

TS03

SUPPLEMENTAL INSTRUCTION
CZTSFDO
TEST

AH-9440D-MC

COPYRIGHT ©75-78

FICHE 1 OF 1

APR 1978

digital

MADE IN USA

801

EOF10ZDVDCSEQ

00010000

780330

PDP10 411

88HDR1CZTSFDSEQ

00010000

780330

P00000001
CZTSFD.P11

15-FEB-78 14:04

CZTSFDD TS03 SPLMTL INSTR

MACY11 30A(1052) 15-FEB-78 14:05 PAGE 1

.REM %

IDENTIFICATION

PRODUCT CODE: AC-9439D-MC
PRODUCT NAME: CZTSFDD TS03 SPLMTL INSTR
PROGRAM DATE: MARCH 1978
MAINTAINER: DIAGNOSTIC ENGINEERING

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

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

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

COPYRIGHT (C) 1975, 1978 BY DIGITAL EQUIPMENT CORPORATION

00010000 780330 P00000001 CZTSFD.P11 15-FEB-78 14:04 MACY11 30A(1052) 15-FEB-78 14:05 PAGE 1

CO1

CZTSFDD TS03 SPLMTL INSTR
CZTSFDD.F11 15-FEB-78 14:04

MACY11 30A(1052) 15-FEB-78 14:05 PAGE 2

SEQ 0002

TABLE OF CONTENTS

PARAGRAPH	SUBJECT	PAGE
1.	ABSTRACT	3
2.	REQUIREMENTS	3
3.	LOADING PROCEDURE	3
4.	STARTING PROCEDURE	3
5.	SWITCH SETTINGS	5
6.	ERROR PRINTOUT	5
7.	OPERATION	6
8.	TEST DESCRIPTION	7
9.	LISTING	

የግንባታው ዋጋ 100 ሺህ ብር ነው።

B. 210(8): STARTING AT THIS ADDRESS WILL NOT PRINT THE HEADER OR THE UNIT SELECT REQUEST AND IS INTENDED AS A RESTART ADDRESS ONLY.

5.

THIS PROGRAM HAS BEEN MODIFIED TO RUN ON A PROCESSOR WITH OR WITHOUT A HARDWARE SWITCH REGISTER. WHEN FIRST EXECUTED THE PROGRAM TESTS THE EXISTENCE OF A HARDWARE SWITCH REGISTER. IF NOT FOUND A SOFTWARE SWITCH REGISTER LOCATION (SWREG=LOC. 176) IS DEFAULTED TO. IF THIS IS THE CASE, UPON EXECUTION THE CONTENTS OF THE SWREG ARE DUMPED IN OCTAL ON THE CONSOLE TTY AND ANY CHANGES ARE REQUESTED

(IE) SWR=XXXXXX NEW=

POSSIBLE RESPONSES ARE:

1. <CR> IF NO CHANGES ARE TO BE MADE
2. 6 DIGITS 0-7 TO REPRESENT IN OCTAL THE NEW SWITCH REGISTER VALUE .LAST DIGIT FOLLOWED BY <CR>.
3. ↑U TO ALLOW REENTERING VALUE IF ERROR IS COMMITTED KEYING IN SWREG VALUE.
4. <LF> ONLY VALID FOR ACT-11 SYSTEMS-DO NOT USE

BUILT INTO THE PROGRAM IS THE ABILITY TO DYNAMICALLY CHANGE THE CONTENTS OF SWREG DURING PROGRAM EXECUTION. BY STRIKING ↑G (CNTL G) ON CONSOLE TTY THE OPERATOR SETS A REQUEST FLAG TO CHANGE THE CONTENTS OF SWREG, WHICH IS PROCESSED IN KEY AREAS OF THE PROGRAM CODE (IE) ERROR ROUTINES, AFTER HALTS END OF PASS, AND OTHER APPLICABLE AREAS.

NOTE ANOTHER MODIFICATION TO THE PROGRAM IS THE TESTING OF 1-8 TS03 UNITS UNDER ACT11M.

5.1

CONSOLE SWITCH SETTING

ALL SWITCHES EXCEPT 3-9 ARE USED AND THE NORMAL, OR DEFAULT, RUN IS DONE WITH ALL SWITCHES SET TO ZERO (0). ALL SWITCHES ARE DYNAMIC AND MAY BE CHANGED AT ANY TIME.

SW15: 1=HALT ON ERROR
0=CONTINUE
SW14: 1=LOOP ON ERROR (SCOPE)
0=CONTINUE
SW13: 1=INHIBIT ERROR TIME OUT
0=PRINT ALL ERRORS
SW12: 1=INHIBIT ITERATION
0=ITERATE EACH TEST ITS ASSIGNED AMOUNT
SW11: 1=CONTINUOUS CYCLE
0=HALT AT END OF PASS
SW10: 1=HALT AT END OF CURRENT TEST
0=CONTINUE
SW9-3: NOT USED
SW2-0: SELECT INDIVIDUAL TEST (1-4)** 00 = DO ALL TESTS

6. ERROR PRINTOUTS

THERE ARE THREE (3) TYPES OF ERROR PRINTOUTS WHICH MAY
APPEAR: STATUS ERROR, DATA ERROR, POSITION ERROR.

- A. STATUS ERROR: ANY READ, WRITE, OR SPACE OPERATION WHICH
RESULTS IN SOME BAD STATUS (BIT 15 OF MTC),
OR UNEXPECTED BUS ADDRESS, OR INCORRECT
BYTE COUNT, WILL BE PRINTED.
- B. DATA ERROR: ANY READ OPERATION WHICH RESULTS IN UNEXPECTED
DATA WILL BE PRINTED.
- C. POSITION ERROR: ANY SPACE OR REWIND OPERATION RESULTING IN
UNEXPECTED STATUS WILL BE PRINTED.

EXAMPLES***

1. THE FOLLOWING EXAMPLE SHOWS A TYPICAL STATUS ERROR.

TEST1: WRITE FROM ODD BYTE
WRITE ERROR THIS PRINT SHOWS THAT WHILE EXECUTING
MTS: 10101 TEST 1 ON UNIT 2 AT 800 BPI, A WRITE
MTC: 161204 PARITY ERROR OCCURED. THE BYTE COUNT
MTBC: 0 IS ZERO AS IT SHOULD BE AND THE CURRENT
MTCA: 6003 6003 ADDRESS IS AS EXPECTED.

2. THE FOLLOWING EXAMPLE SHOWS A TYPICAL DATA ERROR.

TEST 2: READ TO ODD BYTE
DATA ERROR THIS PRINT SHOWS THAT A SINGLE BIT WAS
CN: 0 PICKED UP IN BOTH CHARACTER NUMBER ZERO
G: 00000000 (0) AND CHARACTER NUMBER THREE (3).
B: 01000000
CN: 3
G: 00000011
B: 01000011

3. THE FOLLOWING EXAMPLE SHOWS AN ERROR DURING A REWIND OPERATION.

TEST4: OPI TOO LONG
REWIND ERROR: NO BOT

11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
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
569
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
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
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
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086

THE PROCEDURES FOR OPERATING THIS PROGRAM ARE QUITE SIMPLE AND REQUIRE ONLY A FEW STEPS:

1. LOAD ADDRESS 200 OR 210
2. SET SWITCHES FOR DESIRED TEST SEQUENCE
3. PRESS START

ALL CONSOLE SWITCHES ARE DYNAMIC AND MAY BE CHANGED AT ANY TIME. THE NORMAL OPERATING SEQUENCE IS ALL SWITCHES DOWN (0). THE PROGRAM WILL TAKE APPROXIMATELY 1.25 MINUTES TO RUN; HOWEVER, IF ITERATIONS ARE INHIBITED (SW1=1), THE PROGRAM WILL RUN IN ABOUT .75 MINUTES. THE END OF PASS IS NOTED BY A PRINTOUT STATING END OF PASS AND THE NUMBER OF THAT PASS.

SINGLE TEST SELECTION: (SWO-SW3)

WHEN SWO-3 ARE SET TO ZERO (0), THE SCHEDULER WILL EXECUTE ALL TESTS (1-4) IN SEQUENCE AS A SINGLE PASS. IF SWO-3 ARE SET TO SOME NUMBER BETWEEN 1 AND 4, THEN THAT PARTICULAR TEST WILL BE EXECUTED CONTINUOUSLY. THE PROGRAM MAY BE STOPPED AT THE END OF THE CURRENT TEST (EITHER IN SEQUENCE OR SINGLE TEST MODE) BY SETTING SWITCH TEN (SW10) TO A ONE (1). YOU MAY SELECT TEST NUMBERS IN ANY ORDER (UP OR DOWN) BECAUSE EACH TEST IS SELF CONTAINED.

[illegible]

%

CZTSFDD TS03 SPLMTL INSTR
CZTSFDD.F11 15-FEB-78 14:04

MACY:1 30A(1052) 15-FEB-78 14:05 PAGE 8

SEQ 0008

```
.TITLE CZTSFDD TS03 SPLMTL INSTR
:ZZ - CZTSFDD
:15 APR 75/MARCH 1976/JAN 1978
:R. BARNES/RON PLATUKIS/CLEM WALSH
.ABS
```

```
:*****
:NOTE: PROGRAM HAS BEEN MODIFIED TO RUN WITH OR WITHOUT
:      A HARDWARE SWITCH REGISTER-REFER TO DOCUMENT
:*****
```

```
:CONSOLE SWITCHES
```

```
:SW15: 1=HALT ON ERROR
:      0=CONTINUE
:SW14: 1=LOOP ON ERROR (SCOPE)
:      0=CONTINUE
:SW13: 1=DO NOT PRINT ERRORS
:      0=PRINT ALL ERRORS
:SW12: 1=INHIBIT ITERATIONS
:      0=DO ITERATIONS
:SW11: 1=CONTINUOUS CYCLE
:      0=SINGLE PASS
:SW10: 1=HALT AT END OF EACH TEST
:      0=CONTINUE
:SW0-5: SELECT TEST NUMBER:: 00=ALL TESTS
```

```
:REGISTER EQUIVS*****
```

```
R0=%0
R1=%1
R2=%2
R3=%3
R4=%4
R5=%5
SP=%6
PC=%7
```

```
:TRAP CATCHERS*****
```

```
000000
000001
000002
000003
000004
000005
000006
000007
```


CZTSFDO TS03 SPLMTL INSTR
CZTSFD.P11 15-FEB-78 14:04

SEQ 0009

[illegible]

	001000
000040	000040 000
000041	000041 000
000042	000042 000000
000046	000046 002006
000052	000052 000000
	001000

```

*****
MODIFIED JAN 24 1978
*****

++
--
    $SVPC=.                                ;SAVE PC

DRIVE:  .=40                                ;DRIVE # FOR XXDP LOAD MEDIUM
        .BYTE    0                          ;ASSEMBLE AS A 0

MEDIUM: .=41                                ;XXDP LOAD MEDIUM
        .BYTE    0                          ;ASSEMBLE AS A 0

        .=42                                ;AUTO MAN/MODE INDICATOR
        .WORD    0                          ;ASSEMBLE AS A 0

        .=46                                ;SET TO SENDAD IN .SEOP
        .WORD    SENDAD

        .=52                                ;CHARACTERISTICS OF PROGRAM
        .WORD    0                          ;ASSEMBLE AS A 0

        .=$SVPC                            ;RESTORE PC
*****

```


K01

CZTSFDD TS03 SPLMTL INSTR
CZTSFDD.F11 15-FEB-78 14:04

MACY11 30A(1052) 15-FEB-78 14:05 PAGE 10

SEQ 0010

```
353  
354 : *****  
355 : MODIFIED JAN 24 1978  
356 :  
357 : ++  
358 : ACT11 AND XXDP MODE INDICATORS  
359 : --  
360 :  
361 001000 000000 AUTOM: .WORD 0 ;AUTOMATIC MODE INDICATOR  
362 001002 000 ACT11M: .BYTE 0 ;ACT11 AUTO MODE INDICATOR  
363 001003 000 XXDPM: .BYTE 0 ;XXDP AUTO MODE INDICATOR  
364 001004 000 ADUMPM: .BYTE 0 ;ACT11 DUMP MODE INDICATOR  
365 001005 000 XDUMPM: .BYTE 0 ;XXDP DUMP MODE INDICATOR  
366 :  
367 : *****  
368 :
```


L01

CZTSFDO TS03 SPLMTL INSTR
CZTSFD.P11 15-FEB-78 14:04

MACY11 30A(1052) 15-FEB-78 14:05 PAGE 11

SEQ 0011

369


```

370                                     ;TTY INTERRUPT VECTOR*****
371                                     .=60
372 000060 000060                      ;TTY INTERRUPT HANDLER
373 000060 004544                      0
374 000062 000000
375
376                                     ;SOFTWARE SWITCH REGISTER LOCATIONS*****
377                                     .=174
378                                     DISPREG:0
379 000174 000000                      SWREG: 0
380 000176 000000
381
382                                     ;STARTING ADDRESS*****
383                                     .=200
384 000200 000200                      CLR      R0
385 000200 005000                      JMP      START      ;PROGRAM START
386 000202 000167 000572
387
388                                     .=210
389 000210 000210                      MOV      #1,R0
390 000214 012700 000001              JMP      START      ;NO HEADER START
391 000214 000167 000560
392
393                                     ;TMA-11 INTERRUPT VECTOR*****
394                                     .=224
395 000224 000224                      MTINT
396 000224 004524                      340      ;TAPE INTERRUPT HANDLER
397 000226 000340
398

```


NO1

CZTSF00 TS03 SPLMTL INSTR
CZTSFD.P11 15-FEB-78 14:04

MACY11 30A(1052) 15-FEB-78 14:05 PAGE 13

SEQ 0013

399 000600
400
401
402 000600 172520
403 000602 172522
404 000604 172524
405 000606 172526
406 000610 000000
407 000612 000020
408 000614 177760
409 000616 177776
410 000620 177570
411 000622 177570
412 000624 177560
413 000626 177562
414 000630 177564
415 000632 177566
416 000634 000010
417 000636 000040
418 000640 172520
419 000642 000224
420

. =600
;CONSTANTS*****
MTS: 172520 ;TAPE STATUS REGISTER
MTC: 172522 ;TAPE COMMAND REGISTER
MTBC: 172524 ;TAPE BYTE COUNTER
MTBA: 172526 ;TAPE BUS ADDRESS
UDES: 0 ;UNIT DESCRIPTION
RCNT: 20 ;RECORD COUNT
CCNT: -20 ;CHARACTER COUNT
PSW: 177776 ;PROCESSOR STATUS
SWR: 177570 ;CONSOLE SWITCH REGISTER
DISPLAY: 177570 ;CONSOLE DISPLAY REGISTER
TKS: 177560 ;TTY READ STATUS
TKB: 177562 ;TTY READ BUFFER
TPS: 177564 ;TTY PUNCH STATUS
TPB: 177566 ;TTY OUTPUT BUFFER
ITAMT: 10 ;NUMBER OF ITERATIONS
STALL: 40 ;READY DELAY MULTIPLIER
REGS: 172520 ;UNIBUS ADDRESS
VECT: 224 ;VECTOR ADDRESS

;FLAGS AND COUNTERS*****

421			
422			
423	000644	000000	TINF: 0
424	000646	000000	TOB: 00
425	000650	000000	TIB: 00
426	000652	000000	TEMP1: 00
427	000654	000000	TEMP2: 00
428	000656	000000	TEMP3: 00
429	000660	000000	EMADDR: 00
430	000662	000000	ERRAD: 00
431	000664	000000	LTADD: 00
432	000666	000000	ITRLP: 00
433	000670	000000	SPFLG: 00
434	000672	000000	STFLG: 00
435	000674	000000	PCNTR: 00
436	000676	000000	BADR: 00
437	000700	000000	BYTES: 00
438	000702	000000	SCNT: 00
439	000704	000000	FUN: 00
440	000706	000000	ITCNT: 00
441	000710	000000	CRCNT: 00
442	000712	000000	DERFL: 00
443	000714	000000	HDRFL: 00
444	000716	000000	PFLG: 00
445	000720	000000	UNP: 00
446	000722	000000	BCNT: 00
447	000724	000000	COUNT: 00
448	000726	000000	TEMPST: 00
449	000730	000000	RDSW: 0
450			

;TEST ENTRY TABLE*****

451			
452			
453	000732	000000	TSTTBL: 0
454	000734	000000	0
455	000736	002064	T1AD: LT1
456	000740	002064	T1IAD: LT1
457	000742	002312	T2AD: LT2
458	000744	002312	T2IAD: LT2
459	000746	002534	T3AD: LT3
460	000750	002632	T3IAD: LT3IT
461	000752	002764	T4AD: LT4
462	000754	003062	T4IAD: LT4IT
463	000756	000000	0
464			


```

4665      001000      . = 1000
4666      *****
4667      PROGRAM START AND HOUSEKEEPING
4668      *****
4669
4670
4671      001000      000240      START:  NOP
4672      001002      012777      000340      177606      MOV      #340, @PSW      ; SET PRIORITY
4673      001010      012706      000500      MOV      #500, SP      ; SET STACK POINTER
4674      001014      004767      004524      JSR      PC, SUSWR      ; SEE IF NO HARDWARE SWITCH REG.
4675      001020      004767      005160      JSR      PC, CKMODE      ; CHECK FOR MODE OF OPERATION ++ C.W
4676      001024      105767      177752      TSTB     ACT11M      ; ACT MODE? ++ C.W
4677      001030      001404      BEQ      1$      ; BRANCH - IF NO ++ C.W
4678      001032      005067      177636      CLR      PCNTR      ; INIT PASS COUNTER ++ C.W
4679      001036      000167      000256      JMP      $ACT      ; EXECUTE PROGRAM IN ACT MODE ++ C.W
4680      001042      005700      1$:      TST      R0      ; SEE IF SKIP HEADER
4681      001044      001402      BEQ      2$      ; BRANCH - IF NO ++ C.W
4682      001046      000167      000416      JMP      ST4      ; DO TEST FROM RESTART ++ C.W
4683      001052      012704      006410      2$:      MOV      #MSG1, R4      ; PRINT HEADER
4684      001056      004767      003730      JSR      PC, TTOUT      ; REQUEST UNIBUS ADDRESS
4685      001062      012704      007041      MOV      #MSG22, R4
4686      001066      004767      003720      JSR      PC, TTOUT
4687      001072      016703      177542      MCV      REGS, R3      ; PRINT CURRENT ADDRESS
4688      001076      004767      004106      JSR      PC, OCTP      ; GET ADDRESS OF ENTRY
4689      001102      012705      000640      MOV      #REGS, R5      ; SET SIZE OF ENTRY
4690      001106      012701      000006      MOV      #6, R1      ; SET UPPER LIMIT
4691      001112      012702      172700      MOV      #172700, R2      ; SET LOWER LIMIT
4692      001116      012703      172300      MOV      #172300, R3      ; GO GET RESPONSE
4693      001122      004767      003426      JSR      PC, TTR      ; REQUEST VECTOR
4694      001126      012704      007063      MOV      #MSG23, R4
4695      001132      004767      003654      JSR      PC, TTOUT      ; PRINT CURRENT VECTOR
4696      001136      016703      177500      MOV      VECT, R3      ; GET ADDRESS OF ENTRY
4697      001142      004767      004042      JSR      PC, OCTP      ; SET SIZE OF ENTRY
4698      001146      012705      000642      MOV      #VECT, R5      ; SET UPPER LIMIT
4699      001152      012701      000003      MOV      #3, R1      ; SET LOWER LIMIT
4700      001156      012702      000250      MOV      #250, R2      ; GO GET RESPONSE
4701      001162      012703      000150      MOV      #150, R3      ; GET VECTOR
4702      001166      004767      003362      JSR      PC, TTR      ; SET INTERRUPT ADDRESS IN VECTOR
4703      001172      016700      177444      MOV      VECT, R0      ; SET INTERRUPT PRIORITY
4704      001176      012720      004524      MOV      #MTINT (R0)+      ; SET NUMBER OF REGISTER
4705      001202      012710      000340      MOV      #340, (R0)      ; GET FIRST ADDRESS OF TABLE
4706      001206      012701      000004      MOV      #4, R1      ; GET FIRST REGISTER
4707      001212      012702      000600      MOV      #MTS, R2      ; LOAD TABLE
4708      001216      016700      177416      MOV      REGS, R0      ; BUMP ADDRESS
4709      001222      010022      000002      STOA:  MOV      R0, (R2)+      ; SEE IF DONE
4710      001224      062700      ADD      #2, R0      ; IF NOT: BE
4711      001230      005301      DEC      R1      ; POWER CLEAR
4712      001232      001373      BNE     STOA
4713      001234      012777      010000      177340      MOV      #10000, @MTC
4714      001242      012704      006444      STO:  MOV      #MSG2, R4      ; REQUEST UNIT NUMBER
4715      001246      004767      003540      JSR      PC, TTOUT      ; PRESET UNIT 0
4716      001252      005067      177332      CLR      UDES      ; GET UNIT NUMBER
4717      001256      016703      177326      MOV      UDES, R3      ; POSITION
4718      001262      000303      SWAB     R3      ; MASK UNIT NUMBER
4719      001264      042703      177770      BIC      #177770, R3      ; PRINT CURRENT VALUE
4720      001270      004767      003714      JSR      PC, OCTP

```


521	001274	012705	000656		MOV	#TEMP3,R5	:SET SAVE LOCATION
522	001300	012701	000001		MOV	#1,R1	:SET SIZE OF ENTRY
523	001304	012702	000007		MOV	#7,R2	:SET UPPER LIMIT
524	001310	012703	000000		MOV	#0,R3	:SET LOWER LIMIT
525	001314	004767	003234		JSR	PC,TTR	:GO GET UNIT NUMBER
526	001320	105767	177456		TSTB	ACT11M	:ACT MODE? ++ C.W
527	001324	001413		\$ACT:	BEQ	NOACT	:BRANCH - IF NO ++ C.W
528	001326	005067	177256		CLR	UDES	:INIT UNIT INDICATOR IN ACT MODE ++ C.W
529	001332	122767	000010	177250	ST:	CMPB	#10,UDES
530	001340	001002			BNE	2\$	:ARE ALL UNITS DONE? ++ C.W
531	001342	000167	000410		JMP	\$DONE	:BRANCH - IF NO ++ C.W
532	001346	116767	177236	177302	2\$:	MOVB	UDES,TEMP3
533	001354	000367	177276		NOACT:	SWAB	TEMP3
534	001360	042767	003400	177222		BIC	#3400,UDES
535	001366	056767	177264	177214		BIS	TEMP3,UDES
536	001374	016777	177210	177200		MOV	UDES,AMTC
537	001402	005000			CLR	RO	:SELECT UNIT
538	001404	022767	000176	177206		CMP	#SWREG,SWR
539	001412	001002			BNE	ST1	
540	001414	004767	004256		JSR	PC,CNTLU	:GO ASK FOR SWR SETTINGS
541	001420	032777	000100	177152	ST1:	BIT	#100,AMTS
542	001426	001015			BNE	ST3	:SEE IF SELECT REMOTE
543	001430	005300			DEC	RO	:IF SO: BR
544	001432	001372			BNE	ST1	:DELAY FOR SELECT REMOTE
545	001434	105767	177342		TSTB	ACT11M	:ACT MODE? ++ C.W
546	001440	001403			BEQ	1\$	:BRANCH - IF NO ++ C.W
547	001442	105267	177142		INCB	UDES	:GET NEXT UNIT ++ C.W
548	001446	000731			BR	ST	:CONTINUE ++ C.W
549	001450	012704	007020		1\$:	MOV	#MSG21,R4
550	001454	004767	003332		JSR	PC,TOUT	:PRINT NOT AVAILABLE
551	001460	000670			BR	ST0	:REDO REQUEST
552	001462	052767	060000	177120	ST3:	BIS	#60000,UDES
553	001470	000240			ST4:	NOP	:SET TO 800 BPI 9 CHAN
554	001472	012777	010000	177102		MOV	#10000,AMTC
555	001500	012702	000646			MOV	#TOB,R2
556	001504	012700	000027			MOV	#27,RO
557	001510	005022			ST5:	CLR	(R2)+
558	001512	005300				DEC	RO
559	001514	001375			BNE	ST5	:DO ALL
560	001516	105767	177260		TSTB	ACT11M	:ACT MODE? ++ C.W
561	001522	001002			BNE	TSCD	:BRANCH - IF YES ++ C.W
562	001524	005067	177144		CLR	PCNTR	:CLEAR PASS COUNTER
563							


```

564                                     ;TEST SCHEDULAR*****
565
566 001530 000240 TSCD: NOP
567 001532 005067 177134 CLR STFLG ;CLEAR SINGLE TEST FLAG
568 001536 017700 177056 MOV @SWR,RO ;GET SWITCH REGISTER
569 001542 042700 177700 BIC #177700,RO ;MASK TEST SELECT
570 001546 005700 TST RO ;SEE IF SINGLE TEST SELECT
571 001550 001046 BNE STSCD ;IF SO: BR
572 001552 012767 000732 177104 MOV #TSTTBL,LTADD ;GET TABLE START
573 001560 062767 000004 177076 TSCD0: ADD #4,LTADD
574 001566 016767 177072 177076 LTADD,ITRLP ;SET ITERATION ADDRESS
575 001574 062767 000002 177064 ADD #2,ITRLP
576 001602 005777 177056 TST @LTADD ;SEE IF END OF CYCLE
577 001606 001002 BNE TSCD1 ;IF NOT: BR
578 001610 000167 000124 JMP TEND ;GO TO END ROUTINE
579 001614 005067 177074 TSCD1: CLR HDRFL ;CLEAR HEADER FLAG
580 001620 017700 177040 MOV @LTADD,RO ;GET TEST ADDRESS
581 001624 000110 JMP (RO) ;GO TO TEST
582 001626 032777 002000 176764 TSCD2: BIT #2000,@SWR ;SEE IF HALT ON TEST
583 001634 001401 BEQ TSCD3 ;IF NOT: BR
584 001636 000000 HALT
585 001640 004767 003760 TSCD3: JSR PC,CKSWR ;TEST FOR IG
586 001644 005767 177022 TST STFLG ;SEE IF SINGLE TEST
587 001650 001743 BEQ TSCD0 ;IF NOT: BR
588 001652 017700 176742 MOV @SWR,RO
589 001656 042700 177760 BIC #177760,RO ;GET TEST NUMBER
590 001662 005700 TST RO ;SEE IF ALL TESTS
591 001664 001721 BEQ TSCD ;IF SO: BR
592 001666 012767 000001 176776 STSCD: MOV #1,STFLG ;SET SINGLE TEST FLAG
593 001674 022700 000005 CMP #5,RO ;SEE IF EXCEEDED TEST NUMBER
594 001700 003417 BLE TEND ;IF SO: BR
595 001702 000241 CLC
596 001704 006100 ROL RO
597 001706 006100 ROL RO ;POSITION NUMBER
598 001710 012767 000732 176746 MOV #TSTTBL,LTADD ;GET START OF TABLE
599 001716 060067 176742 ADD RO,LTADD ;SET POINTER
600 001722 016767 176736 176736 MOV LTADD,ITRLP
601 001730 062767 000002 176730 ADD #2,ITRLP
602 001736 000726 BR TSCD1 ;SET ITERATION ADDRESS
603 001740 105767 177036 TEND: TSTB ACT11M ;GO DO TEST
604 001744 001404 BEQ $DONE ;ACT MODE? ++ C.W
605 001746 105267 176636 INCB UDES ;BRANCH - IF NO ++ C.W
606 001752 000167 177354 JMP ST ;GET NEXT UNIT ++ C.W
607 001756 012704 006470 $DONE: MOV #MSG3,R4 ;AND CONTINUE TESTING ++ C.W
608 001762 004767 003024 JSR PC,TROUT ;PRINT END OF PASS
609 001766 016703 176702 MOV PC,NTR,R3
610 001772 004767 003212 JSR PC,OCIP
611 001776 013700 000042 MOV @#42,RO
612 002002 001405 BEQ HERE
613 002004 000005 RESET
614 002006 004710 $ENDAD: JSR PC,(RO) ;PRINT PASS NUMBER
615 002010 000240 NOP ;GET MONITOR ADDRESS ++ C.W
616 002012 000240 NOP ;BRANCH - IF NOT AUTO MODE ++ C.W
617 002014 000240 NOP ;CLEAR THE WORLD ++ C.W
618 002016 000240 NOP ;GO TO MONITOR'S ADDRESS ++ C.W
619 002020 032777 004000 176572 HERE: BIT #4000,@SWR ;SEE IF HALT ON PASS

```


F02

CZTSFDD TS03 SPLMTL INSTR
CZTSFD.P11 15-FEB-78 14:04

MACY11 30A(1052) 15-FEB-78 14:05 PAGE 18

SEQ 0018

620	002026	001001	
621	002030	000000	
622	002032	105767	176744
623	002036	001006	
624	002040	004767	003560
625	002044	005267	176624
626	002050	000167	177454
627	002054	005267	176614
628	002060	000167	177234
629			

	BNE	TENDX
	HALT	
TENDX:	TSTB	ACT11M
	BNE	IS
	JSR	PC,CKSWR
	INC	PCNTR
	JMP	TSCU
IS:	INC	PCNTR
	JMP	\$ACT

```

: IF NOT: BR
: ACT MODE? ++ C.W
: BRANCH - IF YES ++ C.W
: TEST FOR IG
: BUMP PASS COUNTER
: RESTART
: BUMP COUNTER IN ACT MODE ++ C.W
: RESTART IN ACT MODE ++ C.W

```


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
674
675002064 000240
002066 012767 007075 176564
002074 012702 007352
002100 112722 000377
002104 005000
002106 110022
002110 005200
002112 022700 000006
002116 001373
002120 004767 001046
002124 012767 000004 176552
002132 012767 007353 176536
002140 012767 177772 176532
002146 012767 006715 176506
002154 004767 001262
002160 000240
002162 004767 001504
002166 012767 177777 176506
002174 004767 001156
002200 012702 007454
002204 012700 000010
002210 012722 177777
002214 005300
002216 001374
002220 012767 000002 176456
002226 012767 007454 176442
002234 012767 177772 176436
002242 012767 006732 176412
002250 004767 001166
002254 000240
002256 004767 001410
002262 012701 007353
002266 012702 007454
002272 012700 000006
002276 004767 001656
002302 004767 002146
002306 000167 177314LT1:

LT1B:

LT1C:

LT1D:

LT1E:*****
:TEST 1: WRITE FROM ODD BYTE
:
:THIS TEST WILL WRITE A SIX (6) BYTE RECORD
:FROM AN ODD BYTE STARTING ADDRESS. THE RECORD
:WILL BE READ BACK INTO AN EVEN STARTING ADDRESS
:TO TEST FOR PROPER TRANSFER.
:*****
:LT1MSG,EMADDR ;SET HEADER
:WDATA,R2 ;GET BUFFER START
:377,(R2)+ ;INSERT BACKGROUND DATA
:CLR R0
:MOV R0,(R2)+ ;LOAD WRITE BUFFER (0,1,2,3,4,5)
:INC R0
:CMP #6,R0
:BNE LT1B
:JSR PC,RWIND ;GO REWIND
:MOV #4,FUN ;SET WRITE FUNCTION CODE
:MOV WDATA+1,BADR ;SET DATA POINTER
:MOV #-6,BYTES ;SET SIZE OF RECORD
:MOV MSG17,ERRAD ;SET WRITE ERROR CODE
:JSR PC,EXEC ;GO EXECUTE COMMAND
:NOP
:JSR PC,ERCHK ;GO CHECK FOR STATUS ERROR
:MOV #-1,SCNT
:JSR PC,BKSP ;GO BACKSPACE ONE RECORD
:MOV WDATA,R2
:MOV #10,R0
:MOV #-1,(R2)+ ;BACKGROUND READ BUFFER
:DEC R0
:BNE LT1D
:MOV #2,FUN ;DO ALL
:MOV WDATA,BADR ;SET READ FUNCTION CODE
:MOV #-6,BYTES ;SET READ POINTER
:MOV MSG18,ERRAD ;SET SIZE OF RECORD
:JSR PC,EXEC ;SET READ ERROR CODE
:NOP
:JSR PC,ERCHK ;GO CHECK ERRORS
:MOV WDATA+1,R1 ;SET EXPT DATA POINTER
:MOV WDATA,R2 ;SET RCVD DATA POINTER
:MOV #6,R0 ;SET SIZE OF RECORD
:JSR PC,DCHK ;GO CHECK DATA
:JSR PC,ITER ;GO SEE IF ITERATIONS
:JMP TS0D2 ;RETURN TO SCHEDULAR


```

676
677
678
679
680
681
682
683
684
685 002312 000240
686 002314 012767 007133 176336 LT2: NOP
687 002322 012702 007352 MOV #LT2MSG,EMADDR ;SET HEADER POINTER
688 002326 005000 CLR #WDATA,R2 ;POINT TO START OF WRITE BUFFER
689 002330 110022 LT2B: MOVB R0,(R2)+ ;LOAD DATA PATTERN
690 002332 005200 INC R0 ;BUMP PATTERN
691 002334 022700 000006 CMP #6,R0 ;SEE IF DONE
692 002340 001373 BNE LT2B ;IF NOT: BR
693 002342 004767 000624 JSR PC,RWND ;GO REWIND TO BOT
694 002346 012767 000004 176330 MOV #4,FUN ;SET WRITE OP-CODE
695 002354 012767 007352 176314 MOV #WDATA,BADR ;SET STARTING ADDRESS
696 002362 012767 177772 176310 MOV #-6,BYTES ;SET SIZE OF RECORD
697 002370 004767 001046 JSR PC,EXEC ;GO EXECUTE COMMAND
698 002374 000240 LT2C: NOP
699 002376 012767 006715 176256 MOV #MSG17,ERRAD ;SET ERROR CODE
700 002404 004767 001262 JSR PC,ERCHK ;GO CHECK FOR STATUS ERROR
701 002410 012767 177777 176264 MOV #-1,SCNT
702 002416 004767 000734 JSR PC,BKSP ;GO BACKSPACE ONE RECORD
703 002422 012702 007454 MOV #RDATA,R2 ;GET READ BUFFER POINTER
704 002426 012700 000010 MOV #10,R0 ;SET SIZE
705 002432 012722 177777 LT2D: MOV #-1,(R2)+ ;BACKGROUND POINTER
706 002436 005300 DEC R0 ;SEE IF DONE
707 002440 001374 BNE LT2D ;IF NOT: BR
708 002442 012767 000002 176234 MOV #2,FUN ;SET READ FUNCTION CODE
709 002450 012767 007455 176220 MOV #RDATA+1,BADR ;SET START OF READ BUFFER
710 002456 012767 177772 176214 MOV #-6,BYTES ;SET SIZE OF RECORD
711 002464 004767 000752 JSR PC,EXEC ;GO EXECUTE COMMAND
712 002470 000240 LT2E: NOP
713 002472 012767 006732 176162 MOV #MSG18,ERRAD ;SET ERROR CODE
714 002500 004767 001166 JSR PC,ERCHK ;GO CHECK FOR STATUS ERROR
715 002504 012701 007352 MOV #WDATA,R1 ;POINT TO EXPT DATA
716 002510 012702 007455 MOV #RDATA+1,R2 ;POINT TO RCVD DATA
717 002514 012700 000006 MOV #6,R0 ;SET SIZE OF RECORD
718 002520 004767 001434 JSR PC,DCHK ;GO CHECK DATA
719 002524 004767 001724 JSR PC,ITER ;GO SEE IF ITERATION
720 002530 000167 177072 JMP TS0D2 ;RETURN TO SCHEDULAR
721

```


722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766

:*****

:TEST 3: OPI TOO LONG

:THIS TEST WILL ERASE APPROXIMATELY TEN POINT FIVE (10.5)
:FEET OF TAPE BY WRITING WITH IRG, BACKSPACING
:AND REPEATING THE SEQUENCE 32(10) TIMES. TAPE
:WILL REWIND AND A READ FORWARD ISSUED. THE
:OPI TIMER SHOULD SHUTDOWN THE UNIT BEFORE
:REACHING THE FIRST RECORD ON TAPE.

:*****

002534	000240			LT3:	NOP		
002536	012767	007166	176114		MOV	#LT3MSG,EMADDR	:SET TEST HEADER
002544	012700	000040			MOV	#40,RO	:SET NUMBER OF WRITE IRG/BACKSPACE
002550	004767	000416			JSR	PC,RWND	:GO REWIND UNIT
002554	012767	000014	176122	LT3A:	MOV	#14,FUN	:SET WRITE IRG FUNCTION CODE
002562	012767	007352	176106		MOV	#WDATA,BADR	:SET BUS ADDRESS
002570	012767	177760	176102		MOV	#-20,BYTES	:SET SIZE OF RECORD
002576	004767	000640			JSR	PC,EXEC	:GO EXECUTE COMMAND
002602	012767	006715	176052	LT3B:	MOV	#MSG17,ERRAD	:SET ERROR CODE
002610	004767	001056			JSR	PC,ERCHK	:GO CHECK FOR STATUS ERROR
002614	012767	177777	176060		MOV	#-1,SCNT	
002622	004767	000530			JSR	PC,BKSP	:GO BACKSPACE ONE RECORD
002626	005300				DEC	RO	:SEE IF DONE ALL
002630	001351				BNE	LT3A	:IF NOT: BR
002632	000240			LT3IT:	NOP		
002634	004767	000332			JSR	PC,RWND	:GO REWIND
002640	012767	000230	175770		MOV	#230,STALL	:SET OPI STALL
002646	012767	007454	176022		MOV	#RDATA,BADR	:SET START OF READ BUFFER
002654	012767	177760	176016		MOV	#-20,BYTES	:SET SIZE OF RECORD
002662	012767	000002	176014		MOV	#2,FUN	:SET READ FUNCTION CODE
002670	012767	006746	175764		MOV	#MSG19,ERRAD	:SET ERROR CODE
002676	004767	000540			JSR	PC,EXEC	:GO EXECUTE COMMAND
002702	000240			LT3C:	NOP		
002704	012767	000040	175724		MOV	#40,STALL	:RESET NORMAL STALL
002712	032777	000400	175660		BIT	#400,QMTS	:SEE IF BTE IS SET
002720	001007				BNE	LT3X	:IF SO: BR
002722	012767	000001	175740		MOV	#1,SPFLG	:SET NO BA PRINT FLAG
002730	004767	001014			JSR	PC,ERPT	:GO PRINT ERROR
002734	005067	175730			CLR	SPFLG	:RESET FLAG
002740	012767	000002	175666	LT3X:	MOV	#2,ITAMT	:SET TO TWO (2) ITERATIONS
002746	004767	001502			JSR	PC,ITER	:GO SEE IF ITERATION
002752	012767	000010	175654		MOV	#10,ITAMT	:RESET ITERATIONS
002760	000167	176642			JMP	TSC02	:RETURN TO SCHEDULAR

767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806

002764 000240
002766 012767 007215 175664
002774 004767 000172
003000 012700 000014
003004 012767 000014 175672
003012 012767 007352 175656
003020 012767 177760 175652
003026 012767 006715 175626
003034 004767 000402
003040 004767 000626
003044 012767 177777 175630
003052 004767 000300
003056 005300
003060 001351
003062 000240
003064 004767 000102
003070 012767 000200 175540
003076 012767 007454 175572
003104 012767 177760 175566
003112 012767 000002 175564
003120 012767 006773 175534
003126 004767 000310
003132 004767 000534
003136 000240
003140 012767 000040 175470
003146 012767 000002 175460
003154 004767 001274
003160 012767 000010 175446
003166 000167 176434

LT4:

LT4A:

LT4B:

LT4IT:

LT4C:

TEST 4: OPI TOO SHORT

THIS TEST WILL ERASE APPROXIMATELY FOUR (4) FEET
OF TAPE BY WRITING WITH IRG, BACKSPACING
ONE (1) RECORD AND REPEATING THIS SEQUENCE
12(10) TIMES. TAPE WILL REWIND AND BE READ
FORWARD. THE FIRST RECORD ON TAPE SHOULD BE
REACHED BEFORE OPI TIMES OUT.

NOP
MOV #LT4MSG, EMADDR ;SET HEADER
JSR PC, RWND ;GO REWIND
MOV #14, RO ;SET NUMBER OF WRITE IRG/BACKSPACES
LT4A: MOV #14, FUN ;SET WRITE IRG FUNCTION CODE
MOV #WDATA, BADR ;SET START OF WRITE BUFFER
MOV #-20, BYTES ;SET SIZE OF RECORD
MOV #MSG17, ERRAD ;SET ERROR CODE
JSR PC, EXEC ;GO EXECUTE COMMAND
LT4B: JSR PC, ERCHK ;GO CHECK FOR STATUS ERROR
MOV #-1, SCNT
JSR PC, BKSP ;GO BACKSPACE ONE RECORD
DEC RO ;SEE IF DONE ALL SEQUENCES
LT4A BNE LT4A ;IF NOT: BR
LT4IT: NOP
JSR PC, RWND ;REWIND
MOV #200, STALL ;SET OPI STALL
MOV #RDATA, BADR ;SET START OF READ BUFFER
MOV #-20, BYTES ;SET SIZE OF RECORD
MOV #2, FUN ;SET READ FUNCTION CODE
MOV #MSG20, ERRAD ;SET ERROR CODE
JSR PC, EXEC ;GO EXECUTE COMMAND
LT4C: JSR PC, ERCHK ;GO CHECK FOR STATUS ERRORS
NOP
MOV #40, STALL ;RESET NORMAL STALL
MOV #2, ITAMT ;SET TO TWO (2) ITERATIONS
JSR PC, ITER ;GO SEE IF ITERATIONS
MOV #10, ITAMT ;RESET ITERATIONS
JMP TSCD2 ;RETURN TO SCHEDULAR

						:REWIND SUBROUTINE*****	
807							
808							
809	003172	000240			RWND:	NOP	
810	003174	016777	175410	175400		MOV UDES,AMTC	;SELECT UNIT
811	003202	032777	000040	175370		BIT #40,AMTS	;SEE IF AT BOT
812	003210	001056				BNE RWNDXX	;IF SO: BR
813	003212	052777	000017	175362		BIS #17,AMTC	;START REWIND
814	003220	105777	175356		1\$:	TSTB AMTC	
815	003224	100375				BPL 1\$	;AWAIT CUR
816	003226	032777	000001	175344	RWND1:	BIT #1,AMTS	;AWAIT TUR
817	003234	001774				BEQ RWND1	
818	003236	032777	000040	175334		BIT #40,AMTS	;SEE IF BOT SET
819	003244	001040				BNE RWNDXX	;IF SO: BR
820	003246	032777	020000	175344		BIT #20000,ASWR	;SEE IF PRINT ERROR
821	003254	001030				BNE RWNDX	;IF NOT: BR
822	003256	016704	175376			MOV EMADDR,R4	
823	003262	004767	001524			JSR PC,TTOUT	;PRINT HEADER
824	003266	012704	006511			MOV #MSG4,R4	
825	003272	004767	001514			JSR PC,TTOUT	;PRINT REWIND ERROR
826	003276	012704	006541			MOV #MSG5,R4	
827	003302	004767	001504			JSR PC,TTOUT	;PRINT MTS TAG
828	003306	017703	175266			MOV AMTS,R3	
829	003312	004767	001662			JSR PC,OCTPE	;PRINT MTS
830	003316	012704	006550			MOV #MSG6,R4	
831	003322	004767	001464			JSR PC,TTOUT	;PRINT MTC TAG
832	003326	017703	175250			MOV AMTC,R3	
833	003332	004767	001642			JSR PC,OCTPE	;PRINT MTC
834	003336	005777	175256		RWNDX:	TST ASWR	;SEE IF HALT ON ERROR
835	003342	100001				BPL RWNDXX	;IF NOT: BR
836	003344	000000				HALT	
837	003346	004767	002252		RWNDXX:	JSR PC,CKSWR	;TEST FOR IG
838	003352	000240				NOP	
839	003354	000207				RTS PC	;RETURN
840							
841						:BACKSPACE SUBROUTINE*****	
842							
843	003356	000240			BKSP:	NOP	
844	003360	016777	175224	175214		MOV UDES,AMTC	;SELECT UNIT
845	003366	016777	175310	175210		MOV SCNT,AMTBC	;SET NUMBER OF RECORD
846	003374	052777	000013	175200		BIS #13,AMTC	;START SPACE REVERSE
847	003402	032777	000001	175170	BKSP1:	BIT #1,AMTS	
848	003410	001774				BEQ BKSP1	;AWAIT TUR
849	003412	012767	000001	175250		MOV #1,SPFLG	;SET SPACE FLAG
850	003420	012767	006577	175234		MOV #MSG9,ERRAD	
851	003426	004767	000240			JSR PC,ERCHK	;GO CHECK FOR ERROR
852	003432	005067	175232			CLR SPFLG	;CLEAR SPACE FLAG
853	003436	000240				NOP	
854	003440	000207				RTS FC	;RETURN
855							


```

856                                     ;COMMAND EXECUTE SUBROUTINE*****
857
858 003442 000240 EXEC: NOP
859 003444 005005 CLR R5
860 003446 032777 000200 175126 EXEC0: BIT #200,AMTC ;SEE IF CUR
861 003454 001021 BNE EXEC2 ;IF SO: BR
862 003456 005305 DEC R5 ;SEE IF TIMED OUT
863 003460 001372 BNE EXEC0 ;IF NOT: BR
864 003462 005767 175226 TST HDRFL ;SEE IF DONE HEADER
865 003466 001004 BNE EXEC1 ;IF SO: BR
866 003470 016704 175164 MOV EMADDR,R4
867 003474 004767 001312 JSR PC,TTOUT ;ELSE PRINT HEADER
868 003500 012704 006622 EXEC1: MOV #MSG10,R4
869 003504 004767 001302 JSR PC,TTOUT ;PRINT NOT READY ERROR
870 003510 005777 175104 TST ASWR ;SEE IF HALT ON ERROR
871 003514 100001 BPL EXEC2 ;IF NOT: BR
872 003516 000000 HALT
873 003520 004767 002100 EXEC2: JSR PC,CKSWR ;TEST FOR IG
874 003524 000240 NOP
875 003526 016777 175056 175046 MOV UDES,AMTC ;SELECT UNIT
876 003534 016777 175136 175044 MOV BADR,AMTBA ;SET BUS MEMORY ADDRESS
877 003542 016777 175132 175034 MOV BYTES,AMTBC ;SET BYTE COUNT
878 003550 016701 175130 MOV FUN,R1 ;GET FUNCTION
879 003554 052701 000101 BIS #101,R1 ;SET IN GO BIT AND INTERRUPT ENABLE
880 003560 050177 175016 BIS R1,AMTC ;LOAD COMMAND+GO+IE
881 003564 000240 NOP
882 003566 005077 175024 CLR APSW ;ALLOW INTERRUPTS
883 003572 016767 175040 175052 MOV STALL,TEMP1 ;SET READY STALL
884 003600 005001 CLR R1
885 003602 005301 EXEC3: DEC R1
886 003604 001376 BNE EXEC3 ;AWAIT INTERRUPT
887 003606 005367 175040 DEC TEMP1
888 003612 001373 BNE EXEC3
889 003614 032777 020000 174776 BIT #20000,ASWR ;SEE IF PRINT ERROR
890 003622 001013 BNE EXECX ;IF NOT: BR
891 003624 005767 175064 TST HDRFL ;SEE IF DONE HEADER
892 003630 001004 BNE EXEC4 ;IF SO: BR
893 003632 016704 175022 MOV EMADDR,R4
894 003636 004767 001150 JSR PC,TTOUT ;PRINT HEADER
895 003642 012704 006637 EXEC4: MOV #MSG11,R4
896 003646 004767 001140 JSR PC,TTOUT ;PRINT NO INTERRUPT MESSAGE
897 003652 005777 174742 EXECX: TST ASWR ;SEE IF HALT ON ERROR
898 003656 100001 BPL EXECXX ;IF NOT: BR
899 003660 000000 HALT
900 003662 004767 001736 EXECXX: JSR PC,CKSWR ;TEST FOR IG
901 003666 000240 NOP
902 003670 000207 RTS PC ;RETURN TO CALLER
903

```


				; STATUS ERROR CHECK SUBROUTINE*****	
904					
905					
906	003672	005777	174704	ERCHK:	TST @MTC ;SEE IF ANY ERROR BITS
907	003676	100002			BPL ERCHK1 ;IF NOT: BR
908	003700	000167	000044		JMP ERPT ;ELSE PRINT ERROR
909	003704	005777	174674	ERCHK1:	TST @MTBC ;SEE IF BYTE COUNT IS ZERO
910	003710	001402			BEQ ERCHK2 ;IF SO: BR
911	003712	000167	000032		JMP ERPT ;ELSE PRINT ERROR
912	003716	016703	174756	ERCHK2:	MOV BYTES,R3
913	003722	005403			NEG R3
914	003724	066703	174746		ADD @ADR,R3 ;SET EXPT BUS ADDRESS
915	003730	005767	174734		TST SPFLG ;SEE IF SPACE OPERATION
916	003734	001401			BEQ ERCHK3 ;IF NOT: BR
917	003736	000207			RTS PC
918	003740	020377	174642	ERCHK3:	CMP R3,@MTBA ;SEE IF EXPT=RCVD
919	003744	001001			BNE ERPT ;IF NOT: BR
920	003746	000207			RTS PC ;ELSE EXIT
921	003750	000240		ERPT:	NOP
922	003752	032777	020000 174640		BIT #20000,@SWR ;SEE IF SHOULD PRINT
923	003760	001067			BNE ERPTX ;IF NOT: BR
924	003762	005767	174726		TST HDRFL ;SEE IF DONE HEADER
925	003766	001006			BNE ERPT1 ;IF SO: BR
926	003770	016704	174664		MOV @MADDR,R4
927	003774	004767	001012		JSR PC,TTOUT ;ELSE PRINT HEADER
928	004000	005267	174710		INC HDRFL ;SET FLAG
929	004004	016704	174652	ERPT1:	MOV @ERRAD,R4
930	004010	004767	000776		JSR PC,TTOUT ;PRINT ERROR CODE
931	004014	012704	006541		MOV @MSG5,R4
932	004020	004767	000766		JSR PC,TTOUT ;PRINT MTS TAG
933	004024	017703	174550		MOV @MTS,R3
934	004030	004767	001144		JSR PC,OTPE ;PRINT MTS
935	004034	012704	006550		MOV @MSG6,R4
936	004040	004767	000746		JSR PC,TTOUT ;PRINT MTC TAG
937	004044	017703	174532		MOV @MTC,R3
938	004050	004767	001124		JSR PC,OTPE ;PRINT MTC
939	004054	012704	006557		MOV @MSG7,R4
940	004060	004767	000726		JSR PC,TTOUT ;PRINT BYTE COUNT TAG
941	004064	017703	174514		MOV @MTBC,R3
942	004070	004767	001114		JSR PC,OTPE ;PRINT BYTE COUNT
943	004074	005767	174570		TST SPFLG ;SEE IF PRINT BA
944	004100	001017			BNE ERPTX ;IF NOT: BR
945	004102	012704	006567		MOV @MSG8,R4
946	004106	004767	000700		JSR PC,TTOUT ;PRINT BUS ADDRESS TAG
947	004112	017703	174470		MOV @MTBA,R3
948	004116	004767	001066		JSR PC,OTPE ;PRINT CURRENT ADDRESS
949	004122	016703	174552		MOV BYTES,R3
950	004126	005403			NEG R3
951	004130	066703	174542		ADD @ADR,R3
952	004134	004767	001050		JSR PC,OTPE ;PRINT EXPT ADDRESS
953	004140	005777	174454	ERPTX:	TST @SWR ;SEE IF HALT ON ERROR
954	004144	100001			BPL ERPTXX ;IF NOT: BR
955	004146	000000			HALT
956	004150	004767	001450	ERPTXX:	JSR PC,CKSWR ;TEST FOR 1G
957	004154	000167	000240		JMP SCOPE ;GO SEE IF SCOPE ON ERROR


```

958                                     ;DATA CHECK SUBROUTINE*****
959
960 004160 000240 DCHK: NOP
961 004162 005067 174522 DCHKD: CLR CRCNT ;CLEAR COUNTER
962 004166 121112 BNE DCHKE ;SEE IF EXPT DATA=RCVD DATA
963 004170 001007 DCHK1: INC DCHKE ;IF NOT: BR
964 004172 005267 174512 DCHK1: INC CRCNT ;BUMP CHARACTER COUNTER
965 004176 122122 DCHK1: CMPB (R1)+,(R2)+
966 004200 005300 DEC R0 ;SEE IF DONE
967 004202 001371 BNE DCHKD ;IF NOT: BR
968 004204 000167 000150 JMP DCHKX ;ELSE GO TO EXIT ROUTINE
969 004210 000240 DCHKE: NOP
970 004212 012767 000001 174472 MOV #1,DERFL ;SET ERROR FLAG
971 004220 032777 020000 174372 BIT #20000,DSWR ;SEE IF PRINT ERROR
972 004226 001054 BNE DCHKX ;IF NOT: BR
973 004230 005767 174460 TST HDRFL ;SEE IF DONE HEADER
974 004234 001007 BNE DCHKE1 ;IF SO: BR
975 004236 016704 174416 MOV EMADDR,R4
976 004242 004767 000544 JSR PC,TTOUT ;PRINT HEADER
977 004246 012767 000001 174440 MOV #1,HDRFL ;SET HEADER FLAG
978 004254 012704 006657 DCHKE1: MOV MSG12,R4
979 004260 005767 174432 TST PFLG ;SEE IF PRINTED DATA ERROR TAG
980 004264 001004 BNE DCHKE2 ;IF SO: BR
981 004266 005267 174424 INC PFLG
982 004272 004767 000514 JSR PC,TTOUT ;ELSE PRINT DATA ERROR TAG
983 004276 012704 006673 DCHKE2: MOV MSG13,R4
984 004302 004767 000504 JSR PC,TTOUT ;PRINT CHAR NUMBER TAG
985 004306 016703 174376 MOV CRCNT,R3 ;PRINT CHAR NUMBER
986 004312 004767 000672 JSR PC,OCIP
987 004316 012704 006701 MOV MSG14,R4 ;PRINT GOOD TAG
988 004322 004767 000464 JSR PC,TTOUT
989 004326 111103 MOVB (R1),R3 ;PRINT GOOD CHARACTER
990 004330 004767 001102 JSR PC,DOUT
991 004334 012704 006706 MOV MSG15,R4 ;PRINT BAD TAG
992 004340 004767 000446 JSR PC,TTOUT
993 004344 111203 MOVB (R2),R3 ;PRINT BAD CHARACTER
994 004346 004767 001064 JSR PC,DOUT
995 004352 000240 NOP
996 004354 000167 177612 JMP DCHK1 ;CONTINUE FOR ALL BYTES
997 004360 000240 DCHKX: NOP
998 004362 005767 174324 TST DERFL ;SEE IF ANY ERROR
999 004366 001404 BEQ DCHKXX ;IF NOT: BR
1000 004370 005777 174224 TST DSWR ;SEE IF HALT ON ERROR
1001 004374 100001 BPL DCHKXX ;IF NOT: BR
1002 004376 000000 HALT
1003 004400 004767 001220 DCHKXX: JSR PC,CKSWR ;TEST FOR IG
1004 004404 000240 NOP
1005 004406 005067 174304 CLR PFLG ;CLEAR PRINT FLAG
1006 004412 005067 174274 CLR DERFL ;CLEAR DATA ERROR FLAG
1007 004416 000207 RTS PC ;RETURN

```



```

1008                                     ;SCOPE LOOP ON ERROR SUBROUTINE*****
1009
1010 004420 004767 001200 SCOPE: JSR PC,CKSWF ;TEST FOR IG
1011 004424 000240 174164 BIT #40000,ASWR ;SEE IF LOOP ON ERROR
1012 004426 032777 040000 BNE SCOPE1 ;IF SO: BR
1013 004434 001001 RTS PC ;ELSE EXIT
1014 004436 000207 SCOPE1: NOP
1015 004440 000240 TST (SP)+ ;RESET STACK
1016 004442 005726 NOP
1017 004444 000240 MOV #ALTADD,R3
1018 004446 017703 174212 JMP (R3) ;LOOP ON ERROR
1019 004452 000113
1020
1021                                     ;TEST ITERATION SUBROUTINE*****
1022
1023 004454 000240 ITER: NOP
1024 004456 004767 001142 JSR PC,CKSWF ;TEST FOR IG
1025 004462 032777 010000 174130 BIT #10000,ASWR ;SEE IF ITERATIONS
1026 004470 001403 BEQ ITER1 ;IF SO: BR
1027 004472 005067 174210 ITER0: CLR ITCNT ;CLEAR ITERATION COUNTER
1028 004476 000207 RTS PC ;ELSE EXIT
1029 004500 005267 174202 ITER1: INC ITCNT ;BUMP COUNTER
1030 004504 026767 174176 174122 CMP ITCNT,ITAMT ;SEE IF DONE ALL
1031 004512 001767 BEQ ITER0 ;IF SO: BR
1032 004514 005726 TST (SP)+ ;RESET STACK
1033 004516 017700 174144 MOV #ITRLP,R0 ;SET ITERATION POINTER
1034 004522 000110 JMP (R0) ;GO ITERATE
1035
1036                                     ;MAG TAPE INTERRUPT HANDLER*****
1037
1038 004524 000240 MTINT: NOP
1039 004526 022626 000100 174044 CMP (SP)+,(SP)+ ;RESET STACK POINTER
1040 004530 042777 BIC #100,AMTC ;CLEAR INTERRUPT ENABLE
1041 004536 000240 NOP
1042 004540 000240 NOP
1043 004542 000207 RTS PC ;RETURN TO CALLER
1044
1045                                     ;TTY INTERRUPT HANDLER*****
1046
1047 004544 000240 TTINT: NOP
1048 004546 000240 NOP
1049 004550 000240 NOP
1050 004552 000002 RTI
1051

```


1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102

004554 005067 174072
004560 005000
004562 004767 000152
004566 122767 000215 174054
004574 001005
004576 005767 174050
004602 001446
004604 000167 000066
004610 122767 000260 174032
004616 101402
004620 000167 000076
004624 122767 000270 174016
004632 101002
004634 000167 000062
004640 005267 174006
004644 000241
004646 006100
004650 000241
004652 006100
004654 000241
004656 006100
004660 042767 177770 173762
004666 056700 173756
004672 005301
004674 001332
004676 020002
004700 101402
004702 000167 000014
004706 020300
004710 101402
004712 000167 000004
004716 010015
004720 000207

TTR:
TTR0:

TTR1:

TTR1A:

TTR1B:

TTR2:

TTR3:

TTR4:
TTR5:

```

*****
TTY ENTRY SUBROUTINE:

THIS SUBROUTINE IS USED BY THE TEST CONDITION
ENTRY ROUTINE TO READ THE RESPONSE ENTERED
AT THE TTY AND CHECK THEM FOR LEGALITY AND
LIMITS. ALL RESPONSE MUST BE TYPED IN OCTAL
(0-7) AND MUST FALL WITHIN THE LIMITS SET BY
THE CALLING ROUTINE.
IF AN ENTRY IS ILLEGAL OR OUTSIDE THE LIMITS,
A QUESTION MARK IS TYPED (?) AND THE RESPONSE
MAY BE REENTERED.
ENTRIES MAY NOT EXCEED SIX (6) CHARACTERS AND
MAY BE TERMINATED AT LESS THAN SIX BY TYPING A
CARRIAGE RETURN
*****

CLR      TEMP1      ;CLEAR FIRST CHARACTER FLAG
CLR      RO
JSR      PC,TTIN    ;GO READ CHARACTER
CMPB     #215,TIB    ;SEE IF CR
BNE      TTR1        ;IF NOT: BR
TST      TEMP1       ;SEE IF FIRST CHARACTER
BEQ      TTR5        ;IF SO: BR
JMP      TTR2        ;ELSE GO LOAD VALUE
CMPB     #260,TIB    ;SEE IF CHAR IS LESS THAN 0
BLOS     TTR1A       ;IF NOT: BR
JMP      TTR5        ;ELSE GO TO ERROR
CMPB     #270,TIB    ;SEE IF CHAR IS GREATER THAN 7
BHI      TTR1B       ;IF NOT: BR
JMP      TTR5        ;ELSE GO TO ERROR
INC      TEMP1       ;SET FIRST CHARACTER FLAG
CLC
ROL      RO
CLC
ROL      RO          ;SHIFT 3 LEFT
CLC
ROL      RO
BIC      #177770,TIB ;STRIP ASCII
BIS      TIB,RO      ;LOAD CHARACTER
DEC      R1          ;SEE IF DONE
BNE      TTR0        ;IF NOT: BR
CMP      RO,R2       ;SEE IF EXCEEDED MAXIMUM LIMIT
BLOS     TTR3        ;IF OT: BR
JMP      TTR5        ;ELSE GO TO ERROR
CMP      R3,RO       ;SEE IF BELOW MINIMUM LIMIT
BLOS     TTR4        ;IF NOT: BR
JMP      TTR5        ;ELSE GO TO ERROR
MOV      RO,(R5)     ;LOAD VALUE
RTS      PC          ;EXIT

```



```

1103 ;TTY ENTRY ERROR SUBROUTINE*****
1104
1105 004722 012704 006713 T1NER: MOV #MSG16,R4
1106 004726 004767 000060 JSR PC,TTOUT ;PRINT?
1107 004732 162716 000020 SUB #20,(SP) ;RESET SP TO START OF VALUE ROUTINE
1108 004736 000207 RTS PC ;REDO VALUE ENTRY
1109
1110 ;TTY READ SUBROUTINE*****
1111
1112 004740 005077 173660 TTIN: CLR @TKS
1113 004744 005077 173656 CLR @TKB
1114 004750 005067 173674 CLR TIB
1115 004754 005277 173644 INC @TKS
1116 004760 105777 173640 TTIN1: TSTB @TKS
1117 004764 100375 BPL TTIN1
1118 004766 017767 173634 173654 MOV @TKB,TIB
1119 004774 105777 173630 TTIN2: TSTB @TPS
1120 005000 100375 BPL TTIN2
1121 005002 116777 173642 173622 MOVB TIB,@TPB
1122 005010 000207 RTS PC
1123
1124 ;TTY OUTPUT SUBROUTINE*****
1125
1126 005012 112467 173630 TTOUT: MOVB (R4)+,TOB
1127 005016 122767 000043 173622 CMPB #43,TOB
1128 005024 001452 BEQ TEX
1129 005026 122767 000045 173612 CMPB #45,TOB
1130 005034 001407 BEQ TCRLF
1131 005036 122767 000041 173602 CMPB #41,TOB
1132 005044 001443 BEQ TBELL
1133 005046 004767 000064 JSR PC,TOG
1134 005052 000757 BR TTOUT
1135 005054 112767 000015 173564 TCRLF: MOVB #15,TOB
1136 005062 004767 000050 JSR PC,TOG
1137 005066 012703 000004 MOV #4,R3
1138 005072 005067 173550 TCRLFA: CLR TOB
1139 005076 004767 000034 JSR PC,TOG
1140 005102 005303 DEC R3
1141 005104 001372 BNE TCRLFA ;DO FILLERS
1142 005106 112767 000012 173532 MOVB #12,TOB
1143 005114 004767 000016 JSR PC,TOG
1144 005120 105767 173604 TSTB RDSW
1145 005124 100401 BMI 1$
1146 005126 000731 BR TTOUT
1147 005130 005067 173574 1$: CLR RDSW
1148 005134 000406 BR TEX
1149 005136 105777 173466 TOG: TSTB @TPS
1150 005142 100375 BPL TOG
1151 005144 116777 173476 173460 MOVB TOB,@TPB
1152 005152 000207 TEX: RTS PC
1153 005154 012703 000002 TBELL: MOV #2,R3
1154 005160 012767 000007 173460 TBELA: MOV #7,TOB
1155 005166 004767 177744 JSR PC,TOG
1156 005172 005303 DEC R3
1157 005174 001371 BNE TBELA
1158 005176 000705 BR TTOUT

```


:OCTAL OUTPUT SUBROUTINE*****

```

1159
1160
1161 005200 012767 000001 000226 OCTPE: MOV #1,OFL
1162 005206 000402 BR OCTPE1
1163 005210 005067 000220 OCTP: CLR OFL ;CLEAR FLAG FOR LEADING ZERO
1164 005214 010304 OCTPE1: MOV R3,R4 ;SEE IF NUMBER IS ZERO
1165 005216 001007 BNE OCTP0 ;IF NOT ZERO: BR
1166 005220 005767 000210 TST OFL ;SEE IF PRINT ALL 0
1167 005224 001004 BNE OCTP0 ;IF SO: BR
1168 005226 004767 000162 JSR PC,OCTPG1 ;ELSE PRINT ZERO
1169 005232 000167 000120 JMP OCTP3 ;SPACE AND EXIT
1170 005236 032704 100000 OCTP0: BIT #100000,R4 ;SEE IF MSD = 1
1171 005242 001406 BEQ OCTP1 ;IF NOT: BR
1172 005244 012704 000001 MOV #1,R4
1173 005250 004767 000116 JSR PC,OCTPG ;PRINT 1
1174 005254 000167 000006 JMP OCTP2
1175 005260 005004 OCTP1: CLR R4
1176 005262 004767 000104 JSR PC,OCTPG ;PRINT 0
1177 005266 010304 OCTP2: MOV R3,R4
1178 005270 006004 ROR R4
1179 005272 006004 ROR R4
1180 005274 006004 ROR R4 ;POSITION DIGIT
1181 005276 006004 ROR R4
1182 005300 000304 SWAB R4
1183 005302 004767 000064 JSR PC,OCTPG ;PRINT DIGIT 2
1184 005306 010304 MOV R3,R4
1185 005310 006004 ROR R4
1186 005312 000304 SWAB R4
1187 005314 004767 000052 JSR PC,OCTPG ;PRINT DIGIT 3
1188 005320 010304 MOV R3,R4
1189 005322 006104 ROL R4
1190 005324 006104 ROL R4
1191 005326 000304 SWAB R4
1192 005330 004767 000036 JSR PC,OCTPG ;PRINT DIGIT 4
1193 005334 010304 MOV R3,R4
1194 005336 006004 ROR R4
1195 005340 006004 ROR R4
1196 005342 006004 ROR R4
1197 005344 004767 000022 JSR PC,OCTPG
1198 005350 010304 MOV R3,R4
1199 005352 004767 000014 JSR PC,OCTPG ;PRINT DIGIT 5
1200 005356 012767 000240 173262 OCTP3: MOV #240,T09
1201 005364 004767 177546 JSR PC,T09 ;PRINT SPACE
1202 005370 000207 RTS PC ;EXIT

```



```

1203
1204
1205
1206 005372 042704 177770
1207 005376 001004
1208 005400 005767 000030
1209 005404 001001
1210 005406 000207
1211 005410 005267 000020
1212 005414 052704 000260
1213 005420 010467 173222
1214 005424 004767 177506
1215 005430 010304
1216 005432 000207
1217 005434 000000
1218
1219
1220
1221 005436 005067 173204
1222 005442 012704 000010
1223 005446 110367 173174
1224 005452 105777 173152
1225 005456 100375
1226 005460 132767 000200 173160
1227 005466 001404
1228 005470 012777 000061 173134
1229 005476 000403
1230 005500 012777 000060 173124
1231 005506 006167 173134
1232 005512 005304
1233 005514 001356
1234 005516 000207
1235 005520 016703 173132
1236 005524 000303
1237 005526 004767 177704
1238 005532 016703 173120
1239 005536 004767 177674
1240 005542 000207
1241
1242
1243
1244
1245 005544 013746 000006
1246 005550 013746 000004
1247 005554 012737 005574 000004
1248 005562 022777 177777 173030
1249 005570 001402
1250 005572 000407
1251 005574 022626
1252 005576 012767 000176 173014
1253 005604 012767 000174 173010
1254 005612 012637 000004
1255 005616 012637 000006
1256 005622 000207
1257
1258

;OCTAL PRINT SUBROUTINE*****
OCTPG: BIC #177770,R4
      BNE OCTPG0
      TST OFL
      BNE OCTPG0
      RTS PC
OCTPG0: INC OFL
OCTPG1: BIS #260,R4
      MOV R4,T0B
      JSR PC,T0G
      MOV R3,R4
      RTS PC
OFL: 0 ;FIRST CHAR FLAG

;DATA CHARACTER OUTPUT SUBROUTINE*****
DOUT: CLR T0B
      MOV #10,R4 ;SET NUMBER TO PRINT
      MOV R3,T0B
DOUT1: TSTB
      BPL DOUT1
      BITB #200,T0B
      BEQ DOUT2
      MOV #061,ATPB
      BR DOUT3
DOUT2: MOV #060,ATPB
DOUT3: ROL T0B
      DEC R4
      BNE DOUT1
      RTS PC
DOUTD: MOV TEMP3,R3
      SWAB R3
      JSR PC,DOUT
      MOV TEMP3,R3
      JSR PC,DOUT
      RTS PC

SUSWR: MOV @#6,-(SP) ;SAVE VECTORS
      MOV @#4,-(SP)
      MOV #1$,@#4 ;SET UP FOR TIMEOUT
      CMP #-1,@SWR ;REFERENCE HARDWARE SWITCH REGISTER
      BEQ 2$
      BR 3$
1$: CMP (SP)+,(SP)+ ;ADJUST STACK
2$: MOV #SWREG,SWR ;POINT TO SOFTWARE SWITCH REG
      MOV #DISPREG,DISPLAY ;POINT TO SOFT DISPLAY REG
3$: MOV (SP)+,@#4 ;RESTORE VECTORS
      MOV (SP)+,@#6
      RTS PC

```


1259	005624	022767	000176	172766	CKSWR:	CMP	#SWREG,SWR	:SOFTWARE SWITCH REG PRESENT
1260	005632	001041				BNE	OUT	:NO, GET OUT
1261	005634	105777	172764			TSTB	ATKS	:YES, WAIT FOR
1262	005640	100036				BPL	OUT	:READY, GET CHARACTER
1263	005642	017767	172760	173000		MOV	ATKB,TIB	:AND STRIP OFF
1264	005650	042767	177600	172772		BIC	#177600,TIB	:THE GARBAGE
1265	005656	022767	000007	172764		CMP	#7,TIB	:IS IT A <1G>
1266	005664	001024				BNE	OUT	
1267	005666	012704	007245			MOV	#SCNTG,R4	
1268	005672	004767	177114			JSR	PC,TTOUT	
1269	005676	012704	007251		CNTLU:	MOV	#MSWR,R4	
1270	005702	004767	177104			JSR	PC,TTOUT	
1271	005706	017703	172706			MOV	ASWR,R3	
1272	005712	004767	177262			JSR	PC,OCPE	
1273	005716	012704	007263			MOV	#MNEW,R4	
1274	005722	004767	177064			JSR	PC,TTOUT	
1275	005726	005037	000726			CLR	ASTEMPST	
1276	005732	004767	000002			JSR	PC,\$READ	:GO READ A LINE
1277	005736	000207			OUT:	RTS	PC	:RETURN TO MAIN BODY OF PROGRAM
1278								
1279	005740	005067	172762		\$READ:	CLR	TEMPST	
1280	005744	012767	000007	172752		MOV	#7,COUNT	
1281	005752	004767	176762		1\$:	JSR	PC,TTIN	:GO READ A CHARACTER
1282	005756	042767	177600	172664		BIC	#177600,TIB	:STRIP OFF GARBAGE
1283	005764	122767	000025	172656		CMPB	#25,TIB	:IS IT A <U>
1284	005772	001002				BNE	11\$	:BRANCH IF NOT
1285	005774	005726			3\$:	TST	(SP)+	:POP THE STACK
1286	005776	000737				BR	CNTLU	:START OVER
1287	006000	122767	000012	172642	11\$:	CMPB	#12,TIB	:IS IT A <LF>?
1288	006006	001016				BNE	2\$	:BRANCH IF NOT
1289	006010	005767	172032			TST	46	:TEST FOR ACT-11 HOOKS
1290	006014	001406				BEQ	10\$	:NO HOOKS-<LF> WAS MISTAKE
1291	006016	005726				TST	(SP)+	:POP THE STACK
1292	006020	016716	172022			MOV	46,(SP)	:MAKE ACT-11 ADDRESS
1293	006024	062716	000010			ADD	#10,(SP)	
1294	006030	000742				BR	OUT	:GO TO THAT ADDRESS
1295	006032	012704	007273		10\$:	MOV	#NOACT,R4	:LET OPR KNOW IT
1296	006036	004767	176750			JSR	PC,TTOUT	:AND RETURN
1297	006042	000754				BR	3\$	:FOR ANOTHER TRY
1298	006044	122767	000015	172576	2\$:	CMPB	#15,TIB	:IS IT A <CR>?
1299	006052	001013				BNE	4\$	:BRANCH IF NOT
1300	006054	012767	000200	172646		MOV	#200,RDSW	
1301	006062	004767	176766			JSR	PC,TCRLF	:ECHO IT WITH <LF>
1302	006066	022767	000007	172630		CMP	#7,COUNT	:WAS IT FIRST CHARACTER
1303	006074	001037				BNE	7\$	:CHANGE SWR IF NOT FIRST ONE
1304	006076	005726			8\$:	TST	(SP)+	:POP THE STACK
1305	006100	000716				BR	OUT	:GET OUT
1306	006102	122767	000060	172540	4\$:	CMPB	#60,TIB	
1307	006110	003004				BGT	5\$	
1308	006112	122767	000067	172530		CMPB	#67,TIB	
1309	006120	003005				BGT	6\$	
1310	006122	012704	007345		5\$:	MOV	#SQUEST,R4	
1311	006126	004767	176660			JSR	PC,TTOUT	
1312	006132	000720				BR	3\$	:START OVER IF NOT LEGAL CHARACTER
1313	006134	006367	172566		6\$:	ASL	TEMPST	
1314	006140	006367	172562			ASL	TEMPST	

H03

CZTSFDO TS03 SPLMTL INSTR
CZTSFD.P11 15-FEB-78 14:04

MACY11 30A(1052) 15-FEB-78 14:05 PAGE 33

SEQ 0033

1315	006144	006367	172556		ASL	TEMPST	
1316	006150	142767	000060	172472	BICB	#60,TIB	:GET NITTY-GRITTY
1317	006156	156767	172466	172542	BISB	TIB,TEMPST	
1318	006164	005367	172534		DEC	COUNT	:ONLY WANT 6 DIGITS
1319	006170	001754			BEG	5\$	
1320	006172	000667			BR	1\$	
1321	006174	016777	172526	172416 73:	MOV	TEMPST,DSWR	:CHANGE SWITCH REGISTER CONTENTS
1322	006202	000735			BR	8\$	
1323							

CZTSFDD TS03 SPLMTL INSTR
CZTSFDD.F11 15-FEB-78 14:04

MACY11 30A(1052) 15-FEB-78 14:05 PAGE 34

SEQ 0034

```

1324
1325
1326
1327
1328
1329
1330
1331
1332 006204 005067 172570
1333 006210 105067 172566
1334 006214 105067 172563
1335 006220 105067 172560
1336 006224 105067 172555
1337 006230 005737 000042
1338 006234 001425
1339 006236 005267 172536
1340 006242 032737 020000 000052
1341 006250 001402
1342 006252 000167 000054
1343 006256 023737 000042 000046 6$:
1344 006264 001403
1345 006266 105267 172511
1346 006272 000416
1347 006274 105267 172502 1$:
1348 006300 012777 177777 172312
1349 006306 000410
1350 006310 105737 000041 2$:
1351 006314 001003
1352 006316 105267 172462
1353 006322 000402
1354 006324 105267 172455 3$:
1355 006330 000207 5$:
1356
1357
1358

```

MODIFIED JAN 24 1978

++
CHECK FOR DUMP MODE OR AUTOMATIC/ACT11-XXDP MODE
--

CKMODE: CLR AUTOM : INIT AUTOMATIC MODE INDICATOR
 CLRB ACT11M : INIT ACT11 AUTO MODE INDICATOR
 CLRB XXDPM : INIT XXDP AUTO MODE INDICATOR
 CLRB ADUMPM : INIT ACT11 DUMP MODE INDICATOR
 CLRB XDUMPM : INIT XXDP DUMP MODE INDICATOR
 TST @#42 : AUTO MODE?
 BEQ 2\$: BRANCH - IF NO
 INC AUTOM : SET AUTO MODE INDICATOR
 BIT #20000,@#52 : MANUAL INTERVENTION?
 BEQ 6\$: BRANCH - IF NO
 JMP ABORT : ABORT THE PROGRAM
 CMP @#42,@#46 : ACT11 MODE?
 BEQ 1\$: BRANCH - IF YES
 INCB XXDPM : INDICATE XXDP AUTO MODE
 BR 5\$: AND EXIT
 INCB ACT11M : INDICATE ACT11 AUTO MODE
 MOV #177777,@SWR : SET SWITCH REGISTER
 BR 5\$: AND EXIT
 TSTB @#41 : MAN/MODE VIA ACT11/PAPER TAPE?
 BNE 3\$: BRANCH - IF NOT
 INCB ADUMPM : INDICATE MAN/MODE VIA ACT11/PAPER TAPE
 BR 5\$: AND EXIT
 INCB XDUMPM : INDICATE MANUAL MODE VIA XXDP
 RTS PC : RETURN

J03

CZTSFDD TS03 SPLMTL INSTR
CZTSFDD.F11 15-FEB-78 14:04

MACY11 30A(1052) 15-FEB-78 14:05 PAGE 35

SEQ 0035

1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378006332 000005
006334 012704 006366
006340 004767 176446
006344 105767 172433
006350 001405
006352 013700 000042
006356 005037 000042
006362 004700
006364 000777

```

: *****
:                                     MODIFIED JAN 24 1978
:
: ++
:                                     CHECK FOR DUMP MODE OR AUTOMATIC/ACT11-XXDP MODE
: --
ABORT: RESET
MOV      #MSG0,R4
JSR      PC,TROUT
TSTB     XXDPM
BEQ      1$
MOV      @#42,R0
CLR      @#42
JSR      PC,R0
1$: BR
:
: *****

```

```

; CLEAR THE WORLD
; GET THE MESSAGE
; PRINT ABORT MESSAGE
; XXDP AUTO MODE?
; BRANCH - IF NO
; GET MONITOR EXIT ADDRESS
; USE AS ABORT FLAG
; EXIT TO XXDP MONITOR
; AND HANG

```


K03

CZTSFDD TS03 SPLMTL INSTR
CZTSFD.P11 15-FEB-78 14:04

MACY11 30A(1052) 15-FEB-78 14:05 PAGE 36

SEQ 0036

1379

;MESSAGE TABLE*****

1380										
1381										
1382	006366	022445	051120	043517	MSG0:	.ASCII	/%PROGRAM ABORTED#			
1383	006374	040522	020115	041101						
1384	006402	051117	042524	021504						
1385	006410	022445	055103	051524	MSG1:	.ASCII	/%CZTSFDO TS03 SPLMTL INSTR#			
1386	006416	042106	020060	051524						
1387	006424	031460	051440	046120						
1388	006432	052115	020114	047111						
1389	006440	052123	021522							
1390	006444	042445	052116	051105	MSG2:	.ASCII	/%ENTER UNIT NUMBER: #			
1391	006452	052440	044516	020124						
1392	006460	052516	041115	051105						
1393	006466	021472								
1394	006470	022445	042441	042116	MSG3:	.ASCII	/%!END OF PASS: #			
1395	006476	047440	020106	040520						
1396	006504	051523	020072	043						
1397	006511	045	020441	042522	MSG4:	.ASCII	/%!!REWIND ERROR: NO BOT#			
1398	006516	044527	042116	042440						
1399	006524	051122	051117	020072						
1400	006532	047516	041040	052117						
1401	006540	043								
1402	006541	045	052115	035123	MSG5:	.ASCII	/%MTS: #			
1403	006546	021440								
1404	006550	046445	041524	020072	MSG6:	.ASCII	/%MTC: #			
1405	006556	043								
1406	006557	045	052115	041502	MSG7:	.ASCII	/%MTBC: #			
1407	006564	020072	043							
1408	006567	045	052115	040503	MSG8:	.ASCII	/%MTCA: #			
1409	006574	020072	043							
1410	006577	045	020441	040502	MSG9:	.ASCII	/%!!BACKSPACE ERROR#			
1411	006604	045503	050123	041501						
1412	006612	020105	051105	047522						
1413	006620	021522								
1414	006622	020445	047041	052117	MSG10:	.ASCII	/%!!NOT READY#			
1415	006630	051040	040505	054504						
1416	006636	043								
1417	006637	045	020441	047516	MSG11:	.ASCII	/%!!NO INTERRUPT#			
1418	006644	044440	052116	051105						
1419	006652	052522	052120	043						
1420	006657	045	040504	040524	MSG12:	.ASCII	/%DATA ERROR#			
1421	006664	042440	051122	051117						
1422	006672	043								
1423	006673	045	047103	020072	MSG13:	.ASCII	/%CN: #			
1424	006700	043								
1425	006701	045	035107	021440	MSG14:	.ASCII	/%G: #			
1426	006706	041045	020072	043	MSG15:	.ASCII	/%B: #			
1427	006713	077	043		MSG16:	.ASCII	/%?#			
1428	006715	045	051127	052111	MSG17:	.ASCII	/%WRITE ERROR#			
1429	006722	020105	051105	047522						
1430	006730	021522								
1431	006732	051045	040505	020104	MSG18:	.ASCII	/%READ ERROR#			
1432	006740	051105	047522	021522						
1433	006746	047045	020117	050117	MSG19:	.ASCII	/%NO OPI IN 10.5 FEET#			
1434	006754	020111	047111	030440						
1435	006762	027060	020065	042506						

MO3

CZTSFDO TS03 SPLMTL INSTR
CZTSFD.P11 15-FEB-78 14:04

MACY11 30A(1052) 15-FEB-78 14:05 PAGE 38

SEQ 0038

1436	006770	052105	043		
1437	006773	045	050117	020111	MSG20: .ASCII /%OPI WITHIN 4.0 FEET#
1438	007000	044527	044124	047111	
1439	007006	032040	030056	043040	
1440	007014	042505	021524		
1441	007020	020440	047041	052117	MSG21: .ASCII / !!NOT AVAILABLE#
1442	007026	040440	040526	046111	
1443	007034	041101	042514	043	
1444	007041	045	042522	044507	MSG22: .ASCII /%REGISTER START: #
1445	007046	052123	051105	051440	
1446	007054	040524	052122	020072	
1447	007062	043			
1448	007063	045	042526	052103	MSG23: .ASCII /%VECTOR: #
1449	007070	051117	020072	043	
1450					
1451					;TEST HEADER*****
1452					
1453	007075	045	052045	051505	LT1MSG: .ASCII /%TEST 1: WRITE FROM ODD BYTE#
1454	007102	020124	035061	053440	
1455	007110	044522	042524	043040	
1456	007116	047522	020115	042117	
1457	007124	020104	054502	042524	
1458	007132	043			
1459	007133	045	052045	051505	LT2MSG: .ASCII /%TEST 2: READ TO ODD BYTE#
1460	007140	020124	035062	051040	
1461	007146	040505	020104	047524	
1462	007154	047440	042104	041040	
1463	007162	052131	021505		
1464	007166	022445	042524	052123	LT3MSG: .ASCII /%TEST 3: OPI TOO LONG#
1465	007174	031440	020072	050117	
1466	007202	020111	047524	020117	
1467	007210	047514	043516	043	
1468	007215	045	052045	051505	LT4MSG: .ASCII /%TEST 4: OPI TOO SHORT#
1469	007222	020124	035064	047440	
1470	007230	044520	052040	047517	
1471	007236	051440	047510	052122	
1472	007244	043			
1473					
1474	007245	045	047536	043	\$CNTG: .ASCII /%IG#
1475	007251	045	020445	051441	\$MSWR: .ASCII /%!!SWR= #
1476	007256	051127	020075	043	
1477	007263	040	047040	053505	\$MNEW: .ASCII / NEW= #
1478	007270	020075	043		
1479	007273	045	047045	020117	\$NOACT: .ASCII /%NO ACT-11 HOOKS <LF> =INVALID --RETRY%#
1480	007300	041501	026524	030461	
1481	007306	044040	047517	051513	
1482	007314	036040	043114	020076	
1483	007322	044475	053116	046101	
1484	007330	042111	026440	051055	
1485	007336	052105	054522	022445	
1486	007344	043			
1487	007345	045	022477	021445	\$QUEST: .ASCII /%?%#
1488					.EVEN
1489					
1490	007352	177777			WDATA: -1
1491		007454			.=.+100

N03

CZTSFDD TS03 SPLMTL INSTR
CZTSFD.P11 15-FEB-78 14:04

MACY11 30A(1052) 15-FEB-78 14:05 PAGE 39

SEQ 0039

1492 007454 000000
1493 000001

RDATA: 0
.END

804

CZTSFDO TS03 SPLMTL INSTR
CZTSFDP11 15-FEB-78 14:04MACY11 30A(1052) 15-FEB-78 14:05 PAGE 41
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0040

ABORT	006332	1342	1367*										
ACT11M	001002	362*	476	526	545	560	603	622	1333*	1347*			
ADUMPM	001004	364*	1335*	1352*									
AUTOM	001000	361*	1332*	1339*									
BADR	000676	436*	650*	664*	695*	709*	738*	750*	783*	795*	876	914	951
BCNT	000722	446*											
BKSP	003356	657	702	744	789	843*							
BKSP1	003402	847*	848										
BYTES	000700	437*	651*	665*	696*	710*	739*	751*	784*	796*	877	912	949
CCNT	000614	408*											
CKMODE	006204	475	1332*										
CKSWR	005624	585	624	837	873	900	956	1003	1010	1024	1259*		
CNTLU	005676	540	1269*	1286									
COUNT	000724	447*	1280*	1302	1318*								
CRCNT	000710	441*	961*	964*	985								
DCHK	004160	673	718	960*									
DCHKE	004210	963	969*										
DCHKE1	004254	974	978*										
DCHKE2	004276	980	983*										
DCHKX	004360	968	972	997*									
DCHKXX	004400	999	1001	1003*									
DCHKO	004166	962*	967										
DCHK1	004172	964*	996										
DERFL	000712	442*	970*	998	1006*								
DISPLA	000622	411*	1253*										
DISPRE	000174	380*	1253										
DOUT	005436	990	994	1221*	1237	1239							
DOUTD	005520	1235*											
DOUT1	005452	1224*	1225	1233									
DOUT2	005500	1227	1230*										
DOUT3	005506	1229	1231*										
DRIVE	000040	331*											
EMADDR	000660	429*	640*	686*	734*	779*	822	866	893	926	975		
ERCHK	003672	655	669	700	714	742	787	800	851	906*			
ERCHK1	003704	907	909*										
ERCHK2	003716	910	912*										
ERCHK3	003740	916	918*										
ERPT	003750	760	908	911	919	921*							
ERPTX	004140	923	944	953*									
ERPTXX	004150	954	956*										
ERPT1	004004	925	929*										
ERRAD	000662	430*	652*	666*	699*	713*	741*	753*	785*	798*	850*	929	
EXEC	003442	653	667	697	711	740	754	786	799	858*			
EXECX	003652	890	897*										
EXECXX	003662	898	900*										
EXEC0	003446	860*	863										
EXEC1	003500	865	868*										
EXEC2	003520	861	871	873*									
EXEC3	003602	885*	886	888									
EXEC4	003642	892	895*										
FUN	000704	439*	649*	663*	694*	708*	737*	752*	782*	797*	879		
HDRFL	000714	443*	579*	864	891	924	928*	973	977*				
HERE	002016	612	618*										
ITAMT	000634	416*	762*	764*	803*	805*	1030						
ITCNT	000706	440*	1027*	1029*	1030								
ITER	004454	674	719	763	804	1023*							

C04

CZTSFDO TS03 SPLMTL INSTR
CZTSFDO.F11 15-FEB-78 14:04MACY11 30A(1052) 15-FEB-78 14:05 PAGE 42
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0041

ITER0	004472	1027#	1031									
ITER1	004500	1026	1029#									
ITRLP	000666	432#	574*	575*	600*	601*	1033					
LTADD	000664	431#	572*	573*	574	576	580	598*	599*	600	1018	
LT1	002064	455	456	639#								
LT1B	002106	644#	647									
LT1C	002167	654#										
LT1D	002210	660#	662									
LT1E	002254	668#										
LT1MSG	007075	640	1453#									
LT2	002312	457	458	685#								
LT2B	002330	689#	692									
LT2C	002374	698#										
LT2D	002432	705#	707									
LT2E	002470	712#										
LT2MSG	007133	686	1459#									
LT3	002534	459	733#									
LT3A	002554	737#	746									
LT3B	002602	741#										
LT3C	002702	755#										
LT3IT	002632	460	747#									
LT3MSG	007165	734	1464#									
LT3X	002740	758	762#									
LT4	002764	461	778#									
LT4A	003004	782#	791									
LT4B	003040	787#										
LT4C	003132	800#										
LT4IT	003062	462	792#									
LT4MSG	007215	779	1468#									
MEDIUM	000041	335#										
MSG0	006366	1368	1382#									
MSG1	006410	483	1385#									
MSG10	006622	868	1414#									
MSG11	006637	895	1417#									
MSG12	006657	978	1420#									
MSG13	006673	983	1423#									
MSG14	006701	987	1425#									
MSG15	006706	991	1426#									
MSG16	006713	1105	1427#									
MSG17	006715	652	699	741	785	1428#						
MSG18	006732	666	713	1431#								
MSG19	006746	753	1433#									
MSG2	006444	514	1390#									
MSG20	006773	798	1437#									
MSG21	007020	549	1441#									
MSG22	007041	485	1444#									
MSG23	007063	494	1448#									
MSG3	006470	607	1394#									
MSG4	006511	824	1397#									
MSG5	006541	826	931	1402#								
MSG6	006550	830	935	1404#								
MSG7	006557	939	1406#									
MSG8	006567	945	1408#									
MSG9	006577	850	1410#									
MTBA	000606	405#	876*	918	947							
MTBC	000604	404#	845*	877*	909	941						

SEQ 0042

[illegible]

E04

CZTSFDO TS03 SPLMTL INSTR
CZTSFD.P11 15-FEB-78 14:04

MACY11 30A(1052) 15-FEB-78 14:05 PAGE 44
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0043

TENDX	002032	620	622#											
TEX	005152	1128	1148	1152#										
TIB	000650	425#	1072	1077	1080	1090*	1091	1114*	1118*	1121	1263*	1264*	1265	1282*
		1283	1287	1298	1306	1308	1316*	1317						
TINER	004722	1079	1082	1096	1099	1105#								
TINF	000644	423#												
TKB	000626	413#	1113*	1118	1263									
TKS	000624	412#	1112*	1115*	1116	1261								
TOB	000646	424#	555	1126*	1127	1129	1131	1135*	1138*	1142*	1151	1154*	1200*	1213*
		1221*	1223*	1226	1231*									
TOG	005136	1133	1136	1139	1143	1149#	1150	1155	1201	1214				
TPB	000632	415#	1121*	1151*	1228*	1230*								
TPS	000630	414#	1119	1149	1224									
TSCD	001530	561	566#	591	626									
TSCD0	001560	573#	587											
TSCD1	001614	577	579#	602										
TSCD2	001626	582#	675	720	765	806								
TSCD3	001640	583	585#											
TSTTBL	000732	453#	572	598										
TTIN	004740	1071	1112#	1281										
TTINT	004544	373	1047#											
TTIN1	004760	1116#	1117											
TTIN2	004774	1119#	1120											
TTOUT	005012	484	486	495	515	550	608	823	825	827	831	867	869	894
		896	927	930	932	936	940	946	976	982	984	988	992	1106
		1126#	1134	1146	1158	1268	1270	1274	1296	1311	1369			
		493	502	525	1069#									
TTR	004554	1071#	1093											
TTR0	004562	1073	1077#											
TTR1	004610	1078	1080#											
TTR1A	004624	1081	1083#											
TTR1B	004640	1076	1094#											
TTR2	004676	1095	1097#											
TTR3	004706	1098	1100#											
TTR4	004716	1075	1101#											
TTR5	004720	455#												
T1AD	000736	456#												
T1IAD	000740	457#												
T2AD	000742	458#												
T2IAD	000744	459#												
T3AD	000746	460#												
T3IAD	000750	461#												
T4AD	000752	462#												
T4IAD	000754	406#	516*	517	528*</									

F04

CZTSFD0 TS03 SPLMTL INSTR
CZTSFD.P11 15-FEB-78 14:04MACY11 30A(1052) 15-FEB-78 14:05 PAGE 45
CROSS REFERENCE TABLE -- USER SYMBOLS

SEG 0044

\$NOACT	007273	1295	1479#															
\$QUEST	007345	1310	1487#															
\$READ	005740	1276	1279#															
\$SVPC =	001000	328#	349															
.	007456	319#	328	330#	334#	338#	342#	345#	349#	372#	379#	385#	389#	395#				
		399#	465#	1375	1491#													

. ABS. 007456 000

ERRORS DETECTED: 0

CZTSFD,CZTSFD.SEG/CRF/SOL/NL:TOC=CZTSFD.P11

RUN-TIME: 13.5 SECONDS

RUN-TIME RATIO: 96/6=15.3

CORE USED: 7K (13 PAGES)

DOCUMENT PAGES: 44