TEST SETUP
      CALL DENCHK ;CALL DENSITY CHECK
      CALL SDENC ;CALL SET DENSITY CONTROL=DENSITY STATUS
      MOV #ITK!STK,TKSCFG ;SET INITIALIZE & SEQUENCE TRACKS FLAG (TRACK/SE
BBT32: NOP ;
BCT32: CALL GETTRK ;CALL GET TRACK ADR
      BIC #ITK!STK,TKSCFG ;CLEAR INITIALIZE TRACKS FLAG
      CALL READ ;CALL READ SECTOR
      CALL RDERCD ;CALL READ ERROR CODE
IDT32: TST FIN ;IF FINI FLAG
      BNE UCT32 ;NOT SET, THEN
IET32: CMPB TRACK,TTRK ;IF TRACK ADR & TARGET TRACK
      BEQ EET32 ;NOT EQUAL, THEN
      MOV #TRKAER,ERRNBR ;SETUP ERR NBR=TRACK ADDRESS ERROR
      CALL ERROR ;CALL ERROR
EET32: MOV DRVOFF,R5 ;SET R5=DRIVE BYTE OFFSET
IFT32: BIT #ILLGAL,FLAGST ;IF ILLEGAL FLAG
      BNE IIT32 ;NOT SET, THEN
IGT32: CMPB TRACK,CTK0(R5) ;IF TRACK ADR & CURRENT TRACK OF SELECTED DRV (R
      BEQ IHT32 ;NOT EQUAL, THEN
      MOV #TRKAER,ERRNBR ;SETUP ERR NBR=TRACK ADDRESS ERROR
      CALL ERROR ;CALL ERROR
IHT32: BR UCT32 ;BR TO DOUNTIL 'C'
      CMPB #40,XERUUT ;IF ERR CODE
      BNE UCT32 ;SET=40, THEN
      MOV #RECERR,ERRNBR ;SETUP ERRNBR=READ ERR CODE-ERR WRG
      CALL ERROR ;CALL ERROR
IIT32: BR UCT32 ;BR TO DOUNTIL 'C'
      CMPB TRACK,CTK0(R5) ;IF TRACK ADR & CURRENT TRACK OF SELECTED DRV (R
      BNE IJT32 ;EQUAL, THEN
      MOV #TRKAER,ERRNBR ;SETUP ERR NBR=TRACK ADR ERROR
      BR UCT32 ;BR TO DOUNTIL 'C'
IJT32: CMPB #40,XERUUT ;IF ERR CODE NOT
      BEQ UCT32 ;SET=40
      MOV #RECERR,ERRNBR ;SETUP ERR NBR=READ ERR CODE-ERR WRG
UCT32: TST FIN ;DOUNTIL FINI FLAG
      BNE ECT32 ;SET OR
      BIT #TRKDON,FLAGST ;TRACKS DONE FLAG
      BEQ BCT32 ;SET
ECT32: BIC #TRKDON,FLAGST ;CLEAR TRACKS DONE FLAG
      CLR TKSCFG ;CLR TRACK FLAGS
      BIS #ILTK,TKSCFG ;SETUP ILLEGAL TRACKS FLAG
IKT32: BIT #ILLGAL,FLAGST ;IF ILLEGAL FLAG
      BNE LKT32 ;NOT SET, THEN
      BIS #ILLGAL,FLAGST ;SET ILLEGAL FLAG
      BR UBT32 ;BR TO DOUNTIL 'C'
LKT32: BIC #ILLGAL,FLAGST ;CLEAR ILLEGAL FLAG
UBT32: BIT #ILLGAL,FLAGST ;DOUNTIL ILLEGAL FLAG CLEAR
XT32: EXIT
      REGTBL
      TTBL
0,0

```

T32TBL: .WORD T32MSG

(2) 033340 000000
(2) 033342 000000
(2) 033344 177777
(3) 033346
(3) 033346 177777
6900 033350
(2) 033350
(2) 033350 006654
(2) 033352 177777
6901 033354

FRUTBL CTLONL

ENDTST

T32RTB:

T32FTB:

.WORD 0
.WORD 0
.WORD -1
.WORD -1
.WORD CTLONL
.WORD -1

6904
(2) 033356 000417
(2) 033360 051040 040505 020104
(2) 033416
6905
6906
6907
6908
6909
6910
6911
6912
6913
6914
6915
6916
6917
6918
6919
6920
6921
6922
6923
6924
6925
6926
6927
6928
6929
6930
6931
6932
6933
6934
6935

.SBTTL TEST 33 - READ SECTOR-PRT:2 - LGC TST
BR BGNT33 :BR TO BGN TST
T33MSG: .ASCIZ / READ SECTOR-PRT:2 - LGC TST/
.EVEN

++
: TEST TO VERIFY THAT THE CONTROLLER WILL COMPLETE A READ SECTOR IN
: BOTH DENSITIES & RETURN A VALID ERROR CODE, SIMILAR TO
: READ SECTOR PRT:1, BUT WITH DISKETTE IN OPPOSITE DENSITY.

: BGNTST
: IF LOGIC TEST
: THEN-SETUP TEST IDENT
: CALL DEVICE DENSITY CK
: SET DENSITY CONTROL=DISK DEN
: CALL READ SECTOR
: CALL READ ERROR CODE
: IF NO COMMAND ERRORS
: THEN-CALL ERROR CK
: CALL COMPLIMENT DENSITY
: CALL READ SECTOR
: CALL READ ERROR CODE
: IF NO COMMAND ERRORS
: THEN-CALL ERROR CK
: ENDIF
: NOP
: ENDIF
: ENDTST

BOARD CALLOUT:
1. CONTROLLER
2. R/W ELECTRONICS

--

```
6938 033416 TSETUP
(2) 033416 012737 033534 002466 BGNT33: MOV #T33TBL,TSTID ;SETUP TEST ID TBL-TEST# 33
(2) 033424 032737 000001 002324 IAT33: BIT #LOGICT,TSTMOD ;IF TEST MODE=LOGIC TEST
(2) 033432 001436 BEQ XT33 ;BIT SET, THEN
(2) 033434 004737 020736 CALL LTSTUP ;CALL LOGIC TEST SETUP
6939 033440 004737 017350 CALL DENCHK ;CALL DENSITY CHECK
6940 033444 004737 020472 CALL SDENC ;CALL SET DENSITY CONTROL=DENSITY STATUS
6941 033450 004737 011062 CALL READ ;CALL READ SECTOR
6942 033454 004737 011340 CALL RDERCD ;CALL READ ERROR CODE
6943 033460 ESCAPE TST ;IF NO COMMAND ERRORS, THEN
6944 033464 004737 017724 CALL ERRCHK ;CALL ERROR CHECK
6945 033470 004737 020430 CALL CDENC ;CALL COMPLIMENT DENSITY CONTROL
6946 033474 052737 004000 002476 BIS #NEGST,FLAGST ;SET FLAGST=NEG TEST FLAG
6947 033502 012737 000030 002464 MOV #DENERR,NGTSER ;SETUP NEGTEST SET ERROR=DEN ERROR
6948 033510 004737 011062 CALL READ ;CALL READ SECTOR
6949 033514 004737 011340 CALL RDERCD ;CALL READ ERROR CODE
6950 033520 ESCAPE TST ;IF NO COMMAND ERRORS, THEN
6951 033524 004737 017724 CALL ERRCHK ;CALL ERROR CHECK
6952 033530 XT33: EXIT TST
6953 033534 REGTBL CSESAL
(1) 015036 REGS1=CSESAL
6954 033534 TTBL REGCK,0
(2) 033534 033360 T33TBL: .WORD T33MSG
(2) 033536 000001 .WORD REGCK
(2) 033540 000000 .WORD 0
(2) 033542 177777 .WORD -1
(3) 033544 T33RTB:
(3) 033544 015036 .WORD REGS1
(3) 033546 177777 .WORD -1
6955 033550 FRUTBL CTLRWE
(2) 033550 T33FTB:
(2) 033550 006651 .WORD CTLRWE
(2) 033552 177777 .WORD -1
6956
6957 033554 ENDTST
```


6960
(2) 033556 000417
(2) 033560 053440 044522 042524
(2)
6961
6962
6963
6964
6965
6966
6967
6968
6969
6970
6971
6972
6973
6974
6975
6976
6977
6978
6979
6980
6981
6982
6983
6984
6985
6986
6987
6988
6989
6990
6991

.SBTTL TEST 34 - WRITE SECTOR-PRT:2 - LGC TST
BR BGNT34 :BR TO BGN TST
T34MSG: .ASCIZ / WRITE SECTOR-PRT:2 - LGC TST/
.EVEN

++
: TEST TO VERIFY THAT THE CONTROLLER WILL COMPLETE A WRITE SECTOR IN
: BOTH DENSITIES & RETURN A VALID ERROR CODE, SIMILAR TO WRITE
: SECTOR PRT:1, BUT WITH DISKETTE IN OPPOSITE DENSITY.

: BGNTST
: IF LOGIC TEST
: THEN-SETUP TEST IDENT
: CALL DEVICE DENSITY CHECK
: SET DENSITY CONTROL=DISK DEN
: CALL WRITE SECTOR
: IF NO COMMAND ERROR (ESCAPE TEST)
: THEN-CALL ERROR CHECK
: CALL COMPLIMENT DENSITY CONTROL
: CALL WRITE SECTOR
: IF NO COMMAND ERROR (ESCAPE TEST)
: THEN-CALL ERROR CHECK
: ENDIF
: NOP
: ENDIF
: NOP
: ENDIF

: ENDTST
: BOARD CALLOUT:
: 1. CONTROLLER
: 2. R/W ELECTRONICS
:-----

```
6994 033616          TSETUP
(2) 033616 012737 033730 002466      BGNT34: MOV #T34TBL,TSTID :SETUP TEST ID TBL-TEST# 34
(2) 033624 032737 000001 002324      IAT34:  BIT #LOGICT,TSTMOD :IF TEST MODE=LOGIC TEST
(2) 033632 001434          BEQ XT34 :BIT SET, THEN
(2) 033634 004737 020736          CALL LTSTUP :CALL LOGIC TEST SETUP
6995
6996 033640 004737 017350          CALL DENCHK :CALL DENSITY CHECK
6997 033644 004737 020472          CALL SDENC :CALL SET DENSITY CONTROL=DENSITY STATUS
6998 033650 004737 010744          CALL WRITE :CALL WRITE SECTOR
6999 033654          ESCAPE TST :IF NO COMMAND ERROR, THEN
7000 033660 004737 017724          CALL ERRCHK :CALL ERROR CHECK
7001 033664 004737 020430          CALL CDENC :CALL COMPLIMENT DENSITY CONTROL
7002 033670 052737 004000 002476      BIS #NEGST,FLAGST :SET FLAGST-NEG TEST FLAG
7003 033676 012737 000030 002464      MOV #DENERR,NGTSER :SETUP NEG TEST ERR ERR=DENSITY ERR
7004 033704 004737 010744          CALL WRITE :CALL WRITE
7005 033710 004737 011340          CALL RDERCD :CALL READ ERROR CODE
7006 033714          ESCAPE TST :IF NO COMMAND ERROR
7007 033720 004737 017724          CALL ERRCHK :CALL ERROR CHECK
7008 033724          XT34: EXIT TST
7009 033730          REGTBL CSESAL
(1) 015036          REGS1=CSESAL
7010 033730          TTBL REGCK,0
(2) 033730 033560          T34TBL: .WORD T34MSG
(2) 033732 000001          .WORD REGCK
(2) 033734 000000          .WORD 0
(2) 033736 177777          .WORD -1
(3) 033740          T34RTB:
(3) 033740 015036          .WORD REGS1
(3) 033742 177777          .WORD -1
7011 033744          FRUTBL CTLRWE
(2) 033744          T34FTB:
(2) 033744 006651          .WORD CTLRWE
(2) 033746 177777          .WORD -1
7012
7013 033750          ENDTST
```

7016
(2) 033752 000422
(2) 033754 042040 046105 052105
(2)
7017
7013
7019
7020
7021
7022
7023
7024
7025
7026
7027
7028
7029
7030
7031
7032
7033
7034
7035
7036
7037
7038
7039
7040
7041
7042
7043
7044
7045
7046
7047

.SBTTL TEST 35 - DELETED DATA WRITE PRT:2 - LGC TST
BR BGNT35 :BR TO BGN TST
T35MSG: .ASCIZ / DELETED DATA WRITE PRT:2 - LGC TST/
.EVEN

++
: TEST TO VERIFY THAT THE DEVICE LET A DELETED DATA MARK ON THE DISKETTE
: HEADER AND RETRIEVE THIS DATA ON A KNOWN GOOD DISKETTE. THIS IS DONE
: IN OPPOSITE DENSITY OF TEST 1.

: BGNST
: IF LOGIC TEST
: THEN-SETUP TEST IDENT
: CALL DEVICE DENSITY CK
: SET DENSITY CONTROL=DENSITY STATUS
: SET DELETED DATA FLAG (BIT#3-CMD)
: CALL WRITE SECTOR
: IF NO COMMAND ERROR (ESCAPE TEST)
: THEN-CALL READ SECTOR
: IF NO COMMAND ERROR (ESCAPE TEST)
: THEN-IF RXESR-DELETED DATA B.T NOT SET
: THEN-SET ERROR NUMBER=DELETED DATA ERR
: CALL ERROR
: ENDIF
: ENDIF
: ENDIF
: ENDTST

: BOARD CALLOUT:
: 1. CONTROLLER
: 2. R/W ELECTRONICS
:-----
:--

7050	034020				TSETUP			
(2)	034020	012737	034150	002466	BGNT35: MOV	#T35TBL,TSTID	:	SETUP TEST ID TBL-TEST# 35
(2)	034026	032737	000001	002324	IAT35: BIT	#LOGICT,TSTMOD	:	IF TEST MODE=LOGIC TEST
(2)	034034	001443			BEQ	XT35	:	BIT SET, THEN
(2)	034036	004737	020736		CALL	LTSTUP	:	CALL LOGIC TEST SETUP
7051	034042	004737	017350		CALL	DENCHK	:	CALL DEVICE DENSITY CHECK
7052	034046	004737	020472		CALL	SDENC	:	CALL SET DENSITY CONTROL=DENSITY STATUS
7053	034052	012737	000010	002402	MOV	#DLDCMD,DELDT	:	SET DELETED DATA FLAG
7054	034060	004737	010744		CALL	WRITE	:	CALL WRITE SECTOR
7055	034064	004737	011340		CALL	RDERCD	:	CALL READ ERROR CODE
7056	034070				IBT35: ESCAPE	TST	:	IF NO COMMAND ERROR, THEN
7057	034074	004737	017724		CALL	ERRCHK	:	CALL ERROR CHECK
7058	034100	004737	011062		CALL	READ	:	CALL READ SECTOR
7059	034104				ICT35: ESCAPE	TST	:	IF NO COMMAND ERROR, THEN
7060	034110	032777	000100	146234	IDT35: BIT	#DLDBIT,@RXDB	:	IF RXESR-DELETED DATA BIT
7061	034116	001006			BNE	LDT35	:	NOT SET, THEN
7062	034120	012737	000032	002520	MOV	#DLDTERR,ERRNBR	:	SETUP ERROR NUMBER=DELETED DATA ERROR
7063	034126	004737	003060		CALL	ERROR	:	CALL ERROR
7064	034132	000404			BR	XT35	:	BR TO EXIT TST
7065	034134	005037	002402		LDT35: CLR	DELDT	:	CLEAR DELETED DATA MODE
7066	034140	004737	010744		CALL	WRITE	:	CALL WRITE SECTOR - CLR DEL DAT FIELD
7067	034144				XT35: EXIT	TST	:	
7068	034150				REGTBL	CSESAL		
(1)		015036						REGS1=CSESAL
7069	034150				TTBL	REGCK,0		
(2)	034150	033754					T35TBL:	.WORD T35MSG
(2)	034152	000001						.WORD REGCK
(2)	034154	000000						.WORD 0
(2)	034156	177777						.WORD -1
(3)	034160						T35RTB:	
(3)	034160	015036						.WORD REGS1
(3)	034162	177777						.WORD -1
7070	034164				FRUTBL	CTLRWE		
(2)	034164						T35FTB:	
(2)	034164	006651						.WORD CTLRWE
(2)	034166	177777						.WORD -1
7071								
7072	034170				ENDTST			

7075
(2) 034172 000425
(2) 034174 042040 051511 042513
(2) 034246

.SBTTL TEST 36 - DISKETTE & DENSITY DATA CHECK - LGC TST
BR BGNT36 ;BR TO BGN TST
T36MSG: .ASCIZ / DISKETTE & DENSITY DATA CHECK - LGC TST/
.EVEN

7076
7077
7078
7079
7080
7081
7082
7083
7084
7085
7086
7087
7088
7089
7090
7091
7092
7093
7094
7095
7096
7097
7098
7099
7100
7101
7102
7103
7104
7105
7106
7107
7108
7109
7110
7111
7112
7113
7114
7115
7116
7117
7118
7119

++
: TEST TO VERIFY WITH A KNOWN GOOD DISKETTE THAT THE DEVICE WILL READ
: AND WRITE TO THE DISKETTE WITHOUT DATA ERRORS. BOTH DENSITIES WILL
: BE DONE.

```
-----  
: BGNTST  
: IF LOGIC TEST  
: THEN-SETUP TEST IDENT  
: CALL DENSITY CHECK  
: CALL SETUP DENSITY CONTROL=DENSITY STATUS  
: CLEAR DO FLAG  
: BGND0  
: : SET DATA PATTERN=RANDOM  
: : CALL DATA PATTERN GENERATOR  
: : SET TRACK & SECTOR INITIALIZE FLAG  
: : SET TRACK & SECTOR=SEQUENCE MODE  
: : BGND0  
: : : CALL GET TRACK & GET SECTOR  
: : : CALL FILL BUFFER  
: : : CALL WRITE SECTOR  
: : : SETUP TO CLEAR RX INTERNAL BUFFER  
: : : CALL FILL BUFFER-CLEAR INTERNAL BUFFER  
: : : SETUP DATA BUFFER  
: : : CALL READ SECTOR  
: : : CALL EMPTY BUFFER  
: : : CALL DATA CHECK  
: : : IF ERROR  
: : : : THEN-CALL DATA ANYLSIS ERROR  
: : : ENDF  
: : DUNTIL TRACK & SECTOR DONE OR DATA ERRORS=10  
: : CALL CHANGE DENSITY  
: : SET DENSITY CONTROL=DENSITY STATUS  
: : INCREMENT DO FLAG  
: DUNTIL DO FLAG=2 OR ABORT FLAG SET  
: NOP
```

ENDIF
ENDTST

: BOARD CALLOUT:
: 1. CONTROLLER
: 2. R/W ELECTRONICS
:-----

```

7122
7123 034246          TSETUP
(2) 034246 012737 034504 002466  BGNT36: MOV #T36TBL,TSTID ;SETUP TEST ID TBL-TEST# 36
(2) 034254 032737 000001 002324  IAT36: BIT #LOGICT,TSTMOD ;IF TEST MODE=LOGIC TEST
(2) 034262 001506          BEQ XT36 ;BIT SET, THEN
(2) 034264 004737 020736          CALL LTSTUP ;CALL LOGIC TEST SETUP
7124 034270 005037 002504          CLR TTEMP1 ;CLEAR COUNTER (TEST TEMP #1)
7125 034274 004737 017350          CALL DENCHK ;CALL DENSITY CHECK
7126 034300 004737 020472          CALL SDENC ;CALL SETUP DENSITY CONTROL=DENSITY STATUS
7127 034304 005037 012660  BBT36: CLR PAT ;SETUP DATA PATTERN=RANDOM
7128 034310 004737 012306          CALL STDATP ;CALL SET DATA PATTERN
7129 034314 052737 001400 002510  BIS #ITK!ISC,TKSCFG ;SET TRACK & SECTOR INITIALIZE FLAGS
7130 034322 052737 000003 002510  BIS #STK!SSC,TKSCFG ;SET TRACK & SECTOR SEQUENCE MODE FLAGS
7131 034330 004737 012662  BCT36: CALL GETTRK ;CALL GET TRACK
7132 034334 004737 013104          CALL GETSEC ;CALL GET SECTOR
7133 034340 004737 010510          CALL FILBUF ;CALL FULL BUFFER
7134 034344 004737 010744          CALL WRITE ;CALL WRITE SECTOR
7135 034350 004737 013642          CALL CLRDAT ;CALL CLEAR DATA BUFFER
7136 034354 012737 036622 002362  MOV #DATBUF,FILADR ;SETUP TO CLEAR RX INTERNAL BUFFER
7137 034362 004737 010510          CALL FILBUF ;CLEAR THE BUFFER
7138 034366 012737 036222 002362  MOV #DATPAT,FILADR ;SETUP DATA BUFFER ADDRESS
7139 034374 004737 011062          CALL READ ;CALL READ SECTOR
7140 034400 004737 010626          CALL EMPBUF ;CALL EMPTY BUFFER
7141 034404 004737 013246          CALL DATAK ;CALL DATA CHECK
7142 034410 022737 000012 013520  UCT36: CMP #10,DAERCT ;DOUNTIL DATA ERROR COUNT
7143 034416 001410          BEQ ECT36 ;EQUALS 10, OR
7144 034420 032737 001000 002476  BIT #TRKDON,FLAGST ;TRACKS DONE FLAG
7145 034426 001740          BEQ BCT36 ;SET, AND
7146 034430 032737 002000 002476  BIT #SECDON,FLAGST ;SECTORS DONE FLAG
7147 034436 001734          BEQ BCT36 ;SET
7148 034440 004737 020430  ECT36: CALL CDENC ;CALL COMPLIMENT DENSITY CONTROL
7149 034444 004737 011172          CALL SETDN ;CALL SET DENSITY
7150 034450 005237 002504          INC TTEMP1 ;INCREMENT COUNTER
7151 034454 012737 000100 002370  MOV #64.,WDCNT ;SET WORD COUNT
7152 034462 005737 002454  UBT36: TST FIN ;DOUNTIL FIN FLAG
7153 034466 001004          BNE XT36 ;SET OR
7154 034470 022737 000002 002504  CMP #2,TTEMP1 ;COUNT
7155 034476 001302          BNE BBT36 ;EAUALS 2
7156 034500          XT36: EXIT TST
7157
7158 034504          REGTBL CSESAL
(1) 015036          REGS1=CSESAL
7159 034504          TTBL REGCK,0
(2) 034504 034174          T36TBL: .WORD T36MSG
(2) 034506 000001          .WORD REGCK
(2) 034510 000000          .WORD 0
(2) 034512 177777          .WORD -1
(3) 034514          T36RTB:
(3) 034514 015036          .WORD REGS1
(3) 034516 177777          .WORD -1
7160 034520          FRUTBL CTLRWE
(2) 034520          T36FTB:
(2) 034520 006651          .WORD CTLRWE
(2) 034522 177777          .WORD -1
7161
7162 034524          ENDTST

```


HARDWARE TESTS MACY11 30(1046) 12-APR-82 13:23 PAGE 145
CZRFB.P11 09-APR-82 15:14 TEST 36 - DISKETTE & DENSITY DATA CHECK - LGC TST

B 14

SEQ 0170

7170 034526
7171

ENDMOD

7174
7175
7186
7187
7215
7216 034526
7217
7218
7219
7220
7221
7222
7223
7224
7225
7226
7227 034526
7228
7229 034530
7230 034540
7231 034550
7232 034562
7233 034574
7234
7240 034606
7241
7242 034606 054122 041040 051525
7243 034621 126 041505 047524
7244 034634 051104 053111 020105
7245 034647 105 050130 053440
7246 034662 051102 046055 053105
7247 034676
7248
7249
7256

```
.NLIST BEX,ME
.TITLE PARAMETER CODING

.SBTTL HARDWARE PARAMETER CODING SECTION

      BGNMOD

:++
: THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
: THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
: MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
: INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
: MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
: WITH THE OPERATOR.
:--

      BGNHRD

      GPRMA MSG1,0,0,0,177777,YES
      GPRMA MSG2,2,0,0,177777,YES
      GPRMD MSG3,4,0,177777,0,1,YES
      GPRMD MSG4,6,0,177777,0,1,YES
      GPRMD MSG4A,10,0,177777,0,7,YES

      ENDHRD
-----
MSG1: .ASCIZ /RX BUS ADR/
MSG2: .ASCIZ /VECTOR ADR/
MSG3: .ASCIZ /DRIVE # /
MSG4: .ASCIZ /EXP WRD-CR/
MSG4A: .ASCIZ /BR-LEVEL /
      .EVEN
-----
```

7259
7260
7261
7262
7263
7264
7265
7266
7267
7268
7269
7270 034676
7271
7272 034700
7273 034706
7274 034710
7275 034716
7276 034724
7277 034732
7278 034744
7279 034754
7280 034766
7281 034774
7282 034776
7283 035004
7290 035012
7291
7292 000015
7293 000012
7294 035012 054105 040520 051516
7295 035044 042524 052123 044040
7296 035057 104 040511 047107
7297 035106 020040 020040 020040
7298 035160 020040 026440 052506
7299 035213 040 020040 020040
7300 035301 040 020040 046055
7301 035332 020040 020040 020040
7302 035403 040 020040 020040
7303 035461 040 020040 020040
7304 035552 020040 020040 020040
7308 035651 124 050131 020105
7309 035677 114 043517 041511
7310 035722 052506 041516 044524
7311 035745 110 051101 020104
7312 036014 047516 026516 054105
7313 036053 105 052130 047105
7314 036124 042524 052123 041440
7315 036150 020040 050040 044522
7316
7317 036222

.SBTTL SOFTWARE PARAMETER CODING SECTION

:++
: THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
: THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
: MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
: INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
: MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
: WITH THE OPERATOR.
:--

BGNSFT

GPRML MSG6,2,1,YES
XFERF 1\$
1\$: GPRML MSG7,2,2,YES
GPRML MSG8,4,LOGICT,YES
GPRML MSG9,4,FUNCTT,YES
GPRMD MSG10,0,0,177777,1,177777,YES
GPRMA MSG14,24,0,0,177777,YES
GPRMD MSG15,26,0,030000,0,3,YES
GPRML MSG17,2,100,YES
XFERF 6\$
6\$: GPRML MSG20,12,20,YES
GPRML MSG5,12,SIDFLG,YES
ENDSFT

CR==15 ;CARRIAGE RETURN
LF==12 ;LINE FEED
MSG5: .ASCIZ /EXPANSION WORD TYPE <CR> /
MSG6: .ASCIZ /TEST HELP /
MSG7: .ASCII /DIAGNOSTIC MODES ARE: /<CR><LF>
.ASCII / LOGIC TEST, FUNCTION TEST, OR 30TH/<CR><LF>
.ASCII / -FUNCTION TESTS (1-10)/<CR><LF>
.ASCII / ACT AS QUICK VERIFY & REPORT FAILING FUNCTIONS/<CR><LF>
.ASCII / -LOGIC TESTS (11-36)/<CR><LF>
.ASCII / ANALYZE FAILURE & GIVE ERROR INFO/<CR><LF>
.ASCII / REPORT FIELD REPLACEABLE UNITS 'FRU'S' /<CR><LF>
.ASCII / ->DEVICE FATAL THRESHOLD LEVEL (DVTL) IS SET = 1/<CR><LF>
.ASCII / 'DVTL' = NO. OF HARD ERRS THAT CAUSE DEVICE FATAL ERR/<
.ASCIZ /TYPE "CR" TO CONTINUE/
MSG8: .ASCIZ /LOGIC TEST MODE /
MSG9: .ASCIZ /FUNCTION TEST MODE/
MSG10: .ASCIZ /HARD ERR -> DEVICE FATAL THRESHOLD LVL/
MSG14: .ASCIZ /NON-EXISTANT MEM ADR (NXM TST)/
MSG15: .ASCIZ /EXTENDED ADR BITS: 13 & 12 (NPR-NXM TST)/
MSG17: .ASCIZ /TEST CONTROL FLAGS /
MSG20: .ASCIZ / PRINT ONLY 10 DATA ERRORS & CONTINUE /

.EVEN

7320
7321
7322
7323
7324 036222 000400
7327
7328
7329
7330
7331
7332 036622 000400
7335 037222 000000
7336 037224 000000
7337
7344
7345
7346 037226 000000
7347 037630
7348
7349 037630
(3) 037634
7350 037634
7351
7352 037634
7353 037634
7354 037640 177170
7355 037642 000264
7356 037644 000000
7357 037646 000000
7358 037650 000005
7359 037652
7360 037652
7361 037656 177170
7362 037660 000264
7363 037662 000001
7364 037664 000000
7365 037666 000005
7366 037670
7367 037670
7368 000001

.SBTTL - RX02 FILL BUFFER AREA

DATPAT:

.SBTTL - RX02 EMPTY BUFFER AREA

DATBUF:

.WORD 0
.WORD 0

.SBTTL - PATCH AREA

PATCH: 0 ;PATCH AREA
.=.+400

LASTAD

LSLAST::
ENDMOD

BGNSETUP 2

BGNPTAB

177170

264

0

0

5

ENDPTAB

BGNPTAB

177170

264

1

0

5

ENDPTAB

ENDSETUP

.END

BGNT34	033616	6960	6994#							
BGNT35	034020	7016	7050#							
BGNT36	034246	7075	7123#							
BGNT4	022766	5218	5235#							
BGNT5	023102	5246	5264#							
BGNT6	023230	5276	5299#							
BGNT7	023414	5317	5351#							
BGNT8	023664	5381	5404#							
BGNT9	024036	5423	5458#							
BITCNT	014672	3896*	3899	3909*	3912	3914*	3918#			
BITLIM	014674	3793*	3814*	3912	3919#					
BITOFF	014676	3792*	3813*	3889	3891*	3894	3920#			
BIT0	= 000001	G 1566#	1589	1592	1597	1687	1705			
BIT00	= 000001	G 1566#								
BIT01	= 000002	G 1566#								
BIT02	= 000004	G 1566#								
BIT03	= 000010	G 1566#								
BIT04	= 000020	G 1566#								
BIT05	= 000040	G 1566#								
BIT06	= 000100	G 1566#								
BIT07	= 000200	G 1566#								
BIT08	= 000400	G 1566#								
BIT09	= 001000	G 1566#								
BIT1	= 000002	G 1566#	1588	1593	1598	1688	1706			
BIT10	= 002000	G 1566#	1580	1696	1720					
BIT11	= 004000	G 1566#	1573	1579	1697	1719				
BIT12	= 010000	G 1566#	1594	1698	1712	3342				
BIT13	= 020000	G 1566#	1699	3352	4446					
BIT14	= 040000	G 1566#	1572	1700	1713	2038	2570	4057		
BIT15	= 100000	G 1566#	1571	1701	1714					
BIT2	= 000004	G 1566#	1587	1601	1689	1707	2105			
BIT3	= 000010	G 1566#	1570	1586	1690	1708				
BIT4	= 000020	G 1566#	1578	1585	1691	1718	3354	3390	3430	
BIT5	= 000040	G 1566#	1577	1584	1692	4239	4272			
BIT6	= 000100	G 1566#	1583	1693	1709					
BIT7	= 000200	G 1566#	1576	1582	1694	1710				
BIT8	= 000400	G 1566#	1575	1581	1595	1711				
BIT9	= 001000	G 1566#	1574	1596	1695					
BOE	= 000400	G 1566#								
BRONPT	012372	3493*	3501*	3507#						
BTRK	002451	1793#	4078							
BTRP4	= 000004	1590#	5081	5083	6076	6083				
BTRP6	= 000006	1591#								
BYTCNT	013514	3659*	3660*	3694*	3695	3700#				
BYTNUM	013516	3661*	3662	3689	3693*	3701#				
CABLES	= 000010	1673#								
CDENC	020430	4516#	5363*	5371*	5469*	6442*	6547*	6945*	7001*	7148*
CDERCK	011544	2928*	2986*	3022*	3125*	3189*	3256#			
CDMS	010177	2698	2716#							
CEINIT	= 015046	3993#	6228	6396						
CKBITS	014522	3795*	3816*	3882#						
CKERR	= 020000	1699#								
CLRCR	021122	4600*	4621*	4666#						
CLRDAT	013642	3697*	3716#	7135*						
CLRDEV	010472	2949#								
CLRERR	010300	2051*	2735#							

ERMS46	005153	2265	2323#											
ERMS47	005202	2266	2324#											
ERMS48	005230	2267	2325#											
ERMS49	005260	2268	2326#											
ERMS5	003752	2224	2282#											
ERMS50	005307	2269	2327#											
ERMS51	005337	2270	2328#											
ERMS6	003763	2225	2283#											
ERMS7	003774	2226	2284#											
ERNBEV	003344	2013*	2091#											
ERNTCK	020170	4423*	4434*	4477#										
ERRBIT=	100000	1571#	4204	4419	5734									
ERRBLK	002524	1828#	2014*	2739*										
ERRCHK	017724	4411#	5148*	5151*	5179*	5183*	5211*	5239*	5269*	5305*	5307*	5360*	5362*	5367*
		5369*	5410*	5412*	5415*	5462*	5465*	5468*	5471*	5473*	5476*	5479*	5516*	5752*
		5785*	5838*	5845*	5891*	5894*	5896*	5899*	5971*	5973*	6081*	6226*	6388*	6393*
		6441*	6448*	6546*	6553*	6599*	6671*	6678*	6684*	6768*	6788*	6944*	6951*	7000*
		7007*	7057*											
ERRCMD	002422	1771#	2569	3425*										
ERRCTR	003342	2044*	2045	2050*	2056#									
ERRFLG=	100000	1701#	2742	3796	3817	4010	4425							
ERRMSG	002522	1828#	4499	4602*	4623*									
ERRNBR	002520	1828#	2010	2012	2018	2092	2096	2100	2206	2737*	3146	3173	3259	3341*
		3351*	3355*	3389*	3426*	3432*	3677*	4210*	4240*	4293*	4352*	4359*	4367*	4427
		4442*	4444*	5573*	5580*	5630*	5637*	5687*	5694*	5738*	5743*	5748*	5954*	5976*
		5982*	5988*	6166*	6172	6182	6188*	6339*	6604*	6688*	6762*	6785*	6862*	6869*
		6874*	6879*	6883*	7062*									
ERRNST=	000016	1629#												
ERROLD	003340	2042	2048*	2055#										
ERROR	003060	2009#	3263*	3678*	4211*	4241*	4294*	4353*	4361*	4369*	4451*	5574*	5581*	5631*
		5638*	5688*	5695*	5739*	5744*	5749*	5955*	5977*	5983*	5989*	6120*	6189*	6195*
		6343*	6605*	6689*	6764*	6786*	6863*	6870*	6875*	7063*				
ERRREG	015444	4081*	4084#											
ERRSAV	003336	2012*	2042	2048	2049*	2054#								
ERRTYP	002516	1828#	2094*	2098*	2102*	2103	2110*	2738*						
ERTKMS	010241	2709	2718#											
ESERTB	014746	3815	3949#											
ESRCHK	014112	3800#												
ESRCMP	014264	3775*	3804*	3805	3812	3829#								
ESRMSK	014266	3776*	3802	3803	3830#									
ESRSET	014272	3803*	3804	3832#	3840*	3859*	3862*	3865*	3873*					
EVL =	000004	1566#												
EXMS	010156	2650	2689	2715#										
EXTADR	002366	1755#	2975	3011	3178									
ESEND =	002100	1100#												
ESLOAD=	000035	1100#	1146											
FBCMD =	000000	1605#												
FILADR	002362	1753#	2984	4645*	7136*	7138*								
FILBUF	010510	2970#	5210*	5304*	5893*	5970*	6274*	6333*	7133*	7137*				
FILERR=	000012	1625#												
FIN	002454	1798#	2378	2740*	2971	2978	2982	3007	3014	3018	3043	3051	3055	3079
		3086	3090	3114	3121	3181	3185	3257	3391*	3423*	3762	4277	4667*	6758
		6780	6858	6884	7152									
FLAGSP	002500	1810#	2016	2026	2029	2032	2404	2582	3658*	3674	3676*	3849	4360*	4368*
		4371*	4417	4452*	4498	4635*	4781*	4786*	4789*	4791*	4792*	4796*	5181*	5184*
		5565	5889*	5892*	6073	6184	6390*	6394*	6763*	6765*	6784*			

FSCLEA= 000007	1100#	4907	4916											
FSDU = 000016	1100#	4935	4948											
FSEND = 000041	1100#	1127	1245	1559	1920	1924	1928	2213	4689	4733	4747	4831	4916	
	4948	4971	4990	5003	5054	5124	5152	5155	5158	5185	5188	5191	5212	
	5215	5218	5240	5243	5246	5270	5273	5276	5311	5314	5317	5375	5378	
	5381	5417	5420	5423	5480	5483	5486	5520	5523	5527	5582	5586	5589	
	5639	5642	5645	5697	5701	5704	5753	5757	5760	5786	5791	5794	5846	
	5856	5859	5900	5905	5908	5990	5994	6044	6085	6090	6093	6123	6127	
	6130	6197	6203	6206	6227	6231	6234	6275	6277	6286	6290	6293	6334	
	6336	6352	6356	6359	6392	6395	6399	6402	6440	6447	6449	6453	6456	
	6497	6502	6506	6509	6545	6552	6554	6558	6561	6598	6601	6609	6613	
	6616	6670	6675	6683	6694	6698	6701	6790	6794	6797	6897	6901	6904	
	6943	6950	6952	6957	6960	6999	7006	7008	7013	7016	7056	7059	7067	
	7072	7075	7156	7162	7170	7216	7240	7290	7350	7352	7353	7359	7360	
	7366	7367												
FSHARD= 000004	1100#	7227	7240	7273	7281									
FSHW = 000013	1100#	1185	1196											
FSINIT= 000006	1100#	4768	4831											
FSJMP = 000050	1100#	5152	5185	5212	5240	5270	5311	5375	5417	5480	5520	5582	5639	
	5697	5753	5786	5846	5900	5990	6085	6123	6197	6227	6286	6352	6395	
	6449	6502	6554	6609	6694	6790	6897	6952	7008	7067	7156			
FSMOD = 000000	1100#	1127	1245	1559	4689	4733	5003	5054	7170	7216	7350			
FSMSG = 000011	1100#	1918	1920	1922	1924	1926	1928	2205	2213					
FSPROT= 000021	1100#	1205	1209											
FSPWR = 000017	1100#													
FSRPT = 000012	1100#	4740	4747											
FSSEG = 000003	1100#													
FSSOFT= 000005	1100#	7270	7273	7281	7290									
FSSRV = 000010	1100#													
FSSUB = 000002	1100#													
FSSW = 000014	1100#	1217	1236											
FSTEST= 000001	1100#	5124	5155	5158	5188	5191	5215	5218	5243	5246	5273	5276	5314	
	5317	5378	5381	5420	5423	5483	5486	5523	5527	5586	5589	5642	5645	
	5701	5704	5757	5760	5791	5794	5856	5859	5905	5908	5994	6044	6090	
	6093	6127	6130	6203	6206	6231	6234	6290	6293	6356	6359	6399	6402	
	6453	6456	6506	6509	6558	6561	6613	6616	6698	6701	6794	6797	6901	
	6904	6957	6960	7013	7016	7072	7075	7162						
GETREG 012244	2929*	2987*	3023*	3059*	3094*	3126*	3153*	3188*	3451#	5147*				
GETSEC 013104	3628#	5514*	6755*	7132*										
GETTRK 012662	3585#	5513*	6494*	6854*	7131*									
GTECEN 015106	4003#	4421*												
GTECOF 017106	2386*	4011*	4102*	4165#										
GSCNTO= 000200	1100#													
GSDELM= 000372	1100#													
GSDISP= 000003	1100#													
GSEXCP= 000400	1100#													
GSHILI= 000002	1100#													
GSLOLI= 000001	1100#													
GSNO = 000000	1100#													
GSOFFS= 000400	1100#	5832	5840	7229	7230	7231	7232	7233	7272	7274	7275	7276	7277	
	7278	7279	7280	7282	7283									
GSOFSI= 000376	1100#	5832	5840	7229	7230	7231	7232	7233	7272	7274	7275	7276	7277	
	7278	7279	7280	7282	7283									
GSPRMA= 000001	1100#	7229	7230	7278										
GSPRMD= 000002	1100#	7231	7232	7233	7277	7279								
GSPRML= 000000	1100#	5832	5840	7272	7274	7275	7276	7280	7282	7283				

IAT23	030314	6271#		
IAT24	030516	6330#		
IAT25	030770	6386#		
IAT26	031146	6435#		
IAT27	031340	6489#		
IAT28	031536	6541#		
IAT29	031740	6593#		
IAT3	022656	5207#		
IAT30	032140	6665#		
IAT31	032436	6750#		
IAT32	033010	6849#		
IAT33	033424	6938#		
IAT34	033624	6994#		
IAT35	034026	7050#		
IAT36	034254	7123#		
IAT4	022774	5235#		
IAT5	023110	5264#		
IAT6	023236	5299#		
IAT7	023422	5351#		
IAT8	023672	5404#		
IAT9	024044	5458#		
IBDCK	013312	3665#		
IBDDC	017410	4279#		
IBDSC	017152	4206#		
IBE =	010000 G	1566#		
IBECK	017742	4413	4415#	
IBEMB	010672	3014#		
IBENC	020204	4481#		
IBENV	003364	2093	2096#	
IBERR	003140	2017	2019	2021#
IBFLB	010554	2978#		
IBFRU	005424	2379	2382#	
IBGEN	015116	4006#		
IBGSC	013140	3630	3634#	
IBGTK	012770	3600#		
IBI1	021666	4863	4867#	
IBNAT	027006	6022#		
IBPCE	007122	2580#		
IBREC	011414	3181#		
IBRED	011124	3086#		
IBSDN	011234	3121#		
IBSRC	014416	3857#		
IBSTA	015270	4055#		
IBTKE	017540	4344#		
IBTKP	010042	2693#		
IBT11	024502	5568	5570#	
IBT12	024714	5627#		
IBT13	025140	5683#		
IBT14	025342	5734#		
IBT16	025710	5829#		
IBT18	026506	5952#		
IBT19	027170	6073#		
IBT29	031776	6598#		
IBT30	032334	6687	6690#	
IBT35	034070	7056#		
IBUWCH	011704	3337#	3346	

IBWRT	011014	3051#			
ICDDC	017464	4291#			
ICDSC	017160	4208#			
ICECK	020012	4416	4418	4420	4425#
ICEMB	010712	3018#			
ICENC	020214	4483#			
ICERR	003214	2030	2032#		
ICFLB	010574	2982#			
ICGTK	012742	3587	3595#		
ICNAT	027046	6031#			
ICPCE	007130	2582#			
ICREC	011434	3185#			
ICRED	011144	3090#			
ICSRC	014432	3858	3860#		
ICSTA	015276	4057#			
ICTKE	017550	4346#			
ICT11	024546	5577#			
ICT12	024756	5634#			
ICT13	025210	5691#			
ICT14	025352	5736#			
ICT16	025754	5833#			
ICT18	026652	5974#			
ICT21	027640	6172#			
ICT23	030372	6280#			
ICT24	030566	6337#			
ICT29	032012	6601#			
ICT31	032510	6758#			
ICT35	034104	7059#			
ICUWCH	011712	3339#			
ICWRT	011034	3055#			
ID	002336 G	1225#	3589	3606	
IDCOMP	013020	3606#			
IDDDC	017354	4272#			
IDDSMS	020642	4580	4584#		
IDECK	020022	4427#			
IDENC	020226	4484	4487#		
IDENT1	015446	4049	4086#		
IDENV	003404	2097	2100#		
IDERR	003242	2025	2038#		
IDGSC	013224	3646#			
IDSRC	014446	3861	3863#		
IDSSMS	020657	4575	4585#		
IDTKE	017556	4348#			
IDT11	024454	5565#			
IDT14	025376	5737	5741#		
IDT16	026050	5841#			
IDT18	026700	5980#			
IDT24	030640	6347#			
IDT29	032016	6602#			
IDT31	032516	6760#			
IDT32	033066	6858#			
IDT35	034110	7060#			
IDU	= 000040 G	1566#			
IEDCK	013322	3668#			
IEERR	003164	2026#			
IER	= 020000 G	1566#			

LSHW	002276	G	1146	1185#
LSICP	002104	G	1146#	
LSINIT	021214	G	1146	4768#
LSLADP	002026	G	1146#	
LSLAST	037634	G	1146	7349# 7367
LSLOAD	002100	G	1146#	
LSLUN	002074	G	1146#	
LSMREV	002050	G	1146#	
LSNAME	002000	G	1146#	
LSPRIO	002042	G	1146#	
LSPROT	002310	G	1146	1205#
LSPRT	002112	G	1146#	
LSREPP	002062	G	1146#	
LSREV	002010	G	1146#	
LSRPT	021212	G	4740#	
LSSOFT	034700	G	1146	7270#
LSSFC	002056	G	1146#	
LSSPCP	002020	G	1146#	
LSSPTP	002024	G	1146#	
LSSTA	002030	G	1146#	
LSSW	002320	G	1146	1217#
LSTEST	002114	G	1146#	
LSTIML	002014	G	1146#	
LSUNIT	002012	G	1146#	4802
L10000	002310		1185	1196#
L10002	002350		1217	1236#
L10003	002532		1920#	
L10004	002540		1924#	
L10005	002546		1928#	
L10006	003532		2213#	
L10007	021212		4747#	
L10010	021522		4831#	
L10011	021774		4916#	
L10012	022010		4948#	
L10013	022060		4971#	
L10014	022064		4990#	
L10015	022454		5152	5155#
L10016	022614		5185	5188#
L10017	022730		5212	5215#
L10020	023046		5240	5243#
L10021	023164		5270	5273#
L10022	023350		5311	5314#
L10023	023612		5375	5378#
L10024	024002		5417	5420#
L10025	024220		5480	5483#
L10026	024362		5520	5523#
L10027	024620		5582	5586#
L10030	025030		5639	5642#
L10031	025260		5697	5701#
L10032	025500		5753	5757#
L10033	025630		5786	5791#
L10034	026232		5846	5856#
L10035	026426		5900	5905#
L10036	026772		5990	5994#
L10037	027336		6085	6090#
L10040	027500		6123	6127#

PRTFRU	005404	2035*	2377#												
PRTGMS	007620	2622	2630#												
PRTREG	007564	2020*	2028*	2621#											
PRTSEC	007674	2031*	2646#												
PRTSTA	015240	2036*	2040*	4049#											
PRTRK	010002	2034*	2685#												
PRTX0S	002714	1945#	4104*												
PRTX1S	002734	1948#	2586*	4099*											
PRTX2S	002756	1951#	6193*												
PRTX3S	003002	1954#	4073*												
PRTX4S	003030	1957#	4079*												
PTERTY	003474	2015*	2128#												
PTUTMS	020560	4552	4556#												
PWDNRY	026175	5832	5852#												
PWRMS	026122	5831	5839	5851#											
PWUPRY	026214	5840	5853#												
RANDAT	012566	3514	3553#	3558											
RANGEN	010344	2887#	3553	3604*	3642*										
RANUM	010436	2902*	2907#	3554	3605*	3606	3609	3612	3614	3643*	3644	3646	3648*	3649	
RAN1	010432	2888	2894*	2899	2905#										
RAN2	010434	2889	2896	2901*	2906#										
RDERR =	000003	3172#	5182*	5751*	5890*	5951*	6078*	6335*	6391*	6430*	6446*	6496*	6551*	6674*	
RDSTAT	011266	6757*	6779*	6857*	6942*	6949*	7005*	7055*							
READ	011062	1618#	4028	4030	4031	4032	4035	4036	6775						
		3145#	4236*	5268*	5898*										
		3078#	4276*	4285*	5361*	5368*	5411*	5464*	5467*	5472*	5475*	5515*	6438*	6445*	
		6495*	6600*	6756*	6778*	6856*	6941*	6948*	7058*	7139*					
READ1	011112	3084#													
RECADR	002364	1754#	3183	4653*	5180*	6077*									
RECCMD=	000016	1612#													
RECERN	002462	1801#	4013*												
RECERR=	000021	1632#	6874	6883											
RECFLG=	000200	1694#	2384	2647	2686	3187	4065	4080	4344	4415	4452	5956	6080		
RECTST=	000200	1710#	3849	4417	5181	5184	5889	5892	6390	6394	6452	5956	6080		
REGACT	002440	1778#	2623	2626*	3785*	5572*	5579*	5629*	5636*	5685*	5686*	5693*	6340*	6341*	
REGCK =	000001	1687#	4412	4603	5755	5789	5848	5903	5992	6088	6229	6397	6451	6504	
		6556	6611	6696	6792	6954	7010	7069	7159	6088	6229	6397	6451	6504	
REGEXP	002436	1777#	2624	2627*	3784*	5567*	5569*	5570	5575*	5577	5626*	5627	5632*	5634	
		5682*	5683	5689*	5691	6342*									
REGSCK	013664	3760#	4414*												
REGS1 =	015036	5153#	5154	5186#	5187	5213#	5214	5241#	5242	5271#	5272	5312#	5313	5376#	
		5377	5418#	5419	5481#	5482	5521#	5522	5583#	5584	5640	5699	5754#	5755	
		5788#	5789	5847#	5848	5902#	5903	5991#	5992	6086#	6088	6124#	6125	6200#	
		6201	6228#	6229	6287#	6288	6353#	6354	6396#	6397	6450#	6451	6503#	6504	
		6555#	6556	6610#	6611	6695#	6696	6791#	6792	6898#	6899	6953#	6954	7009#	
		7010	7068#	7069	7158#	7159									
REGS2 =	000000	5153#	5154	5186#	5187	5213#	5214	5241#	5242	5271#	5272	5312#	5313	5376#	
		5377	5418#	5419	5481#	5482	5521#	5522	5583#	5584	5640	5699	5754#	5755	
		5788#	5789	5847#	5848	5902#	5903	5991#	5992	6086#	6088	6124#	6125	6200#	
		6201	6228#	6229	6287#	6288	6353#	6354	6396#	6397	6450#	6451	6503#	6504	
		6555#	6556	6610#	6611	6695#	6696	6791#	6792	6898#	6899	6953#	6954	7009#	
		7010	7068#	7069	7158#	7159									
REGS3 =	000000	5153#	5154	5186#	5187	5213#	5214	5241#	5242	5271#	5272	5312#	5313	5376#	
		5377	5418#	5419	5481#	5482	5521#	5522	5583#	5584	5640	5699	5754#	5755	
		5788#	5789	5847#	5848	5902#	5903	5991#	5992	6086#	6088	6124#	6125	6200#	
		6201	6228#	6229	6287#	6288	6353#	6354	6396#	6397	6450#	6451	6503#	6504	

SETDN	011172	3113#	5461*	5470*	5478*	6669*	6673*	6682*	6693*	7149*				
SETSCD	011502	2974*	3010*	3047*	3082*	3117*	3149*	3176*	3228#					
SETUP	021362	4793	4797	4800#										
SFPTBL	002320	1217#												
SFTSTS	002450	1792#	4077											
SIDE	002410	1765#	3211	4869*	4872*									
SIDE1 =	001000	1574#	3843	3857	3859	4872								
SIDFLG=	010000	1594#	4577	7283										
SIDPRT	002515	1818#	4579	4870*	4873*									
SIDRDY=	000002	1588#												
SIDWRG=	000026	1638#	3958											
SSC =	000002	1598#	6754	7130										
STAFLG=	100000	1714#	4781	4792										
START	021252	4782#												
STARTO	021262	4784#												
START1	021322	4788	4790	4792#										
STATER	017222	4212	4218#											
STCMD =	000012	1610#												
STDATP	012306	3493#	5303*	5959*	6273*	7128*								
STDNER=	000035	1645#	6688											
STK =	000001	1597#	3600	6852	6855	7130								
STTK76	021174	4684#	5372*	5405*	5463*	5474*								
SUDVCD	021014	4604*	4625*	4643#	5209*	5237*	5267*	5301*						
SUM	012654	3494*	3565*	3570	3571*	3572	3576#							
SURGCK	014276	3764*	3839#											
SUTSFG	020772	4624*	4631#											
SVCGBL=	000000	1100#	1109#	1146	1147	1157	1170	1185	1205	1217	1828	1918	1922	1926
		2205	4740	4768	4907	4935	4969	4983	7227	7270	7349#			
SVCINS=	177777	1100#	1106#	1146	1147	1157	1170	1185	1217	1920	1924	1928	1930	1933
		1936	1939	1942	1945	1948	1951	1954	1957	2129	2213	3264	3265	3313
		3334	3344	3356	3383	3417	4214	4215	4295	4296	4747	4775	4777	4778
		4779	4780	4782	4783	4784	4785	4794	4795	4798	4799	4804	4805	4808
		4810	4812	4831	4908	4909	4916	4948	4971	4990	5081	5083	5088	5089
		5145	5152	5155	5185	5188	5212	5215	5240	5243	5270	5273	5311	5314
		5375	5378	5417	5420	5480	5483	5520	5523	5561	5582	5586	5623	5639
		5642	5679	5697	5701	5733	5753	5757	5786	5791	5829	5830	5831	5832
		5839	5840	5846	5856	5900	5905	5990	5994	6076	6083	6085	6090	6117
		6123	6127	6164	6197	6203	6227	6231	6275	6277	6286	6290	6334	6336
		6352	6356	6392	6395	6399	6440	6447	6449	6453	6497	6502	6506	6545
		6552	6554	6558	6598	6601	6609	6613	6670	6675	6683	6694	6698	6790
		6794	6897	6901	6943	6950	6952	6957	6999	7006	7008	7013	7056	7059
		7067	7072	7156	7162	7227	7229	7230	7231	7232	7233	7240	7270	7272
		7273	7274	7275	7276	7277	7278	7279	7280	7281	7282	7283	7290	7349
		7353	7360											
SVCSUB=	177777	1100#	1108#											
SVCTAG=	177777	1100#	1110#	1196	1236	1920	1924	1928	2213	4747	4831	4916	4948	4971
		4990	5155	5188	5215	5243	5273	5314	5378	5420	5483	5523	5586	5642
		5701	5757	5791	5832	5840	5856	5905	5994	6090	6127	6203	6231	6290
		6356	6399	6453	6506	6558	6613	6698	6794	6901	6957	7013	7072	7162
		7240	7290	7353	7359	7360	7366							
SVCTST=	177777	1100#	1107#	5124	5158	5191	5218	5246	5276	5317	5381	5423	5486	5527
		5589	5645	5704	5760	5794	5859	5908	6044	6093	6130	6206	6234	6293
		6359	6402	6456	6509	6561	6616	6701	6797	6904	6960	7016	7075	
SWREG	002332	1223#	2038	3670	3786	3807	4446	4577						
SYFERR=	002000	1720#												
SYSERR	002456	1799#	3354*	3390*	3427*	3430*	4239*	4272						

TSEXCP= 000000	7229#	7230#	7231#	7232#	7233#	7277#	7278#	7279#					
TSFLAG= 000040	5152#	5185#	5212#	5240#	5270#	5311#	5375#	5417#	5480#	5520#	5582#	5639#	5697#
	5753#	5786#	5846#	5900#	5990#	6085#	6123#	6197#	6227#	6275#	6277#	6286#	6334#
	6336#	6352#	6392#	6395#	6440#	6447#	6449#	6497#	6502#	6545#	6552#	6554#	6598#
	6601#	6609#	6670#	6675#	6683#	6694#	6790#	6897#	6943#	6950#	6952#	6999#	7006#
	7008#	7056#	7059#	7067#	7156#								
TSFREE= 037670	7349	7367#											
TSGMAN= 000000	1100#												
TSHILI= 000003	7229#	7230#	7231#	7232#	7233#	7277#	7278#	7279#					
TSLAST= 000001	1100#	7349#	7352										
TSLOLI= 000000	7229#	7230#	7231#	7232#	7233#	7277#	7278#	7279#					
TSLSYM= 010000	1100#	1196	1236	1920	1924	1928	2213	4747	4831	4916	4948	4971	4990
	5155	5188	5215	5243	5273	5314	5378	5420	5483	5523	5586	5642	5701
	5757	5791	5856	5905	5994	6090	6127	6203	6231	6290	6356	6399	6453
	6506	6558	6613	6698	6794	6901	6957	7013	7072	7162	7240	7290	
TSLTNO= 000044	7349#												
TSNEST= 177777	1100#	1127#	1185#	1196#	1205#	1209#	1217#	1236#	1245#	1559#	1918#	1920#	1922#
	1924#	1926#	1928#	2205#	2213#	4689#	4733#	4740#	4747#	4768#	4831#	4907#	4916#
	4935#	4948#	4969#	4971#	4983#	4990#	5003#	5054#	5124#	5155#	5158#	5188#	5191#
	5215#	5218#	5243#	5246#	5273#	5276#	5314#	5317#	5378#	5381#	5420#	5423#	5483#
	5486#	5523#	5527#	5586#	5589#	5642#	5645#	5701#	5704#	5757#	5760#	5791#	5794#
	5856#	5859#	5905#	5908#	5994#	6044#	6090#	6093#	6127#	6130#	6203#	6206#	6231#
	6234#	6290#	6293#	6356#	6359#	6399#	6402#	6453#	6456#	6506#	6509#	6558#	6561#
	6613#	6616#	6698#	6701#	6794#	6797#	6901#	6904#	6957#	6960#	7013#	7015#	7072#
	7075#	7162#	7170#	7216#	7227#	7240#	7270#	7273	7281	7290#	7350#		
	1127#	1245	1559#	4689	4733#	5003	5054#	7170	7216#	7350			
TSNSO = 000000	1185#	1196	1205#	1209	1217#	1236	1918#	1920	1922#	1924	1926#	1928	2205#
TSNS1 = 000005	2213	4740#	4747	4768#	4831	4907#	4916	4935#	4948	4969#	4971	4983#	4990
	5124#	5155	5158#	5188	5191#	5215	5218#	5243	5246#	5273	5276#	5314	5317#
	5378	5381#	5420	5423#	5483	5486#	5523	5527#	5586	5589#	5642	5645#	5701
	5704#	5757	5760#	5791	5794#	5856	5859#	5905	5908#	5994	6044#	6090	6093#
	6127	6130#	6203	6206#	6231	6234#	6290	6293#	6356	6359#	6399	6402#	6453
	6456#	6506	6509#	6558	6561#	6613	6616#	6698	6701#	6794	6797#	6901	6904#
	6957	6960#	7013	7016#	7072	7075#	7162	7227#	7240	7270#	7273	7281	7290
TSPCNT= 000000	7352#	7353#	7360#										
TSPTAB= 010066	7353#	7360#											
TSPTHV= 000002	1146	7367#											
TSPTNU= 000002	1100#	7353#	7360#	7367									
TS SAVL= 177777	1100#												
TSSEGL= 177777	1100#												
TSSIZE= 000016	7349	7367#											
TSSUBN= 000000	1100#	5124#	5158#	5191#	5218#	5246#	5276#	5317#	5381#	5423#	5486#	5527#	5589#
	5645#	5704#	5760#	5794#	5859#	5908#	6044#	6093#	6130#	6206#	6234#	6293#	6359#
	6402#	6456#	6509#	6561#	6616#	6701#	6797#	6904#	6960#	7016#	7075#		
TSTAGL= 177777	1100#												
TSTAGN= 010070	1100#	1185#	1205#	1217#	1918#	1922#	1926#	2205#	4740#	4768#	4907#	4935#	4969#
	4983#	5124#	5158#	5191#	5218#	5246#	5276#	5317#	5381#	5423#	5486#	5527#	5589#
	5645#	5704#	5760#	5794#	5859#	5908#	6044#	6093#	6130#	6206#	6234#	6293#	6359#
	6402#	6456#	6509#	6561#	6616#	6701#	6797#	6904#	6960#	7016#	7075#	7227#	7270#
	7352#	7353#	7360#										
TSTEMP= 000000	1170#	1196#	1209#	1236#	1245#	1920#	1924#	1928#	2213#	4689#	4747#	4831#	4916#
	4948#	4971#	4990#	5003#	5152#	5155#	5185#	5188#	5212#	5215#	5240#	5243#	5270#
	5273#	5311#	5314#	5375#	5378#	5417#	5420#	5480#	5483#	5520#	5523#	5582#	5586#
	5639#	5642#	5697#	5701#	5753#	5757#	5786#	5791#	5832#	5840#	5846#	5856#	5900#
	5905#	5990#	5994#	6085#	6090#	6123#	6127#	6197#	6203#	6227#	6231#	6275#	6277#
	6286#	6290#	6334#	6336#	6352#	6356#	6392#	6395#	6399#	6440#	6447#	6449#	6453#

TOFT16	007006	2511	2534#
TOFT17	007012	2512	2535#
TOFT2	006742	2499	2526#
TOFT20	007016	2513	2536#
TOFT22	007022	2515	2537#
TOFT23	007025	2516	2538#
TOFT24	007030	2517	2539#
TOFT25	007035	2518	2540#
TOFT4	006746	2501	2527#
TOFT40	006644	2480	2481#
TOFT41	006646	2483	2484#
TOFT42	006651	2486	2487#
TOFT43	006654	2489	2490#
TOFT5	006751	2502	2528#
TOFT7	006755	2504	2529#
TOMSG	022172	5088	5092#
TORT1	015026	3984#	3991
TORT2	015036	3985#	3992
TORT3	015046	3986#	3993
TORT4	015056	3987#	3994
TORT5	015066	3988#	3995
TORT6	015076	3989#	3996
T1	022332 G	1170	5124#
T1MSG	022334	5124#	5154
T1RTB	022450	5154#	
T1RT1 =	***** U	5154	
T1RT2 =	***** U	5154	
T1RT3 =	***** U	5154	
T1RT4 =	***** U	5154	
T1TBL	022444	5144	5154#
T10	024222 G	1170	5486#
T10MSG	024224	5486#	5522
T10RTB	024356	5522#	
T10RT1=	***** U	5522	
T10RT2=	***** U	5522	
T10RT3=	***** U	5522	
T10RT4=	***** U	5522	
T1CTBL	024352	5509	5522#
T11	024364 G	1170	5527#
T11FTB	024614	5585#	
T11FT1=	***** U	5585	
T11FT2=	***** U	5585	
T11FT3=	***** U	5585	
T11FT4=	***** U	5585	
T11FT5=	***** U	5585	
T11FT6=	***** U	5585	
T11MSG	024366	5527#	5584
T11RTB	024612	5584#	
T11RT1=	***** U	5584	
T11RT2=	***** U	5584	
T11RT3=	***** U	5584	
T11RT4=	***** U	5584	
T11TBL	024602	5560	5584#
T12	024622 G	1170	5589#
T12FTB	025024	5641#	
T12FT1=	***** U	5641	

T12FT2=	***** U	5641	
T12FT3=	***** U	5641	
T12FT4=	***** U	5641	
T12FT5=	***** U	5641	
T12FT6=	***** U	5641	
T12MSG	024624	5589#	5640
T12RTB	025022	5640#	
T12RT1=	***** U	5640	
T12RT2=	***** U	5640	
T12RT3=	***** U	5640	
T12RT4=	***** U	5640	
T12TBL	025012	5622	5640#
T13	025032 G	1170	5645#
T13FTB	025254	5700#	
T13FT1=	***** U	5700	
T13FT2=	***** U	5700	
T13FT3=	***** U	5700	
T13FT4=	***** U	5700	
T13FT5=	***** U	5700	
T13FT6=	***** U	5700	
T13MSG	025034	5645#	5699
T13RTB	025252	5699#	
T13RT1=	***** U	5699	
T13RT2=	***** U	5699	
T13RT3=	***** U	5699	
T13RT4=	***** U	5699	
T13TBL	025242	5678	5699#
T14	025262 G	1170	5704#
T14FTB	025474	5756#	
T14FT1=	***** U	5756	
T14FT2=	***** U	5756	
T14FT3=	***** U	5756	
T14FT4=	***** U	5756	
T14FT5=	***** U	5756	
T14FT6=	***** U	5756	
T14MSG	025264	5704#	5755
T14RTB	025472	5755#	
T14RT1=	***** U	5755	
T14RT2=	***** U	5755	
T14RT3=	***** U	5755	
T14RT4=	***** U	5755	
T14TBL	025462	5732	5755#
T15	025502 G	1170	5760#
T15FTB	025624	5790#	
T15FT1=	***** U	5790	
T15FT2=	***** U	5790	
T15FT3=	***** U	5790	
T15FT4=	***** U	5790	
T15FT5=	***** U	5790	
T15FT6=	***** U	5790	
T15MSG	025504	5760#	5789
T15RTB	025620	5789#	
T15RT1=	***** U	5789	
T15RT2=	***** U	5789	
T15RT3=	***** U	5789	
T15RT4=	***** U	5789	

T15TBL	025610		5782	5789#
T16	025632	G	1170	5794#
T16FTB	026116		5849#	
T16FT1=	*****	U	5849	
T16FT2=	*****	U	5849	
T16FT3=	*****	U	5849	
T16FT4=	*****	U	5849	
T16FT5=	*****	U	5849	
T16FT6=	*****	U	5849	
T16MSG	025634		5794#	5848
T16RTB	026112		5848#	
T16RT1=	*****	U	5848	
T16RT2=	*****	U	5848	
T16RT3=	*****	U	5848	
T16RT4=	*****	U	5848	
T16TBL	026102		5827	5848#
T17	026234	G	1170	5859#
T17FTB	026422		5904#	
T17FT1=	*****	U	5904	
T17FT2=	*****	U	5904	
T17FT3=	*****	U	5904	
T17FT4=	*****	U	5904	
T17FT5=	*****	U	5904	
T17FT6=	*****	U	5904	
T17MSG	026236		5859#	5903
T17RTB	026414		5903#	
T17RT1=	*****	U	5903	
T17RT2=	*****	U	5903	
T17RT3=	*****	U	5903	
T17RT4=	*****	U	5903	
T17TBL	026404		5888	5903#
T18	026430	G	1170	5908#
T18FTB	026766		5993#	
T18FT1=	*****	U	5993	
T18FT2=	*****	U	5993	
T18FT3=	*****	U	5993	
T18FT4=	*****	U	5993	
T18FT5=	*****	U	5993	
T18FT6=	*****	U	5993	
T18MSG	026432		5908#	5992
T18RTB	026762		5992#	
T18RT1=	*****	U	5992	
T18RT2=	*****	U	5992	
T18RT3=	*****	U	5992	
T18RT4=	*****	U	5992	
T18TBL	026752		5949	5992#
T19	027104	G	1170	6044#
T19FTB	027332		6089#	
T19FT1=	*****	U	6089	
T19FT2=	*****	U	6089	
T19FT3=	*****	U	6089	
T19FT4=	*****	U	6089	
T19FT5=	*****	U	6089	
T19FT6=	*****	U	6089	
T19MSG	027106		6044#	6088
T19RTB	027326		6088#	

T19RT1	027306		6087#	6088
T19RT2=	*****	U	6088	
T19RT3=	*****	U	6088	
T19RT4=	*****	U	6088	
T19TBL	027316		6072	6088#
T2	022456	G	1170	5158#
T2MSG	022460		5158#	5187
T2RTB	022610		5187#	
T2RT1 =	*****	U	5187	
T2RT2 =	*****	U	5187	
T2RT3 =	*****	U	5187	
T2RT4 =	*****	U	5187	
T2TBL	022604		5177	5187#
T20	027340	G	1170	6093#
T20FTB	027474		6126#	
T20FT1=	*****	U	6126	
T20FT2=	*****	U	6126	
T20FT3=	*****	U	6126	
T20FT4=	*****	U	6126	
T20FT5=	*****	U	6126	
T20FT6=	*****	U	6126	
T20MSG	027342		6093#	6125
T20RTB	027472		6125#	
T20RT1=	*****	U	6125	
T20RT2=	*****	U	6125	
T20RT3=	*****	U	6125	
T20RT4=	*****	U	6125	
T20TBL	027462		6115	6125#
T21	027502	G	1170	6130#
T21FTB	030120		6202#	
T21FT1=	*****	U	6202	
T21FT2=	*****	U	6202	
T21FT3=	*****	U	6202	
T21FT4=	*****	U	6202	
T21FT5=	*****	U	6202	
T21FT6=	*****	U	6202	
T21MSG	027504		6130#	6201
T21RTB	030116		6201#	
T21RT1=	*****	U	6201	
T21RT2=	*****	U	6201	
T21RT3=	*****	U	6201	
T21RT4=	*****	U	6201	
T21TBL	030106		6161	6201#
T22	030126	G	1170	6206#
T22FTB	030240		6230#	
T22FT1=	*****	U	6230	
T22FT2=	*****	U	6230	
T22FT3=	*****	U	6230	
T22FT4=	*****	U	6230	
T22FT5=	*****	U	6230	
T22FT6=	*****	U	6230	
T22MSG	030130		6206#	6229
T22RTB	030234		6229#	
T22RT1=	*****	U	6229	
T22RT2=	*****	U	6229	
T22RT3=	*****	U	6229	

T22RT4=	***** U	6229	
T22TBL	030224	6224	6229#
T23	030246 G	1170	6234#
T23FTB	030442	6289#	
T23FT1=	***** U	6289	
T23FT2=	***** U	6289	
T23FT3=	***** U	6289	
T23FT4=	***** U	6289	
T23FT5=	***** U	6289	
T23FT6=	***** U	6289	
T23MSG	030250	6234#	6288
T23RTB	030440	6288#	
T23RT1=	***** U	6288	
T23RT2=	***** U	6288	
T23RT3=	***** U	6288	
T23RT4=	***** U	6288	
T23TBL	030430	6271	6288#
T24	030450 G	1170	6293#
T24FTB	030702	6355#	
T24FT1=	***** U	6355	
T24FT2=	***** U	6355	
T24FT3=	***** U	6355	
T24FT4=	***** U	6355	
T24FT5=	***** U	6355	
T24FT6=	***** U	6355	
T24MSG	030452	6293#	6354
T24RTB	030700	6354#	
T24RT1=	***** U	6354	
T24RT2=	***** U	6354	
T24RT3=	***** U	6354	
T24RT4=	***** U	6354	
T24TBL	030670	6330	6354#
T25	030710 G	1170	6359#
T25FTB	031072	6398#	
T25FT1=	***** U	6398	
T25FT2=	***** U	6398	
T25FT3=	***** U	6398	
T25FT4=	***** U	6398	
T25FT5=	***** U	6398	
T25FT6=	***** U	6398	
T25MSG	030712	6359#	6397
T25RTB	031064	6397#	
T25RT1=	***** U	6397	
T25RT2=	***** U	6397	
T25RT3=	***** U	6397	
T25RT4=	***** U	6397	
T25TBL	031054	6386	6397#
T26	031100 G	1170	6402#
T26FTB	031272	6452#	
T26FT1=	***** U	6452	
T26FT2=	***** U	6452	
T26FT3=	***** U	6452	
T26FT4=	***** U	6452	
T26FT5=	***** U	6452	
T26FT6=	***** U	6452	
T26MSG	031102	6402#	6451

T26RTB	031266		6451#	
T26RT1=	*****	U	6451	
T26RT2=	*****	U	6451	
T26RT3=	*****	U	6451	
T26RT4=	*****	U	6451	
T26TBL	031256		6435	6451#
T27	031300	G	1170	6456#
T27FTB	031462		6505#	
T27FT1=	*****	U	6505	
T27FT2=	*****	U	6505	
T27FT3=	*****	U	6505	
T27FT4=	*****	U	6505	
T27FT5=	*****	U	6505	
T27FT6=	*****	U	6505	
T27MSG	031302		6456#	6504
T27RTB	031456		6504#	
T27RT1=	*****	U	6504	
T27RT2=	*****	U	6504	
T27RT3=	*****	U	6504	
T27RT4=	*****	U	6504	
T27TBL	031446		6489	6504#
T28	031470	G	1170	6509#
T28FTB	031656		6557#	
T28FT1=	*****	U	6557	
T28FT2=	*****	U	6557	
T28FT3=	*****	U	6557	
T28FT4=	*****	U	6557	
T28FT5=	*****	U	6557	
T28FT6=	*****	U	6557	
T28MSG	031472		6509#	6556
T28RTB	031652		6556#	
T28RT1=	*****	U	6556	
T28RT2=	*****	U	6556	
T28RT3=	*****	U	6556	
T28RT4=	*****	U	6556	
T28TBL	031642		6541	6556#
T29	031664	G	1170	6561#
T29FTB	032072		6612#	
T29FT1=	*****	U	6612	
T29FT2=	*****	U	6612	
T29FT3=	*****	U	6612	
T29FT4=	*****	U	6612	
T29FT5=	*****	U	6612	
T29FT6=	*****	U	6612	
T29MSG	031666		6561#	6611
T29RTB	032066		6611#	
T29RT1=	*****	U	6611	
T29RT2=	*****	U	6611	
T29RT3=	*****	U	6611	
T29RT4=	*****	U	6611	
T29TBL	032056		6593	6611#
T3	022616	G	1170	5191#
T3MSG	022620		5191#	5214
T3RTB	022724		5214#	
T3RT1 =	*****	U	5214	
T3RT2 =	*****	U	5214	

T3RT3 =	*****	U	5214	
T3RT4 =	*****	U	5214	
T3TBL	022720		5207	5214#
T30	032100	G	1170	6616#
T30FTB	032372		6697#	
T30FT1=	*****	U	6697	
T30FT2=	*****	U	6697	
T30FT3=	*****	U	6697	
T30FT4=	*****	U	6697	
T30FT5=	*****	U	6697	
T30FT6=	*****	U	6697	
T30MSG	032102		6616#	6696
T30RTB	032366		6696#	
T30RT1=	*****	U	6696	
T30RT2=	*****	U	6696	
T30RT3=	*****	U	6696	
T30RT4=	*****	U	6696	
T30TBL	032356		6665	6696#
T31	032400	G	1170	6701#
T31FTB	032744		6793#	
T31FT1=	*****	U	6793	
T31FT2=	*****	U	6793	
T31FT3=	*****	U	6793	
T31FT4=	*****	U	6793	
T31FT5=	*****	U	6793	
T31FT6=	*****	U	6793	
T31MSG	032402		6701#	6792
T31RTB	032740		6792#	
T31RT1=	*****	U	6792	
T31RT2=	*****	U	6792	
T31RT3=	*****	U	6792	
T31RT4=	*****	U	6792	
T31TBL	032730		6750	6792#
T32	032752	G	1170	6797#
T32FTB	033350		6900#	
T32FT1=	*****	U	6900	
T32FT2=	*****	U	6900	
T32FT3=	*****	U	6900	
T32FT4=	*****	U	6900	
T32FT5=	*****	U	6900	
T32FT6=	*****	U	6900	
T32MSG	032754		6797#	6899
T32RTB	033346		6899#	
T32RT1=	*****	U	6899	
T32RT2=	*****	U	6899	
T32RT3=	*****	U	6899	
T32RT4=	*****	U	6899	
T32TBL	033336		6849	6899#
T33	033356	G	1170	6904#
T33FTB	033550		6955#	
T33FT1=	*****	U	6955	
T33FT2=	*****	U	6955	
T33FT3=	*****	U	6955	
T33FT4=	*****	U	6955	
T33FT5=	*****	U	6955	
T33FT6=	*****	U	6955	

T33MSG	033360	6904#	6954
T33RTB	033544	6954#	
T33RT1=	***** U	6954	
T33RT2=	***** U	6954	
T33RT3=	***** U	6954	
T33RT4=	***** U	6954	
T33TBL	033534	6938	6954#
T34	033556 G	1170	6960#
T34FTB	033744	7011#	
T34FT1=	***** U	7011	
T34FT2=	***** U	7011	
T34FT3=	***** U	7011	
T34FT4=	***** U	7011	
T34FT5=	***** U	7011	
T34FT6=	***** U	7011	
T34MSG	033560	6960#	7010
T34RTB	033740	7010#	
T34RT1=	***** U	7010	
T34RT2=	***** U	7010	
T34RT3=	***** U	7010	
T34RT4=	***** U	7010	
T34TBL	033730	6994	7010#
T35	033752 G	1170	7016#
T35FTB	034164	7070#	
T35FT1=	***** U	7070	
T35FT2=	***** U	7070	
T35FT3=	***** U	7070	
T35FT4=	***** U	7070	
T35FT5=	***** U	7070	
T35FT6=	***** U	7070	
T35MSG	033754	7016#	7069
T35RTB	034160	7069#	
T35RT1=	***** U	7069	
T35RT2=	***** U	7069	
T35RT3=	***** U	7069	
T35RT4=	***** U	7069	
T35TBL	034150	7050	7069#
T36	034172 G	1170	7075#
T36FTB	034520	7160#	
T36FT1=	***** U	7160	
T36FT2=	***** U	7160	
T36FT3=	***** U	7160	
T36FT4=	***** U	7160	
T36FT5=	***** U	7160	
T36FT6=	***** U	7160	
T36MSG	034174	7075#	7159
T36RTB	034514	7159#	
T36RT1=	***** U	7159	
T36RT2=	***** U	7159	
T36RT3=	***** U	7159	
T36RT4=	***** U	7159	
T36TBL	034504	7123	7159#
T4	022732 G	1170	5218#
T4MSG	022734	5218#	5242
T4RTB	023042	5242#	
T4RT1 =	***** U	5242	

T4RT2 = ***** U	5242		
T4RT3 = ***** U	5242		
T4RT4 = ***** U	5242		
T4TBL 023036	5235	5242#	
T5 023050 G	1170	5246#	
T5MSG 023052	5246#	5272	
T5RTB 023160	5272#		
T5RT1 = ***** U	5272		
T5RT2 = ***** U	5272		
T5RT3 = ***** U	5272		
T5RT4 = ***** U	5272		
T5TBL 023154	5264	5272#	
T6 023166 G	1170	5276#	
T6MSG 023170	5276#	5313	
T6RTB 023344	5313#		
T6RT1 = ***** U	5313		
T6RT2 = ***** U	5313		
T6RT3 = ***** U	5313		
T6RT4 = ***** U	5313		
T6TBL 023340	5299	5313#	
T7 023352 G	1170	5317#	
T7MSG 023354	5317#	5377	
T7RTB 023606	5377#		
T7RT1 = ***** U	5377		
T7RT2 = ***** U	5377		
T7RT3 = ***** U	5377		
T7RT4 = ***** U	5377		
T7TBL 023602	5351	5377#	
T8 023614 G	1170	5381#	
T8MSG 023616	5381#	5419	
T8RTB 023776	5419#		
T8RT1 = ***** U	5419		
T8RT2 = ***** U	5419		
T8RT3 = ***** U	5419		
T8RT4 = ***** U	5419		
T8TBL 023772	5404	5419#	
T9 024004 G	1170	5423#	
T9MSG 024006	5423#	5482	
T9RTB 024214	5482#		
T9RT1 = ***** U	5482		
T9RT2 = ***** U	5482		
T9RT3 = ***** U	5482		
T9RT4 = ***** U	5482		
T9TBL 024210	5458	5482#	
UACDB 013660	3721#		
UADCK 013500	3695#		
UAM = 000200 G	1566#		
UAUWCH 011750	3348#		
UBRCR 013710	3766#	3768	
UBT10 024334	5517#		
UBT21 027670	6175	6178#	
UBT23 030410	6281	6283#	
UBT24 030654	6345	6348	6350#
UBT27 031432	6500#		
UBT31 032566	6759	6767	6769#
UBT32 033324	6894	6896#	

XPTFRU	005606	2417#					
XPTSTA	015436	4066	4081#				
XRDERC	011454	3174	3182	3186	3189#		
XRSTA	011336	3147	3154#				
XREAD	011164	3080	3087	3091	3094#		
XREGCK	014256	3825#					
XSCP	007756	2648	2659#				
XSDC	020540	4535	4538#				
XSETDN	011254	3115	3122	3125#			
XSRC	014520	3875#					
XTKECK	017722	4372#					
XTKPRT	010154	2707	2713#				
XT1	022440	5144	5152#				
XT10	024346	5509	5520#				
XT11	024576	5560	5578	5582#			
XT12	025006	5622	5635	5639#			
XT13	025236	5678	5692	5697#			
XT14	025456	5732	5735	5740	5745	5750	5753#
XT15	025604	5782	5786#				
XT16	026076	5827	5830	5834	5842	5846#	
XT17	026400	5888	5900#				
XT18	026746	5949	5978	5984	5987	5990#	
XT19	027302	6072	6074	6085#			
XT2	022600	5177	5185#				
XT20	027456	6115	6123#				
XT21	027774	6161	6185	6187	6194	6196#	
XT22	030220	6224	6227#				
XT23	030424	6271	6286#				
XT24	030664	6330	6352#				
XT25	031050	6386	6395#				
XT26	031252	6435	6449#				
XT27	031442	6489	6502#				
XT28	031636	6541	6554#				
XT29	032052	6593	6606	6609#			
XT3	022714	5207	5212#				
XT30	032352	6665	6691	6694#			
XT31	032724	6750	6790#				
XT32	033332	6849	6897#				
XT33	033530	6938	6952#				
XT34	033724	6994	7008#				
XT35	034144	7050	7064	7067#			
XT36	034500	7123	7153	7156#			
XT4	023032	5235	5240#				
XT5	023150	5264	5270#				
XT6	023334	5299	5311#				
XT7	023573	5351	5375#				
XT8	023766	5404	5417#				
XT9	024204	5458	5480#				
XUWCH	012014	3340	3343	3353	3356#		
XWAIT	011632	3287	3289#				
XWRITE	011054	3044	3052	3056	3059#		
XXPG	012540	3544#	3548	3551			
XSALWA=	000000	1100#					
XSALS=	000040	1100#	7273	7281			
XSOFFS=	000400	1100#	7273	7281			
X\$TRUE=	000020	1100#					

BCOMPL	4778	4780																
BGNAU	4983																	
BGNAUT	4969																	
BGNCLN	4907																	
BGNDU	4935																	
BGNHRD	7227																	
BGNHW	1185																	
BGINI	4768																	
BGNMOD	1127	1559	4733	5054	7216													
BGNMSG	1918	1922	1926	2205														
BGNPRO	1205																	
BGNPTA	7353	7360																
BGNRPT	4740																	
BGNSET	7352																	
BGNSFT	7270																	
BGNSW	1217																	
BGNTST	5124	5158	5191	5218	5246	5276	5317	5381	5423	5486	5527	5589	5645	5704	5760			
	5794	5859	5908	6044	6093	6130	6206	6234	6293	6359	6402	6456	6509	6561	6616			
	6701	6797	6904	6960	7016	7075												
BNCOMP	4783	4785	4795	4799	4805	5830												
BREAK	3344	3383	3417															
BRESET	3313	4909	5145	5561	5623	5679	5733											
CLRVEC	4908	5083	6083															
DESCRI	1147																	
DEVTYP	1157																	
DISPAT	1170																	
DOCLN	3265	4215	4296	4812														
DODU	3264	4214	4295	5089														
ENDAU	4990																	
ENDAUT	4971																	
ENDCLN	4916																	
ENDDU	4948																	
ENDHRD	7240																	
ENDHW	1196																	
ENDINI	4831																	
ENDMOD	1245	4689	5003	7170	7350													
ENDMSG	1920	1924	1928	2213														
ENDPRO	1209																	
ENDPTA	7359	7366																
ENDRPT	4747																	
ENDSET	7367																	
ENDSFT	7290																	
ENDSW	1236																	
ENDTST	5155	5188	5215	5243	5273	5314	5378	5420	5483	5523	5586	5642	5701	5757	5791			
	5856	5905	5994	6090	6127	6203	6231	6290	6356	6399	6453	6506	6558	6613	6698			
	6794	6901	6957	7013	7072	7162												
EQUALS	1566																	
ERROR	2129																	
ERRSF	5088																	
ERRTBL	1828																	
ESCAPE	6275	6277	6334	6336	6392	6440	6447	6497	6545	6552	6598	6601	6670	6675	6683			
	6943	6950	6999	7006	7056	7059												
EXIT	5152	5185	5212	5240	5270	5311	5375	5417	5480	5520	5582	5639	5697	5753	5786			
	5846	5900	5990	6085	6123	6197	6227	6286	6352	6395	6449	6502	6554	6609	6694			
	6790	6897	6952	7008	7067	7156												
FRUCO	1530#	2478	2481	2484	2487	2490	2525	2526	2527	2528	2529	2530	2531	2532	2533			

PARAMETER CODING	MACY11 30(1046)		12-APR-82 13:23 PAGE 148-1										SEQ 0212		
CZRFB.P11	09-APR-82 15:14		CROSS REFERENCE TABLE -- MACRO NAMES												
FRUTB	2534 1524#	2535 2478	2536 2481	2537 2484	2538 2487	2539 2490	2540 2525	2526	2527	2528	2529	2530	2531	2532	2533
FRUTBL	2534 1488#	2535 5585	2536 5641	2537 5700	2538 5756	2539 5790	2540 5849	5904	5993	6089	6126	6202	6230	6289	6355
FUTABL	6398 1494#	6452 5585	6505 5641	6557 5700	6612 5756	6697 5790	6793 5849	6900 5904	6955 5993	7011 6089	7070 6126	7160 6202	6230	6289	6355
GMANIL	5832	5840	6505	6557	6612	6697	6793	6900	6955	7011	7070	7160			
GPHARD	4804														
GPRMA	7229	7230	7278												
GPRMD	7231	7232	7233	7277	7279										
GPRML	5832#	5840#	7272	7274	7275	7276	7280	7282	7283						
HEADER	1146														
LASTAD	7349														
MANUAL	5829														
MSBYTE	1146#														
MSCHEC	5152#	5185#	5212#	5240#	5270#	5311#	5375#	5417#	5480#	5520#	5582#	5639#	5697#	5753#	5786#
	5846#	5900#	5990#	6085#	6123#	6197#	6227#	6286#	6352#	6395#	6449#	6502#	6554#	6609#	6694#
	6790#	6897#	6952#	7008#	7067#	7156#									
MSCNTO	5832#	5840#	7229#	7230#	7231#	7232#	7233#	7272#	7274#	7275#	7276#	7277#	7278#	7279#	7280#
	7282#	7283#													
MSCOUN	1930#	1933#	1936#	1939#	1942#	1945#	1948#	1951#	1954#	1957#	4810#	5831#	5839#		
MSDATA	1146#	1147#	1157#												
MSDECR	1196#	1209#	1236#	1245#	1920#	1924#	1928#	2213#	4689#	4747#	4831#	4916#	4948#	4971#	4990#
	5003#	5155#	5188#	5215#	5243#	5273#	5314#	5378#	5420#	5483#	5523#	5586#	5642#	5701#	5757#
	5791#	5856#	5905#	5994#	6090#	6127#	6203#	6231#	6290#	6356#	6399#	6453#	6506#	6558#	6613#
	6698#	6794#	6901#	6957#	7013#	7072#	7162#	7170#	7240#	7290#	7350#	7353#	7360#	7360#	7360#
MSDEFA	5832#	5840#	7229#	7230#	7231#	7232#	7233#	7272#	7274#	7275#	7276#	7277#	7278#	7279#	7280#
	7282#	7283#													
MSENDE	1196#	1236#	1245#	1920#	1924#	1928#	2213#	4689#	4747#	4831#	4916#	4948#	4971#	4990#	5003#
	5155#	5188#	5215#	5243#	5273#	5314#	5378#	5420#	5483#	5523#	5586#	5642#	5701#	5757#	5791#
	5856#	5905#	5994#	6090#	6127#	6203#	6231#	6290#	6356#	6399#	6453#	6506#	6558#	6613#	6698#
	6794#	6901#	6957#	7013#	7072#	7162#	7170#	7240#	7290#	7350#					
MSERRI	5088#														
MSESCA	6275#	6277#	6334#	6336#	6392#	6440#	6447#	6497#	6545#	6552#	6598#	6601#	6670#	6675#	6683#
	6943#	6950#	6999#	7006#	7056#	7059#									
MSESCS	6275#	6277#	6334#	6336#	6392#	6440#	6447#	6497#	6545#	6552#	6598#	6601#	6670#	6675#	6683#
	6943#	6950#	6999#	7006#	7056#	7059#									
MSEXCP	7229#	7230#	7231#	7232#	7233#	7277#	7278#	7279#							
MSEXIT	5152#	5185#	5212#	5240#	5270#	5311#	5375#	5417#	5480#	5520#	5582#	5639#	5697#	5753#	5786#
	5846#	5900#	5990#	6085#	6123#	6197#	6227#	6286#	6352#	6395#	6449#	6502#	6554#	6609#	6694#
	6790#	6897#	6952#	7008#	7067#	7156#									
MSEXSE	5152#	5185#	5212#	5240#	5270#	5311#	5375#	5417#	5480#	5520#	5582#	5639#	5697#	5753#	5786#
	5846#	5900#	5990#	6085#	6123#	6197#	6227#	6286#	6352#	6395#	6449#	6502#	6554#	6609#	6694#
	6790#	6897#	6952#	7008#	7067#	7156#									
MSEXTJ	5152#	5185#	5212#	5240#	5270#	5311#	5375#	5417#	5480#	5520#	5582#	5639#	5697#	5753#	5786#
	5846#	5900#	5990#	6085#	6123#	6197#	6227#	6286#	6352#	6395#	6449#	6502#	6554#	6609#	6694#
	6790#	6897#	6952#	7008#	7067#	7156#									
MSGEN	1146#	1147#	1157#	1170#	1185#	1196#	1205#	1217#	1236#	1828#	1918#	1920#	1922#	1924#	1926#
	1928#	2205#	2213#	4740#	4747#	4768#	4831#	4907#	4916#	4935#	4948#	4969#	4971#	4983#	4990#
	5124#	5155#	5158#	5188#	5191#	5215#	5218#	5243#	5246#	5273#	5276#	5314#	5317#	5378#	5381#
	5420#	5423#	5483#	5486#	5523#	5527#	5586#	5589#	5642#	5645#	5701#	5704#	5757#	5760#	5791#
	5794#	5832#	5840#	5856#	5859#	5905#	5908#	5994#	6044#	6090#	6093#	6127#	6130#	6203#	6206#
	6231#	6234#	6290#	6293#	6356#	6359#	6399#	6402#	6453#	6456#	6506#	6509#	6558#	6561#	6613#
	6616#	6698#	6701#	6794#	6797#	6901#	6904#	6957#	6960#	7013#	7016#	7072#	7075#	7162#	7227#
	7240#	7270#	7290#	7349#	7353#	7359#	7360#	7366#							

MSGENB	5832#	5840#													
MSGETS	1196#	1209#	1236#	1245#	1920#	1924#	1928#	2213#	4689#	4747#	4831#	4916#	4948#	4971#	4990#
	5003#	5155#	5188#	5215#	5243#	5273#	5314#	5378#	5420#	5483#	5523#	5586#	5642#	5701#	5757#
	5791#	5856#	5905#	5994#	6090#	6127#	6203#	6231#	6290#	6356#	6399#	6453#	6506#	6558#	6613#
MSGETT	6698#	6754#	6901#	6957#	7013#	7072#	7162#	7170#	7240#	7273#	7281#	7290#	7350#		
	5152#	5185#	5212#	5240#	5270#	5311#	5375#	5417#	5480#	5520#	5582#	5639#	5697#	5753#	5786#
	5846#	5900#	5990#	6085#	6123#	6197#	6227#	6275#	6277#	6286#	6334#	6336#	6352#	6392#	6395#
	6440#	6447#	6449#	6497#	6502#	6545#	6552#	6554#	6598#	6601#	6609#	6670#	6675#	6683#	6694#
MSGNGB	6790#	6897#	6943#	6950#	6952#	6999#	7006#	7008#	7056#	7059#	7067#	7156#	7273#	7281#	
	1127#	1146#	1147#	1157#	1170#	1185#	1205#	1217#	1559#	1828#	1918#	1922#	1926#	2205#	4733#
MSGNIN	4740#	4768#	4907#	4935#	4969#	4983#	5054#	7216#	7227#	7270#	7349#				
	1146#	1147#	1157#	1170#	1185#	1217#	1920#	1924#	1928#	1930#	1933#	1936#	1939#	1942#	1945#
	1948#	1951#	1954#	1957#	2129#	2213#	3264#	3265#	3313#	3334#	3344#	3356#	3383#	3417#	4214#
	4215#	4295#	4296#	4747#	4775#	4777#	4778#	4779#	4780#	4782#	4783#	4784#	4785#	4794#	4795#
	4798#	4799#	4804#	4805#	4808#	4810#	4812#	4831#	4908#	4909#	4916#	4948#	4971#	4990#	5081#
	5083#	5088#	5089#	5145#	5152#	5155#	5185#	5188#	5212#	5215#	5240#	5243#	5270#	5273#	5311#
	5314#	5375#	5378#	5417#	5420#	5480#	5483#	5520#	5523#	5561#	5582#	5586#	5623#	5639#	5642#
	5679#	5697#	5701#	5733#	5753#	5757#	5786#	5791#	5829#	5830#	5831#	5832#	5839#	5840#	5846#
	5856#	5900#	5905#	5990#	5994#	6076#	6083#	6085#	6090#	6117#	6123#	6127#	6164#	6197#	6203#
	6227#	6231#	6275#	6277#	6286#	6290#	6334#	6336#	6352#	6356#	6392#	6395#	6399#	6440#	6447#
	6449#	6453#	6497#	6502#	6506#	6545#	6552#	6554#	6558#	6598#	6601#	6609#	6613#	6670#	6675#
	6683#	6694#	6698#	6790#	6794#	6897#	6901#	6943#	6950#	6952#	6957#	6999#	7006#	7008#	7013#
	7056#	7059#	7067#	7072#	7156#	7162#	7227#	7229#	7230#	7231#	7232#	7233#	7240#	7270#	7272#
	7273#	7274#	7275#	7276#	7277#	7278#	7279#	7280#	7281#	7282#	7283#	7290#	7349#	7353#	7360#
MSGNLS	5832#	5840#													
MSGNTA	1196#	1236#	1920#	1924#	1928#	2213#	4747#	4831#	4916#	4948#	4971#	4990#	5155#	5188#	5215#
	5243#	5273#	5314#	5378#	5420#	5483#	5523#	5586#	5642#	5701#	5757#	5791#	5856#	5905#	5994#
	6090#	6127#	6203#	6231#	6290#	6356#	6399#	6453#	6506#	6558#	6613#	6698#	6794#	6901#	6957#
	7013#	7072#	7162#	7240#	7290#	7353#	7359#	7360#	7366#						
MSGNTE	5124#	5158#	5191#	5218#	5246#	5276#	5317#	5381#	5423#	5486#	5527#	5589#	5645#	5704#	5760#
	5794#	5859#	5908#	6044#	6093#	6130#	6206#	6234#	6293#	6359#	6402#	6456#	6509#	6561#	6616#
	6701#	6797#	6904#	6960#	7016#	7075#									
MSHAPT	1146#														
MSHNP	1146#														
MSINCR	1127#	1185#	1205#	1217#	1559#	1918#	1920#	1922#	1924#	1926#	1928#	1930#	1933#	1936#	1939#
	1942#	1945#	1948#	1951#	1954#	1957#	2129#	2205#	2213#	3264#	3265#	3313#	3334#	3344#	3356#
	3383#	3417#	4214#	4215#	4295#	4296#	4733#	4740#	4747#	4768#	4775#	4777#	4779#	4782#	4784#
	4794#	4798#	4804#	4808#	4810#	4812#	4831#	4907#	4908#	4909#	4916#	4935#	4948#	4969#	4971#
	4983#	4990#	5054#	5081#	5083#	5088#	5089#	5124#	5145#	5152#	5155#	5158#	5185#	5188#	5191#
	5212#	5215#	5218#	5240#	5243#	5246#	5270#	5273#	5276#	5311#	5314#	5317#	5375#	5378#	5381#
	5417#	5420#	5423#	5480#	5483#	5486#	5520#	5523#	5527#	5561#	5582#	5586#	5589#	5623#	5639#
	5642#	5645#	5679#	5697#	5701#	5704#	5733#	5753#	5757#	5760#	5786#	5791#	5794#	5829#	5831#
	5832#	5839#	5840#	5846#	5856#	5859#	5900#	5905#	5908#	5990#	5994#	6044#	6076#	6083#	6085#
	6090#	6093#	6117#	6123#	6127#	6130#	6164#	6197#	6203#	6206#	6227#	6231#	6234#	6275#	6277#
	6286#	6290#	6293#	6334#	6336#	6352#	6356#	6359#	6392#	6395#	6399#	6402#	6440#	6447#	6449#
	6453#	6456#	6497#	6502#	6506#	6509#	6545#	6552#	6554#	6558#	6561#	6598#	6601#	6609#	6613#
	6616#	6670#	6675#	6683#	6694#	6698#	6701#	6790#	6794#	6797#	6897#	6901#	6904#	6943#	6950#
	6952#	6957#	6960#	6999#	7006#	7008#	7013#	7016#	7056#	7059#	7067#	7072#	7075#	7156#	7162#
	7216#	7227#	7270#	7352#	7353#	7360#									
MSLDRO	3264#	3334#	3356#	4214#	4295#	4777#	4779#	4782#	4794#	4798#	4804#	4908#	5083#	5089#	6083#
	6117#	6164#													
MSMCHI	1100#														
MSMCLO	1100#														
MSPOP	1196#	1209#	1236#	1245#	1920#	1924#	1928#	2213#	4689#	4747#	4831#	4916#	4948#	4971#	4990#
	5003#	5155#	5188#	5215#	5243#	5273#	5314#	5378#	5420#	5483#	5523#	5586#	5642#	5701#	5757#
	5791#	5856#	5905#	5994#	6090#	6127#	6203#	6231#	6290#	6356#	6399#	6453#	6506#	6558#	6613#

	1930	1933	1936	1939	1942										
PRINTB	1930	1933	1936	1939	1942										
PRINTF	4810	5831	5839												
PRINTX	1945	1948	1951	1954	1957										
READBU	4784														
READEF	4777	4779	4782	4794	4798										
REG	1404#	3984	3985	3986	3987	3988	3989	6087							
REGTB	1398#	3984	3985	3986	3987	3988	3989	6087							
REGTBL	1453#	5153	5186	5213	5241	5271	5312	5376	5418	5481	5521	5583	5754	5788	5847
	5902	5991	6086	6124	6200	6228	6287	6353	6396	6450	6503	6555	6610	6695	6791
	6898	6953	7009	7068	7158										
RFLAGS	4775														
RGTABL	1416#	5154	5187	5214	5242	5272	5313	5377	5419	5482	5522	5584	5640	5699	5755
	5789	5848	5903	5992	6088	6125	6201	6229	6288	6354	6397	6451	6504	6556	6611
	6696	6792	6899	6954	7010	7069	7159								
SETPRI	3334	3356	6117	6164											
SETVEC	4808	5081	6076												
SVC	1099#	1100													
TSETUP	1337#	5144	5177	5207	5235	5264	5299	5351	5404	5458	5509	5560	5622	5678	5732
	5782	5827	5888	5949	6072	6115	6161	6224	6271	6330	6386	6435	6489	6541	6593
	6665	6750	6849	6938	6994	7050	7123								
TSTABL	1369#	5154	5187	5214	5242	5272	5313	5377	5419	5482	5522	5584	5640	5699	5755
	5789	5848	5903	5992	6088	6125	6201	6229	6288	6354	6397	6451	6504	6556	6611
	6696	6792	6899	6954	7010	7069	7159								
TSTITL	1299#	5124	5158	5191	5218	5246	5276	5317	5381	5423	5486	5527	5589	5645	5704
	5760	5794	5859	5908	6044	6093	6130	6206	6234	6293	6359	6402	6456	6509	6561
	6616	6701	6797	6904	6960	7016	7075								
TSTSUP	1345#	5144	5177	5207	5235	5264	5299	5351	5404	5458	5509	5560	5622	5678	5732
	5782	5827	5888	5949	6072	6115	6161	6224	6271	6330	6386	6435	6489	6541	6593
	6665	6750	6849	6938	6994	7050	7123								
TSTTL	1307#	5124	5158	5191	5218	5246	5276	5317	5381	5423	5486	5527	5589	5645	5704
	5760	5794	5859	5908	6044	6093	6130	6206	6234	6293	6359	6402	6456	6509	6561
	6616	6701	6797	6904	6960	7016	7075								
TTBL	1363#	5154	5187	5214	5242	5272	5313	5377	5419	5482	5522	5584	5640	5699	5755
	5789	5848	5903	5992	6088	6125	6201	6229	6288	6354	6397	6451	6504	6556	6611
	6696	6792	6899	6954	7010	7069	7159								
XFER	5152#	5185#	5212#	5240#	5270#	5311#	5375#	5417#	5480#	5520#	5582#	5639#	5697#	5753#	5786#
	5846#	5900#	5990#	6085#	6123#	6197#	6227#	6286#	6352#	6395#	6449#	6502#	6554#	6609#	6694#
	6790#	6897#	6952#	7008#	7067#	7156#									
XFERF	7273	7281													

. ABS. 037670 000

ERRORS DETECTED: 0

,CZRXFB/CRF/NL:TOC=SVC.SML/ML,CZRXFB.P11
 RUN-TIME: 34 34 6 SECONDS
 RUN-TIME RATIO: 162/75=2.1
 CORE USED: 36K (71 PAGES)