

LA36

LA36 DH11 DJ11 DIAG
CZLADD0

AH 8903D MC

NOV 1979

COPYRIGHT 74 79

digital

FICHE 1 OF 1

MADE IN USA

This image shows a microfiche card with a grid of frames. The frames are arranged in approximately 10 columns and 15 rows. Most of the frames are either blank or contain extremely faint, illegible data. The overall appearance is that of a standard microfiche card used for data storage and retrieval.

1
2
3
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
46
47
48
49

.REM @

IDENTIFICATION

PRODUCT CODE: AC-8902D-MC

PRODUCT NAME: CZLADDO LA36 DH11/DJ11 DIAG

PRODUCT DATE: JUNE 1979

MAINTAINER: ENGINEERING

DIAGNOSTIC ENGINEER: ROBERT BAKER

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 RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C): 1974,1979 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	DECX/11

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

TABLE OF CONTENTS

2.0	REQUIREMENTS
2.1	EQUIPMENT
2.2	STORAGE
2.3	PRELIMINARY PROGRAMS
3.0	LOADING PROCEDURE & INITIALIZATION
4.0	STARTING PROCEDURE
4.1	STARTING ADDRESS
5.0	OPERATING PROCEDURE
5.1	SWITCH REGISTER CONTROL
5.2	KEYBOARD CONTROL
5.3	CONTROL ^G
6.0	TEST DESCRIPTIONS
6.1	PRINTING TESTS
6.1.1	TEST0 - DATA PATH TEST
6.1.2	TEST1 - PRINTABLE CHARACTER TEST
6.1.3	TEST2 - NON-PRINTABLE CHARACTER TEST
6.1.4	TEST3 - CARRIAGE RETURN TEST
6.1.5	TEST4 - MULTIPLE LINE FEED TEST
6.1.6	TEST5 - SINGLE LINE FEED TEST
6.1.7	TEST6 - BACKSPACE TEST
6.1.8	TEST7 - OVERPRINT TEST
6.1.9	TEST10 - PRINTING FREQUENCY SWEEP TEST
6.1.10	TEST11 - RIBBON FEED TEST
6.1.11	TEST12 - PRINTER BELL TEST
6.1.12	TEST17 - LIFE TEST
6.2	ECHO TESTS
6.2.1	TEST20 - CHARACTER ECHO TEST
6.2.2	TEST21 - LINE ECHO TEST, FAST RATE
6.2.3	TEST22 - LINE ECHO TEST, SLOW RATE
6.2.4	TEST23 - CHARACTER/CODE ECHO TEST
6.2.5	TEST24 - SELECTED PATTERN ECHO TEST
6.2.6	TEST25 - BELL ECHO TEST
6.3	OPTION TESTS
6.3.1	TEST30 - SECONDARY CHARACTER SET OPTION
6.3.2	TEST31 - SELECTIVE ADDRESSING OPTION
6.3.3	TEST32 - AUTO ANSWER BACK OPTION
6.3.4	TEST33 - TOP OF FORM OPTION
6.3.5	TEST34 - HORIZONTAL TAB OPTION

106

6.3.6 TEST35 - VERTICAL TAB OPTION

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

1.0 ABSTRACT

THIS DIAGNOSTIC IS DIVIDED INTO FOUR BASIC SECTIONS:

1. A CHECK OF THE CONSOLE TERMINAL INTERFACE LOGIC
2. A CHECK OF THE PRINTING CHARACTERISTICS AND CONTROL LOGIC
3. AN ECHO PORTION DESIGNED TO CHECK THE KEYBOARD AND TO AID IN THE DIAGNOSIS OF TERMINAL PROBLEMS.
4. A CHECK OF THE VARIOUS LA36 OPTIONS.

PATTERNS USED BY THE PRINTING TESTS WERE CHOSEN FOR EASE OF VISUAL VERIFICATION. THE ECHO TESTS WERE DESIGNED FOR MAXIMUM FLEXIBILITY, WITH TEST 24 ALLOWING ANY DESIRED PATTERN TO BE USED.

2.0 REQUIREMENTS

2.1 EQUIPMENT

THE DIAGNOSTIC IS WRITTEN TO RUN ON ALL MODELS OF THE PDP-11 COMPUTER WITH EITHER DH11 OR DJ11 MULTIPLEXER. UP TO 16 MULTIPLEXERS + 256 TERMINALS ARE DRIVEN. THE DIAGNOSTIC IS SET TO TEST THE TERMINALS AT 300 BAUD. TO CHANGE ANY PARAMETERS FOR THE DH11 INTERFACE REFER TO THE DH11 PARAMETER TABLE IN THE LISTING.

2.2 STORAGE

THE DIAGNOSTIC PROGRAM USES ALL OF 4K OF MEMORY WITH EXCEPTION OF THE AREA USED BY THE ABSOLUTE LOADER.

2.3 PRELIMINARY PROGRAMS

ANY APPLICABLE PDP-11 DIAGNOSTICS SHOULD BE RUN ON THE PROCESSOR. IF ANY ERRORS ARE ENCOUNTERED DURING THE INTERFACE CHECK, REFER TO THE APPROPRIATE INTERFACE DIAGNOSTIC FOR FURTHER HELP IN LOCATING THE PROBLEM IF NEEDED.

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

3.0 LOADING PROCEDURE & INITIALIZATION

LOAD THE LA36 DIAGNOSTIC PROGRAM TAPE FOLLOWING NORMAL PROCEDURES. BEFORE STARTING THE PROGRAM, REFER TO THE EXISTING LINE TABLE (ELTAB) AND CLEAR THE PROPER BITS IN THE TABLE TO INDICATE WHICH TERMINALS ARE TO BE TESTED. A DETAILED DESCRIPTION IS CONTAINED IN THE PROGRAM LISTING. ALSO, REFER TO THE DESCRIPTION OF THE ROUTINE 'DLY'. TIME DELAYS USED BY THE PROGRAM ARE A FUNCTION OF THE CPU MODEL AND MEMORY TYPE AND SHOULD BE SET UP BEFORE RUNNING THE DIAGNOSTIC. THE ROUTINE IS PRESET FOR A PDP-11/40 WITH CORE MEMORY.

IF A HARDWARE SWITCH REGISTER DOES NOT EXIST, THE PROGRAM WILL USE THE CONTENTS OF LOCATION 176 AS THE VALUE OF THE SWITCHES. THE PROGRAM WILL PRINT OUT THE PRESENT CONTENTS OF THE SOFTWARE SWITCH REGISTER WHEN THE PROGRAM IS STARTED. IT WILL THEN ASK FOR THE NEW CONTENTS TO BE INPUT TO THE SOFTWARE SWITCH REGISTER. TYPE CARRIAGE RETURN TO FINISH INPUT.

4.0 STARTING PROCEDURE

4.1 STARTING ADDRESSES

200(8) = EXECUTE WITH DH11 MULTIPLEXER
204(8) = EXECUTE WITH DJ11 MULTIPLEXER

4.1.1 EXECUTE WITH DH11 MULTIPLEXER

- A. REFER TO SECTION 3.0 AND MAKE SURE THE PROPER BITS IN THE ELTAB TABLE INDICATING WHAT TERMINALS ARE TO BE TESTED HAVE BEEN CLEARED AND THE CORRECT DELAY COUNT FOR THE CPU AND MEMORY TYPE IN USE HAS BEEN SET IN TIMER.
- B. SET SWITCH REGISTER = 200(8) AND PRESS THE LOAD ADDRESS SWITCH
- C. SET THE SWITCH REGISTER BITS 7-0 EQUAL TO THE PAPER WIDTH IN TERMS OF THE NUMBER OF COLUMNS (OCTAL). REFER TO SECTION 5.1.4.
- D. SET SWITCH 8 UP IF IT IS DESIRED TO SELECT A SPECIFIC TEST RATHER THAN BEGIN THE NORMAL PRINTING TEST SEQUENCE. OTHERWISE, LEAVE SWITCH 8 DOWN.
- E. PRESS THE START SWITCH. IF BIT 8 WERE ZERO WHEN STARTING THE NUMBER OF DH11'S UNDER TEST WILL BE PRINTED ON ALL EXISTING TERMINALS AND THE PRINTER TESTS ARE EXECUTED SEQUENTIALLY.

208
209
210
211
212
213
214
215

PAGE 5

F. IF BIT 8 WERE 1 WHEN STARTING, THE NUMBER OF DH11'S UNDER TEST WILL BE INDICATED AND THE MESSAGE "SELECT TEST NUMBER" WILL BE PRINTED ON ALL EXISTING TERMINALS. THE PROGRAM WILL THEN BE WAITING FOR A TEST SELECTION VIA ANY TERMINAL KEYBOARD (IF SWITCH 13 IS DOWN), REFER TO SECTION 5.2

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

4.1.2 EXECUTE WITH DJ11 MULTIPLEXER

- A. SAME INSTRUCTIONS AS 4.1.1 EXCEPT THAT THE STARTING ADDRESS IN B IS 204 AND THE FIRST MESSAGE PRINTED WILL BE THE NUMBER OF DJ11'S UNDER TEST.

5.0 OPERATING PROCEDURE

THE PROGRAM IS GENERALLY CONTROLLED FROM A MULTIPLEXER TERMINAL, BUT A FEW SWITCH REGISTER CONTROLS ARE AVAILABLE. THE PRINTER TEST WILL OUTPUT TO ALL TERMINALS OR TO THE ONE UNDER TEST AS A FUNCTION OF SR BIT 13. ECHO TESTS WILL REFERENCE ONLY THE TERMINAL SELECTING THE TEST, OR ALL TERMINALS DEPENDING ON THE SPECIFIC TEST AND THE SETTING OF SWITCH 13.

5.1 SWITCH REGISTER CONTROL

THE VARIOUS SWITCHES AND THEIR FUNCTIONS ARE LISTED BELOW. SWITCHES MAY BE CHANGED AND SET AS DESIRED EXCEPT AS NOTED IN THE SPECIFIC SWITCH DESCRIPTIONS. REFER TO THE DETAILED SWITCH DESCRIPTIONS FOR FURTHER, MORE COMPLETE INFORMATION.

SWITCH NUMBER	DESCRIPTION
15	1(UP) = HALT AT END OF TEST 0(DOWN) = CONTINUE TEST SEQUENCE
13	1(UP) = DRIVE ONLY SELECTED TERMINAL 0(DOWN) = DRIVE ALL TERMINALS
8	1(UP) = SELECT TEST (AT START-UP ONLY) 0(DOWN) = START NORMAL TEST SEQUENCE
7-0	NUMBER OF COLUMNS AT START-UP

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

5.1.1 SWITCH 15

WITH SWITCH 15 IN THE UP POSITION, THE PROGRAM WILL HALT AT THE END OF THE CURRENT TEST. REPLACING SWITCH 15 TO THE DOWN POSITION AND PRESSING CONTINUE WILL CONTINUE THE NORMAL TEST OPERATION.

5.1.2 SWITCH 13

PLACING SWITCH 13 IN THE DOWN POSITION WILL CAUSE THE DRIVING OF ALL TERMINALS. IF SWITCH 13 IS UP, ONLY THE TERMINAL UNDER TEST IS DRIVEN.

NOTE

SWITCH 13 CAN ONLY BE CHANGED WHEN THE PROGRAM IS WAITING FOR A TEST SELECTION.

5.1.3 SWITCH 8 (AT START-UP ONLY)

TO SELECT A SPECIFIC TEST RATHER THAN START THE PRINTING TEST SEQUENCE, PLACE SWITCH 8 UP BEFORE STARTING THE DIAGNOSTIC. OTHERWISE, LEAVE SWITCH 8 DOWN TO RUN THE NORMAL TEST SEQUENCE. THIS SWITCH IS ONLY EFFECTIVE AT START-UP OF THE PROGRAM.

5.1.4 SWITCHES 7 TO 0 (AT START-UP ONLY)

AT START-UP ONLY, SWITCHES 7 TO 0 ARE USED TO SET THE DESIRED MAXIMUM NUMBER OF COLUMNS THE DIAGNOSTIC IS TO TEST. IF THE NUMBER SET IS GREATER THAN 132(10) OR LESS THAN 30(10), THE PROGRAM WILL DEFAULT TO 132(10). THE VALUE SET MUST BE IN OCTAL FORM.

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

5.2 KEYBOARD CONTROL

THE PROGRAM WILL ALWAYS BE UNDER KEYBOARD CONTROL. CONTROL FROM THE SWITCH REGISTER DURING PROGRAM EXECUTION IS ONLY POSSIBLE WITH SWITCHS 13 AND 15 AS STATED ABOVE.

TYPING THE "RUBOUT" (DEL) KEY ON ANY TERMINAL KEYBOARD WILL TERMINATE THE TEST IMMEDIATELY. AFTER TERMINATION OF THE TEST THE FOLLOWING MESSAGE WILL BE TYPED:

SELECT TEST

AT THIS TIME, TYPE THE DESIRED TEST NUMBER FOLLOWED BY ANY ONE OF THE FOLLOWING CONTROL CHARACTERS:

- . (PERIOD) - RUN THE SELECTED TEST ONCE AND RETURN FOR ANOTHER TEST SELECTION.
- L = LOOP ON THE SELECTED TEST UNTIL A "RUBOUT" IS TYPED.
- S - START THE TEST SEQUENCE WITH THE SELECTED TEST. CONTINUE TO LOOP ON THE PRINTING TEST SEQUENCE UNTIL A "RUBOUT" IS TYPED.

THE L OR S MAY BE EITHER UPPER OF LOWER CASE, BUT THE TEST NUMBER MUST ALWAYS BE A 2 DIGIT OCTAL NUMBER. FOR ALL ECHO TESTS, THE "L" AND "S" WILL ONLY RUN THE TEST ONCE (THE SAME AS IF TYPING A PERIOD). FOR ALL OPTION TESTS, THE "S" WILL ONLY RUN THE TEST ONCE (THE SAME AS IF TYPING A PERIOD), HOWEVER, TYPING AN "L" WILL CAUSE THE PROGRAM TO LOOP ON THE SELECTED TEST. IF AN ERROR IS DETECTED IN THE TEST SELECTION (ILLEGAL TEST NUMBER OR CONTROL CHARACTER) A QUESTION MARK IS PRINTED AND THE MESSAGE WILL BE REPEATED.

5.3 ^G FUNCTION

THE PROGRAM WILL SENSE ^G COMMANDS WHEN STARTED AT LOCATION 200 OR 204. CONSOLE SWITCH REGISTER VALUES WILL BE PRINTED ON THE TTY AT START UP TIME AND THE NEW VALUE ENTERED BY OPERATOR. TO CHANGE A VALUE, TYPE ^G. AT THE END OF A TEST THE NEW VALUE CAN BE ENTERED. ^U ALLOWS INCORRECT DATA TO BE DELETED AND A NEW ENTRY STARTED. ALL INPUTS ARE TERMINATED BY CARRIAGE RETURN.

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

EXAMPLE:

```
...  
!!! AAA AAA  
.... BBB BBB  
### CCC CCC  
$$$ DDD DDD  
%% EEE EEE  
888 FFF FFF  
''' GGG GGG  
((( HHH HHH  
))) III III  
*** JJJ JJJ  
+++ KKK KKK  
... LLL LLL  
--- MMM MMM  
... NNN NNN  
/// OOO OOO  
000 PPP PPP  
111 QQQ QQQ  
222 RRR RRR  
333 SSS SSS  
444 TTT TTT  
555 UUU UUU  
666 VVV VVV  
777 WWW WWW  
888 XXX XXX  
999 YYY YYY  
::: ZZZ ZZZ  
::: [[[  
<<< \\ \\ \\  
-=- ]]]  
>>>  
???
```

6.1.3 TEST 2 - NON-PRINTABLE CHARACTER TEST

THIS TEST CHECKS ALL NON-PRINTABLE CHARACTERS THAT HAVE NO CONTROL FUNCTION IN THE LA36 TERMINAL OR THE LA36 OPTIONS (SUCH AS CR, LF, BS, & BEL). FIRST THE ASCII CODE WILL BE PRINTED FOLLOWED BY THE MNEMONIC AFTER A FEW SEPARATING SPACES. FOLLOWING THE MNEMONIC, THE ACTUAL CONTROL CHARACTER WILL BE SENT THREE TIMES AND NOTHING SHOULD HAPPEN AT THE PRINTER. THIS PATTERN IS REPEATED, THREE TIMES ON A LINE, UNTIL ALL OF THE NON-PRINTING CHARACTERS HAVE BEEN TESTED.

WITH THE AUTO LINE FEED OPTION SET TO PRODUCE AN AUTOMATIC LINE FEED AFTER EVERY RECEIVED CARRIAGE RETURN, THERE WILL BE A BLANK LINE BETWEEN EACH PRINTED LINE.

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

EXAMPLE:

000	NUL	001	SOM	002	STX
006	ACK	020	DLE	021	DC1
022	DC2	023	DC3	024	DC4
025	NAK	026	SYN	027	ETB
030	CAN	031	EM	032	SUB
034	FS	035	GS	036	RS
037	US	177	DEL		

6.1.4 TEST 3 - CARRIAGE RETURN TEST

THIS TEST CHECKS THE CARRIAGE RETURN FROM ALL EVEN NUMBERED COLUMNS AND THE SPACING OF THE SOLENOID HEAD FROM THE LEFT MARGIN. IT IS ALSO A GOOD CHECK FOR PROPER OPERATION OF THE POSITION DECODER.

THE TEST PRINTS A FULL LINE OF ALTERNATING O'S AND SPACES, STARTING WITH A 0. AT THE END OF THE LINE THE PRINT HEAD IS RETURNED TO THE LEFT MARGIN WITH A CARRIAGE RETURN. THE SPACES ARE THEN FILLED IN BY SPACING THE PRINT HEAD OUT FROM THE LEFT MARGIN TO THE FIRST SPACE, PRINTING AN 'X', AND EXECUTING A CARRIAGE RETURN. THIS PATTERN IS REPEATED UNTIL THE LINE IS COMPLETED. CHECK TO SEE THAT ALL X'S ARE IN THE MIDDLE OF THE SPACE BETWEEN THE TWO ZEROES ON EITHER SIDE OF IT.

EXAMPLE:

OXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOX

WITH THE AUTO LINE FEED OPTION SET TO PRODUCE AN AUTOMATIC LINE FEED AFTER EVERY RECEIVED CARRIAGE RETURN, THIS TEST WILL PRINT A LINE OF O'S AND SPACES, THEN PRINT A DIAGINAL LINE OF X'S. TO CORRECTLY CHECK THE ENCODER, THE AUTO LINE FEED OPTION SHOULD BE DISABLED.

EXAMPLE:

```
0 0 0 0 0 0 0 0 0
 X
  X
   X
    X
     X
      X
       X
        X
```

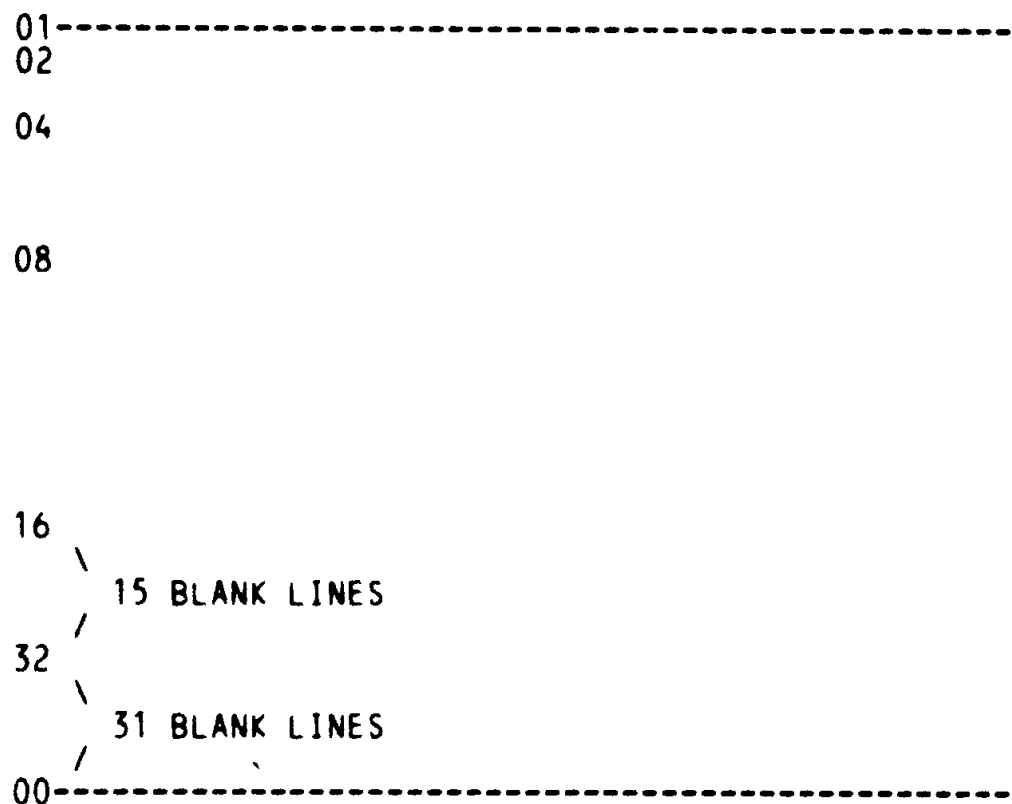
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554

6.1.5 TEST 4 - MULTIPLE LINE FEED TEST

THIS TEST CHECKS THE LINE FEED CAPABILITY OF THE PRINTER BY SENDING VARIOUS GROUPS OF LINE FEEDS INTERSPACED WITH REFERENCE LINES. THE NUMBER PRINTED AS THE REFERENCE LINE INDICATES THE NUMBER OF LINE FEEDS THAT FOLLOW. THE FIRST AND LAST LINES ALSO CONTAIN A STRING OF DASHES AS REFERENCE POINTS FOR MEASURING, THE TOTAL DISTANCE IS 63(10) LINES BETWEEN THE TWO DASHED LINES.

WITH THE AUTO LINE FEED OPTION SET TO PRODUCE AN AUTOMATIC LINE FEED AFTER EVERY CARRIAGE RETURN, THE NUMBER PRINTED WILL INDICATE ONE LESS THAN THE NUMBER OF LINE FEEDS (THE NUMBER OF BLANK LINES) THAT FOLLOW. THE TOTAL DISTANCE BETWEEN THE TWO DASHED LINES WILL THEN BE 69 LINES.

EXAMPLE:



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

PAGE 13

6.1.6 TEST 5 - SINGLE LINE FEED TEST

THIS TEST IS DESIGNED TO CHECK THE TIMING OF SINGLE LINE FEEDS AND THE CAPABILITY OF DOING LINE FEEDS IN ALL COLUMNS. TWO REFERENCE LINES ARE USED BY THIS TEST (AND TEST 6) WHICH ALSO CAN BE USED TO EASILY CHECK THE NUMBER OF COLUMNS THE PRINTER IS PRINTING.

THE FIRST REFERENCE LINE CONTAINS 130(10) ZEROES FOLLOWED BY TWO 2'S IF TESTING 132(10) COLUMNS. IF LESS THAN 132 COLUMNS, THE LINE WILL CONTAIN 0'S FOR TWO LESS THAN THE MAXIMUM NUMBER OF COLUMNS FOLLOWED BY THE TWO 2'S. THIS REFERENCE LINE IS A QUICK CHECK FOR 132(10) COLUMNS IF TESTING THE FULL 132(10) COLUMNS. THE SECOND REFERENCE LINE PRINTS A STRING OF NUMBERS (1 TO 9 & 0) REPEATED TO THE MAXIMUM COLUMN. THIS LINE, AGAIN, CAN BE USED AS A QUICK CHECK OF THE NUMBER OF COLUMNS.

THE LINE FEED TEST IS ACCOMPLISHED BY: PRINTING THE FIRST REFERENCE LINE OF 0'S AND TWO 2'S; THEN EITHER SENDING 60(10) 3'S, IF TESTING 132(10) COLUMNS, OR WAITING 1.8 SECONDS FOR AN LCV, IF TESTING LESS THAN 132(10) COLUMNS. IF TESTING 132(10) COLUMNS, NOTHING SHOULD HAPPEN, EXCEPT FOR AN LCV, AT THE END OF THE LINE. THE 3'S SHOULD BE LOST AND NEVER PRINTED. AFTER THE LCV, WITH THE PRINT HEAD AT THE EXTREME RIGHT, A CARRIAGE RETURN - LINE FEED WILL BE SENT FOLLOWED BY REPEATED BACKSLASHES "\" AND LINEFEEDS TO PRINT A DIAGONAL LINE DOWN THE PAPER. WHEN A BACKSLASH IS PRINTED IN THE MAXIMUM COLUMN, A CARRIAGE RETURN WILL BE SENT IMMEDIATELY AFTER THE LINE FEED AND THE SECOND REFERENCE LINE OF SEQUENTIAL NUMBERS WILL BE PRINTED. AFTER COMPLETING THE LINE, A CARRIAGE RETURN - LINE FEED WILL BE SENT AND THE PROGRAM WILL WAIT ONE SECOND FOR THE CARRIAGE RETURN FUNCTION TO COMPLETE. AFTER THE DELAY, THE REFERENCE LINE WILL BE REPEATED, THE LAST LINE BEING GUARANTEED TO BE CORRECT. ANY TIMING PROBLEMS DURING THE LINE FEEDS WILL SHOW AS MISS PRINTS OR MISSING CHARACTERS DURING THE FIRST 16(10) CHARACTERS OF THE MIDDLE REFERENCE LINE. ALSO, ANY PAPER FEED PROBLEMS WILL CAUSE MISS-ALIGNMENT OF THE SLASHES FORMING THE DIAGONAL LINE.

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

EXAMPLE:

000000000000000000000000000022

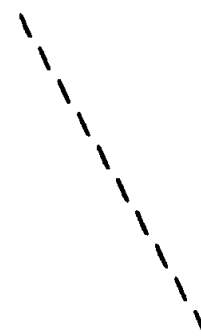
123456789012345678901234567890
123456789012345678901234567890

WITH THE AUTO LINE FEED OPTION SET TO PRODUCE AN AUTOMATIC LINE FEED AFTER EVERY RECEIVED CARRIAGE RETURN, THERE WILL BE A BLANK LINE EVERY PLACE A CARRIAGE RETURN IS EXECUTED.

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

EXAMPLE:

000000022



1234567890

1234567890

6.1.7 TEST 6 - BACKSPACE TEST

THIS TEST IS DESIGNED TO TEST THE PRINT TIMING AS IN TEST 5 AS WELL AS THE BACKWARD AND FORWARD MOVEMENT OF THE PRINT SOLENOID HEAD.

THE TEST CONSISTS OF THE SAME FIRST REFERENCE LINE AS IN TEST 5 THEN A CARRIAGE RETURN-LINE FEED. A FULL LINE IS THEN PRINTED USING THE FOLLOWING PATTERN:

FORWARD SLASH "/"
BACKSPACE
BACK SLASH "\"

THIS PATTERN PRODUCES A LINE OF ALL X'S. THE TWO SLASHES SHOULD CROSS EXACTLY AT THE MIDDLE, PRODUCING THE X CHARACTER. WHEN THE LINE IS COMPLETED A CARRIAGE RETURN-LINE FEED IS SENT AND THE LAST TWO REFERENCE LINES ARE PRINTED AS IN TEST 5. ANY TIMING PROBLEMS WILL SHOW IN THE FIRST 16(10) CHARACTERS OF THE MIDDLE REFERENCE LINE; AGAIN AS IN TEST 5.

WITH THE AUTO LINE FEED OPTION SET TO PRODUCE AN AUTOMATIC LINE FEED AFTER EVERY RECEIVED CARRIAGE RETURN, THERE WILL BE A BLANK LINE BETWEEN EACH PRINTED LINE.

EXAMPLE:

000000000000000000000000000000022
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
123456789012345678901234567890
123456789012345678901234567890

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

6.1.11 TEST 12 - PRINTER BELL TEST

THIS TEST CHECKS THE PRINTER BELL BUFFER TO INSURE THAT EIGHT BELLS ARE DISTINCTLY HEARD, EVEN WHEN SENT AT THE MAXIMUM TRANSFER RATE. THE PROGRAM SENDS 8 BELL CODES AT THE MAXIMUM RATE TO THE PRINTER THEN WAITS 2.5 SECONDS TO ALLOW THE OPERATOR TO HEAR THE BELLS.

6.1.12 TEST 17 - LIFE TEST

THIS TEST RUNS CONTINUOUSLY AND IS RUN AS AN INDIVIDUAL, SPECIAL TEST. IT IS NOT PART OF THE STANDARD PRINTING TEST SEQUENCE.

THIS TEST PRINTS 2 LINES OF EACH PRINTABLE CHARACTER AND THEN REPEATS CONTINUOUSLY. THE SECOND LINE OF EACH CHARACTER IS OVERPRINTED 4 TIMES TO CONSERVE PAPER. AT THE END OF EACH COMPLETE PASS THROUGH THE CHARACTER SET A MESSAGE IS PRINTED INDICATING THE NUMBER OF PASSES EXECUTED. IF ANY CHARACTER (EXCEPT 'RUBOUT') IS TYPED ON THE KEYBOARD DURING THIS TEST, THE PATTERN WILL CHANGE AND RESTART WITH THE TYPED CHARACTER. THIS WILL ONLY HAPPEN IF KEYBOARD CONTROL IS IN USE.

EXAMPLE:

```
AAAAAAAAAAAAAAAAAAAAAAAAAAAA  
AAAAAAAAAAAAAAAAAAAAAAAAAAAA  
BBBBBBBBBBBBBBBBBBBBBBBBBB  
BBBBBBBBBBBBBBBBBBBBBBBBBB
```

IF THE AUTO LINE FEED OPTION IS SET TO PRODUCE AN AUTOMATIC LINE FEED AFTER EVERY RECEIVED CARRIAGE RETURN, THE TEST WILL PRINT SIX LINES OF EACH CHARACTER WITH A BLANK LINE BETWEEN THE FIRST AND SECOND LINES AS WELL AS BETWEEN EACH GROUP OF CHARACTERS.

EXAMPLE:

```
AAAAAAAAAAAAAAAA  
  
AAAAAAAAAAAAAAAA  
AAAAAAAAAAAAAAAA  
AAAAAAAAAAAAAAAA  
AAAAAAAAAAAAAAAA  
AAAAAAAAAAAAAAAA  
  
BBBBBBBBBBBBBBBB  
  
BBBBBBBBBBBBBBBB  
BBBBBBBBBBBBBBBB  
BBBBBBBBBBBBBBBB  
BBBBBBBBBBBBBBBB
```

853

BBBBBBBBBBBBBBBB

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

6.2 ECHO TESTS

THESE TESTS ARE DESIGNED AS A TEST OF THE KEYBOARD AND AN AID IN ISOLATING TROUBLES WITHIN THE TERMINAL. AT THE BEGINNING OF EACH TEST, THE TEST NUMBER WILL BE PRINTED INDICATING WHICH TEST IS BEING EXECUTED. TYPING A 'RUBOUT' OR 'DELETE' AT ANY TIME, WHETHER IN KEYBOARD CONTROL OR NOT, WILL EXIT THE CURRENT ECHO TEST AND PRINT A TEST TERMINATION MESSAGE. IF IN KEYBOARD CONTROL, THE SELECT TEST MESSAGE WILL BE PRINTED AND THE PROGRAM WILL AWAIT A TEST SELECTION AS USUAL. IN SWITCH REGISTER CONTROL, THE PROGRAM WILL HALT (AT SELLHLT) WAITING FOR CONTROL VIA THE SWITCH REGISTER. A DETAILED DESCRIPTION OF EACH TEST FOLLOWS:

6.2.1 TEST 20 - CHARACTER ECHO TEST

THIS TEST IS DESIGNED TO OPERATE THE TERMINAL IN A SIMULATED LOCAL MODE. ANY CHARACTER TYPED ON THE KEYBOARD (EXCEPT A 'RUBOUT') WILL BE ECHOED TO THE PRINTER.

6.2.2 TEST 21 - LINE ECHO TEST, FAST RATE

THIS TEST CONTINUALLY SENDS FULL LINES OF ANY CHARACTER UP TO THE MAXIMUM COLUMN WIDTH. THE TEST PRINTS A 'O' CHARACTER WHEN STARTED UNTIL A KEY IS TYPED ON THE KEYBOARD. THE PROGRAM WILL THEN SEND THE TYPED CHARACTER UNTIL ANOTHER CHARACTER IS TYPED OR THE TEST IS TERMINATED BY TYPING A 'RUBOUT'. THE CHARACTERS ARE TRANSMITTED AT THE MAXIMUM RATE WITH A CARRIAGE RETURN-LINE FEED INSERTED AFTER EVERY 132(10) PRINTABLE CHARACTERS.

IF THE LA36 IS IN HALF DUPLEX WHEN RUNNING THIS TEST, CHARACTERS MAY BE LOST OR GARBLED WHENEVER A CHARACTER IS TYPED ON THE KEYBOARD.

WITH THE AUTO LINE FEED OPTION SET TO PRODUCE AN AUTOMATIC LINE FEED AFTER EVERY CARRIAGE RETURN, THERE WILL BE A BLANK LINE BETWEEN EACH PRINTED LINE.

6.2.3 TEST 22 - LINE ECHO TEST, SLOW RATE

THIS TEST IS IDENTICAL TO TEST 21 EXCEPT A DELAY OF 1.8 SECONDS IS INSERTED BETWEEN EACH CHARACTER TO ALLOW THE PRINT HEAD TO PERFORM AN LCV BETWEEN CHARACTERS.

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

6.2.4 TEST 23 - CHARACTER/CODE ECHO TEST

THIS TEST WILL PRINT THE OCTAL CODE RECEIVED BY THE PROCESSOR FOLLOWED BY THE CHARACTER OR THE MNEMONIC OF THE CHARACTER EVERY TIME A KEY IS PRESSED ON THE KEYBOARD. THE PARITY OF THE RECEIVED CODE WILL BE INDICATED AS EITHER ODD OR EVEN. ALLOW SUFFICIENT TIME BETWEEN CHARACTERS FOR THE LINE TO BE PRINTED.

WITH THE AUTO LINE FEED OPTION SET TO PRODUCE AN AUTOMATIC LINE FEED AFTER EVERY RECEIVED CARRIAGE RETURN, THERE WILL BE A BLANK LINE BETWEEN EACH PRINTED LINE.

EXAMPLE:

301	A	ODD
263	3	ODD
215	CR	EVEN
240	SP	EVEN

6.2.5 TEST 24 - SELECTED PATTERN ECHO TEST

THIS TEST IS DESIGNED TO GIVE MAINTENANCE THE FLEXIBILITY TO CHOOSE THEIR OWN PATTERNS FOR ISOLATING ANY SPECIFIC PROBLEMS WHICH MAY ARISE IN THE FIELD.

TYPE ANY CHARACTERS (EXCEPT CONTROL-C AND RUBOUT) AND EACH CHARACTER WILL BE ECHOED AS TYPED. A MAXIMUM OF 256(10) CHARACTERS MAY BE INPUTTED. NO CARRIAGE RETURNS OR LINE FEEDS ARE INSERTED BY THE PROGRAM, ALL CHARACTERS MUST BE INPUTTED BY THE OPERATOR. TO TERMINATE THE INPUT STRING TYPE A CONTROL-C, THE PROGRAM WILL THEN CONTINUALLY ECHO THE INPUTTED PATTERN. TO STOP THE PRINTING, TYPE CONTROL-C. THE PROGRAM WILL STOP PRINTING THE PATTERN AND WILL WAIT FOR EITHER ANOTHER PATTERN INPUT TERMINATED BY A CONTROL-C, OR THE SAME PATTERN MAY BE USED AGAIN BY TYPING CONTROL-C. TO EXIT THE TEST AT ANY TIME, TYPE A "RUBOUT".

WHEN ANY OPTIONS ARE AVAILABLE, BE CAREFUL WHAT CHARACTERS OR CHARACTER SEQUENCES ARE SELECTED.

6.2.6 TEST 25 - BELL ECHO TEST

THIS TEST IS DESIGNED TO TEST THE BELL ON COLUMN 64 IF TYPING HAS OCCURED ON THE LINE. THE TEST PRINTS A MESSAGE:

TYPE ANY PRINTABLE CHARACTER AND LISTEN FOR BELL

AFTER THE TEST MESSAGE IS PRINTED, TYPE ANY PRINTABLE CHARACTER ON THE KEYBOARD. THE CHARACTER WILL BE ECHOED AND THE BELL SHOULD RING. THE

960
961

MESSAGE WILL THEN BE TYPED AGAIN. TYPE THE 'RUBOUT' KEY TO TERMINATE THE TEST AT ANY TIME.

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

6.3 OPTION TESTS

THESE TESTS ARE DESIGNED AS A TEST OF THE VARIOUS OPTIONS IN WHATEVER COMBINATIONS THEY ARE AVAILABLE IN THE LA36. AT THE BEGINNING OF EACH TEST, THE TEST NUMBER WILL BE PRINTED INDICATING WHICH TEST IS BEING EXECUTED. TYPING A "RUBOUT" OR "DELETE" AT ANY TIME, WHETHER IN KEYBOARD CNTRL OR NOT, WILL EXIT THE CURRENT OPTION TEST. A DETAILED DESCRIPTION OF EACH TEST FOLLOWS:

6.3.1 TEST 30 - SECONDARY CHARACTER SET OPTION

THIS TEST IS DESIGNED TO TEST THE SECONDARY CHARACTER SET OPTION, TESTING THE ABILITY TO SELECT EITHER CHARACTER SET UNDER SOFTWARE CONTROL FROM THE CPU AND PRINTING THE CORRECT CHARACTERS WITHIN EACH CHARACTER SET.

A NUMBER IS PRINTED AT THE LEFT MARGIN INDICATING WHICH CHARACTER SET IS BEING PRINTED, #1 INDICATES THE PRIMARY SET AND #2 INDICATES THE SECONDARY SET (APL). AFTER THE NUMBER, THE APPROPRIATE SHIFT IN (SI) OF SHIFT OUT (SO) WILL BE SENT FOLLOWED BY THE ENTIRE PRINTABLE CHARACTER SET. IF LESS THAN 98 COLUMNS ARE BEING TESTED, A CARRIAGE RETURN - LINE FEED WILL BE INSERTED IN THE APPROPRIATE PLACES. THIS WILL BE REPEATED, ALTERNATING BETWEEN PRIMARY AND SECONDARY SETS UNTIL 16 LINES HAVE BEEN PRINTED (IF USING 98 OR MORE COLUMNS). THERE WILL BE A BLANK LINE BETWEEN EACH PAIR OF LINES TO SEPARATE EACH GROUPING. CHANGE LOCATION "T30SC" AT THE END OF TEST30 TO 377 (8) IF USING 8 BIT SELECTION CODE RATHER THAN THE SI AND SO TO SELECT CHARACTER SETS. THE TEST WILL THEN SET OR CLEAR BIT 8 INSTEAD OF SENDING THE SI OR SO TO SELECT CHARACTER SETS.

WITH THE AUTO LINE FEED OPTION SET TO PRODUCE AN AUTOMATIC LINE FEED AFTER EACH RECEIVED CARRIAGE RETURN, THERE WILL BE EXTRA BLANK LINES EVERY PLACE A CARRIAGE RETURN IS SENT.

EXAMPLE:

#1= !"#S%&'().....PRIMARY CHARACTER SET.....
#2= !"#S%&'().....SECONDARY CHARACTER SET.....

#1= !"#S%&'().....PRIMARY CHARACTER SET.....
#2= !"#S%&'().....SECONDARY CHARACTER SET.....

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

6.3.2 TEST 31 - SELECTIVE ADDRESSING OPTION

THIS TEST IS DESIGNED TO TEST THE VARIOUS FUNCTIONS OF THE SELECTIVE ADDRESSING OPTION. THE TEST FIRST SENDS AN 'EOT' <004> TO DISABLE ALL TERMINALS AND TRIES TO PRINT AN ERROR MESSAGE. THE ERROR MESSAGE SHOULD NOT BE PRINTED ON ANY TERMINAL WITH THE SELECTIVE ADDRESSING OPTION. THEN A 'BEL' <007> AND 'STX' <002> ARE SENT TO SELECT ALL TERMINALS. AT THIS POINT THE TEST NUMBER IS PRINTED ON ALL TERMINALS. THUS, IF AN ERROR MESSAGE IS PRINTED BEFORE THE TEST NUMBER, THE EOT DID NOT DE-SELECT THE TERMINAL WHERE THE MESSAGE WAS PRINTED.

THE TEST NEXT SENDS AN EOT DIRECTLY FOLLOWED BY A STX, WITH NO SELECT CHARACTER. AGAIN, THE ERROR MESSAGE IS SENT TO ALL TERMINALS, WHICH SHOULD NOW BE ALL DE-SELECTED. THE ERROR MESSAGE SHOULD NOT BE PRINTED ON ANY TERMINAL WITH THE SELECTIVE ADDRESSING OPTION.

THE NEXT SERIES OF CHECKS ARE MADE ON THE GROUP SELECT CHARACTER. A TABLE LOCATED AT THE END OF TEST 31 IN THE LISTING IS USED TO TEST VARIOUS GROUP SELECT CHARACTERS. THE FIRST ZERO ENCOUNTERED IN THE TABLE WILL INDICATE THE END OF THE TABLE AND THE TEST WILL GO TO THE NEXT SERIES OF CHECKS ON THE OPTION. THE TABLE IS PRESET WITH A SINGLE GROUP SELECT CHARACTER, THE LETTER 'G', BUT ALLOWS ROOM TO TEST UP TO 8 DIFFERENT SELECT CODES. THIS TABLE SHOULD BE CHANGED TO CONTAIN THE VARIOUS GROUP SELECT CHARACTERS DESIRED TO TEST WITH ONE ASCII CODE PER LOCATION. THE TEST WILL THEN USE THE VARIOUS GROUP SELECT CHARACTERS TO SELECT TERMINALS AND PRINT A MESSAGE ON EACH SELECTED TERMINAL INDICATING THE GROUP SELECT CHARACTER USED. CHECK THAT THE CORRECT GROUP SELECT CHARACTER HAS ENABLED EACH TERMINAL. ALSO, IT MAY BE HELPFUL TO PLACE ;UNUSED SELECT CHARACTERS IN THE TABLE TO CHECK THAT THEY DO NOT SELECT TERMINALS. IF AN ERROR MESSAGE WAS PRINTED BETWEEN THE TEST NUMBER AND THE GROUP SELECT MESSAGE, THE TERMINAL WHERE THE MESSAGE WAS PRINTED WAS SELECTED BY AN EOT AND STX WITH NO SELECT CHARACTER BETWEEN THEM.

THE LAST SERIES OF CHECKS ARE MADE ON THE UNIQUE SELECT CHARACTER. A TABLE LOCATED AT THE END OF TEST 31 IN THE LISTING IS USED TO TEST VARIOUS UNIQUE SELECT CHARACTERS. THE FIRST ZERO ENCOUNTERED IN THE TABLE WILL INDICATE THE END OF THE TABLE. THE PROGRAM WILL SELECT ALL TERMINALS USING THE BEL CODE BEFORE EXITING THE TEST. THE TABLE IS PRESET WITH A SINGLE UNIQUE SELECT CHARACTER, THE LETTER 'U', BUT ALLOWS ROOM TO TEST UP TO 16 DIFFERENT UNIQUE SELECT CODES. THIS TABLE SHOULD BE CHANGED TO CONTAIN THE VARIOUS UNIQUE SELECT CHARACTERS DESIRED TO TEST, WITH ONE ASCII CODE PER LOCATION. MAKE SURE THAT EACH CHARACTER IN THE TABLE IS A VALID UNIQUE SELECT CODE OR THE DIAGNOSTIC WILL HANG DURING THIS PORTION OF THE TEST. USING EACH UNIQUE SELECT CHARACTER IN TURN, THE TEST WILL PERFORM THE REMAINING CHECKS OF THE SELECTIVE ADDRESSING OPTION.

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

THE TEST WILL SEND AN EOT FOLLOWED BY THE CURRENT UNIQUE SELECT CHARACTER. BEFORE THE STX IS SENT, THE TEST WILL TRY TO PRINT THE ERROR MESSAGE ON ALL TERMINALS. THEN THE STX WILL BE SENT AND A MESSAGE WILL BE PRINTED TO INDICATE THE UNIQUE SELECT CHARACTER USED. CHECK THAT THE CORRECT UNIQUE SELECT CHARACTER HAS ENABLED EACH TERMINAL. IF AN ERROR MESSAGE IS PRINTED BEFORE THE UNIQUE SELECT MESSAGE, THE TERMINAL WHERE THE MESSAGE WAS PRINTED WAS ENABLED BEFORE THE STX WAS RECEIVED. A MESSAGE WILL THEN BE PRINTED TELLING THE OPERATOR TO TYPE ANY PRINTABLE CHARACTER TO CHECK THAT THE KEYBOARD IS ENABLED. WHATEVER CHARACTER IS TYPED WILL BE ECHOED TO THE TERMINAL.

THE FINAL SECTION OF THE TEST WILL USE A DUMMY SELECT CHARACTER. THE ASCII CODE FOR THIS SELECT CHARACTER IS LOCATED BETWEEN THE TWO SELECT CHARACTER TABLE AT THE END OF THE TEST. THIS LOCATION SHOULD CONTAIN THE ASCII CODE OF ANY UNUSED SELECT CHARACTER. THE TEST WILL SEND AN EOT FOLLOWED BY THE DUMMY SELECT CHARACTER AND AN STX. THE ERROR MESSAGE WILL BE LOADED TO ALL TERMINALS AND SHOULD NOT BE PRINTED ON ANY TERMINALS SINCE ALL SHOULD BE DE-SELECTED. NEXT AN ETX <003> FOLLOWED BY THE CURRENT UNIQYUE SELECT CHARACTER AND AN STX WILL BE SENT AND A PRINTED MESSAGE WILL INDICATE THE SELECT CHARACTER USED. ANOTHER EXT WILL BE SENT, FOLLOWED BY THE DUMMY SELECT CHARACTER AND AN STX THIS TIME. A MESSAGE WILL AGAIN BE PRINTED INDICATING THE CURRENT UNIQUE SELECT CHARACTER. ALL SELECTED TERMINALS SHOULD REMAIN SELECTED AND NO OTHER TERMINALS SHOULD GET SELECTED.

6.3.3 TEST 32 - ANSWER BACK OPTION

THIS TEST IS DESIGNED TO TEST THAT THE ANSWER BACK OPTION SENDS THE CORRECT MESSAGE UPON RECEIPT OF AN ENQ (005) OR UPON TYPING CONTROL-E OR THE HERE IS KEY ON THE KEYBOARD. THE TEST WILL SEND AN ENQ (005), READ THE MESSAGE, AND THEN PRINT OUT THE MESSAGE ON THE LA36. THE TEST WILL THEN ASK THE OPERATOR TO DEPRESS THE HERE IS KEY, READ THE MESSAGE, AND THEN PRINT OUT THE MESSAGE. FINALLY, THE TEST WILL TELL THE OPERATOR TO DEPRESS THE CONTROL-E KEY, READ THE MESSAGE, AND PRINT OUT THE MESSAGE. IF THE SELECTIVE ADDRESSING OPTION IS AVAILABLE, THE AUTO ANSWER BACK OPTION WILL NOT RESPOND TO ANOTHER ENQ AFTER THE FIRST ONE RECEIVED. THUS, YOU MAY HAVE TO DEPRESS THE RUBOUT KEY TO EXIT THE TEST.

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

6.3.4 TEST 33 - TOP OF FORM OPTION

THIS TEST IS DESIGNED TO TEST THE FORM FEED CAPABILITY OF THE TOP OF FORM OPTION. A SET OF INSTRUCTIONS IS PRINTED FOR THE OPERATOR TO REMIND HIM TO DEPRESS THE TOP OF FORM RESET SWITCH AFTER MAKING EACH SWITCH SETTING. UPON COMPLETION OF EACH SETTING, AFTER DEPRESSING THE RESET SWITCH, TYPE ANY CHARACTER (EXCEPT RUBOUT) ON THE KEYBOARD TO TEST THAT SWITCH SETTING. THE REFERENCE LINES PRINTED WILL INDICATE THE LENGTH FORM FEED JUST EXECUTED AND THE NEXT SWITCH SETTING TO MAKE. THE 3 INCH FORM FEED IS TESTED TWICE BEFORE TESTING THE REMAINING POSITIONS. THE FIRST TIME, 16 OR 17 LINE FEEDS ARE EXECUTED BEFORE DOING THE FORM FEED, DEPENDING ON HOW THE AUTO LINE FEED OPTION IS SET UP. THE DIAGNOSTIC WILL THEN TEST EACH POSITION IN SEQUENCE FROM 3 TO 14 INCHES. THE SINGLE STEP POSITION IS NOT CHECKED.

6.3.5 TEST 34 - HORIZONTAL TAB OPTION

THIS TEST CHECKS THE ABILITY TO SET A TAB IN EVERY COLUMN AND AT PREDETERMINED INTERVALS, AS WELL AS THE ABILITY TO CLEAR ALL TABS. THE PROGRAM SETS A TAB IN THE PREDETERMINED COLUMN, DOES A BACKSPACE, AND PRINTS AN 'O'. AFTER THE LINE IS PRINTED AND THE TABS ARE SET, A CARRIAGE RETURN IS SENT AND THEN THE PRINT HEAD IS POSITIONED USING TABS AND X'S ARE PRINTED OVER THE O'S. SINCE THE FIRST LINE OF THE TEST SETS A TAB IN EVERY COLUMN, THE PRINT HEAD IS TABED ACROSS THE PAGE TWICE TO TEST ALL TABS. THE FIRST PASS CHECKS THE EVEN NUMBERED COLUMNS WHILE THE SECOND PASS CHECKS THE ODD NUMBERED COLUMNS. THE TEST SETS TABS IN EVERY COLUMN, EVERY OTHER COLUMN, AND EVERY 4, 8, 16, 32, 64, 128, & 132 COLUMNS. ALL HORIZONTAL TABS WILL BE CLEARED AT THE END OF THE TEST IF THE TEST IS RUN TO COMPLETION. IF A RUBOUT IS USED TO EXIT THE TEST BEFORE COMPLETION, THE TABS WILL STILL BE SET.

EXAMPLE:

```
0000000000
 0 0 0 0 0
   0 0
    0
```

WHEN THE AUTO LINE FEED OPTION IS SET UP TO PRODUCE AN AUTOMATIC LINE FEED AFTER EVERY RECEIVED CARRIAGE RETURN, THERE WILL BE A BLANK LINE BEFORE EACH REFERENCE LINE OF O'S AND THE X'S WILL BE PRINTED ON THE NEXT LINE UNDER THE O'S. THE FIRST LINE OF O'S WILL HAVE 2 LINES OF X'S UNDER IT, THE FIRST HAVING X'S IN ALL EVEN NUMBERED COLUMNS AND THE SECOND HAVING X'S IN ALL ODD NUMBERED COLUMNS.

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

EXAMPLE:

```
000000000000  
X X X X X X  
X X X X X X  
  
O O O O O O  
X X X X X X  
  
O O O  
X X X  
  
O  
X
```

6.3.6 TEST 35 - VERTICAL TAB OPTION

THIS TEST CHECKS THE VERTICAL TAB OPTION BY TESTING THE ABILITY TO SET TABS IN VARIOUS POSITIONS OF A 14 INCH FORM. AN INSTRUCTION IS PRINTED TELLING THE OPERATOR TO SET A 14 INCH FORM LENGTH AND DEPRESS THE TOP OF FORM RESET SWITCH. WHEN READY, TYPE ANY CHARACTER (EXCEPT RUBOUT) ON THE KEYBOARD TO CONTINUE. THE TEST WILL SEND LINE FEEDS, SET TABS, AND PRINT REFERENCE LINES WHEREVER A TAB IS SET. AT THE END OF THE FORM, A MESSAGE WILL INDICATE TO EITHER REMOVE THE REFERENCE PAGE (WITHOUT TOUCHING THE KEYBOARD) OR RESET THE FIRST REFERENCE LINE. TO RESET THE REFERENCE PAGE IN THE PRINTER, OPEN THE PAPER TRACTORS AND PLACE THE FIRST REFERENCE LINE INFRONT OF THE PRINT HEAD. WHEN READY TO CONTINUE, TYPE ANY CHARACTER (EXCEPT RUBOUT) ON THE KEYBOARD. THE TEST WILL THEN REPRINT THE REFERENCE LINES, USING THE TABS INSTEAD OF LINE FEEDS TO ADVANCE THE PAPER. IF THE FIRST REFERENCE PAGE WAS REMOVED, HOLD IT AGAINST THE SECOND REFERENCE PAGE TO CHECK FOR PROPER PAPER ADVANCING USING TABS. IF THE REFERENCE PAGE WAS RESET IN THE PRINTER, THE SECOND SET OF REFERENCE LINES SHOULD HAVE PRINTED DIRECTLY OVER THE FIRST SET EXCEPT ON THE FIRST LINE WHERE THEY SHOULD BE SIDE-BY-SIDE. ALLOW FOR A SLIGHT VARIANCE IN PAPER POSITION WHEN CHECKING THAT THE REFERENCE LINES ARE CORRECT, LOOK FOR FULL LINE DIFFERENCES. THE TEST PRODUCES 0,1,2,3,4,5,6,7,8,9, & 10 BLANK LINES BETWEEN THE REFERNECES LINES, IN THAT ORDER.

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

:LA36 TERMINAL DIAGNOSTIC
:DH11 AND DJ11 INTERFACES

:AUTHOR: ROBERT W. BAKER

:COPYRIGHT 1974,1979, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754

.SBTTL SWITCH REGISTER OPTIONS

SWITCH	POSITION	FUNCTION
15	UP (1) DOWN (0)	HALT AT END OF CURRENT TEST CONTINUE NORMAL TEST SEQUENCE
13	UP (1) DOWN (0)	DRIVE ONLY SELECTED TERMINAL DRIVE ALL TERMINALS
8	UP (1) DOWN (0)	SELECT TEST (AT START UP ONLY) START NORMAL TEST SEQUENCE
00 - 07		# OF COLUMNS AT START-UP
00 - 05		# OF TEST SELECTION

.SBTTL SPECIAL OPERATIONAL NOTES

:1.-- BEFORE START UP REFER TO THE DESCRIPTION OF THE ROUTINE "DLY"
:TIMING IS A FUNCTION OF THE PDP11 MODEL AND MEMORY TYPE AND
:SHOULD BE SET UP BEFORE RUNNING THE DIAGNOSTIC.

:2.-- THE DIAGNOSTIC WILL NOT RUN UNLESS THE ELTAB TABLE IS MANUALLY INITIALIZED.

:*****

:3.-- ECO TABLE

- :CHGD1 - 3 ADDITIONAL EMT ROUTINES
- :CHGD2 - ADDED DEFINITIONS
- :CHGD3 - NEW START UP PROCEDURE ALLOWS ENTERING SOFT SWITCH REG
- :CHGD4 - FLAG FOR CHECKING ^G FROM KEYBOARD
- :CHGD5 - PREVIOUS DEPO TO ADD 4 FILL CHARACTERS AFTER EA CARRIAGE RET
- :CHGD6 - PART OF DEPO ON 4 FILL CHARACTERS

```
1310 ;CHGD7 - ADDED $CKSWR ROUTINE TO CHECK SWITCH REGISTER, PRINTOUT
1311 ;          CONTENTS, ENTER NEW DATA TO SWITCH REGISTER
1312 ;CHGD8 - NEW PRINT MESSAGES FOR ^G ADDITION
1313 ;CHGD9 - EXTENSION OF CHGD3.  WAIT AND SEE IF ANY DH OR DJ DEVICES
1314 ;CHGD10 - FLAG FOR CHECKING ^G DURING PRINT OPERATION
1315 ;*****
1316 ;          .SBITL EQUATES
1317
1318
1319 ;          REGISTER EQUATES
1320
1321          000000          R0=%0
1322          000001          R1=%1
1323          000002          R2=%2
1324          000003          R3=%3
1325          000004          R4=%4
1326          000005          R5=%5
1327          000006          SP=%6
1328          000007          PC=%7
1329          177776          PSW=177776
1330
1331
1332
1333 ;          SYSTEM EQUATES
1334
1335          000001          BIT0=1
1336          000002          BIT1=2
1337          000004          BIT2=4
1338          000010          BIT3=10
1339          000020          BIT4=20
1340          000040          BIT5=40
1341          000100          BIT6=100
1342          000200          BIT7=200
1343          000400          BIT8=400
1344          001000          BIT9=1000
1345          002000          BIT10=2000
1346          004000          BIT11=4000
1347          010000          BIT12=10000
1348          020000          BIT13=20000
1349          040000          BIT14=40000
1350          100000          BIT15=100000
1351
1352          000000          OPEN=0
1353          022626          POPSP2=22626          ;POP STACK TWICE
1354          000200          ACRLF=200
1355          005726          POPSP=5726          ;POP STACK ONCE
```

```
1356          :      PROGRAM TRAP EQUATES
1357
1358          104000      TYPE=EMT+0
1359          104001      CHAIN=EMT+1
1360          104002      TYPEM=EMT+2
1361          104003      DELAY=EMT+3
1362          104004      TTYCTL=EMT+4
1363          104005      TTYJTL=EMT+5
1364          104006      CRLF=EMT+6
1365          104007      SCRLF=EMT+7
1366          104010      LF=EMT+10
1367          104011      PRINTC=EMT+11
1368          104012      PRTHDR=EMT+12
1369          104013      READ=EMT+13
1370          104014      CR=EMT+14
1371          104015      BTOASC=EMT+15
1372          104016      FORWD=EMT+16
1373          104017      CLEAN=EMT+17
1374          104020      TESTC=EMT+20
1375          104021      ECHO=EMT+21
1376          104022      INRDY=EMT+22
1377          104023      FOUR=EMT+23          ;CHGD1
1378          104024      CKSWR=EMT+24
1379          104025      CNTLU=EMT+25
1380          .SBTTL TRAP CATCHER & STARTING ADDRESSES
1381
1382
1383
1384          .ENABL ABS
1385          .ENABL AMA          ;ENABLE ABSOLUTE ADDRESSING
1386
1387          000000      .=0
1388
1389          000000      000000      HALT
1390          000002      000000      HALT
1391
1392          000004      .=4
1393
1394          000004      MACHER:
1395
1396          000030      .=30
1397
1398          000030      003346      EMTINT
1399          000032      000340      340
1400
1401          000046      .=46
1402
1403          000046      010740      LOGICAL
1404
1405          000052      .=52
1406
1407          000052      010000      010000
1408
1409          000174      .=174
1410
1411          000174      000000      DISPREG:          .WORD 0          ;SOFTWARE DISPLAY REG
```

1412	000176	000000		SWREG:	.WORD	0		:SOFTWARE SWITCH REGISTER
1413								
1414				:STARTING ADDRESSES				
1415								
1416	000200	000137	001276	JMP	START		:RUN	DH11 TESTS
1417	000204	000137	001266	JMP	START1		:RUN	DJ11 TESTS
1418								
1419								
1420		001100			.=1100			
1421								
1422	001100	000000		SPBOT:	0			:BOTTOM OF STACK

.SBTTL PROGRAM INITIALIZATION & CONTROL

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

```
*****  
:PROGRAM START  
:*****  
START1: MOV #177777,IOSW ;SET UP FOR DJ11 TEST  
BR STARTX  
START: CLR IOSW ;SET UP FOR DH11 TESTS  
STARTX: MOV #SPBOT,SP ;SET STACK POINTER  
MOV 6,-(SP) ;SAVE CURRENT VECTORS  
MOV 4,-(SP)  
:*****  
CHGD3: MOV #64$,@#4 ;SET UP LOC 4 FOR NON-EXIS MEM TRAP  
CMP #-1,@SR ;REFERENCE HDWE SW REG  
BEQ 65$ ;IF IT = -1,USE SOFT SW REG  
BR 66$ ;THEN USE HDWE SWITCH REG  
64$: CMP (SP)+,(SP)+ ;CORRECT THE STACK, NO HDWE SW REG  
65$: MOV #SWREG,SR ;POINT TO SOFT SW REG  
MOV #DISPREG,DISPLAY ;POINT TO SOFT SW REG  
66$: MOV (SP)+,@#4 ;RESTORE VECTORS  
MOV (SP)+,@#6 ;SAME  
60$: CLEAN ;CLEAN UP  
CLR CNTLSW ;INITIALIZE TERMINAL CONTROL SWITCH  
MOV #PFAIL,24 ;SET ADDR POWER FAIL ROUTINE  
:*****
```



```
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518 001400 012737 001432 000004      MOV    #END3,MACHER ;INIT TIME OUT TRAP
1519 001406 005037 001226                CLR    DJCNT         ;CLEAR DJ11 COUNTER
1520 001412 012700 160010                MOV    #160010,RO   ;ADDR OF FIRST DJ11
1521 001416 005710                1$:   TST    (RO)      ;REF DJ11
1522 001420 062700 000010                ADD    #10,RO       ;SET RO TO NEXT DJ11
1523 001424 005237 001226                INC    DJCNT        ;INCREMENT COUNT OF DJ11'S
1524 001430 000772                BR     1$           ;TEST PRESENCE OF NEXT DJ11
1525
1526 001432 022626                END3: POPSP2        ; POP 2 FROM STACK
1527 001434 030027 000010                BIT    RO,#10       ;CHECK IF RO IS MULTIPLE OF 20
1528 001440 001402                BEQ    1$           ;SKIP IF YES
1529 001442 062700 000010                ADD    #10,RO       ;MAKE RO FIRST DH11 ADDR
1530 001446 010037 001134                1$:   MOV    RO,DHADR ;SAVE AS FIRST DH11 ADDRES
1531 001452 012737 001500 000004      MOV    #END4,MACHER ;SET TIME OUT TRAP
1532 001460 005037 001230                CLR    DHCNT        ;CLEAR COUNT OF DH11'S
1533 001464 005710                2$:   TST    (RO)      ;TEST IF DH11 IS PRESENT
1534 001466 062700 000020                ADD    #20,RO       ;YES, SET RO TO NEXT DH11
1535 001472 005237 001230                INC    DHCNT        ;INCREMENT COUNT OF DH11'S
1536 001476 000772                BR     2$           ;CHECK IF NEXT ONE IS PRESENT
1537
1538 001500 022626                END4: POPSP2        ;POP 2 FROM STACK
1539
1540
1541 001502 032737 000077 001230      CHGD9: BIT    #77,DHCNT ;ANY DH11'S PRESENT?
1542 001510 001005                BNE    20$         ;IF YES, BRANCH
1543 001512 032737 000077 001226      BIT    #77,DJCNT   ;ANY DJ11'S PRESENT?
1544 001520 001001                BNE    20$         ;IF YES, BRANCH
1545 001522 000000                HALT                ;NO DJ11 OR DH11 AT ADDR 160010 OR 20
1546 001524 022737 000176 001244      20$:  CMP    #SWREG,SR  ;IS SOFT SW REG SELECTED?
1547 001532 001001                BNE    60$         ;IF NOT, SKIP OVER NEXT OPER
1548 001534 104025                CNTLU                ;SOFT SWITCH REG INPUT ROUTINE
1549 001536                60$:
```

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

001536 017701 177502
001542 042701 177400
001546 020127 000204
001552 101003
001554 020127 000035
001560 101002
001562 012701 000204
001566 010137 001112
001572 012700 017654
001576 012702 000003
001602 104015
001604 005037 001106
001610 000401
001612 000405
001614 104002
001616 017442
001620 012737 000240 001610
001626 005737 001224
001632 001440
001634 013701 001226
001640 012700 017613
001644 012702 000002
001650 104015
001652 104006
001654 104002
001656 017613

:READ THE PAPER WIDTH, NUMBER OF COLUMNS,
:FROM SWITCH REGISTER POSITIONS 0-7. SAVE AND
:CONVERT TO 3 ASCII CHARACTERS. A WIDTH GT132
:OR LT30 COLUMNS (DECIMAL) WILL BE ABORTED TO 132.

MOV @SR,R1 ;PUT (SR) INTO R1
BIC #177400,R1 ;SAVE ONLY BITS 0-7
CMP R1,#204 ;TEST NO. COLUMN GT132
BHI 2\$;COLUMNS GT132, DEFAULT TO 132
1\$: CMP R1,#35 ;CHECK IF NO. COLUMNS LT 30
BHI 3\$;NOT LT 30 NOR GT 132
2\$: MOV #204,R1 ;COLUMNS LT 30 OR GT 132, DEFAULT
3\$: MOV R1,WIDTH ;SAVE NO. COLUMNS IN WIDTH
MOV #HDRO,R0 ;ADDR TO STORE ASCII COLUMN VALUE
MOV #3,R2 ;DO A 3 CHAR. CONVERSION
BTOASC ;CONVERT NO. COLUMNS TO ASCII
CLR RTNNO ;SET ROUTINE NO = 0
4\$: BR 5\$;PRINT TITLE FIRST TIME
BR 6\$;SKIP IT AFTER FIRST TIME
5\$: TYPEN ;PRINT DIAGNOSTIC HEADER
STARTM
MOV #NOP,4\$;TAKE OUT BRANCH INSTR
6\$: TST IOSW ;CHECK IF DJ11 OR DH11 UNDER TEST
BEQ 10\$;BRANCH IF DH11
MOV DJCNT,R1 ;GET NUMBER OF DJ11'S
MOV #DJ11S,R0 ;ADDR TO STOR ASCII CHAR
MOV #2,R2 ;NO. OF ASCII CHARS (=2)
BTOASC ;CONVERT TO ASCII
CRLF
TYPEN
DJ11S ;OUTPUT MESSAGE

```
1585  
1586  
1587  
1588  
1589  
1590  
1591 001660 012701 160010  
1592 001664 013703 001226  
1593 001670 001002  
1594 001672 000000  
1595 001674 000776  
1596 001676 052711 000010  
1597 001702 012700 000005  
1598 001706 104003  
1599 001710 031127 000020  
1600 001714 001372  
1601 001716 052711 000001  
1602 001722 062701 000010  
1603 001726 005303  
1604 001730 001362  
1605 001732 000412  
1606 001734 013701 001230  
1607 001740 012700 017564  
1608 001744 012702 000002  
1609 001750 104015  
1610 001752 104006  
1611 001754 104002  
1612 001756 017564  
1613 001760 032777 000400 177256  
1614 001766 001004  
1615 001770 012737 006560 001110  
1616 001776 000431  
1617 002000 104002  
1618 002002 020066  
1619 002004 005037 001222  
1620 002010 005737 001224  
1621 002014 001453  
1622 002016 000436  
1623  
1624  
1625  
1626  
1627  
1628  
1629  
1630  
1631 002020 012737 000006 000004 $CLEAN: MOV #6,MACHER ;SET UP MACHINE ERROR VECTOR  
1632 002026 005066 000002 CLR 2(SP) ;CLEAR PROCESSOR STATUS WORD  
1633 002032 000002 RTI ;RETURN
```

: START ALL DJ11 RECEIVER SCANNERS

7\$: HALT ;DO NOT CONTINUE
8\$: BR 7\$;CLEAR MOS
9\$: MOV #5,R0 ;DELAY TIME TO R0
DELAY
BIT (R1),#20 ;TEST IF CLEAR STILL BUSY
BNE 9\$;BRANCH IF BUSY
BIS #1,(R1) ;START SCANNER
ADD #10,R1 ;ADDR OF NEXT SCR
DEC R3 ;DEC COUNT OF DJ11
BNE 8\$;NOT ZERO, START NEXT ONE
BR 11\$
10\$: MOV DHCNT,R1 ;NO. OF DH11'S TO R1
MOV #DH11S,R0 ;ADDR TO STORE ASCII CHAR
MOV #2,R2 ;NO. OF CHAR TO CONVERT
BTOASC ;CONVERT TO ASCII
CRLF
TYPEM ;OUTPUT MESSAGE
DH11S
11\$: BIT #BIT8,@SR ;CHECK IF WANT TEST SELECTION?
BNE 12\$;BRANCH IF BIT 8 IS SET (YES)
MOV #PTO,NXTST ;STARTUP PRINTING TESTS
BR CHAINY
12\$: TYPEM ;TYPE SELECT TEST NO. MESSAGE
MSG3
CLR ACTIV ;SET TEST ACTIVE STATE NOT ACTIVE
TST IOSW ;DH11 OR DJ11
BEQ TTY1A ;WAIT FOR TERMINAL CONTROL FROM DH11
BR TTYJA ;WAIT FOR TERMINAL CONTROL FROM DJ11

: CLEAN--INITIALIZES POINTERS BEFORE ENTERING A TEST

```

1634
1635
1636
1637
1638
1639
1640
1641
1642 002034
1643 002034 104024
1644
1645 002036 032737 000001 001104
1646 002044 001405
1647 002046 005737 001224
1648 002052 001401
1649 002054 104005
1650 002056 104004
1651 002060 022626
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661 002062 005777 177156
1662 002066 100003
1663 002070 113700 001106
1664 002074 000000
1665 002076 104017
1666 002100 012706 001100
1667 002104 104016
1668 002106 000177 177006
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679 002112 022626
1680 002114 005737 001222
1681 002120 001402
1682 002122 000137 002500
1683 002126 004737 003106
1684 002132 004737 003214
1685 002136 000137 002174

:*****
:CHAINM--THIS PORTION IS THE COMMON RETURN
:FOR ALL TESTS.
:*****

:*****
CHGD4:
CHAINN: CKSWR ;SEE IF A ^G HAS BEEN GIVEN
:*****
BIT #1,CNTLSW ;CHECK IF TERMINAL CONTROL
BEQ 2$ ;BRANCH IF NOT
TST IOSW ;DH11 OR DJ11?
BEQ 1$ ;BRANCH IF DH11
TTYJTL ;WAIT FOR DJ11 TERMINAL CONTROL
1$: TTYCTL ;WAIT FOR DH11 TERMINAL CONTROL
2$: POPSP2 ;CORRECT STACK

:*****
: IF THE SR BIT 15 IS SET,THE CPU WILL HALT HERE WITH
: THE TEST NUMBER IN RO. PRESS CONTINUE TO
: RUN NEXT TEST
:*****

CHAINY: TST @SR ;CHECK SW REG.
BPL 1$ ;BRANCH IF NO HALT
MOVB RTNNO,RO ;CURRENT TEST NUMBER TO RO
1$: HALT
CLEAN ;CLEAN UP
MOV #SPBOT,SP ;SET UP STACK POINTER
FORWD ;SET UP VALUES FOR NEXT TEST
JMP @CURTST ;GO TO TEST

:*****
:TTYJ-- THIS ROUTINE IS USED WHEN THE DJ11S' ARE UNDER
:TEST, OTHERWISE THE COMMENTS AND INSTRUCTIONS ARE
:THE SAME AS FOR TTY1.
:*****

TTYJ: POPSP2 ;CORRECT STACK
TTYJA: TST ACTIV ;TEST IF ENTRY IS FROM A TEST
BEQ 1$ ;BRANCH IF NOT
JMP TTY1G
1$: JSR PC,SCANDJ ;LOOK FOR INPUT
JSR PC,SETDJ ;SET TERMINAL AS CONSOLE
JMP TTY1B ;GO TO CONTROL

```

1686
 1687
 1688
 1689
 1690
 1691
 1692
 1693
 1694
 1695
 1696
 1697
 1698
 1699
 1700
 1701
 1702 002142 022626
 1703 002144 005737 001230
 1704 002150 001002
 1705 002152 000000
 1706 002154 000776
 1707 002156 005737 001222
 1708 002162 001146
 1709 002164 004737 002674
 1710 002170 004737 002750
 1711 002174 042700 177600
 1712 002200 010037 001124
 1713 002204 020027 000040
 1714 002210 001002
 1715 002212 104013
 1716 002214 000773
 1717 002216 012700 000036
 1718 002222 104003
 1719 002224 013700 001124
 1720 002230 104021
 1721 002232 104020
 1722 002234 000421
 1723 002236 010005
 1724 002240 006305
 1725 002242 006305
 1726 002244 006305
 1727 002246 104013
 1728 002250 020027 000040
 1729 002254 001002
 1730 002256 104013
 1731 002260 000773
 1732 002262 012700 000036
 1733 002266 104003
 1734 002270 013700 001124
 1735 002274 104021
 1736 002276 104020
 1737 002300 000555
 1738 002302 060005
 1739 002304 104013
 1740 002306 020027 000040
 1741 002312 001002

```

*****
:TTY1-- THIS ROUTINE IS USED WHEN THE DH11'S ARE UNDER TEST. IT PROCESSES
:THE RESPONSE TO THE MESSAGE "SELECT TEST NO.". THE RESPONSE
: MUST BE THE TWO DIGIT OCTAL TEST NO. FOLLOWED BY:
:         "L" TO LOOP ON TEST
:         "S" TO SEQUENCE ON TEST
:         "" TO EXECUTE TEST ONCE
:
: ALL SPACES ARE IGNORED AND AN ILLEGAL INPUT WILL BE FLAGED
: BY A "?" AND THE RETYPING OF THE ABOVE MESSAGE. THE FIRST
: TERMINAL TO RESPOND IS THE TERMINAL UNDER TEST. ALL
: PRINTER TESTS WILL OUTPUT TO ALL TERMINALS IF SR BIT 13 IS = 1
: WHILE THE ECHO TESTS WILL RESPOND TO THE PRINTER UNDER TEST ONLY.
*****

```

```

*****
TTY1:  POPSP2                ;CORRECT STACK
TTY1A:  TST      DHCNT        ;BE SURE THAT THERE ARE DH11'S
        BNE     2$           ;BRANCH IF YES
1$:     HALT
        BR      1$           ;DO NOT CONTINUE
2$:     TST      ACTIV        ;TEST IF ENTRY IS FROM A TEST
        BNE     TTY1G        ;BRANCH IF IT IS
        JSR     PC,SCANDH     ;LOOK FOR INPUT
        JSR     PC,SETERM     ;SET TERMINAL DATA
TTY1B:  BIC     #177600,R0    ;SAVE ONLY CHAR
        MOV     R0,TEMPCH     ;GET CHAR
1$:     CMP     R0,#40        ;CHECK IF CHAR IS A SPACE
        BNE     2$           ;BRANCH IF NOT
        READ    ;SPACE, LOOP WAITING FOR NEXT CHAR
        BR      1$           ;GOT ONE
2$:     MOV     #30.,R0       ;DELAY FOR HALF DUPLEX
        DELAY
        MOV     TEMPCH,R0     ;GET CHAR
        ECHO    ;ECHO CHAR
        TESTC  ;GO TEST CHAR
        BR      11$          ;ERROR IN CHAR
        MOV     R0,R5         ;OK, SAVE DIGIT IN R5, POS 5-3
        ASL    R5
        ASL    R5
        ASL    R5
        READ
        3$:     CMP     R0,#40 ;GO WAIT FOR NEXT CHAR
        BNE     4$           ;CHECK IF A SPACE
        READ    ;BRANCH IF NOT A SPACE
        BR      3$           ;WAIT FOR CHAR
        4$:     MOV     #30.,R0 ;GOT ONE, ECHO IT
        DELAY ;DELAY FOR HALF DUPLEX
        MOV     TEMPCH,R0    ;GET CHAR
        ECHO    ;ECHO CHAR
        TESTC  ;GO CHECK CHAR
        11$:    BR      NG     ;ERROR IN CHAR
        ADD     R0,R5        ;OK,R5 IS NOW OCTAL TEST NO.
        READ    ;GO WAIT FOR TERMINATOR
        5$:     CMP     R0,#40 ;CHECK IF A SPACE
        BNE     6$           ;BRANCH IF NOT A SPACE

```

1742	002314	104013				READ		:SPACE, WAIT SOME MORE
1743	002316	000773				BR	5\$:GOT ONE, ECHO IT
1744	002320	012700	000036		6\$:	MOV	#30.,RO	:DELAY FOR HALF DUPLEX
1745	002324	104003				DELAY		
1746	002326	013700	001124			MOV	TEMPCH,RO	:GET CHAR
1747	002332	104021				ECHO		:ECHO CHAR
1748	002334	042700	000040			BIC	#BITS,RO	:ALLOW LOWER CASE OR UPPER CASE
1749	002340	020027	000114			CMP	RO,#114	:IS IT AN 'L'
1750	002344	001413				BEQ	7\$:BRANCH IF YES
1751	002346	020027	000123			CMP	RO,#123	:NO, IS IT AN 'S'
1752	002352	001414				BEQ	8\$:BRANCH IF YES
1753	002354	023727	001124	000056		CMP	TEMPCH,#56	:NO, IS IT A '.'
1754	002362	001124				BNE	NG	:NO, ERROROR
1755	002364	012737	000001	001104		MOV	#1,CNTLSW	:SET BIT 0 ONLY IN CNTLSW
1756	002372	000407				BR	9\$	
1757	002374	012737	004001	001104	7\$:	MOV	#4001,CNTLSW	:SET BITS 11 AND 0
1758	002402	000403				BR	9\$	
1759	002404	012737	000401	001104	8\$:	MOV	#401,CNTLSW	:SET BITS 8 AND 0
1760	002412	104017			9\$:	CLEAN		:CLEAN UP
1761	002414	012706	001100			MOV	#SPBOT,SP	:RESET SP
1762	002420	010500				MOV	R5,RO	:TEST NO TO RO
1763	002422	020027	000040			CMP	RO,#40	:CHECK IF TEST NO. IS EQ OR GT 40
1764	002426	103102				BHIS	NG	:ERROR IF YES
1765	002430	020027	000020			CMP	RO,#20	:CHECK IF THIS IS AN ECHO TEST
1766	002434	103406				BLO	10\$:BRANCH IF NOT
1767	002436	020027	000030			CMP	RO,#30	:OPTION TEST?
1768	002442	103003				BHIS	10\$:ALLOW LOOP ON OPTION TEST
1769	002444	012737	000001	001104		MOV	#1,CNTLSW	:YES, FORCE TO ONE TIME ONLY
1770	002452	006300			10\$:	ASL	RO	:TEST NO. * 2
1771	002454	016037	003246	001110		MOV	PRGTAB(RO),NXTST	:ADDR OF TEST TO NXTST
1772	002462	001464				BEQ	NG	:BRANCH IF ILLEGAL TEST
1773	002464	104016				FORWD		:SET UP TEST PARAMETERS
1774	002466	012737	000001	001222		MOV	#1,ACTIV	:SET TEST ACTIVE IND
1775	002474	000177	176420			JMP	@CURTST	:GO TO TEST
1776	002500	017700	176434		TTY1G:	MOV	@NRCRA,RO	:TEST ACTIVE,CHECK INPUT FROM DH11
1777	002504	100040				BPL	TTY1L	:BRANCH IF NO DATA
1778	002506	010004				MOV	RO,R4	:DATA, SAVE IT
1779	002510	000300				SWAB	RO	:RIGHT JUSTIFY LINE NO.
1780	002512	042700	177760			BIC	#177760,RO	:CLEAR ALL BUT LINE NO.
1781	002516	020037	001156			CMP	RO,LINENO	:CHECK IF LINE NO. IS SAME AS TEST LINE
1782	002522	001366				BNE	TTY1G	:NOT SAME, SEE IF ANY MORE IN SILO
1783	002524	010400				MOV	R4,RO	:LINES ARE THE SAME, GET CHAR
1784	002526	042700	177600			BIC	#177600,RO	:SAVE 7 BITS OF CHAR
1785	002532	020027	000177			CMP	RO,#177	:CHECK IF A RUBOUT
1786	002536	001360				BNE	TTY1G	:NOT A RUBOUT, SEE IF ANY MORE
1787	002540	012706	001100		TTY1H:	MOV	#SPBOT,SP	:RESET STACK
1788	002544	012737	000001	001104		MOV	#1,CNTLSW	:CLEAR BITS 11 AND 8
1789	002552	012700	000036			MOV	#30.,RO	:DELAY FOR HALF DUPLEX
1790	002556	104003				DELAY		
1791	002560	104002				TYPEN		:OUTPUT MESSAGE
1792	002562	020066				MSG3		
1793	002564	005037	001222			CLR	ACTIV	:CLEAR TEST ACTIVE STATE
1794	002570	005737	001224			TST	IOSW	:DJ11 OR DH11 ?
1795	002574	001402				BEQ	1\$:BRANCH IF DH11
1796	002576	000137	002114			JMP	TTYJA	:WAIT FOR NEXT TEST FROM DJ11
1797	002602	000137	002144		1\$:	JMP	TTY1A	:WAIT FOR NEXT TEST FROM DH11

```

1798 002606 032737 004000 001104 TTY1L: BIT #BIT11,CNTLSW ;CHECK IF LOOP ON TEST
1799 002614 001401 BEQ 1$ ;BRANCH IF NO LOOP
1800 002616 000002 RTI ;GO LOOP ON TEST
1801 002620 032737 000400 001104 1$: BIT #BIT8,CNTLSW ;CHECK IF LOOP ON SEQUENCE
1802 002626 001744 BEQ TTY1H ;BRANCH IF NO
1803 002630 000137 002062 JMP CHAINY ;GO LOOP ON SEQUENCE
1804 002634 012700 000036 NG: MOV #30.,R0 ;DELAY FOR HALF DUPLEX
1805 002640 104003 DELAY
1806 002642 112700 000077 MOVB #77,R0 ;'"?' TO TEMPCH
1807 002646 104021 ECHO ;PRINT A ?
1808 002650 000733 BR TTY1H ;TRY AGAIN FROM DH11
1809
1810
1811 ;*****
1812 ;
1813 ;FORWARD--THIS ROUTINE TRANSFERS THE 2 ARGUMENTS
1814 ; FROM THE TEST ROUTINE. THEY ARE:
1815 ; 1- ROUTINE NUMBER
1816 ; 2- ADDRESS OF NEXT TEST
1817 ;
1818 ;*****
1819
1820 002652 013705 001110 $FORWD: MOV NXTST,R5 ;ADDR OF NEXT TEST TO R5
1821 002656 012537 001106 MOV (R5)+,RTNNO ;GET NUMBER OF NEXT TEST
1822 002662 012537 001110 MOV (R5)+,NXTST ;GET ADDR OF FOLLOWING TEST
1823 002666 010537 001120 MOV R5,CURTST ;ENTRY POINT TO TEST IN CURTST
1824 002672 000002 RTI ;EXIT
1825
1826
1827 ;*****
1828 ;
1829 ;SCANDH - ROUTINE TO SCAN DH CHANNELS LOOKING FOR INPUT
1830 ;
1831 ;*****
1832
1833 002674 013701 001230 SCANDH: MOV DHCNT,R1 ;COUNT OF DH11'S TO R1
1834 002700 005037 001154 CLR CNTDH ;CLEAR DH11 POSITION COUNTER
1835 002704 013700 001134 MOV DHADR,R0 ;ADDR OF FIRST DH11 TO R0
1836 002710 005720 TST (R0)+ ;ADDR OF NRCRA
1837 002712 010037 001140 MOV R0,NRCRA ;SET UP NRCRA ADDRESS
1838 002716 017700 176216 1$: MOV @NRCRA,R0 ;GET NEXT CHAR FROM SILO
1839 002722 100410 BMI 2$ ;BRANCH IF DATA IS PRESENT
1840 002724 005301 DEC R1 ;DECREMENT COUNT OF DH11'S
1841 002726 001762 BEQ SCANDH ;START OVER IF ALL DONE
1842 002730 062737 000020 001140 ADD #20,NRCRA ;SET UP ADDR FOR NEXT DH11
1843 002736 005237 001154 INC CNTDH ;INC DH11 POSITION COUNTER
1844 002742 000765 BR 1$ ;GO CHECK NEXT DH11 ON BUS
1845 002744 010004 2$: MOV R0,R4 ;SAVE LINE NO. AND CHAR
1846 002746 000207 RTS PC ;RETURN

```

1847
1848
1849
1850
1851
1852
1853
1854 002750 013701 001140
1855 002754 005741
1856 002756 010137 001136
1857 002762 022121
1858 002764 010137 001142
1859 002770 005721
1860 002772 010137 001144
1861 002776 005721
1862 003000 010137 001146
1863 003004 005721
1864 003006 010137 001150
1865 003012 022121
1866 003014 010137 001152
1867 003020 000300
1868 003022 042700 177760
1869 003026 010077 176104
1870 003032 012701 016402
1871 003036 005737 001154
1872 003042 001405
1873 003044 062701 000020
1874 003050 005337 001154
1875 003054 000770
1876 003056 010037 001156
1877 003062 006300
1878 003064 060001
1879 003066 011177 176050
1880 003072 062700 001162
1881 003076 011037 001160
1882 003102 010400
1883 003104 000207

: SETERM - THIS ROUTINE IS USED TO SET UP THE REGISTER ADDRESSES OF
: THE TERMINAL CURRENTLY UNDER TEST.
: *****

SETERM: MOV NRCRA,R1 ;GET ADDR OF CURRENT NRCRA
TST -(R1) ;CAL ADDR OF SCR
MOV R1,SCR ;STORE SCR ADDR
CMP (R1)+,(R1)+ ;ADD 4
MOV R1,LPR ;STORE LPR ADDR
TST (R1)+ ;ADD 2
MOV R1,CARA ;STORE CARA ADDR
TST (R1)+ ;ADD 2
MOV R1,BYCR ;STORE BYCR ADDR
TST (R1)+ ;ADD 2
MOV R1,BAR ;STORE BAR ADDR
CMP (R1)+,(R1)+ ;ADD 4
MOV R1,SSR ;STORE SSR ADDR
SWAB R0 ;RIGHT JUSTIFY LINE NO.
BIC #177760,R0 ;SAVE ONLY LINE NO.
MOV R0,@SCR ;SET UP SCR
MOV #DH1100,R1 ;GET ADDR OF VP TABLE
1\$: TST CNTDH ;CAL WHICH DH11 IS UNDER TEST
BEQ 2\$;BRANCH IF THIS IS THE ONE
ADD #20,R1 ;REPOSITION R1 IN VP TABLE
DEC CNTDH ;DEC DH11 POSITION COUNTER
BR 1\$
2\$: MOV R0,LINENO ;SAVE LINE NO.
ASL R0 ;CREATE POINTER TO TABLE
ADD R0,R1 ;ADD LINE NO. TO VP TAB PNTR
MOV (R1),@LPR ;PUT VP INTO LPR
ADD #BITTAB,R0 ;CAL PNTR INTO LINE MASK TABLE
MOV (R0),MASK ;PUT LINE MASK INTO MASK
MOV R4,R0 ;RESTORE LINE NO AND CHAR
RTS PC ;RETURN


```
1884  
1885  
1886  
1887  
1888  
1889  
1890 003106 013701 001226  
1891 003112 005037 001154  
1892 003116 012700 160012  
1893 003122 012703 017402  
1894 003126 011002  
1895 003130 100410  
1896 003132 005301  
1897 003134 001764  
1898 003136 062700 000010  
1899 003142 005723  
1900 003144 005237 001154  
1901 003150 000766  
1902 003152 010204  
1903 003154 000302  
1904 003156 042702 177760  
1905 003162 010237 001156  
1906 003166 006302  
1907 003170 062702 001162  
1908 003174 011237 001160  
1909 003200 033713 001160  
1910 003204 001352  
1911 003206 110402  
1912 003210 001750  
1913 003212 000207  
1914  
1915  
1916  
1917  
1918  
1919  
1920  
1921  
1922  
1923 003214 010037 001140  
1924 003220 005740  
1925 003222 010037 001236  
1926 003226 022020  
1927 003230 010037 001240  
1928 003234 005720  
1929 003236 010037 001242  
1930 003242 010400  
1931 003244 000207
```

```
*****  
:SCANDJ - ROUTINE TO SCAN CHANNELS LOOKING FOR INPUT  
*****  
SCANDJ: MOV      DJCNT,R1      ;COUNT OF DJ11 TO R1  
          CLR      CNTDH      ;CLEAR POS. COUNTER  
          MOV      #160012,R0  ;ADDR OF FIRST RECV BUFFER TO R0  
          MOV      #ELTAB,R3   ;ADDR OF ELTAB TO R3  
1$:      MOV      (R0),R2     ;GET RECV BUFFER  
          BMI      3$         ;BRANCH IF DATA  
2$:      DEC      R1          ;NO DATA, DEC COUNT OF DJ11  
          BEQ      SCANDJ     ;START OVER IF ALL CHECKED  
          ADD      #10,R0     ;ADDR OF NEXT RBUF  
          TST      (R3)+      ;INCREMENT ELTAB POINTER  
          INC      CNTDH     ;INCREMENT POS COUNTER  
          BR      1$         ;DO NEXT DJ11  
3$:      MOV      R2,R4       ;GET CONTENTS OF RBUF  
          SWAB     R2  
          BIC      #177760,R2  ;SAVE ONLY THE LINE NO.  
          MOV      R2,LINENO  ;SAVE LINE NO.  
          ASL      R2         ;LINE NO.*2  
          ADD      #BITTAB,R2 ;CAL ADDR OF LINE MASK  
          MOV      (R2),MASK  ;SAVE LINE MASK  
          BIT      MASK,(R3)  ;CHECK IF LINE EXISTS  
          BNE      2$         ;BRANCH IF NOT  
          MOVB    R4,R2      ;RESTORE R2 WITH CHAR  
          BEQ      2$         ;BRANCH IF NULL  
          RTS      PC        ;RETURN  
*****  
:SETDJ - THIS ROUTINE IS USED TO SET UP THE PARAMETERS  
:          FOR THE DJ11 TERMINAL CURRENTLY UNDER TEST.  
:          *****  
SETDJ:  MOV      R0,NRCRA     ;SAVE ADDR FOR READ & TTY1H  
          TST      -(R0)      ;CAL ADDR OF CSR  
          MOV      R0,CSR     ;ADDR OF CURRENT CSR  
          CMP      (R0)+,(R0)+ ;CAL ADDR OF TCR  
          MOV      R0,TCR     ;SAVE ADDR OF CURRENT CSR  
          TST      (R0)+      ;CAL ADDR OF TBUF  
          MOV      R0,TBUF    ;SAVE ADDR OF CURRENT TBUF  
          MOV      R4,R0      ;GET CONTENTS OF RECV BUFFER AGAIN  
          RTS      PC        ;RETURN
```

			.SBTTL TEST ADDRESS TABLE	
1932				
1933				
1934	003246	006560	PRGTAB: PT0	:DATA PATH TEST
1935	003250	006634	PT1	:PRINTER CHARACTER TEST
1936	003252	006756	PT2	:NON-PRINTING CHARACTER TEST
1937	003254	007406	PT3	:CARRIAGE RETURN TEST
1938	003256	007526	PT4	:MULTIPLE LINE FEED TEST
1939	003260	007704	PT5	:SINGLE LINE FEED TEST
1940	003262	010110	PT6	:BACKSPACE TEST
1941	003264	010272	PT7	:OVERPRINT TEST
1942	003266	010504	PT10	:PRINTING FREQUENCY SWEEP TEST
1943	003270	010642	PT11	:RIBBON FEED TEST
1944	003272	010674	PT12	:PRINTER BELL TEST
1945	003274	000000	OPEN	:SPARE FOR ADDITIONAL PRINTER TEST
1946	003276	000000	OPEN	:SPARE FOR ADDITIONAL PRINTER TEST
1947	003300	000000	OPEN	:SPARE FOR ADDITIONAL PRINTER TEST
1948	003302	000000	OPEN	:SPARE FOR ADDITIONAL PRINTER TEST
1949	003304	010764	PT17	:LIFE TEST
1950				
1951	003306	011126	E020	:CHARACTER ECHO TEST
1952	003310	011172	E021	:LINE ECHO TEST, FAST RATE
1953	003312	011244	E022	:LINE ECHO TEST, SLOW RATE
1954	003314	011530	E023	:CHARACTER/CODE ECHO TEST
1955	003316	012052	E024	:SELECTIVE PATTERN ECHO TEST
1956	003320	012636	E025	:BELL ECHO TEST
1957	003322	000000	OPEN	:SPARE FOR ADDITIONAL ECHO TESTS
1958	003324	000000	OPEN	:SPARE FOR ADDITIONAL ECHO TESTS
1959				
1960	003326	012740	TEST30	:SECONDARY CHARACTER SET OPTION
1961	003330	013122	TEST31	:SELECTIVE ADDRESSING OPTION
1962	003332	013740	TEST32	:AUTO ANSWER BACK OPTION
1963	003334	014170	TEST33	:FORM FEED OPTION
1964	003336	014724	TEST34	:HORIZONTAL TAB OPTION
1965	003340	015274	TEST35	:VERTICAL TAB OPTION
1966	003342	000000	OPEN	:SPARE FOR ADDITIONAL OPTION TESTS
1967	003344	000000	OPEN	:SPARE FOR ADDITIONAL OPTION TESTS

.SBTTL EMT TRAP DECODER

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

003346 011646
003350 162716 000002
003354 017616 000000
003360 121627 000026
003364 101402
003366 000000
003370 000776
003372 006116
003374 042716 177001
003400 062716 003422
003404 017616 000000
003410 005046
003412 012746 003420
003416 000002
003420 000136

003422 004114
003424 002034
003426 004144
003430 004174
003432 002142
003434 002112
003436 004474
003440 004456
003442 004476
003444 004572
003446 004516
003450 003716
003452 004506
003454 004320
003456 002652
003460 002020
003462 004424
003464 003516
003466 003654

003470 003476
003472 015736
003474 015774

```
*****  
:EMTINT--SERVICE ROUTINES FOR TRAPS THROUGH LOC. 30 AND 34.  
*****  
EMTINT: MOV @SP,-(SP) ;GET SAVED PC  
SUB #2,@SP ;DECREMENT PC BY TWO  
MOV @SP,@SP  
CMPB @SP,#26 ;CHECK THAT CALL IS WITHIN LIMITS  
BLOS 2$ ;BRANCH IF OK  
1$: HALT  
BR 1$  
2$: ROL @SP ;EMT ARGUMENT *2  
BIC #177001,@SP ;REMOVE 7 MSB  
ADD #EMTAB,@SP ;FORM EMT RTN ADDRESS  
MOV @SP,@SP  
CLR -(SP) ;CLEAR PSW  
MOV #3$,-(SP)  
3$: JMP @SP+ ;GO TO EMT ROUTINE  
  
EMTAB: TYP ;MESSAGE OUTPUT ROUTINE  
CHAINN ;COMMON TEST EXIT  
TYPM ;MESSAGE OUTPUT ROUTINE, MULTI DEVICES  
DLY ;DELAY ROUTINE  
TTY1 ;DH11 CONSOLE TERMINAL CONTROL  
TTYJ ;DJ11 CONSOLE TERMINAL CONTROL  
$CRLF ;CARRIAGE RETURN, ALL TERMINALS  
$$CRLF ;CARRIAGE RETURN, CONSOLE TERMINAL  
$LF ;LINE FEED ONLY, ALL TERM'S  
$PRTC ;PRINT CHAR, ALL TERM'S  
$PRHDR ;PRINT TEST HEADER  
$READ ;READ CHAR  
$CR ;CARRIAGE RETURN ONLY, ALL TERM'S  
$BTASC ;BINARY TO ASCII CONVERSION  
  
$TESTC ;CHECK CHAR  
$ECHO ;PRINT CHAR, CONSOLE ONLY  
$INRDY ;CHECK IF READY  
  
:***** CHGD1 *****  
$FOUR ;FOUR FILL CHARACTERS  
$CKSWR ;CHECK SWITCH REG CONTENTS  
$CNTLU ;RE-ENTER POINT FOR CKSWR ROUTINE  
:*****
```

2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035

003476
003476 012705 000004
003502 012700 000006
003506 104011
003510 005305
003512 101373
003514 000002

```
***** CHGDS *****  
:FOUR-- THIS ROUTINE WILL PRINT FOUR FILL CHARACTERS  
:*****  
:*****  
CHGDS:  
$FOUR: MOV #4,R5 :PLACE A FOUR INTO R5  
1$: MOV #6,R0 :PLACE THE ACK CODE INTO R6  
PRINTC :TRANSMIT CODE  
DEC R5 :DEC R5  
BHI 1$ :4 TIMES  
RTI  
:*****
```

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
2087
2088
2089

003516 005737 001224
003522 001026
003524 037737 175420 001160
003532 001371
003534 010037 001124
003540 013777 001156 175370
003546 012777 001124 175370
003554 012777 177777 175364
003562 012777 000020 175362
003570 053777 001160 175352
003576 000002
003600 013777 001160 175432
003606 052777 000400 175422
003614 005777 175416
003620 100375
003622 017704 175414
003626 000304
003630 042704 177760
003634 020437 001156
003640 001362
003642 110077 175374
003646 110037 001124
003652 000002

017700 175260
100015
010037 001124
000300
042700 177760
020037 001156
001365
042737 177600 001124
062716 000002
000002

.SBTTL COMMON ROUTINES

:*****
:ECHO-- THIS ROUTINE ECHOS CHARACTERS ON THE TERMINAL UNDER TEST.
:*****

```
SECHO: TST IOSW ;DJ11 OR DH11
        BNE ECHODJ ;BRANCH IF DJ11
        BIT @BAR,MASK ;CHECK IF OK TO SEND TO DH11
        BNE SECHO ;NO, WAIT UNTIL OK
        MOV RO,TEMPCH ;CHAR INTO TEMPCH
        MOV LINENO,@SCR ;SET LINE NUMBER
        MOV #TEMPCH,@CARA ;OK, PUT ADDR OF CHAR INTO CARA
        MOV #-1,@BYCR ;SET CHAR COUNT TO 1
        MOV #20,@SSR ;SET SILO OVERFLOW TO 16
        BIS MASK,@BAR ;SET TRANSMIT BIT
        RTI ;RETURN
ECHODJ: MOV MASK,@TCR ;SET LINE NO. IN TCR
1$: BIS #BIT8,@CSR ;START TRANSMITTER SCANNER
2$: TST @CSR ;CHECK IF OK
    BPL 2$ ;BRANCH IF NOT READY
    MOV @TBUF,R4 ;GET CONTENTS OF TBUF
    SWAB R4 ;RIGHT JUSTIFY LINE NO.
    BIC #177760,R4 ;SAVE ONLY THE LINE NO.
    CMP R4,LINENO ;BE SURE IT'S THE LINE UNDER TEST
    BNE 1$ ;BRANCH IF NOT
    MOVB RO,@TBUF ;LOAD CHAR TO PRINT
    MOVB RO,TEMPCH ;SAVE CHAR IN TEMPCH
    RTI ;RETURN
```

:*****

:INRDY-- CHECKS IF ANY INPUT FROM TERMINAL UNDER TEST.
: RETURN VIA PC IF NO CHAR OR VIA PC+2 IF VALID CHAR. CHAR WILL
: BE IN BITS 6-0 OF TEMPCH.
: CALLING SEQUENCE:
: INRDY
: NO CHAR RETURN
: VALID CHAR RETURN

:*****

```
SINRDY: MOV @NRCRA,RO ;GET CHAR
        BPL 1$ ;BRANCH IF NO CHAR
        MOV RO,TEMPCH ;SAVE CHAR
        SWAB RO ;RIGHT JUSTIFY LINE NO.
        BIC #177760,RO ;SAVE ONLY LINE NO.
        CMP RO,LINENO ;CHECK IF SAME AS LINE UNDER TEST
        BNE $INRDY ;BRANCH IF NOT LINE UNDER TEST
        BIC #177600,TEMPCH ;SAVE ONLY THE CHAR
        ADD #2,@SP ;SET UP RETURN ADDR.
1$: RTI ;RETURN
```

.....

```
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099 003716 032777 020000 175320 $READ: BIT #BIT13,@SR ;SW13 SET?
2100 003724 001413 BEQ 1$ ;BRANCH IF NOT
2101 003726 017700 175206 MOV @NRCRA,R0 ;GET CHAR
2102 003732 100371 RPL $READ ;NO CHAR, WAIT FOR ONE
2103 003734 010004 MOV R0,R4 ;VALID DATA
2104 003736 000304 SWAB R4 ;RIGHT JUSTIFY LINE NO.
2105 003740 042704 177760 BIC #177760,R4 ;SAVE ONLY LINE NO.
2106 003744 020437 001156 CMP R4,LINENO ;CHECK IF SAME AS LINE UNDER TEST
2107 003750 001362 BNE $READ ;BRANCH IF NOT
2108 003752 000424 BR 4$ ;CONTINUE
2109 003754 010146 1$: MOV R1,-(SP) ;SAVE R1
2110 003756 010246 MOV R2,-(SP)
2111 003760 010346 MOV R3,-(SP)
2112 003762 010446 MOV R4,-(SP)
2113 003764 005737 001224 TST IOSW ;DH OR DJ?
2114 003770 001005 BNE 2$ ;BRANCH IF DJ
2115 003772 004737 002674 JSR PC,SCANDH ;LOOK FOR INPUT
2116 003776 004737 002750 JSR PC,SETERM ;SETUP TERMINAL UNDER TEST
2117 004002 000404 BR 3$ ;CONTINUE
2118 004004 004737 003106 2$: JSR PC,SCANDJ ;LOOK FOR DJ INPUT
2119 004010 004737 003214 JSR PC,SETDJ ;SETUP DJ UNDER TEST
2120 004014 012604 3$: MOV (SP)+,R4 ;RESTORE REGS
2121 004016 012603 MOV (SP)+,R3
2122 004020 012602 MOV (SP)+,R2
2123 004022 012601 MOV (SP)+,R1
2124 004024 010037 001130 4$: MOV R0,PCHAR ;SAVE CHAR WITH PARITY BIT
2125 004030 113737 001130 001127 MOVB PCHAR,PARITY+1 ;GET CODE FOR PARITY CHECK
2126 004036 042737 177400 001130 BIC #177400,PCHAR ;CLEAR UNWANTED BITS
2127 004044 042700 177600 BIC #177600,R0 ;SAVE ONLY THE CHAR
2128 004050 010037 001124 MOV R0,TEMPCH
2129 004054 012700 000011 MOV #11,R0 ;SET SHIFT COUNT
2130 004060 042737 000377 001126 BIC #377,PARITY ;CLEAR PARITY FLAG
2131 004066 005300 5$: DEC R0 ;DECREMENT SHIFT COUNT
2132 004070 001406 BEQ 6$ ;EXIT IF DONE
2133 004072 106337 001127 ASLB PARITY+1 ;ROTATE CODE
2134 004076 103373 BCC 5$ ;CONTINUE IF CARRY BIT WAS A ZERO
2135 004100 105137 001126 COMB PARITY ;COMPLIMENT PARITY FLAG IF BIT WAS ONE
2136 004104 000770 BR 5$ ;CONTINUE
2137 004106 013700 001124 6$: MOV TEMPCH,R0 ;RESTORE R0
2138 004112 000002 RTI ;RETURN
```

```
2139 ;XXXXXXXXXX
2140 ;
2141 ;TYPE-- A COMMON ROUTINE USED TO TYPE MESSAGES ON THE TERMINAL
2142 ; UNDER TEST ONLY. THE NULL CHARACTER TERMINATES
2143 ; THE MESSAGE. CALLED THROUGH AN EMT TRAP.
2144 ; CALLING SEQUENCE
2145 ; TYPE
2146 ; MSG ;ADDRESS OF MESSAGE
2147 ;
2148 ;XXXXXXXXXX
2149 ;
2150 004114 011601 TYP: MOV @SP,R1 ;GET POINTER TO ADDR. OF MSG.
2151 004116 062716 000002 ADD #2,@SP ;SET UP RETURN ADDRESS
2152 004122 011101 MOV @R1,R1 ;ADDR. OF MSG TO R1
2153 004124 112100 1$: MOV (R1)+,R0 ;GET CHAR
2154 004126 100402 BMI 2$ ;BRANCH IF WANT AUTO CR-LF
2155 004130 001003 BNE 3$ ;PRINT CHAR IF NOT NULL
2156 004132 000002 RTI ;EXIT ON NULL CHAR
2157 004134 104007 2$: SCRLF ;SEND A CRLF
2158 004136 000772 BR 1$ ;GET NEXT CHAR
2159 004140 104021 3$: ECHO 1$ ;OUTPUT CHAR
2160 004142 000770 BR 1$ ;GO GET NEXT CHAR
2161 ;
2162 ;XXXXXXXXXX
2163 ;
2164 ;
2165 ;TYPM---MULTI TYPE-A COMMON ROUTINE TO OUTPUT
2166 ; A MESSAGE ON ALL DH11S.
2167 ; THIS ROUTINE IS USED BY
2168 ; THE PRINTER TESTS TO TYPE HEADINGS.
2169 ; CALLING SEQUENCE;
2170 ; TYPM
2171 ; MSGAD ;ADDRESS OF MESSAGE
2172 ;
2173 ;XXXXXXXXXX
2174 ;
2175 004144 011601 TYPM: MOV @SP,R1 ;GET POINTER TO ADDR OF MSG
2176 004146 062716 000002 ADD #2,@SP ;SET UP RETURN ADDRESS
2177 004152 011101 MOV @R1,R1 ;ADDR OF MSG TO R1
2178 004154 112100 1$: MOV (R1)+,R0 ;GET CHAR
2179 004156 100402 BMI 2$ ;BRANCH IF WANT AUTO CR-LF
2180 004160 001003 BNE 3$ ;CONTINUE IF NOT NULL
2181 004162 000002 RTI ;RETURN TO CALLER
2182 004164 104006 2$: CRLF ;SEND CR-LF
2183 004166 000772 BR 1$ ;GET NEXT CHAR
2184 004170 104011 3$: PRINTC ;PRINT CHAR
2185 004172 000770 BR 1$ ;GO GET NEXT CHAR.
```

2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223

:DELAY--A COMMON ROUTINE TO DELAY PROCESSING
: A GIVEN NUMBER OF MSEC.
: CALLING SEQUENCE:
: MOV #5,R0 ;R0 CONTAINS THE NUMBER OF MSEC DELAY DESIRED
: DELAY

: THE DELAY IS EFFECTED BY THE EXECUTION OF THE LOOP:
: 1\$: DEC R1
: BNE 1\$

: SINCE THE EXECUTION TIMES OF THE PDP11 LINE DOES VARY FROM
: MACHINE TO MACHINE, THE VALUE AT SYMBOLIC LOCATION
: "TIMER" MUST BE CHANGED TO THE APPROPRIATE VALUE AS SHOWN BELOW
: BEFORE STARTING THE DIAGNOSTIC. "TIMER" IS INITIALIZED
: FOR AN PDP-11/40 (TIMER = 554).

MACHINE	05810	35840	15820	BIPOLAR	11/45 & 11/70	MOS	CORE
LOOP: DEC R1	3.4	.99	2.3	.30		.51	.90
BNE LOOP	2.5	1.76	2.6	.60		.98	1.13
TIME=	5.9US	2.75	4.9	.90USEC		1.49USEC	
SET TIMER	251	554	314	2127		1237	755

2.03USEC

:XXXXXXXXXX

004174 010146
004176 013701 001102
004202 005301
004204 001376
004206 005300
004210 001372
004212 012601
004214 000002

DLY: MOV R1,-(SP) ;SAVE R1
1\$: MOV TIMER,R1 ;MOV 1MSEC LOOP CNT TO R1
2\$: DEC R1 ;DECREMENT COUNT
BNE 2\$;BRANCH IF NOT ZERO
DEC R0 ;DEC NO. OF MSEC DELAY
BNE 1\$;DELAY AGAIN IF NOT ZERO
MOV (SP)+,R1 ;ALL DONE RESTORE R1
RTI


```

2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235 004216 010046
2236 004220 010146
2237 004222 010246
2238 004224 010346
2239 004226 010446
2240 004230 010546
2241 004232 013746 000024
2242 004236 010637 004252
2243 004242 012737 004254 000024
2244 004250 000000
2245
2246
2247 004252 000000
2248
2249
2250 004254 104002
2251 004256 004310
2252 004260 013706 004252
2253 004264 012637 000024
2254 004270 012605
2255 004272 012604
2256 004274 012603
2257 004276 012602
2258 004300 012601
2259 004302 012600
2260 004304 000137 001302
2261
2262 004310 050200 053517 051105 1$:
2263 004316 000200
2264

```

```

*****
:PFail--POWER FAIL ROUTINE
:SAVE ALL REGISTERS AND SET RESTART ADDRESS
:INTO LOCATION 24
:RESTART--POWER FAIL RECOVERY
:RESTORE ALL REGISTERS AND GO TO START
*****
PFail:  MOV  R0,-(SP)
        MOV  R1,-(SP)
        MOV  R2,-(SP)
        MOV  R3,-(SP)
        MOV  R4,-(SP)
        MOV  R5,-(SP)
        MOV  24,-(SP)
        MOV  SP,SAVR6      ;SAVE STACK POSITION
        MOV  #RESTR,24    ;STORE RESTART ADDRESS
        HALT
SAVR6:  .WORD  0
RESTR:  TYPEN              ;TYPE POWER MESSG
        1$
        MOV  SAVR6,SP      ;RESTORE STACK POINTER
        MOV  (SP)+,24      ;RESTORE PFAIL ADDRESS
        MOV  (SP)+,R5
        MOV  (SP)+,R4
        MOV  (SP)+,R3
        MOV  (SP)+,R2
        MOV  (SP)+,R1
        MOV  (SP)+,R0
        JMP  STARTX
        .ASCIZ <ACRLF>/POWER/<ACRLF>
        .EVEN

```

```
2265  
2266  
2267  
2268  
2269  
2270  
2271  
2272  
2273  
2274  
2275  
2276  
2277  
2278 004320 010237 004404  
2279 004324 006302  
2280 004326 062702 004412  
2281 004332 014237 004410  
2282 004336 005037 004406  
2283 004342 163701 004410  
2284 004346 103403  
2285 004350 005237 004406  
2286 004354 000772  
2287 004356 063701 004410  
2288 004362 062737 000060 004406  
2289 004370 113720 004406  
2290 004374 005337 004404  
2291 004400 001354  
2292 004402 000002  
2293  
2294  
2295 004404 000000  
2296 004406 000000  
2297 004410 000000  
2298  
2299 004412 000001 000012 000144  
2300 004420 001750 023420  
2301  
2302  
2303  
2304  
2305  
2306  
2307  
2308  
2309  
2310 004424 013700 001124  
2311 004430 020027 000060  
2312 004434 103407  
2313 004436 020027 000067  
2314 004442 101004  
2315 004444 062716 000002  
2316 004450 042700 177770  
2317 004454 000002
```

```
*****  
: BINARY TO ASCII CONVERSION (1 TO 5 ASCII CHARACTERS)  
: CALLING SEQUENCE  
:     MOV     ADDRESS OF LOC TO STORE FIRST ASCII CHAR. INTO R0  
:     MOV     BINARY NUMBER TO BE CONVERTED INTO R1  
:     MOV     NUMBER TO BE CONVERTED AS A POWER OF TEN INTO R2  
:     BTOASC  
:*****  
SBTASC: MOV     R2,CNVCTR     ;SAVE TEN POWER  
        ASL     R2           ;R2*2  
        ADI     #ADTENP,R2   ;CALCULATE ADDRESS OF STARTING TEN POWER  
1$:     MOV     -(R2),TENPWR ;POWER OF TEN VALUE TO TENPWR  
        CLR     DIGIT        ;CLEAR CURRENT DIGIT  
2$:     SUB     TENPWR,R1    ;SUBTRACT TEN POWER FROM BINARY VALUE  
        BCS     3$          ;BRANCH IF END  
        INC     DIGIT        ;  
        BR     2$           ;  
3$:     ADD     TENPWR,R1    ;RESTORE SUBTRACTED VALUE  
        ADD     #60,DIGIT    ;CONVERT (DIGIT) TO ASCII  
        MOVB   DIGIT,(R0)+  ;PUT ASCII CHAR INTO USER BUFFER  
        DEC     CNVCTR       ;FINISH ALL CHARS CALLED FOR  
        BNE    1$          ;BRANCH IF NOT FINISHED  
        RTI  
        ;RETURN  
  
CNVCTR: .WORD 0           ;CONVERSION CHARACTER COUNT  
DIGIT:  .WORD 0           ;CONVERTED CHARACTER  
TENPWR: .WORD 0           ;CURRENT TEN POWER  
  
ADTENP: .WORD 1.,10.,100.,1000.,10000.  
  
*****  
: TESTC-- CHECKS FOR INPUTTED OCTAL DIGIT  
:         BETWEEN A 0 AND A 7 INCLUSIVE  
:*****  
STESTC: MOV     TEMPCH,R0   ;GET CHAR  
        CMP     R0,#60      ;CHECK IF NUMERIC AND EQ ,GT 0  
        BLO    1$          ;BRANCH ERROR  
        CMP     R0,#67      ;CHECK IF EQ OR LT 7  
        BHI    1$          ;BRANCH ERROR  
        ADD     #2,@SP      ;SET UP RETURN ADDRESS  
        BIC    #177770,R0  ;SAVE ONLY THE DIGIT  
1$:     RTI  
        ;NORMAL RETURN
```

```
2318  
2319  
2320  
2321  
2322  
2323  
2324  
2325 004456 112700 000015  
2326 004462 104021  
2327 004464 112700 000012  
2328 004470 104021  
2329 004472 000002  
2330  
2331  
2332  
2333  
2334  
2335  
2336  
2337  
2338 004474 104014  
2339 004476 012700 000012  
2340 004502 104011  
2341 004504 000002  
2342  
2343 004506 012700 000015  
2344 004512 104011  
2345 004514 000002  
2346  
2347  
2348  
2349  
2350  
2351  
2352  
2353  
2354 004516 104002  
2355 004520 017642  
2356 004522 013700 001106  
2357 004526 006200  
2358 004530 006200  
2359 004532 006200  
2360 004534 042700 177770  
2361 004540 062700 000060  
2362 004544 104011  
2363 004546 013700 001106  
2364 004552 042700 177770  
2365 004556 062700 000060  
2366 004562 104011  
2367 004564 104006  
2368 004566 104010  
2369 004570 000002
```

```
*****  
:SCRLF-- A COMMON ROUTINE TO OUTPUT A CR AND LF TO  
:THE TEST TERMINAL ONLY.  
*****  
$SCRLF: MOVB #15,R0 ;SEND A CR  
ECHO ;WAIT UNTIL PRINTER IS READY  
MOVB #12,R0 ;SEND A LF  
ECHO  
RTI ;RETURN TO CALLER  
  
:XXXXXXXXXX  
:CRLF-- ROUTINES TO SEND A CR AND/OR LF TO ALL TERMINALS.  
:XXXXXXXXXX  
$CRLF: CR ;SEND CR  
$LF: MOV #12,R0 ;LF TO R0  
PRINTC ;SEND IT  
RTI ;RETURN  
$CR: MOV #15,R0 ;CR TO R0  
PRINTC ;SEND IT  
RTI ;RETURN  
  
:*****  
:ROUTINE TO PRINT TEST HEADER  
:*****  
$PRHDR: TYPEM ;PRINT MESSG  
HDRMSG  
MOV R1NNO,R0 ;GET TEST NUMBER  
ASR R0 ;GET FIRST DIGIT  
ASR R0  
ASR R0  
BIC #177770,R0 ;MASK FIRST DIGIT  
ADD #60,R0 ;MAKE ASCII  
PRINTC ;PRINT IT  
MOV R1NNO,R0 ;GET TEST NUMBER AGAIN  
BIC #177770,R0 ;MASK LAST DIGIT  
ADD #60,R0 ;MAKE ASCII  
PRINTC ;PRINT IT  
CRLF ;CR-LF  
LF ;BLANK LINE  
RTI ;RETURN
```

```
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384 004572
2385 004572 010046
2386 004574 010146
2387 004576 010246
2388 004600 010346
2389 004602 010446
2390 004604 010546
2391 004606 005737 001224
2392 004612 001402
2393 004614 000137 005474
2394 004620 005737 001230
2395 004624 001576
2396 004626 032777 020000 174410
2397 004634 001453
2398 004636 104021
2399 004640 104022
2400 004642 000446
2401
2402 004644 023727 001124 000007
2403 004652 001002
2404 004654 104024
2405 004656 000561
2406
2407 004660 023727 001124 000177
2408 004666 001402
2409 004670 000137 005416
2410 004674 023727 001106 000024
2411 004702 001004
2412 004704 012766 012224 000014
2413 004712 000543
2414 004714 023727 001106 000021
2415 004722 001004
2416 004724 012766 011234 000014
2417 004732 000533
2418 004734 023727 001106 000022
2419 004742 001004
2420 004744 012766 011314 000014
2421 004752 000523
2422 004754 000137 002540
2423 004760 000137 005432
2424 004764 013737 001134 001232
2425 004772 012705 017402

*****
:PRINTC--THIS ROUTINE IS USED TO DRIVE EACH OF THE EXISTING TERMINALS
:ON EACH OF THE EXISTING DH11'S( AS DEFINED BY THE SET UP IN ELTAB).
:IF IN THE MAINTENANCE MODE SR BIT 13 CONTROLS WHETHER OR NOT
:ALL DH11'S ARE DRIVEN OR ONLY THE TERMINAL UNDER TEST. SET
:BIT 13 DOWN TO DRIVE ALL TERMINALS ON ALL DH11'S. SET BIT 13 UP TO
:DRIVE ONLY THE TERMINAL UNDER TEST.
:EACH TERMINAL IS DRIVEN ONE CHARACTER AT A TIME.
:PRINTC WILL LOOP WAITING FOR THE FIRST TERMINAL TO BE READY
:ENTER WITH CHAR TO PRINT IN RO.
*****
CHGD10:
$PRTC:  MOV R0,-(SP)      ;SAVE R0
        MOV R1,-(SP)      ;SAVE R1
        MOV R2,-(SP)      ;SAVE R2
        MOV R3,-(SP)      ;SAVE R3
        MOV R4,-(SP)      ;SAVE R4
        MOV R5,-(SP)      ;SAVE R5
        TST IOSW          ;DH11 OR DJ11?
        BEQ 1$
        JMP PRINTJ        ;GO TO DJ11 ROUTINE
1$:     TST DHCNT         ;ANY DH11'S PRESENT?
        BEQ 12$          ;RETURN IF NONE
        BIT #BIT13,@SR   ;CHECK IF SR BIT13 IS SET
        BEQ 6$           ;DRIVE ALL TERMINALS IF NOT SET
        ECHO              ;OUTPUT CHAR
        INRDY            ;CHECK IF ANY INPUT
        BR 18$           ;NO,RETURN
;***** CHGD10 *****
        CMP TEMPCH,#7    ;INPUT, CHECK IF A ^G
        BNE 20$          ;IF NOT, BRANCH
        CKSWR            ;GO TO SOFT SWITCH REG INPUT
        BR 12$          ;RETURN TO EXIT TEST PROPERLY
;*****
20$:   CMP TEMPCH,#177   ;INPUT,CHECK IF A RUBOUT
        BEQ 2$
        JMP ENDITR      ;NO RUBOUT, RETURN
2$:   CMP RTNNO,#24     ;CHECK IF TEST 24
        BNE 3$         ;BRANCH IF NOT
        MOV #TERM,14(SP) ;SET RETURN ADR
        BR 12$         ;RETURN TO EXIT TEST PROPERLY
3$:   CMP RTNNO,#21     ;TEST 21?
        BNE 4$         ;BRANCH IF NOT
        MOV #EO21B,14(SP) ;SET RETURN TO EXIT TEST PROPERLY
        BR 12$         ;RETURN
4$:   CMP RTNNO,#22     ;TEST 22?
        BNE 5$         ;CONTINUE IF NOT
        MOV #EO22B,14(SP) ;SET RETURN ADR
        BR 12$         ;RETURN TO EXIT TEST PROPERLY
5$:   JMP TTY1H        ;GO WAIT
18$:   JMP ENDIT
6$:   MOV DHADR,SCR1   ;INIT ADDR OF FIRST DH11
        MOV #ELTAB,R5  ;INIT ADDR TO EXISTING TERM TAB
```

2426	004776	012704	016402			MOV	#DH1100,R4	:INIT ADDR TO VP TAB
2427	005002	013703	001230			MOV	DHCNT,R3	:INIT DH11 COUNT
2428	005006	012702	000001		7\$:	MOV	#1,R2	:INIT CURRENT LINE NO.
2429	005012	005001				CLR	R1	:SET UP CURRENT CHANNEL NUMBER
2430	005014	013737	001232	001234		MOV	SCR1,SCR2	:SET SCR2 = ADDR OF CURRENT DH11
2431	005022	062737	000012	001234		ADD	#12,SCR2	:SET SCR2 = ADDR OF BAR
2432	005030	031502			8\$:	BIT	@R5,R2	:TEST IF TERMINAL EXISTS
2433	005032	001155				BNE	17\$:BRANCH IF NO TERMINAL
2434	005034	037702	174174		9\$:	BIT	@SCR2,R2	:TEST IF OK TO SEND
2435	005040	001375				BNE	9\$:TEST AGAIN
2436	005042	062737	000004	001234		ADD	#4,SCR2	:ADDR OF SILO STATUS
2437	005050	112777	000020	174156		MOVB	#20,@SCR2	:SET SILO OVERFLOW TO 16
2438	005056	162737	000016	001234		SUB	#16,SCR2	:SET SCR2 AS ADDR OF SCR
2439	005064	110177	174144			MOVB	R1,@SCR2	:PUT CHANNEL NO. INTO SCR
2440	005070	062737	000002	001234		ADD	#2,SCR2	:SET CHAR BUF ADR
2441	005076	005777	174132			TST	@SCR2	:ANY INPUT?
2442	005102	100072				BPL	16\$:CONTINUE IF NONE
2443	005104	017737	174124	001124		MOV	@SCR2,TEMPCH	:GET CHAR
2444	005112	042737	177600	001124		BIC	#177600,TEMPCH	:MASK CHAR
2445						:***** CHGD10 *****		
2446	005120	023727	001124	000007		CMP	TEMPCH,#7	:CHECK IF ^G
2447	005126	001002				BNE	19\$:BRANCH IF NOT ^G
2448	005130	104024				CKSWR		:GO TO SOFT SWITCH REG INPUT
2449	005132	000433				BR	12\$:RETURN TO EXIT TEST PROPERLY
2450						:*****		
2451	005134	023727	001124	000177	19\$:	CMP	TEMPCH,#177	:CHECK IF RUBOUT
2452	005142	001032				BNE	14\$:BRANCH IF NOT RUBOUT
2453	005144	023727	001106	000024		CMP	RTNNO,#24	:TEST 24?
2454	005152	001004				BNE	10\$:BRANCH IF NOT
2455	005154	012766	012224	000014		MOV	#TERM,14(SP)	:SET RETURN ADR
2456	005162	000520				BR	ENDITD	:RETURN TO EXIT TEST PROPERLY
2457	005164	023727	001106	000021	10\$:	CMP	RTNNO,#21	:TEST 21?
2458	005172	001004				BNE	11\$:BRANCH IF NOT
2459	005174	012766	011234	000014		MOV	#EO21B,14(SP)	:SET RETURN ADR
2460	005202	000510				BR	ENDITD	:RETURN TO EXIT TEST PROPERLY
2461	005204	023727	001106	000022	11\$:	CMP	RTNNO,#22	:TEST 22?
2462	005212	001004				BNE	13\$:BRANCH IF NOT
2463	005214	012766	011314	000014		MOV	#EO22B,14(SP)	:SET RETURN ADR
2464	005222	000500			12\$:	BR	ENDITD	:RETURN TO EXIT TEST PROPERLY
2465	005224	000137	002540		13\$:	JMP	TTY1H	:CONTROL
2466	005230	023727	001124	000003	14\$:	CMP	TEMPCH,#3	:CHAR = CONTROL-C ?
2467	005236	001004				BNE	15\$:CONTINUE IF NOT
2468	005240	023727	001106	000024		CMP	RTNNO,#24	:TEST 24?
2469	005246	001466				BEQ	ENDITD	:EXIT IF TEST 24
2470	005250	013737	001124	001114	15\$:	MOV	TEMPCH,REPT	:SAVE CHAR FOR TESTS 21 AND 22
2471	005256	010046				MOV	R0,-(SP)	:SAVE R0
2472	005260	012700	000036			MOV	#30.,R0	:DELAY FOR HALF DUPLEX
2473	005264	104003				DELAY		
2474	005266	012600				MOV	(SP)+,R0	:RESTORE R0
2475	005270	062737	000002	001234	16\$:	ADD	#2,SCR2	:SCR2 EQ ADDR OF LPR
2476	005276	011477	173732			MOV	(R4),@SCR2	:STORE VP INTO LPR
2477	005302	062737	000002	001234		ADD	#2,SCR2	:ADD 2 TO ADDR IN SCR2
2478	005310	010146				MOV	R1,-(SP)	:SAVE R1
2479	005312	006301				ASL	R1	:FIND TABLE POINTER
2480	005314	006301				ASL	R1	:TO STORE CHAR
2481	005316	006301				ASL	R1	:FOR THIS CHANNEL

2482	005320	006301				ASL	R1	
2483	005322	06030i				ADD	R3,R1	
2484	005324	062701	006157			ADD	#CHARAC-1,R1	
2485	005330	110011				MOVB	R0,(R1)	;STORE CHAR
2486	005332	010177	173676			MOV	R1,@SCR2	;ADDR OF CHAR INTO CARA
2487	005336	012601				MOV	(SP)+,R1	;RESTORE R1
2488	005340	062737	000002	001234		ADD	#2,SCR2	;ADD 2 TO ADDR IN SCR2
2489	005346	012777	177777	173660		MOV	#177777,@SCR2	;SET CHAR COUNT EQ 1
2490	005354	062737	000002	001234		ADD	#2,SCR2	;ADD 2 TO ADDR IN SCR2
2491	005362	050277	173646			BIS	R2,@SCR2	;SET LINE BIT IN BAR
2492	005366	005724			17\$:	TST	(R4)+	;INC PTR TO VPTSB FOR NEXT TERM
2493	005370	005201				INC	R1	;INCREMENT CHANNEL NO.
2494	005372	006302				ASL	R2	;ROTATE LINE NO. MASK TO NEXT POS.
2495	005374	103215				BCC	8\$;DO NEXT TERM. ON SAME DH11
2496	005376	005303				DEC	R3	;DEC COUNT OF DH11'S
2497	005400	001414				BEQ	ENDIT	;BRANCH IF ALL DONE
2498	005402	062737	000020	001232		ADD	#20,SCR1	;SET UP FOR NEXT DH11
2499	005410	005725				TST	(R5)+	;INC PTR TO EXISTING TERM. TBL.
2500	005412	000137	005006			JMP	7\$;DO NEXT DH11
2501								
2502								
2503	005416	013737	001124	001114		ENDITR: MOV	TEMPCH,REPT	;SAVE CHAR FOR TESTS 21 & 22
2504	005424	012700	000036			ENDITD: MOV	#30.,R0	;DELAY FOR HALF DUPLEX
2505	005430	104003				DELAY		
2506								
2507	005432	012605				ENDIT: MOV	(SP)+,R5	
2508	005434	012604				MOV	(SP)+,R4	
2509	005436	012603				MOV	(SP)+,R3	
2510	005440	012602				MOV	(SP)+,R2	
2511	005442	012601				MOV	(SP)+,R1	
2512	005444	012600				MOV	(SP)+,R0	
2513	005446	023727	001124	000003		MOV	(SP)+,R0	
2514	005454	001006				CMP	TEMPCH,#3	;CHAR = CONTROL C ?
2515	005456	023727	001106	000024		BNE	1\$;EXIT IF NOT
2516	005464	001002				CMP	RTNNO,#24	;TEST 24?
2517	005466	012716	012140			BNE	1\$;EXIT IF NOT
2518	005472	000002			1\$:	MOV	#EO24R,(SP)	;YES, SET RETURN ADR
						RTI		;RETURN

2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531 005474 005737 001226
2532 005500 001754
2533 005502 032777 020000 173534
2534 005510 001441
2535 005512 104021
2536 005514 104022
2537 005516 000745
2538 005520 023727 001124 000177
2539 005526 001333
2540 005530 023727 001106 000024
2541 005536 001004
2542 005540 012766 012224 000014
2543 005546 000726
2544 005550 023727 001106 000021 1\$:
2545 005556 001004
2546 005560 012766 011234 000014
2547 005566 000716
2548 005570 023727 001106 000022 2\$:
2549 005576 001004
2550 005600 012766 011314 000014
2551 005606 000706
2552 005610 000137 002540 3\$:
2553 005614 012737 160010 001232 4\$:
2554 005622 012705 017402
2555 005626 013703 001226
2556 005632 012702 000001 5\$:
2557 005636 005001
2558 005640 013737 001232 001234 6\$:
2559 005646 062737 000004 001234
2560 005654 031502
2561 005656 001124
2562 005660 010277 173350
2563 005664 162737 000004 001234
2564 005672 105777 173336
2565 005676 100072
2566 005700 062737 000002 001234
2567 005706 017737 173322 001124
2568 005714 042737 177600 001124
2569 005722 023727 001124 000177
2570 005730 001032
2571 005732 023727 001106 000024
2572 005740 001004
2573 005742 012766 012224 000014
2574 005750 000625

```
*****  
:PRINTJ--THIS ROUTINE IS USED TO DRIVE EACH OF THE EXISTING TERMINALS  
: ON EACH OF THE EXISTING DJ11S' (AS DEFINED BY THE SET UP IN  
: "ELTAB"). IF IN THE MAINTENANCE MODE, SR BIT 13 CONTROLS  
: WHETHER OR NOT ALL DJ11S' ARE DRIVEN OR ONLY THE TERMINAL  
: UNDER TEST. EACH TERMINAL IS DRIVEN ONE AT A TIME. PRINTJ  
: WILL LOOP WAITING FOR THE FIRST ONE TO BE READY TO SEND  
: ENTER WITH CHARACTER TO SEND IN R0.  
*****  
PRINTJ: TST DJCNT ;ANY DJ11S'  
BEQ ENDIT ;NO, RETURN  
BIT #BIT13,@SR ;CHECK IF SR 13 IS SET  
BEQ 4$ ;CLEAR, DRIVE ALL TERMINALS  
ECHO ;OUTPUT CHAR TO TERM UNDER TEST  
INRDY ;CHECK IF ANY INPUT  
BR ENDIT ;NO, RETURN  
CMP TEMPCH,#177 ;YES CHECK IF A RUBOUT  
BNE ENDITR ;NO, RETURN  
CMP RTNNO,#24 ;TEST 24?  
BNE 1$ ;BRANCH IF NOT  
MOV #TERM,14(SP) ;SET RETURN ADR  
BR ENDITD ;RETURN TO EXIT TEST PROPERLY  
1$: CMP RTNNO,#21 ;TEST 21?  
BNE 2$ ;BRANCH IF NOT  
MOV #EQ21B,14(SP) ;SET RETURN ADR  
BR ENDITD ;RETURN TO EXIT TEST PROPERLY  
2$: CMP RTNNO,#22 ;TEST 22?  
BNE 3$ ;BRANCH IF NOT  
MOV #EQ22B,14(SP) ;SET RETURN ADR  
BR ENDITD ;RETURN TO EXIT TEST PROPERLY  
3$: JMP TTY1H ;GO WAIT  
4$: MOV #160010,SCR1 ;INIT ADDR OF FIRST DJ11  
MOV #ELTAB,R5 ;INIT ADDR OF EXISTING TERM TAB  
MOV DJCNT,R3 ;COUNT OF DJ11S' TO R3  
5$: MOV #1,R2 ;INIT CURRENT LINE NO.  
CLR R1 ;INIT CURRENT CHANNEL NO.  
6$: MOV SCR1,SCR2 ;SET SCR2=ADDR OF CURRENT DJ11  
ADD #4,SCR2 ;SCR2 IS ADDR OF TCR  
BIT @R5,R2 ;TEST IF TERMINAL EXISTS  
BNE 15$ ;BRANCH IF NO TERMINAL  
MOV R2,@SCR2 ;YES, SET LINE NO. IN TCR  
SUB #4,SCR2 ;SCR2 IS NOW ADDR OF CSR  
TSTB @SCR2 ;CHECK FOR INPUTS  
BPL 13$ ;CONTINUE IF NO INPUT  
ADD #2,SCR2 ;SET CHAR BUF REG ADR  
MOV @SCR2,TEMPCH ;GET INPUT CHAR  
BIC #177600,TEMPCH ;MASK CHAR  
CMP TEMPCH,#177 ;CHECK CHAR  
BNE 11$ ;BRANCH IF NOT RUBOUT  
CMP RTNNO,#24 ;TEST 24?  
BNE 7$ ;BRANCH IF NOT  
MOV #TERM,14(SP) ;SET RETURN ADR  
BR ENDITD ;RETURN TO EXIT TEST PROPERLY
```

2575	005752	023727	001106	000021	7\$:	CMP	RTNNO,#21	:TEST 21?
2576	005760	001004				BNE	8\$:BRANCH IF NOT
2577	005762	012766	011234	000014		MOV	#EO21B,14(SP)	:SET RETURN ADR
2578	005770	000615				BR	ENDITD	:RETURN TO EXIT TEST PROPERLY
2579	005772	023727	001106	000022	8\$:	CMP	RTNNO,#22	:TEST 22?
2580	006000	001004				BNE	10\$:BRANCH IF NOT
2581	006002	012766	011314	000014		MOV	#EO22B,14(SP)	:SET RETURN ADR
2582	006010	000605			9\$:	BR	ENDITD	:RETURN TO EXIT TEST PROPERLY
2583	006012	000137	002540		10\$:	JMP	TTY1H	:GO TO CONTROL
2584	006016	023727	001124	000003	11\$:	CMP	TEMPCH,#3	:CHAR = CONTROL-C ?
2585	006024	001004				BNE	12\$:BRANCH IF NOT
2586	006026	023727	001106	000024		CMP	RTNNO,#24	:TEST 24?
2587	006034	001765				BEQ	9\$:YES, RETURN TO TEST
2588	006036	013737	001124	001114	12\$:	MOV	TEMPCH,REPT	:SAVE CHAR FOR TESTS 21 & 22
2589	006044	010046				MOV	RO,-(SP)	:SAVE RO
2590	006046	012700	000036			MOV	#30.,RO	:DELAY FOR HALF DUPLEX
2591	006052	104003				DELAY		
2592	006054	012600				MOV	(SP)+,RO	:RESTORE RO
2593	006056	162737	000002	001234		SUB	#2,SCR2	:RESET PROPER ADR FOR PRINT
2594	006064	052777	000400	173142	13\$:	BIS	#BIT8,@SCR2	:START TRANSMITTER SCANNER
2595	006072	005777	173136		14\$:	TST	@SCR2	:CHECK STATE OF CSR
2596	006076	100375				BPL	14\$:BRANCH IF XMITTER NOT READY
2597	006100	062737	000006	001234		ADD	#6,SCR2	:READY, SCR2 NOW=ADDR OF TBUF
2598	006106	017704	173122			MOV	@SCR2,R4	:GET CONTENTS OF TBUF
2599	006112	000304				SWAB	R4	
2600	006114	042704	177760			BIC	#177760,R4	:SAVE ONLY LINE NO.
2601	006120	020401				CMP	R4,R1	:BE SURE SCANNER IS ON CORRECT LINE
2602	006122	001246				BNE	6\$:BRANCH IF NOT, TRY AGAIN
2603	006124	110077	173104			MOVB	RO,@SCR2	:LOAD TBUF WITH CHAR
2604	006130	005201			15\$:	INC	R1	:INCREMENT LINE NO.
2605	006132	006302				ASL	R2	:SET LINE NO. TO NEXT POSITION
2606	006134	103241				BCC	6\$:IF CARRY IS CLEAR DJ11 IS DONE
2607	006136	005303				DEC	R3	:DEC COUNT OF DJ11S
2608	006140	001002				BNE	16\$:BRANCH IF NOT DONE
2609	006142	000137	005432			JMP	ENDIT	:DONE
2610	006146	062737	000010	001232	16\$:	ADD	#10,SCR1	:SET SCR1 TO ADDR OF NEXT DJ11
2611	006154	005725				TST	(R5)+	:SET R5 TO NEXT ELTAB ENTRY
2612	006156	000625				BR	5\$:GO DO NEXT DJ11
2613								
2614								
2615								
2616	006160	000400				CHARAC:	.BLKB 256.	:CHARACTER STORAGE FOR OUTPUT

2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652

006560 000000
006562 006634
006564 104012
006566 104002
006570 017654
006572 012703 025125
006576 012702 000020
006602 010300
006604 013701 001112
006610 104011
006612 000300
006614 005301
006616 001374
006620 000303
006622 104006
006624 005302
006626 001365
006630 104001
006632 000763

.SBTTL PRINTER TESTS

:XXXXXXXXXX
:PTO -- DATA PATH TEST---FOUR LINES OF ALTERNATING
: '*'' AND 'U' ARE PRINTED, OUT TO THE GIVEN PAPER
: WIDTH THE PATTERN WILL APPEAR AS FOLOWS.
:
: *U*U*U*U*U*U
: U*U*U*U*U*U*
: *U*U*U*U*U*U
: U*U*U*U*U*U*
:XXXXXXXXXX

PTO: 0 ;TEST NUMBER
PT1 ;NEXT TEST
PRTHDR ;PRINT TEST HEADER
TYPEM ;PRINT COLUMN MESSG
HDRO
1\$: MOV #'U',R3 ;SET FIRST CHAR PAIR
MOV #16.,R2 ;SET LINE COUNT
2\$: MOV R3,R0 ;SET CHAR PAIR
MOV WIDTH,R1 ;SET COLUMN COUNT
3\$: PRINTC ;PRINT CHAR
SWAB R0 ;SET NEXT CHAR
DEC R1 ;DEC COLUMN COUNT
BNE 3\$;FINISH LINE
SWAB R3 ;SET NEXT START CHAR
CRLF ;CR-LF
DEC R2 ;DEC LINE COUNT
BNE 2\$;FINISH TEST
CHAIN ;ALL DONE, EXIT
BR 2\$;REPEAT TEST

2653
2654
2655
2656
2657
2658
2659 006634 000001
2660 006636 006756
2661 006640 104012
2662 006642 012701 000040
2663 006646 012702 000100
2664 006652 012703 000140
2665 006656 010100
2666 006660 004737 006726
2667 006664 010200
2668 006666 004737 006726
2669 006672 012704 000003
2670 006676 010300
2671 006700 104011
2672 006702 005304
2673 006704 001375
2674 006706 104006
2675 006710 122122
2676 006712 105723
2677 006714 020327 000200
2678 006720 103756
2679 006722 104001
2680 006724 000746
2681
2682 006726 012704 000003
2683 006732 104011
2684 006734 005304
2685 006736 001375
2686
2687 006740 012700 000040
2688 006744 104011
2689
2690 006746 012700 000040
2691 006752 104011
2692 006754 000207

```

:XXXXXXXXXX
:PT1 -- PRINTER CHARACTER TEST --- PRINTS ALL PRINTABLE CHARACTERS
:XXXXXXXXXX
PT1: 1 ;TEST NUMBER
PT2 ;NEXT TEST
PRTHDR ;PRINT TEST HEADER
1$: MOV #40,R1 ;SPACE TO R1
MOV #100,R2 ;@ TO R2
MOV #140,R3 ;\ TO R3
2$: MOV R1,R0 ;FIRST CHAR TO R0
JSR PC,SPSP ;SEND TWO SPACES
MOV R2,R0 ;SECOND CHAR TO R0
JSR PC,SPSP ;SEND TWO SPACES
MOV #3,R4 ;CHAR COUNT TO R4
MOV R3,R0 ;THIRD CHAR TO R0
3$: PRINTC ;PRINT CHAR
DEC R4 ;THREE TIMES ?
BNE 3$ ;BRANCH IF NOT
CRLF ;CR-LF
CMPB (R1)+,(R2)+ ;SET NEXT CHARS
TSTB (R3)+
CMP R3,#200 ;CHECK IF ALL DONE
BLO 2$ ;BRANCH IF NOT
CHAIN ;EXIT TO NEXT TEST
BR 1$ ;REPEAT TEST

SPSP: MOV #3,R4 ;SET PRINT COUNT
1$: PRINTC ;PRINT CHAR
DEC R4 ;DEC PRINT COUNT
BNE 1$ ;FINISH CHAR

SP2: MOV #40,R0 ;SPACE TO R0
PRINTC ;PRINT FIRST SPACE

SPC: MOV #40,R0 ;SPACE TO R0
PRINTC ;PRINT SECOND SPACE
RTS PC ;RETURN
```

2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705 006756 000002
2706 006760 007406
2707 006762 104012
2708 006764 005003
2709 006766 012701 007144
2710 006772 012702 000003
2711 006776 012704 000010
2712 007002 112100
2713 007004 104011
2714 007006 005304
2715 007010 001374
2716 007012 022703 000002
2717 007016 001420
2718 007020 022703 000004
2719 007024 001414
2720 007026 022703 000033
2721 007032 001412
2722 007034 022703 000007
2723 007040 001002
2724 007042 012703 000020
2725 007046 022703 000040
2726 007052 001003
2727 007054 000421
2728 007056 005203
2729 007060 005203
2730 007062 010300
2731 007064 012704 000003
2732 007070 104011
2733 007072 005304
2734 007074 001375
2735 007076 005203
2736 007100 005302
2737 007102 001404
2738 007104 004737 006740
2739 007110 104011
2740 007112 000731
2741 007114 104006
2742 007116 000725
2743 007120 012704 000003
2744 007124 012700 000177
2745 007130 104011
2746 007132 005304
2747 007134 001375
2748 007136 104006

:XXXXXXXXXX

:PT2 -- NON-PRINTING CHARACTER TEST. THIS TEST
PRINTS THE OCTAL CODE FOLLOWED BY THE MNEMONIC
OF ALL NON-PRINTING CHARACTERS. FOLLOWING EACH
MNEMONIC, THE PRINTER IS DRIVEN BY THE NON-PRINTING
CODE (000 THROUGH 037 PLUS 177). ALL CONTROL CHARACTERS
(INCLUDING THOSE FOR OPTIONS) WILL BE SKIPPED.
REFER TO THE DOCUMENT FOR A LIST OF THOSE PRINTED.

:XXXXXXXXXX

PT2: 2 ;TEST NUMBER
PT3 ;NEXT TEST
PRTHDR ;PRINT TEST HEADER
1\$: CLR R3 ;NON-PRINTABLE CHAR..NULL IS FIRST
MOV #LINE2,R1 ;ADDR OF CHAR STRING TO R1
2\$: MOV #3,R2 ;NO. OF CHAR GROUPS PER LINE
3\$: MOV #10,R4 ;NO. OF CHARS PER GROUP
4\$: MOVB (R1)+,R0 ;CHAR INTO R0
PRINTC ;PRINT CHAR
DEC R4 ;9 CHARS. PRINTED?
BNE 4\$;BRANCH IF NOT
CMP #2,R3 ;CHAR = STX?
BEQ 7\$;YES, SET NEXT CHAR
CMP #4,R3 ;CHAR = EOT?
BEQ 6\$;YES, SET NEXT CHAR
CMP #33,R3 ;CHAR = ESC?
BEQ 7\$;YES, SET NEXT CHAR
CMP #7,R3 ;CHAR = BELL?
BNE 5\$;BRANCH IF NOT A BELL
MOV #20,R3 ;SET NEXT CHAR
5\$: CMP #40,R3 ;IS IT THE LAST?
BNE 8\$;BRANCH IF NO
BR 11\$;YES, OUTPUT LAST CHAR (177)
6\$: INC R3 ;SKIP CHAR
7\$: INC R3 ;SKIP CHAR
8\$: MOV R3,R0 ;NON-PRINTABLE CHAR TO R0
MOV #3,R4 ;A COUNT OF 3 TO R4
9\$: PRINTC ;DRIVE PRINTER WITH NON-PRINTABLE CHAR
DEC R4 ;DECREMENT COUNTER
BNE 9\$;BRANCH IF NOT ZERO (3 TIMES)
INC R3 ;INCREMENT CHAR.CODE
DEC R2 ;DEC. GROUPS PER LINE COUNTER (3)
BEQ 10\$;BRANCH IF ZERO
JSR PC,SP2 ;SEND 3 SPACES
BR 3\$;CONTINUE
10\$: CRLF ;SEND A CR,LF
BR 2\$;GO DO NEXT LINE
11\$: MOV #3,R4 ;A 3 COUNT TO R4
MOV #177,R0 ;A DEL TO R0
12\$: PRINTC ;PRINT CHAR
DEC R4 ;DECREMENT COUNTER
BNE 12\$;BRANCH IF NOT ZERO
CRLF ;SEND A CR, LF

Line	Code 1	Code 2	Code 3	Code 4	Code 5	Code 6	Code 7	Code 8	Code 9	Code 10
2749	007140	104001								
2750	007142	000710								
2751										
2752										
2753	007144	030060	020060	047040	LINE2:	.ASCII	/000	NUL001	SOH002	STX/
2754	007152	046125	030060	020061						
2755	007160	051440	044117	030060						
2756	007166	020062	051440	054124						
2757	007174	030060	020066	040440		.ASCII	/006	ACK020	DLE021	DC1/
2758	007202	045503	031060	020060						
2759	007210	042040	042514	031060						
2760	007216	020061	042040	030503						
2761	007224	031060	020062	042040		.ASCII	/022	DC2023	DC3024	DC4/
2762	007232	031103	031060	020063						
2763	007240	042040	031503	031060						
2764	007246	020064	042040	032103						
2765	007254	031060	020065	047040		.ASCII	/025	NAK026	SYN027	ETB/
2766	007262	045501	031060	020066						
2767	007270	051440	047131	031060						
2768	007276	020067	042440	041124						
2769	007304	031460	020060	041440		.ASCII	/030	CAN031	EM 032	SUB/
2770	007312	047101	031460	020061						
2771	007320	042440	020115	031460						
2772	007326	020062	051440	041125						
2773	007334	031460	020064	043040		.ASCII	/034	FS 035	GS 036	RS /
2774	007342	020123	031460	020065						
2775	007350	043440	020123	031460						
2776	007356	020066	051040	020123						
2777	007364	031460	020067	052440		.ASCII	/037	US 177	DEL /	
2778	007372	020123	033461	020067						
2779	007400	042040	046105	040						
2780	007406					.EVEN				

```
2781 ;XXXXXXXXXX
2782 :
2783 ;PT3 -- CARRIAGE RETURN TEST --
2784 : THE LINE CONSISTS OF A STRING OF O'S AND
2785 : X'S. FIRST, THE O'S ARE PRINTED OUT TO THE LAST
2786 : COLUMN WITH A SPACE SEPARATING EACH. THEN THE
2787 : CARRIAGE IS SPACED TO THE FIRST BLANK SPACE, AND X
2788 : PRINTED AND THE RETURNED TO THE MARGIN. THIS
2789 : PROCESS IS CONTINUE UNTIL ALL SPACES BETWEEN
2790 : THE ZEROES HAVE BEEN FILLED.
2791 :
2792 ;XXXXXXXXXX
2793 :
2794 007406 000003 PT3: 3 ;TEST NUMBER
2795 007410 007526 PT4 ;NEXT TEST
2796 007412 104012 PRTHDR ;PRINT TEST HEADER
2797 007414 005037 001116 1$: CLR SPCNT ;CLEAR SPACE COUNTER
2798 007420 013701 001112 MOV WIDTH,R1 ;POSITION COUNTER TO R1
2799 007424 012700 000117 2$: MOV #117,R0 ;"O" TO R0
2800 007430 104011 PRINTC ;PRINT THE 'O'
2801 007432 005301 DEC R1 ;DECREMENT POSITION COUNTER
2802 007434 001404 BEQ 3$ ;BRANCH IF 0
2803 007436 004737 006746 JSR PC,SPC ;PRINT A SPACE
2804 007442 005301 DEC R1 ;DECREMENT POSITION COUNTER
2805 007444 001367 BNE 2$ ;BRANCH IF NOT ZERO
2806 007446 104014 3$: CR ;SEND CR
2807 007450 012737 000001 001116 MOV #1,SPCNT ;SPACE, COUNTER SET TO 1
2808 007456 013701 001116 4$: MOV SPCNT,R1 ;NO. OF SPACES TO R1
2809 007462 004737 006746 5$: JSR PC,SPC ;PRINT SPACE
2810 007466 005301 DEC R1 ;DECREMENT SPACE COUNTER
2811 007470 001374 BNE 5$ ;BRANCH IF NOT ZERO
2812 007472 012700 000130 MOV #130,R0 ;"X" INTO R0
2813 007476 104011 PRINTC ;PRINT 'X'
2814 007500 104014 CR ;SEND CR
2815 007502 062737 000002 001116 ADD #2,SPCNT ;INCREMENT SPACE COUNT BY 2
2816 007510 023737 001116 001112 CMP SPCNT,WIDTH ;COMPARE POSITION COUNTER WITH COLM. COUNT
2817 007516 103757 BLO 4$ ;BRANCH IF LOWER
2818 007520 104010 LF ;SEND LF
2819 007522 104001 CHAIN ;CHAIN TO NEXT TEST
2820 007524 000733 BR 1$ ;REPEAT TEST
```

```
2821 ;XXXXXXXXXX
2822 ;
2823 ;PT4 -- MULTIPLE LINE FEED TEST -- 63 LINE FEEDS ARE
2824 ; SENT WITH A REFERENCE LINE AT THE START AND END.
2825 ; A NUMBER IS PRINTED WHICH INDICATES THE NUMBER OF LINE
2826 ; FEEDS THAT WILL BE ISSUED BEFORE THE NEXT
2827 ; NUMBER OR REFERENCE LINE IS PRINTED.
2828 ;
2829 ;XXXXXXXXXX
2830 ;
2831 007526 000004 PT4: 4 ;TEST NUMBER
2832 007530 007704 PT5 ;NEXT TEST
2833 007532 104012 PRTHDR ;PRINT TEST HEADER
2834 007534 012737 000001 001132 1$: MOV #1, LFCNT ;LINE FEED COUNT TO 1
2835 007542 013701 001112 MOV WIDTH, R1 ;COLUMN COUNT TO R1
2836 007546 012702 007666 MOV #LINE3, R2 ;ADDR OF NUMBER FIELD TO R2
2837 007552 004737 007636 JSR PC, REF ;PRINT REFERENCE LINE
2838 007556 013701 001132 2$: MOV LFCNT, R1 ;LINE FEED COUNT TO R1
2839 007562 104010 3$: LF ;SEND LF
2840 007564 005301 DEC R1 ;DECREMENT COUNTER
2841 007566 001375 BNE 3$ ;BRANCH IF NOT YET 0
2842 007570 006337 001132 ASL LFCNT ;DOUBLE LINE FEED COUNT
2843 007574 022737 000100 001132 CMP #BIT6, LFCNT ;TEST IF COUNT IS 32
2844 007602 001406 BEQ 4$ ;BRANCH IF =32, END
2845 007604 112200 MOVB (R2)+, R0 ;NUMBER TO R0
2846 007606 104011 PRINTC ;PRINT IT
2847 007610 112200 MOVB (R2)+, R0 ;NUMBER TO R0
2848 007612 104011 PRINTC ;PRINT IT
2849 007614 104014 CR ;SEND CR
2850 007616 000757 BR 2$ ;DRIVE THE LINEFEEDS
2851 007620 013701 001112 4$: MOV WIDTH, R1 ;COLUMN COUNT TO R1
2852 007624 004737 007636 JSR PC, REF ;SEND END REFERENCE LINE
2853 007630 104010 LF ;SEND LF
2854 007632 104001 CHAIN
2855 007634 000737 BR 1$ ;REPEAT TEST
2856 ;
2857 007636 112200 REF: MOVB (R2)+, R0 ;NUMBER TO R0
2858 007640 104011 PRINTC ;PRINT IT
2859 007642 112200 MOVB (R2)+, R0 ;NUMBER TO R0
2860 007644 104011 PRINTC ;PRINT IT
2861 007646 005741 TST -(R1) ;DECREASE COUNTER BY 2
2862 007650 012700 000137 MOV #137, R0 ;DASH (-) TO R0
2863 007654 104011 1$: PRINTC ;PRINT IT
2864 007656 005301 DEC R1 ;DECREMENT COLUMN COUNTER
2865 007660 001375 BNE 1$ ;BRANCH IF NO ZERO
2866 007662 104014 CR ;SEND CR
2867 007664 000207 RTS PC ;RETURN
2868 ;
2869 007666 030460 031060 032060 LINE3: .ASCII /01020408163200/
2870 007674 034060 033061 031063
2871 007702 030060
2872 .EVEN
```

```
2873 ;XXXXXXXXXX
2874 ;PT5-- SINGLE LINE FEED TEST -- TESTS THE LINE FEED
2875 ; CAPABILITY FROM ALL COLUMNS.
2876 ;XXXXXXXXXX
2877
2878 007704 000005 PT5: S ;TEST NUMBER
2879 007706 010110 PT6 ;NEXT TEST
2880 007710 104012 PRTHDR ;PRINT TEST HEADER
2881 007712 013701 001112 1$: MOV WIDTH,R1 ;COLUMN COUNT TO R1
2882 007716 005741 TST -(R1) ;DECREASE BY 2
2883 007720 012700 000060 MOV #60,R0 ;'0' TO R0
2884 007724 104011 2$: PRINTC ;SEND 0
2885 007726 005301 DEC R1 ;DECREMENT COLUMN COUNTER
2886 007730 001375 BNE 2$ ;BRANCH IF NOT ZERO
2887 007732 012700 000062 MOV #62,R0 ;SEND A 2
2888 007736 104011 PRINTC
2889 007740 104011 PRINTC ;SEND A SECOND TWO
2890 007742 023727 001112 000204 CMP WIDTH,#132. ;COMPARE COLUMN COUNT
2891 007750 001404 BEQ 3$ ;BRANCH IF EQ 132
2892 007752 012700 003410 MOV #3410,R0 ;DELAY 1.8 SEC
2893 007756 104003 DELAY
2894 007760 000407 BR 5$
2895 007762 012700 000063 3$: MOV #63,R0 ;3'S TO R0
2896 007766 012701 000100 MOV #100,R1 ;64 TO COUNTER
2897 007772 104011 4$: PRINTC ;SEND CHARACTER
2898 007774 005301 DEC R1 ;DECREMENT COUNT
2899 007776 001375 BNE 4$ ;BRANCH IF NOT ZERO
2900 010000 104006 5$: CRLF ;SEND A CR,LF
2901 010002 013701 001112 MOV WIDTH,R1 ;NO. COLUMNS TO R1
2902 010006 012700 000134 6$: MOV #134,R0 ;BACKSLASH TO R0
2903 010012 104011 PRINTC ;SEND IT
2904 010014 104010 LF ;SEND LF
2905 010016 005301 DEC R1 ;DECREMENT COUNTER
2906 010020 001372 BNE 6$ ;BRANCH IF NOT ZERO.
2907 010022 104014 CR ;SEND CR
2908 010024 004737 010052 JSR PC,PT5AL ;SEND REF LINE #1
2909 010030 104006 CRLF ;SEND A CR,LF
2910 010032 012700 001750 MOV #1750,R0 ;DELAY 1 SEC
2911 010036 104003 DELAY
2912 010040 004737 010052 JSR PC,PT5AL ;SEND A SECOND REF. LINE
2913 010044 104006 CRLF ;SEND A CR,LF
2914 010046 104001 CHAIN ;CHAIN TO NEXT TEST
2915 010050 000720 BR 1$ ;REPEAT TEST
2916 010052 013701 001112 PT5AL: MOV WIDTH,R1 ;COLUMN COUNT TO R1
2917 010056 012700 000061 MOV #61,R0 ;'1' TO R0
2918 010062 104011 1$: PRINTC ;PRINT R0
2919 010064 005301 DEC R1 ;DECREMENT COUNTER
2920 010066 001407 BEQ 2$ ;BRANCH IF=0
2921 010070 005200 INC R0 ;INCREMENT CHARACTER
2922 010072 020027 000071 CMP R0,#71 ;COMP CHAR TO '9'
2923 010076 101771 BLOS 1$ ;BRANCH IF LOWER OR SAME
2924 010100 012700 000060 MOV #60,R0 ;RESET CHAR TO '0'
2925 010104 000766 BR 1$ ;CONTINUE
2926 010106 000207 2$: RTS PC ;FINISHED, RETURN TO CALLER
```

2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938 010110 000006
2939 010112 010272
2940 010114 104012
2941 010116 013701 001112
2942 010122 005741
2943 010124 012700 000060
2944 010130 104011
2945 010132 005301
2946 010134 001375
2947 010136 012700 000062
2948 010142 104011
2949 010144 104011
2950 010146 023727 001112 000204
2951 010154 001404
2952 010156 012700 003410
2953 010162 104003
2954 010164 000407
2955 010166 012700 000063
2956 010172 012701 000100
2957 010176 104011
2958 010200 005301
2959 010202 001375
2960 010204 104006
2961 010206 013701 001112
2962 010212 012700 000134
2963 010216 104011
2964 010220 012700 000010
2965 010224 104011
2966 010226 012700 000057
2967 010232 104011
2968 010234 005301
2969 010236 001365
2970 010240 104010
2971 010242 104014
2972 010244 004737 010052
2973 010250 104006
2974 010252 012700 001750
2975 010256 104003
2976 010260 004737 010052
2977 010264 104006
2978 010266 104001
2979 010270 000712

:XXXXXXXXXX

:PT6-- BACKSPACE TEST -- A REFERENCE LINE SUCH AS IN
:TEST PT5 IS PRINTED. THE SECOND LINE CONSISTS
:OF PRINTING A BACKSLASH, BACKSPACE AND FORWARD
:SLASH COMBINATION OUT TO THE GIVEN COLUMN WIDTH.
:THIS LINE IS THEN FOLLOWED BY THE SAME TWO REFERENCE
:LINES AS PRINTED IN TEST PT5.

:XXXXXXXXXX

PT6: 6 ;TEST NUMBER
PT7 ;NEXT TEST
PRTHDR ;PRINT TEST HEADER
1\$: MOV WIDTH,R1 ;COLUMN COUNT TO R1
TST -(R1) ;DECREMENT BY 2
MOV #60,R0 ;'0' TO R0
2\$: PRINTC ;SEND 0
DEC R1 ;DECREMENT COLUMN COUNTER
BNE 2\$;BRANCH IF NOT ZERO
MOV #62,R0 ;'2' TO R0
PRINTC ;SEND A '2'
PRINTC ;SEND A SECOND '2'
CMP WIDTH,#132. ;COMPARE COLUMN COUNT
BEQ 3\$;BRANCH IF EQ 132
MOV #3410,R0 ;DELAY 1.8 SEC
DELAY
BR 5\$
3\$: MOV #63,R0 ;3'S TO R0
MOV #100,R1 ;64 TO COUNTER
4\$: PRINTC ;SEND CHAR
DEC R1 ;DECREMENT COUNTER
BNE 4\$;CONTINUE IF NOT DONE
5\$: CRLF ;SEND A CR,LF
MOV WIDTH,R1 ;COLUMN COUNT TO R1
6\$: MOV #134,R0 ;BACKSLASH TO R0
PRINTC ;SEND IT
MOV #10,R0 ;BACKSPACE TO R0
PRINTC ;SEND IT
MOV #57,R0 ;FORWARD SLASH TO R0
PRINTC ;SEND IT
DEC R1 ;END OF PAPER
BNE 6\$;BRANCH IF NO
LF ;SEND LF
CR ;SEND CR
JSR PC,PT5AL ;SEND REF LINE #1
CRLF ;SEND A CR,LF
MOV #1750,R0 ;DELAY 1 SEC
DELAY
JSR PC,PT5AL ;SEND SECOND REF LINE
CRLF ;SEND A CR,LF
CHAIN ;CHAIN TO NEXT TEST
BR 1\$;REPEAT TEST


```
2980 ;XXXXXXXXXX
2981 ;
2982 ;PT7-- OVERPRINT TEST-- A ROW OF ALTERNATING M'S AND
2983 ; SPACES ARE PRINTED, OUT TO THE LAST COLUMN AND OVERPRINTED TWICE.
2984 ; A SECOND LINE OF ALTERNATING SPACES AND 'a'S' IS THEN
2985 ; SENT 3 TIMES AS THE FIRST LINE. THIS IS FOLLOWED
2986 ; BY A THIPD AND FINAL LINE OF ALTERNATING '8'
2987 ; AND SPACES.
2988 ;
2989 ;XXXXXXXXXX
2990
2991 010272 000007 PT7: 7 ;TEST NUMBER
2992 010274 010504 PT10 ;NEXT TEST
2993 010276 104012 PRTHDR ;PRINT TEST HEADER
2994 010300 012703 000002 1$: MOV #2,R3 ;2 COUNT TO R3
2995 010304 013701 001112 2$: MOV WIDTH,R1 ;NO. OF COLUMNS TO R1
2996 010310 012700 000115 3$: MOV #115,R0 ;PRINT M
2997 010314 104011 PRINTC
2998 010316 005301 DEC R1 ;END OF LINE
2999 010320 001404 BEQ 4$ ;BRANCH IF YES
3000 010322 004737 006746 JSR PC,SPC ;SEND SPACE
3001 010326 005301 DEC R1 ;END OF LINE?
3002 010330 001367 BNE 3$ ;BRANCH IF NO
3003 010332 022703 000002 4$: CMP #2,R3 ;TEST R3
3004 010336 001003 BNE 6$ ;BRANCH IF NOT FIRST TIME
3005 010340 104014 5$: CR ;SEND CR
3006 010342 005303 DEC R3 ;DECREASE LINE COUNTER
3007 010344 000757 BR 2$ ;REPEAT LINE
3008 010346 005703 6$: TST R3 ;THIRD TIME?
3009 010350 001373 BNE 5$ ;BRANCH IF NOT
3010 010352 104006 CRLF ;NEXT LINE
3011 010354 005723 TST (R3)+ ;REPEAT COUNTER TO R3
3012 010356 013701 001112 7$: MOV WIDTH,R1 ;COLUMN COUNT TO R1
3013 010362 004737 006746 8$: JSR PC,SPC ;SEND SPACE
3014 010366 005301 DEC R1 ;DECREASE COLUMN COUNT
3015 010370 001405 BEQ 9$ ;BRANCH IF 0, END OF LINE
3016 010372 012700 000100 MOV #100,R0 ;'a' TO R0
3017 010376 104011 PRINTC ;SEND IT
3018 010400 005301 DEC R1 ;DECREASE COLUMN COUNT
3019 010402 001367 BNE 8$ ;BRANCH IF NOT 0 (NOT END)
3020 010404 022703 000002 9$: CMP #2,R3 ;END OF LINE, FIRST TIME?
3021 010410 001003 BNE 11$ ;BRANCH IF NOT
3022 010412 104014 10$: CR ;SEND CR
3023 010414 005303 DEC R3 ;DECREASE LINE COUNTER
3024 010416 000757 BR 7$ ;REPEAT LINE
3025 010420 005703 11$: TST R3 ;TEST IF THIRD REPEAT
3026 010422 001373 BNE 10$ ;BRANCH IF NOT
3027 010424 104006 CRLF ;DO NEXT LINE
3028 010426 005723 TST (R3)+ ;LINE REPEAT COUNTER TO R3
3029 010430 013701 001112 12$: MOV WIDTH,R1 ;COLUMN COUNT TO R1
3030 010434 012700 000046 13$: MOV #46,R0 ;'8' TO R0
3031 010440 104011 PRINTC ;PRINT IT
3032 010442 005301 DEC R1 ;DECREASE COLUMN COUNT
3033 010444 001404 BEQ 14$ ;BRANCH IF END
3034 010446 004737 006746 JSR PC,SPC ;SEND SPACE
3035 010452 005301 DEC R1 ;DECREASE COLUMN COUNT
```

3036	010454	001367		BNE	13\$:BRANCH IF NOT END
3037	010456	022703	000002	14\$: CMP	#2,R3	:TEST IF FIRST TIME
3038	010462	001003		BNE	16\$:BRANCH IF =2, FIRST TIME
3039	010464	104014		15\$: CR		:CARRIAGE RETURN
3040	010466	005303		DEC	R3	:DECREASE REPEAT COUNTER
3041	010470	000757		BR	12\$:PRINT LINE AGAIN
3042	010472	005703		16\$: TST	R3	:TEST IF END, R3=0
3043	010474	001373		BNE	15\$:BRANCH IF NOT END
3044	010476	104006		CRLF		:SEND CR,LF
3045	010500	104001		CHAIN		:CHAIN TO NEXT TEST
3046	010502	000676		BR	1\$:REPEAT TEST

3047
 3048
 3049
 3050
 3051
 3052
 3053
 3054
 3055
 3056
 3057
 3058
 3059
 3060
 3061
 3062
 3063
 3064
 3065
 3066
 3067
 3068
 3069
 3070
 3071
 3072
 3073
 3074
 3075
 3076
 3077
 3078
 3079
 3080
 3081
 3082
 3083
 3084
 3085
 3086
 3087
 3088
 3089
 3090
 3091
 3092
 3093
 3094
 3095

010504 000010
 010506 010642
 010510 104012
 010512 012701 000036
 010516 012702 000170
 010522 012737 000036 010540
 010530 012700 000110
 010534 104011
 010536 012700 000036
 010542 104003
 010544 005301
 010546 001426
 010550 005302
 010552 001430
 010554 013704 010540
 010560 006204
 010562 006204
 010564 006204
 010566 006204
 010570 010405
 010572 006204
 010574 006204
 010576 006204
 010600 060405
 010602 022702 000074
 010606 003403
 010610 160537 010540
 010614 000745
 010616 060537 010540
 010622 000742
 010624 104006
 010626 012701 000036
 010632 000746
 010634 104006
 010636 104001
 010640 000724

010540

010540

:XXXXXXXXXX
 :PT10-- PRINTING FREQUENCY TEST-- 120 H'S ARE PRINTED ON 4 LINES
 : 30 PER LINE. THE TEST IS SUCH THAT BETWEEN THE FIRST AND SECOND
 : 'H' A 30 MSEC DELAY IS INTRODUCED. THIS DELAY IS THEN INCREASED
 : BETWEEN CHARACTERS OUT TO 60 CHARACTERS IN AN EXPONENTIAL
 : MANNER. THE DELAY IS THEN DECREASED IN THE SAME MANNER OUT TO THE
 : 120TH CHARACTER. THIS DELAY IS CALCULATED AS FOLLOWS:
 :
 : NEW DELAY = OLD DELAY [+ OR -] (OLD DELAY/16 + OLD DELAY/128)
 :XXXXXXXXXX

```

PT10: 10 ;TEST NUMBER
      PT11 ;NEXT TEST
      PRTHDR ;PRINT TEST HEADER
1$: MOV #36,R1 ;SET R1=30
     MOV #120.,R2 ;SET CHAR COUNT = 120
     MOV #30.,3$+2 ;SET UP DELAY VALUE
2$: MOV #110,R0 ;'H' TO R0
     PRINTC ;SEND IT
3$: MOV #30.,R0 ;DELAY
     DELAY ;DEC. COUNT OF CHARS PER LINE
     DEC R1 ;BRANCH IF 0, END OF LINE
     BEQ 6$ ;BRANCH IF 0, END OF LINE
4$: DEC R2 ;DECREMENT CHAR COUNTER
     BEQ 7$ ;BRANCH IF END
     MOV 3$+2,R4 ;GET OLD DELAY
     ASR R4 ;CAL 1/16 OF OLD DELAY
     ASR R4
     ASR R4
     ASR R4
     MOV R4,R5 ;SAVE 1/16 IN R5
     ASR R4 ;CAL 1/128 OF OLD DELAY
     ASR R4
     ASR R4
     ADD R4,R5 ;1/16 +1/128 TO R5
     CMP #60.,R2 ;TEST WHICH HALF OF THE 120 CHARS.
     BLE 5$ ;BRANCH IF LT OR EQ 60
     SUB R5,3$+2 ;GT 51, DECREASE DELAY BY 34 MEC.
     BR 2$ ;GO PRINT AGAIN
5$: ADD R5,3$+2 ;LT HALF WAY, ADD DELAY OF 34 MEC.
     BR 2$ ;GO PRINT AGAIN
6$: CRLF ;SEND CRLF
     MOV #36,R1 ;SET R1=30
     BR 4$
7$: CRLF ;SEND CR,LF
     CHAIN ;CHAIN TO NEXT TEST
     BR 1$ ;REPEAT TEST
  
```

3096
3097
3098
3099
3100
3101
3102
3103
3104 010642 000011
3105 010644 010674
3106 010646 104012
3107 010650 012701 000030
3108 010654 012700 000130
3109 010660 104011
3110 010662 104006
3111 010664 005301
3112 010666 001372
3113 010670 104001
3114 010672 000766
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124
3125 010674 000012
3126 010676 006560
3127 010700 104012
3128 010702 012701 000010
3129 010706 012700 000007
3130 010712 104011
3131 010714 005301
3132 010716 001373
3133 010720 104010
3134 010722 012700 003720
3135 010726 104003
3136
3137 010730 013700 000042
3138 010734 001405
3139 010736 000005
3140 010740 004710
3141 010742 000240
3142 010744 000240
3143 010746 000240
3144 010750 104001
3145 010752 000753

```
;XXXXXXXXXX
:
:PT11-- RIBBON FEED TEST -- THIS TEST PRINTS A SINGLE COLUMN
:      OF X'S (24 LINES) DOWN THE LEFT MARGIN OF THE PAGE. VISUALLY
:      CHECK THE RIBBON REED MECHANISM FOR PROPER OPERATION.
:
:XXXXXXXXXX
PT11:  11          ;TEST NUMBER
      PT12        ;NEXT TEST
      PRTHDR      ;PRINT TEST HEADER
1$:    MOV        #30,R1  ;SET R1=24
2$:    MOV        #130,R0 ;SET CHAR = X
      PRINTC      ;PRINT X
      CRLF        ;SEND CR-LF
      DEC         R1     ;DEC COUNT
      BNE         2$     ;BRANCH IF NOT DONE TEST
      CHAIN       ;CHAIN TO NEXT TEST
      BR          1$     ;REPEAT TEST

;XXXXXXXXXX
:
:PT12-- PRINTER BELL TEST-- THE LAST TEST IN THE
:      PRINTER TEST SEQUENCE. THIS TEST OUTPUTS
:      EIGHT BELL SIGNALS TO THE PRINTER
:
:XXXXXXXXXX
PT12:  12          ;THIS TEST
      PTO         ;NEXT TEST
      PRTHDR      ;PRINT TEST HEADER
PT12A: MOV        #10,R1  ;COUNTER TO R1
1$:    MOV        #7,R0   ;BELL TO R0
      PRINTC      ;SEND IT
      DEC         R1     ;DECREMENT COUNT
      BNE         1$     ;BRANCH IF NOT ZERO
      LF          ;SEND LF
      MOV        #3720,R0 ;DELAY 2 SEC BEFORE RESTARTING
      DELAY

      MOV        @#42,R0  ;CHECK IF UNDER ACT11 OR XXDP
      BEQ        HERE    ;CONTINUE TEST SEQUENCE
      RESET
LOGICAL: JSR      PC,(R0)
      NOP
      NOP
      NOP
HERE:  CHAIN       ;CHAIN TO NEXT TEST
      BR          PT12A  ;REPEAT TEST
```

3146
3147
3148
3149
3150
3151
3152
3153
3154
3155
3156
3157 010754 000017
3158 010756 010754
3159 010760 000137 010776
3160 010764 000017
3161 010766 010754
3162 010770 005037 011124
3163 010774 104012
3164 010776 012737 000041 001114
3165 011004 013701 001112
3166 011010 013700 001114
3167 011014 104011
3168 011016 005301
3169 011020 001373
3170 011022 104006
3171 011024 012702 000005
3172 011030 013701 001112
3173 011034 013700 001114
3174 011040 104011
3175 011042 005301
3176 011044 001373
3177 011046 104014
3178 011050 005302
3179 011052 001366
3180 011054 104010
3181 011056 005237 001114
3182 011062 022737 000177 001114
3183 011070 001345
3184 011072 005237 011124
3185 011076 012700 017555
3186 011102 013701 011124
3187 011106 012702 000004
3188 011112 104015
3189 011114 104002
3190 011116 017536
3191 011120 104001
3192 011122 000725
3193
3194 011124 000000

:XXXXXXXXXX

:PT17--LIFE TEST

: THIS TEST PRINTS 2 FULL LINES OF EACH PRINTABLE
: CHARACTER AND OVERPRINTS THE SECOND LINE 4 TIMES.
: THIS TEST IS CONTINUOUS RUNNING ONCE INITIATED,
: LOOPING AUTOMATICALLY ON ITSELF.
: END OF PASS COUNT IS CLEARED WHENEVER THE TEST IS RESTARTED.

:XXXXXXXXXX

PT17B: 17 :TEST NUMBER
PT17B :NEXT TEST
JMP PT17D :CONTINUE
PT17: 17 :TEST NUMBER
PT17B :NEXT TEST
CLR PASCNT :CLEAR PASS COUNT
PRTHDR :PRINT TEST HEADER
PT17D: MOV #41,REPT :SET START CHAR
1\$: MOV WIDTH,R1 :SET COLUMN COUNT
2\$: MOV REPT,RO :GET CHAR
PRINTC :SEND CHAR
DEC R1 :DECREMENT COUNT
BNE 2\$:BRANCH IF NOT DONE
CRLF :SEND CR-LF
MOV #5,R2 :SET OVERPRINT COUNT
3\$: MOV WIDTH,R1 :SET COLUMN COUNT
4\$: MOV REPT,RO :GET CHAR
PRINTC :SEND CHAR
DEC R1 :DECREMENT COUNT
BNE 4\$:BRANCH IF NOT DONE
CR :SEND CR
DEC R2 :DONE OVERPRINTS?
BNE 3\$:NO, CONTINUE
LF :SEND LF
INC REPT :SET NEXT CHAR
CMP #177,REPT :DONE CHAR SET?
BNE 1\$:NO, CONTINUE
INC PASCNT :INCREMENT PASS COUNT
MOV #PASMES,RO :SET MESSAGE ADDRESS
MOV PASCNT,R1 :# TO CONVERT
MOV #4,R2 :# DIGITS
BTOASC :CONVERT PASS COUNT TO ASCII
TYPEM :TYPE PASS COUNT
CHAIN :REPEAT TEST
BR PT17D

PASCNT: 0

.SBTTL ECHO TESTS

3195
3196
3197
3198
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3210
3211
3212 011126 000020
3213 011130 011172
3214 011132 104012
3215 011134 104013
3216 011136 012700 000036
3217 011142 104003
3218 011144 013700 001124
3219 011150 022700 000177
3220 011154 001402
3221 011156 104021
3222 011160 000765
3223 011162 104002
3224 011164 020017
3225 011166 104001
3226 011170 000761

```

:XXXXXXXXXX
:
:E020-- CHARACTER ECHO TEST-- ALL PRINTABLE AND
:          NON-PRINTING CHARACTERS TYPED ON THE KEYBOARD
:          ARE USED TO DRIVE THE PRINTER, ONE CHARACTER AT
:          A TIME.  A "RUBOUT" WILL CAUSE THE TEST TO BE
:          TERMINATED.
:
:XXXXXXXXXX
E020:  20          ;TEST NUMBER
      EO21        ;NEXT TEST
      PRTHDR      ;PRINT TEST HEADER
1$:   READ        ;GO WAIT FOR KEYBOARD INPUT
      MOV         #30.,R0 ;DELAY FOR HALF DUPLEX
      DELAY
      MOV         TEMPCH,R0 ;GET CHAR
      CMP         #177,R0 ;CHECK IF RUBOUT
      BEQ         2$ ;BRANCH IF YES
      ECHO
      BR          1$ ;NO, CHECK PRINTER READY
2$:   TYPEM
      ECOEND
      CHAIN
      BR          1$ ;CHAIN TO NEXT TEST
      ;REPEAT TEST

```

```
3227          ;XXXXXXXXXX
3228          ;
3229          ;E021-- LINE ECHO TEST, FAST RATE-- THIS TEST WILL
3230          ;          CAUSE THE CONTINUAL PRINTING OF 'O' AT THE MAXIMUM
3231          ;          RATE UNTIL EITHER ANOTHER CHARACTER IS SELECTED
3232          ;          BY PRESSING A KEY ON THE KEYBOARD OR TERMINATION BY THE
3233          ;          RUBOUT.
3234          ;
3235          ;XXXXXXXXXX
3236
3237 011172 000021          E021: 21          ;TEST NUMBER
3238 011174 011244          E022          ;NEXT TEST
3239 011176 104012          PRTHDR          ;PRINT TEST HEADER
3240 011200 012737 000060 001114 E021A: MOV #60,REPT ;CHARACTER TO BE REPEATED (O)
3241 011206 013737 001112 001122 2$: MOV WIDTH,POS1 ;SET CHAR COUNT
3242 011214 013700 001114          1$: MOV REPT,R0 ;CHAR TO TEMPCH
3243 011220 104011          PRINTC          ;PRINT CHAR
3244 011222 005337 001122          DEC POS1 ;DECREMENT POSITION COUNTER
3245 011226 001372          BNE 1$ ;CONTINUE
3246 011230 104006          CRLF          ;SEND A CR AND LF
3247 011232 000765          BR 2$
3248
3249 011234 104002          E021B: TYPED          ;PRINT TERMINATION MESSG
3250 011236 020017          ECOEND
3251 011240 104001          CHAIN          ;CHAIN TO NEXT TEST
3252 011242 000756          BR E021A ;REPEAT TEST
```

```
3253 ;XXXXXXXXXX
3254 ;
3255 ;E022-- LINE ECHO TEST, SLOW RATE-- SAME AS E021 EXCEPT
3256 ; THAT A DELAY IS INTRODUCED BETWEEN CHARACTERS
3257 ; TO PRODUCE AN LCV ACTION
3258 ;
3259 ;XXXXXXXXXX
3260
3261 011244 000022 E022: 22
3262 011246 011530 E023
3263 011250 104012 PRTHDR ;PRINT TEST HEADER
3264 011252 012737 000060 001114 E022A: MOV #60,REPT ;LOAD 0 AS INITIAL CHARACTER
3265 011260 013737 001112 001122 3$: MOV WIDTH,POS1 ;SET CHAR COUNT
3266 011266 013700 001114 1$: MOV REPT,R0 ;READY,CHAR TO TEMPCH
3267 011272 104011 PRINTC ;OUTPUT CHAR
3268 011274 012700 003410 MOV #3410,R0
3269 011300 104003 DELAY
3270 011302 005337 001122 DEC POS1 ;DECREMENT POSITION COUNTER
3271 011306 001367 BNE 1$ ;BRANCH IF NOT DONE LINE
3272 011310 104006 2$: CRLF ;SEND A CR AND LF
3273 011312 000762 BR 3$
3274
3275 011314 104002 E022B: TYPEM ;PRINT TERMINATION MESSAGE
3276 011316 020017 ECOEND
3277 011320 104001 CHAIN ;CHAIN TO NEXT TEST
3278 011322 000753 BR E022A ;REPEAT TEST
```


3279
3280
3281
3282
3283
3284
3285 011324 052516 020114
3286 011330 047523 020110
3287 011334 052123 020130
3288 011340 052105 020130
3289 011344 047505 020124
3290 011350 047105 020121
3291 011354 041501 020113
3292 011360 042502 020114
3293 011364 051502 020040
3294 011370 052110 020040
3295 011374 043114 020040
3296 011400 052126 020040
3297 011404 043106 020040
3298 011410 051103 020040
3299 011414 047523 020040
3300 011420 044523 020040
3301 011424 046104 020105
3302 011430 041504 020061
3303 011434 041504 020062
3304 011440 041504 020063
3305 011444 041504 020064
3306 011450 040516 020113
3307 011454 054523 020116
3308 011460 052105 020102
3309 011464 040503 020116
3310 011470 046505 020040
3311 011474 052523 020102
3312 011500 051505 020103
3313 011504 051506 020040
3314 011510 051507 020040
3315 011514 051522 020040
3316 011520 051525 020040
3317 011524 050123 020040
3318
3319

: THIS FOLLOWING TABLE IS USED BY TEST E023
:*****

MONIC: .ASCII /NUL /
.ASCII /SOH /
.ASCII /STX /
.ASCII /E /
.ASCII /EOT /
.ASCII /ENQ /
.ASCII /ACK /
.ASCII /BEL /
.ASCII /BS /
.ASCII /HT /
.ASCII /LF /
.ASCII /VT /
.ASCII /FF /
.ASCII /CR /
.ASCII /SO /
.ASCII /SI /
.ASCII /DLE /
.ASCII /DC1 /
.ASCII /DC2 /
.ASCII /DC3 /
.ASCII /DC4 /
.ASCII /NAK /
.ASCII /SYN /
.ASCII /ETB /
.ASCII /CAN /
.ASCII /EM /
.ASCII /SUB /
.ASCII /ESC /
.ASCII /FS /
.ASCII /GS /
.ASCII /RS /
.ASCII /US /
.ASCII /SP /

.EVEN

```
3320 ;XXXXXXXXXX
3321 :
3322 :E023-- CHARACTER CODE TEST-- ALL CHARACTERS SELECTED
3323 : WILL BE ECHOED ALONG WITH ITS OCTAL CODE.
3324 : A MNEMONIC WILL BE PRINTED INSTEAD OF THE CHARACTER
3325 : IF IT IS A NON-PRINTING CHARACTER.
3326 : THE PARITY OF THE RECEIVED CODE WILL BE
3327 : INDICATED AS EITHER EVEN OR ODD.
3328 :
3329 ;XXXXXXXXXX
3330 :
3331 011530 000023 E023: 23 ;TEST NUMBER
3332 011532 012052 E024 ;NEXT TEST
3333 011534 104012 PRTHDR ;PRINT TEST HEADER
3334 011536 104013 1$: READ ;GO WAIT FOR CHARACTER
3335 011540 012700 000036 MOV #30.,RO ;DELAY FOR HALF DUPLEX
3336 011544 104003 DELAY
3337 011546 023727 001124 000041 CMP TEMPCH,#41 ;TEST IF CHAR IS PRINTABLE
3338 011554 103015 BHIS 3$ ;BRANCH IF IT IS
3339 011556 004737 011712 JSR PC,STRLN ;STORE CODE INTO MESSAGE
3340 011562 013700 001124 MOV TEMPCH,RO ;GET CODE AGAIN
3341 011566 006300 ASL RO ;MULT BY 2
3342 011570 006300 ASL RO ;MULT BY 4
3343 011572 062700 011324 ADD #MONIC,RO ;ADD ADDR OF MNEMONIC TABLE
3344 011576 004737 011770 JSR PC,MOVNUM ;MOV MNEMONIC TO MESSAGE
3345 011602 104000 2$: TYPE ;TYPE CODE AND MNEMONIC
3346 011604 020046 E023M ;ADDRESS OF MESSAGE
3347 011606 000753 BR 1$ ;GO WAIT FOR NEXT CHARACTER
3348 011610 023727 001124 000177 3$: CMP TEMPCH,#177 ;TEST IF CHAR IS A RUBOUT
3349 011616 001421 BEQ 4$ ;BRANCH IF RUBOUT
3350 011620 012701 012042 MOV #MG24,R1
3351 011624 113721 001124 MOVB TEMPCH,(R1)+
3352 011630 112721 000040 MOVB #40,(R1)+
3353 011634 112721 000040 MOVB #40,(R1)+
3354 011640 112721 000040 MOVB #40,(R1)+
3355 011644 004737 011712 JSR PC,STRLN ;STORE CODE INTO MESSAGE
3356 011650 012700 012042 MOV #MG24,RO ;ADDR OF CHAR INTO RO
3357 011654 004737 011770 JSR PC,MOVNUM ;MOVE CHAR INTO MESSAGE
3358 011660 000750 BR 2$ ;TYPE MESSAGE
3359 011662 004737 011712 4$: JSR PC,STRLN ;RUBOUT, CONVERT AND STOR CODE
3360 011666 012700 012046 MOV #MG25,RO ;ADDR. OF DEL INTO RO
3361 011672 004737 011770 JSR PC,MOVNUM ;MOVE DEL INTO MESSAGE
3362 011676 104000 TYPE ;TYPE MESSAGE
3363 011700 020046 E023M ;ADDR OF MESSAGE
3364 011702 104002 TYPEN
3365 011704 020017 ECOEND
3366 011706 104001 CHAIN ;CHAIN TO NEXT TEST
3367 011710 000712 BR 1$ ;REPEAT TEST
```


3407
3408
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3420
3421
3422 012052 000024
3423 012054 012636
3424 012056 104012
3425 012060 005001
3426 012062 012702 012234
3427 012066 104013
3428 012070 012700 000036
3429 012074 104003
3430 012076 022737 000177 001124
3431 012104 001447
3432 012106 022737 000003 001124
3433 012114 001416
3434 012116 020127 000400
3435 012122 103361
3436 012124 013700 001124
3437 012130 110022
3438 012132 005201
3439 012134 104021
3440 012136 000753
3441
3442 012140 005037 001124
3443 012144 104006
3444 012146 104010
3445 012150 000743

:XXXXXXXXXX
:E024-- SFLECTED PATTERN ECHO TEST-- SELECT 1 TO 256
:CHARACTERS. EACH WILL BE ECHOED
:AND STORED UNTIL THE CNTL/C IS SELECTED.
:AT THAT TIME ALL CHARACTERS WILL BE PRINTED AS
:A CONTINUOUS STRING UNTIL EITHER THE RUBOUT IS
:SELECTED TO TERMINATE OR THE CNTL/C IS SELECTED
:AGAIN. A TERMINATING CNTL/C FOLLOWED BY A CNTL/C WILL ALWAYS
:CAUSE THE LAST INPUTTED STRING TO BE REPEATED. A TERMINATING
:CNTL/C FOLLOWED BY SOME OTHER CHARACTER WILL START A NEW
:STRING.

:XXXXXXXXXX
E024: 24 ;TEST NUMBER
E025 ;NEXT TEST
PRTHDR ;PRINT TEST HEADER
E024B: CLR R1 ;CLEAR CHARACTER COUNT
MOV #BUFR,R2 ;ADDRESS OF BUFFER TO R2
1\$: READ ;WAIT FOR INPUT
MOV #30.,R0 ;DELAY FOR HALF DUPLEX
DELAY
CMP #177,TEMPCH ;TEST IF RUBOUT
BEQ TERM ;BRANCH IF RUBOUT
CMP #3,TEMPCH ;TEST IF .CNTL-C
BEQ OUTPUT ;BRANCH IF CNTL-C
CMP R1,#256. ;YES, CHECK IF CHAR CNT IS EQ, GT 256
BHS 1\$;BRANCH IF YES, IGNORE CHAR
MOV TEMPCH,R0 ;GET CHAR
MOVB R0,(R2)+ ;STORE CHAR INTO BUFFER
INC R1 ;INCREMENT CHARACTER COUNT
ECHO ;OUTPUT CHAR
BR 1\$;GO WAIT FOR NEXT CHAR
E024R: CLR TEMPCH ;CLEAR CONTROL-C FROM BUFFER
CRLF ;CONTROL-C RETURN FROM PRINT ROUTINE
LF
BR E024B

```
3446  
3447  
3448  
3449  
3450  
3451  
3452 012152 020227 012234  
3453 012156 001405  
3454 012160 113722 001124  
3455 012164 005037 001124  
3456 012170 104006  
3457 012172 012702 012234  
3458 012176 005037 001124  
3459 012202 121227 000003  
3460 012206 001724  
3461 012210 112200  
3462 012212 020027 000003  
3463 012216 001765  
3464 012220 104011  
3465 012222 000772  
3466  
3467 012224 104002  
3468 012226 020017  
3469 012230 104001  
3470 012232 000712  
3471  
3472 012234 000003  
3473 012236 000400
```

```
*****  
SECTION TO OUTPUT CONTINOUS STRING  
*****  
OUTPUT: CMP R2,#BUFR ;CHECK IF POINTER IS AT START OF BUFFER  
BEQ 1$ ;YES, DON'T STORE ^C IN TABLE  
MOVB TEMPCH,(R2)+ ;STORE ^C IN TABLE  
CLR TEMPCH ;CLEAR CONTROL-C FROM BUFFER  
CRLF ;SEND A CR LF  
1$: MOV #BUFR,R2 ;BUFFER ADDRESS TO R2  
CLR TEMPCH ;CLEAR CONTROL-C  
CMPB (R2),#3 ;FIRST CHAR IN TABLE ^C ?  
BEQ E024B ;YES, GO LOOK FOR MORE INPUT  
2$: MOVB (R2)+,R0 ;GET CHAR  
CMP R0,#3 ;DONE STRING?  
BEQ 1$ ;YES, RESTART STRING  
PRINTC ;OUTPUT CHAR  
BR 2$  
TERM: TYPEM ;OUTPUT TERMINATION MESSAGE  
ECOEND  
CHAIN ;CHAIN TO NEXT TEST  
BR E024B ;REPEAT TEST  
BUFR: 3 ;INITIALIZE FIRST CHAR AS CNTL-C IN TALBE  
.BLKB 256. ;256 CHARACTER BUFFER
```

3474
3475
3476
3477
3478
3479
3480
3481
3482
3483
3484
3485 012636 000025
3486 012640 011126
3487 012642 104012
3488 012644 023727 001112 000101
3489 012652 103424
3490 012654 104002
3491 012656 017672
3492 012660 000402
3493 012662 104000
3494 012664 017672
3495 012666 104013
3496 012670 012700 000036
3497 012674 104003
3498 012676 113700 001124
3499 012702 020027 000040
3500 012706 103767
3501 012710 022700 000177
3502 012714 001405
3503 012716 104021
3504 012720 104007
3505 012722 000757
3506 012724 104002
3507 012726 017772
3508 012730 104002
3509 012732 020017
3510 012734 104001
3511 012736 000742

```

:XXXXXXXXXX
:
: E025-- BELL ECHO TEST-- A MESSAGE IS PRINTED AND
: THE TEST WAITS FOR SOME PRINTABLE CHARACTER
: TO BE SELECTED ON THE KEYBOARD (GT040). THIS
: TEST IS VALID ONLY IF THE PAPER WIDTH IS GT 64
: COLUMNS. IF LT64 COLUMNS AN ILLEGAL BELL TEST
: MESSAGE IS PRINTED.
:
:XXXXXXXXXX
E025: 25 ;TEST NUMBER
      E020 ;NEXT TEST HEADER
      PRTHDR ;PRINT TEST HEADER
      1$: CMP WIDTH,#101 ;TEST IF COLUMN COUNT IS EQ,GT 64
          BLO 4$ ;BRANCH IF NOT
          TYPEM ;TYPE TEST MESSG
          E025MA ;ON ALL TERM'S
          BR 3$ ;WAIT FOR CHAR
      2$: TYPE ;TYPE TEST MESSG ON TERM
          E025MA ;CHARACTER WAS RECEIVED ON
      3$: READ ;WAIT FOR OPERATOR RESPONSE
          MOV #30.,R0 ;DELAY FOR HALF DUPLEX
          DELAY
          MOVB TEMPCH,R0 ;CHAR TO R0
          CMP R0,#40 ;TEST IF PRINTABLE
          BLO 3$ ;BRANCH IF NON-PRINTABLE
          CMP #177,R0 ;CHECK IF CHAR IS RUBOUT
          BEQ 5$ ;BRANCH IF YES
          ECHO ;PRINT CHAR
          SCRLF ;SEND A CRLF
          BR 2$ ;REPEAT
      4$: TYPEM ;TYPE ERROR MESSAGE
          E025MB
      5$: TYPEM ;PRINT TERMINATION
          ECOEND
          CHAIN ;EXIT TO NEXT TEST
          BR 1$ ;REPEAT TEST
```

.SBTTL OPTION TESTS

```
3512  
3513  
3514  
3515  
3516  
3517  
3518  
3519  
3520 012740 000030  
3521 012742 012740  
3522 012744 104012  
3523 012746 012704 000010  
3524 012752 104002 013036  
3525 012756 012702 000177  
3526 012762 004737 013050  
3527 012766 104002 013043  
3528 012772 013702 013120  
3529 012776 020227 000377  
3530 013002 001403  
3531 013004 012700 000016  
3532 013010 104011  
3533 013012 004737 013050  
3534 013016 104006  
3535 013020 005304  
3536 013022 001353  
3537 013024 012700 000017  
3538 013030 104011  
3539 013032 104001  
3540 013034 000741  
3541  
3542 013036 021417 036461 000 10$: .ASCIZ <17>/#1=/  
3543 013043 017 031043 000075 20$: .ASCIZ <17>/#2=/  
3544 .EVEN  
3545  
3546 013050 010201 30$: MOV R2,R1 ;GET LIMIT CHAR  
3547 013052 042701 177537 BIC #177537,R1 ;GET START CHAR  
3548 013056 013703 001112 MOV WIDTH,R3 ;GET COLUMN COUNT  
3549 013062 162703 000003 SUB #3,R3 ;SUBTRACT 3  
3550 013066 010100 31$: MOV R1,R0 ;GET CHAR  
3551 013070 104011 PRINTC ;PRINT IT  
3552 013072 005201 INC R1 ;NEXT CHAR  
3553 013074 020102 CMP R1,R2 ;DONE CHAR SET?  
3554 013076 001406 BEQ 32$ ;EXIT IF DONE  
3555 013100 005303 DEC R3 ;DEC COLUMN COUNT  
3556 013102 001371 BNE 31$ ;FINISH LINE  
3557 013104 104006 CRLF ;SEND CR-LF WHEN DONE LINE  
3558 013106 013703 001112 MOV WIDTH,R3 ;RESET COLUMN COUNT  
3559 013112 000765 BR 31$ ;CONTINUE  
3560 013114 104006 32$: CRLF ;SEND CR-LF  
3561 013116 000207 RTS PC ;RETURN  
3562  
3563 013120 000177 T30SC: .WORD 177 ;CHAR SET LIMIT  
3564 ;CHANGE TO 377 WHEN USING 8 BIT CHAR SELECTION
```

```
3565 ;XXXXXXXXXXXXXXXXXX
3566 ;
3567 ;TEST31 - SELECTIVE ADDRESSING OPTION
3568 ;
3569 ;XXXXXXXXXXXXXXXXXX
3570
3571 013122 000031 TEST31: 31
3572 013124 013122 TEST31
3573 013126 104006 CRLF ;SEND CRLF
3574 013130 013700 013712 MOV 34$,R0 ;SET EOT CHAR
3575 013134 104011 PRINTC ;SEND IT
3576 013136 104002 015542 TYPEM, 10$ ;TRY PRINTING ERROR MMSG
3577 013142 012700 000007 MOV #7,R0 ;SET BEL CHAR
3578 013146 104011 PRINTC ;SEND IT
3579 013150 012700 000002 MOV #2,R0 ;SET STX CHAR
3580 013154 104011 PRINTC ;SEND IT
3581 013156 104012 PRTHDR ;PRINT TEST HEADER ON ALL TERMINALS
3582 013160 013700 013712 MOV 34$,R0 ;SET EOT CODE
3583 013164 104011 PRINTC ;SEND IT
3584 013166 012700 000002 MOV #2,R0 ;SET STX CODE
3585 013172 104011 PRINTC ;SEND IT
3586 013174 104002 013542 TYPEM, 10$ ;TRY PRINT ERROR MMSG
3587 013200 012703 013670 MOV #30$,R3 ;SET TABLE POINTER
3588 013204 005713 1$: TST (R3) ;CHECK TABLE POINTER
3589 013206 001416 BEQ 2$ ;NEXT PORTION OF TEST IF DONE
3590 013210 013700 013712 MOV 34$,R0 ;SEND EOT CHAR
3591 013214 104011 PRINTC
3592 013216 011300 MOV (R3),R0 ;SEND GROUP SELECT CHAR
3593 013220 104011 PRINTC ;SEND IT
3594 013222 012700 000002 MOV #2,R0 ;SEND STX CHAR
3595 013226 104011 PRINTC
3596 013230 104002 013606 TYPEM, 14$ ;TYPE MMSG
3597 013234 012300 MOV (R3)+,R0 ;TYPE SELECT CHAR FOR MMSG
3598 013236 104011 PRINTC
3599 013240 104006 CRLF ;CR-LF
3600 013242 000760 BR 1$ ;CONTINUE - NEXT SELECT CHAR
3601 013244 012703 013716 2$: MOV #40$,R3 ;SET TABLE ADR
3602 013250 005713 3$: TST (R3) ;CHECK SELECT CHAR
3603 013252 001517 BEQ 4$ ;CONTINUE TEST
3604 013254 013700 013712 MOV 34$,R0 ;SEND EOT CHAR
3605 013260 104011 PRINTC ;ALL TERMS OFF
3606 013262 011300 MOV (R3),R0 ;GET UNIQUE SELECT CHAR
3607 013264 104011 PRINTC ;SEND IT - THAT TERM ON
3608 013266 104002 013542 TYPEM, 10$ ;TRY PRINTING ERROR MMSG
3609 013272 012700 000002 MOV #2,R0 ;SEND STX
3610 013276 104011 PRINTC
3611 013300 104002 013614 TYPEM, 15$ ;TYPE SELECT CHAR MMSG
3612 013304 011300 MOV (R3),R0 ;PRINT SELECT CHAR FOR MMSG
3613 013306 104011 PRINTC
3614 013310 104006 CRLF
3615 013312 104002 013633 TYPEM, 20$ ;PRINT MMSG
3616 013316 104013 READ ;READ CHAR FROM SELECTED TERM
3617 013320 012700 000036 MOV #30.,R0 ;DELAY FOR HALF DUPLEX
3618 013324 104003 DELAY
3619 013326 022737 000177 001124 CMP #177,TEMPCH ;CHECK CHAR
3620 013334 001013 BNE 5$ ;EXIT IF RUBOUT
```


3621	013336	013700	013712		MOV	34\$,RO	:ENABLE ALL TERMINALS
3622	013342	104011			PRINTC		
3623	013344	012700	000007		MOV	#7,RO	
3624	013350	104011			PRINTC		
3625	013352	012700	000002		MOV	#2,RO	
3626	013356	104011			PRINTC		
3627	013360	000137	002540		JMP	TTY1H	:GO TO KYBD CONTROL
3628	013364	013700	001124	5\$:	MOV	TEMPCH,RO	:GET CHAR
3629	013370	104021			ECHO		:ECHO CHAR
3630	013372	104006			CRLF		:SEND CR-LF
3631	013374	013700	013712		MOV	34\$,RO	:SEND EOT CHAR
3632	013400	104011			PRINTC		
3633	013402	013700	013714		MOV	35\$,RO	:SEND DUMMY SELECT CHAR
3634	013406	104011			PRINTC		
3635	013410	012700	000002		MOV	#2,RO	:SEND STX
3636	013414	104011			PRINTC		
3637	013416	104002	013542		TYPEN,	10\$:TRY PRINTING ERROR MESSAGE
3638	013422	012700	000003		MOV	#3,RO	:SEND ETX
3639	013426	104011			PRINTC		
3640	013430	011300			MOV	(R3),RO	:SEND UNIQUE SELECT CHAR
3641	013432	104011			PRINTC		
3642	013434	012700	000002		MOV	#2,RO	:SEND STX
3643	013440	104011			PRINTC		
3644	013442	104002	013614		TYPEN,	15\$:PRINT SELECT MESSG ON SELECTED TERMINAL
3645	013446	011300			MOV	(R3),RO	:PRINT SELECT CHAR FOR MESSG
3646	013450	104011			PRINTC		
3647	013452	104006			CRLF		:SEND CR-LF
3648	013454	012700	000003		MOV	#3,RO	:SEND ETX
3649	013460	104011			PRINTC		
3650	013462	012700	013714		MOV	#35\$,RO	:SEND DUMMY SELECT CHAR
3651	013466	104011			PRINTC		
3652	013470	012700	000002		MOV	#2,RO	:SEND STX
3653	013474	104011			PRINTC		
3654	013476	104002	013614		TYPEN,	15\$:PRINT MESSG ON SELECTED TERM
3655	013502	012300			MOV	(R3)+,RO	:PRINT SELECT CHAR FOR MESSG
3656	013504	104011			PRINTC		
3657	013506	104006			CRLF		:SEND CR-LF
3658	013510	000657			BR	3\$:CONTINUE
3659	013512	013700	013712	4\$:	MOV	34\$,RO	:ENABLE ALL LINES
3660	013516	104011			PRINTC		:BEFORE EXITING TEST
3661	013520	012700	000007		MOV	#7,RO	
3662	013524	104011			PRINTC		
3663	013526	012700	000002		MOV	#2,RO	
3664	013532	104011			PRINTC		
3665	013534	104001			CHAIN		:CHAIN TO NEXT TEST OR LOOP ON TEST
3666	013536	000137	013122		JMP	TEST31	:LOOP ON TEST

3667	013542	051105	047522	026122	10\$:	.ASCIZ	/ERROR, ALL TERMINALS SHOULD BE OFF/<ACRLF>
3668	013550	040440	046114	052040			
3669	013556	051105	044515	040516			
3670	013564	051514	051440	047510			
3671	013572	046125	020104	042502			
3672	013600	047440	043106	000200			
3673	013606	051107	052517	020120	14\$:	.ASCII	/GROUP /
3674	013614	042523	042514	052103	15\$:	.ASCIZ	/SELECT CHAR = /
3675	013622	041440	040510	020122			
3676	013630	020075	000				
3677	013633	124	050131	020105	20\$:	.ASCIZ	/TYPE ANY PRINTABLE CHAR ... /
3678	013640	047101	020131	051120			
3679	013646	047111	040524	046102			
3680	013654	020105	044103	051101			
3681	013662	027040	027056	000040			
3682						.EVEN	
3683							
3684	013670	000107			30\$:	107	;GROUP SELECT CHAR TABLE
3685	013672	000000				0	;FIRST ZERO - END OF TABLE
3686	013674	000000				0	
3687	013676	000000				0	
3688	013700	000000				0	
3689	013702	000000				0	
3690	013704	000000				0	
3691	013706	000000				0	
3692	013710	000000				0	
3693							
3694	013712	000004			34\$:	004	;DESELECT CHAR = 'EOT'
3695							
3696	013714	000045			35\$:	045	;DUMMY SELECT CHARACTER,
3697							;IF '%' IS USED AS A UNIQUE OR GROUP SELECT
3698							;CHARACTER, REPLACE WITH ANY UNUSED SELECT
3699							;CHARACTER CODE.
3700							
3701	013716	000125			40\$:	125	;UNIQUE SELECT CHAR TABLE
3702	013720	000000				0	;FIRST ZERO = END OF TABLE
3703	013722	000000				0	
3704	013724	000000				0	
3705	013726	000000				0	
3706	013730	000000				0	
3707	013732	000000				0	
3708	013734	000000				0	
3709	013736	000000				0	

```
3710 ;XXXXXXXXXXXXXXXXXX
3711 ;
3712 ;TEST32 - AUTO ANSWER BACK OPTION
3713 ;
3714 ;XXXXXXXXXXXXXXXXXX
3715
3716 013740 000032 TEST32: 32
3717 013742 013740 TEST32
3718 013744 104012 PRTHDR ;PRINT TEST HEADER
3719 013746 012700 000005 MOV #5,R0 ;SEND ENQ CHAR
3720 013752 104021 ECHO
3721 013754 004737 014004 JSR PC,10$ ;READ AND PRINT MSG
3722 013760 104002 014072 TYPEN, 20$ ;TYPE INSTRUCTIONS
3723 013764 004737 014004 JSR PC,10$ ;READ AND PRINT MSG
3724 013770 104002 014117 TYPEN, 30$ ;TYPE INSTRUCTIONS
3725 013774 004737 014004 JSR PC,10$ ;READ AND PRINT MSG
3726 014000 104001 2$: CHAIN ;CHAIN TO NEXT TEST
3727 014002 000756 BR TEST32 ;LOOP ON TEST
3728
3729 014004 012702 014143 10$: MOV #STORE,R2 ;SET TABLE ADR
3730 014010 104013 READ ;READ FIRST CHAR
3731 014012 023727 001124 000177 8$: CMP TEMPCH,#177 ;CHAR = RUBOUT?
3732 014020 001002 BNE 3$ ;CONTINUE IF NOT RUBOUT
3733 014022 000137 002540 JMP TTY1H ;GO TO KYBD CONTROL
3734 014026 113722 001124 3$: MOVB TEMPCH,(R2)+ ;STORE CHAR
3735 014032 012703 000062 MOV #50.,R3 ;SET DELAY COUNT
3736 014036 013701 001102 4$: MOV TIMER,R1
3737 014042 104022 5$: INRDY ;ANY INPUT?
3738 014044 000401 BR 6$ ;NO, WAIT FOR CHAR
3739 014046 000761 BR 8$ ;YES, READ CHAR
3740 014050 005301 6$: DEC R1 ;DELAY WHILE WAITING FOR CHAR
3741 014052 001373 BNE 5$
3742 014054 005303 DEC R3
3743 014056 001367 BNE 4$
3744 014060 105012 7$: CLRB (R2) ;SET NULL AS TERMINATOR IN TABLE
3745 014062 104000 014142 TYPE, STORE-1 ;TYPE MSG ON TERMINAL RECEIVED ON
3746 014066 104007 SCRLF ;SEND CR-LF
3747 014070 000207 RTS PC ;RETURN TO TEST
3748
3749 014072 042504 051120 051505 20$: .ASCIZ /DEPRESS HERE IS KEY/<ACRLF>
3750 014100 020123 042510 042522
3751 014106 044440 020123 042513
3752 014114 100131 000
3753 014117 104 050105 042522 30$: .ASCIZ /DEPRESS CONTROL-E/<ACRLF>
3754 014124 051523 041440 047117
3755 014132 051124 046117 042455
3756 014140 000200
3757 014142 200 .ASCII <ACRLF>
3758
3759 014143 000025 STORE: .BLKB 21. ;20 CHAR + TERMINATOR BUFFER
3760
3761 .EVEN
```

3762
3763
3764
3765
3766
3767
3768 014170 000033
3769 014172 014170
3770 014174 104012
3771 014176 012705 014642
3772 014202 012704 014626
3773 014206 104002 014451
3774 014212 012701 000020
3775 014216 104013
3776 014220 012700 000036
3777 014224 104003
3778 014226 022737 000177 001124
3779 014234 001002
3780 014236 000137 002540
3781 014242 104014
3782 014244 104010
3783 014246 005301
3784 014250 001375
3785 014252 012700 000014
3786 014256 104011
3787 014260 113701 014626
3788 014264 005000
3789 014266 104011
3790 014270 005301
3791 014272 001374
3792 014274 104002 014567
3793 014300 010537 014360
3794 014304 104013
3795 014306 012700 000036
3796 014312 104003
3797 014314 022737 000177 001124
3798 014322 001002
3799 014324 000137 002540
3800 014330 104014
3801 014332 012700 000014
3802 014336 104011
3803 014340 112401
3804 014342 005000
3805 014344 104011
3806 014346 005301
3807 014350 001374
3808 014352 104002 014424
3809 014356 104002
3810 014360 000000
3811 014362 104002 014432
3812 014366 022525
3813 014370 010537 014376
3814 014374 104002
3815 014376 000000
3816 014400 104002 014441
3817 014404 104002 014424

;XXXXXXXXXXXXXXXXXX

;TEST33 - FORM FEED OPTION

;XXXXXXXXXXXXXXXXXX

TEST33: 33

TEST33
PRTHDR ;PRINT TEST HEADER
MOV #50\$,R5 ;SET TABLE POINTER
MOV #60\$,R4 ;SET TABLE POINTER
TYPEM, 40\$;PRINT INSTR
MOV #16.,R1 ;SET LF COUNT TO 16
READ ;WAIT FOR KYBD FLAG
MOV #30.,R0 ;DELAY FOR HALF DUPLEX
DELAY
CMP #177,TEMPCH ;CHECK FOR RUBOUT
BNE 13\$;EXIT IF RUBOUT
JMP TTY1H ;GO TO KYBD CONTROL
13\$: CR ;SEND CR
8\$: LF ;SEND LF
DEC R1 ;DEC COUNT
BNE 8\$;CONTINUE
MOV #14,R0 ;SET FF
PRINTC ;SEND IT
MOVB 60\$,R1 ;SET FILL COUNT
9\$: CLR R0 ;SET NULL
PRINTC ;SEND FILLS
DEC R1 ;DEC FILL COUNT
BNE 9\$;CONTINUE
TYPEM, 45\$;TYPE MESSAGE
5\$: MOV R5,2\$;SET MESH
READ ;WAIT FOR KYBD FLAG
MOV #30.,R0 ;DELAY FOR HALF DUPLEX
DELAY
CMP #177,TEMPCH ;CHECK IF RUBOUT
BNE 15\$;EXIT IF RUBOUT
JMP TTY1H ;GO TO KYBD CONTROL
15\$: CR ;SEND CR
MOV #14,R0 ;SEND FF
PRINTC ;SEND IT
MOVB (R4)+,R1 ;GET FILL COUNT
11\$: CLR R0 ;SET NULL
PRINTC ;SEND FILLS
DEC R1 ;CONTINUE
BNE 11\$;FINISH NULLS
12\$: TYPEM, 10\$;PRINT MESH
TYPEM
2\$: .WORD 0
TYPEM, 20\$
CMP (R5)+,(R5)+ ;INC TABLE POINTER
3\$: MOV R5,4\$;SET MESH
TYPEM ;PRINT MESH
4\$: .WORD 0
TYPEM, 30\$
TYPEM, 10\$

3818	014410	005765	000004							TST	4(R5)	:DONE TEST?
3819	014414	001331								BNE	58	:NO, CONTINUE
3820	014416	104006								CRLF		:YES, SEND CR-LF
3821	014420	104001								CHAIN		:SELECT TEST OR LOOP
3822	014422	000662								BR	TEST33	:LOOP ON TEST
3823												
3824	014424	026455	026455	000055	108:	.ASCIZ	/-----/					
3825	014432	020042	043106	020057	208:	.ASCIZ	/' FF/<57>/ /					
3826	014440	000										
3827	014441	042	047040	054105	308:	.ASCIZ	/' NEXT /					
3828	014446	020124	000									
3829	014451	104	050105	042522	408:	.ASCII	/DEPRESS FORMFEED RESET SWITCH/<ACRLF>/AFTER EACH SWITCH SETTING/<ACRLF>					
3830	014456	051523	043040	051117								
3831	014464	043115	042505	020104								
3832	014472	042522	042523	020124								
3833	014500	053523	052111	044103								
3834	014506	040600	052106	051105								
3835	014514	042440	041501	020110								
3836	014522	053523	052111	044103								
3837	014530	051440	052105	044524								
3838	014536	043516	200									
3839	014541	124	050131	020105		.ASCII	/TYPE SPACE WHEN READY/<ACRLF>					
3840	014546	050123	041501	020105								
3841	014554	044127	047105	051040								
3842	014562	040505	054504	200								
3843	014567	055	026455	020055	458:	.ASCIZ	/---- SET 3 INCH FORMFEED ----/					
3844	014574	042523	020124	031440								
3845	014602	044440	041516	020110								
3846	014610	047506	046522	042506								
3847	014616	042105	026440	026455								
3848	014624	000055										
3849	014626	002	005	010	608:	.BYTE	2,5,8,,17,,20,,26,,32,,35,,50,,56,,68.					
3850	014631	021	024	032								
3851	014634	040	043	062								
3852	014637	070	104									
3853		014642				.EVEN						
3854												
3855	014642	020040	000063		508:	.ASCIZ	/ 3/					
3856	014646	027063	000065			.ASCIZ	/3.5/					
3857	014652	020040	000064			.ASCIZ	/ 4/					
3858	014656	027065	000065			.ASCIZ	/5.5/					
3859	014662	020040	000066			.ASCIZ	/ 6/					
3860	014666	020040	000067			.ASCIZ	/ 7/					
3861	014672	020040	000070			.ASCIZ	/ 8/					
3862	014676	027070	000065			.ASCIZ	/8.5/					
3863	014702	030440	000061			.ASCIZ	/ 11/					
3864	014706	030440	000062			.ASCIZ	/ 12/					
3865	014712	030440	000064			.ASCIZ	/ 14/					
3866	014716	020040	000040			.ASCIZ	/ /					
3867	014722	000000				.WORD	0					:END OF TABLE

```
3868 ;XXXXXXXXXXXXXXXXXX
3869 ;
3870 ;TEST34 - HORIZONTAL TAB OPTION
3871 ;
3872 ;XXXXXXXXXXXXXXXXXX
3873
3874 014724 000034 TEST34: 34
3875 014726 014724 TEST34
3876 014730 104012 PRTHDR ;PRINT TEST HEADER
3877
3878 014732 005004 CHGD6: CLR R4 ;SET TABLE POINTER
3879 ;*****
3880 014734 012737 015134 015132 MOV #16$,12$+2 ;RESET JUMP INSTR OR FIRST TIME THRU
3881 014742 013703 001112 2$: MOV WIDTH,R3 ;SET COLUMN COUNT
3882 014746 012700 000033 MOV #33,R0 ;CLEAR OLD TABS
3883 014752 104011 PRINTC
3884 014754 012700 000062 MOV #62,R0
3885 014760 104011 PRINTC
3886 014762 104014 CR ;DO A CARRIAGE RETURN
3887 ;*****
3888 014764 104023 FOUR ;FOUR FILL CHARACTERS---- CHGD6
3889 ;*****
3890 014766 016401 015224 3$: MOV 20$(R4),R1 ;GET SPACE COUNT FOR TAB
3891 014772 000405 BR 5$ ;SUBTRACT 1 FOR TAB SET
3892 014774 012700 000040 4$: MOV #40,R0 ;GET SPACE
3893 015000 104011 PRINTC ;SEND IT
3894 015002 005303 DEC R3 ;DEC COLUMN COUNT
3895 015004 001420 BEQ 6$ ;CR IF DONE LINE
3896 015006 005301 5$: DEC R1 ;DEC SPACE COUNT
3897 015010 001371 BNE 4$ ;CONTINUE IF NOT DONE
3898 015012 012700 000033 MOV #33,R0 ;SEND IT
3899 015016 104011 PRINTC
3900 015020 012700 000061 MOV #61,R0 ;SEND 1 - SET TAB
3901 015024 104011 PRINTC
3902 015026 012700 000010 MOV #10,R0 ;SET BACKSPACE
3903 015032 104011 PRINTC
3904 015034 012700 000117 MOV #'0,R0 ;PRINT 0
3905 015040 104011 PRINTC
3906 015042 005303 DEC R3 ;DEC COLUMN COUNT
3907 015044 001350 BNE 3$ ;CONTINUE IF NOT DONE LINE
3908 015046 104014 6$: CR ;SEND CR
3909 ;*****
3910 015050 104023 FOUR ;FOUR FILL CHARACTERS---- CHGD6
3911 ;*****
3912 015052 013703 001112 MOV WIDTH,R3 ;RESET COLUMN COUNT
3913 015056 016401 015224 17$: MOV 20$(R4),R1 ;RESET COLUMN COUNT FOR TAB
3914 015062 020127 000001 CMP R1,#1 ;ADD 1 IF FIRST LINE
3915 015066 001001 BNE 11$
3916 015070 005201 INC R1
3917 015072 016402 015250 11$: MOV 30$(R4),R2 ;SET FILL CHAR COUNT
3918 015076 160103 SUB R1,R3 ;SUBTRACT TAB FROM COLUMN COUNT
3919 015100 002413 BLT 12$ ;BRANCH IF TOO MANY COLUMNS
3920 015102 012700 000011 MOV #11,R0 ;SET TAB
3921 015106 104011 PRINTC ;SEND IT
3922 015110 005000 14$: CLR R0 ;SET NULL CHAR - FILLS
3923 015112 104011 PRINTC ;SEND FILL CHARS
```

3924	015114	005302				DEC	R2		:DEC FILL COUNT
3925	015116	001374				BNE	14\$:CONTINUE FILLS
3926	015120	012700	000130			MOV	#'X,R0		:SET X CHAR
3927	015124	104011				PRINTC			:SEND IT
3928	015126	000761				BR	11\$:CONTINUE
3929	015130	000137	015134		12\$:	JMP	@#16\$:SKIP FOLLOWING AFTER FIRST TIME THRU
3930	015134	012737	015164	015132	16\$:	MOV	#15\$,12\$+2		
3931	015142	104014				CR			:SEND CR
3932	015144	104023				FOUR			:FOUR FILL CHARACTERS----
3933	015146	012700	000130			MOV	#'X,R0		:PRINT X
3934	015152	104011				PRINTC			
3935	015154	013703	001112			MOV	WIDTH,R3		:RESET COLUMN COUNT
3936	015160	005303				DEC	R3		:SUBTRACT ONE FOR FIRST X CHAR
3937	015162	000735				BR	17\$:CONTINUE
3938	015164	104006			15\$:	CRLF			:SEND CR-LF
3939	015166	005724				TST	(R4)+		:INC TABLE POINTER
3940	015170	016401	015224			MOV	20\$(R4),R1		:GET COLUMN COUNT FOR TAB
3941	015174	001403				BEQ	13\$:EXIT IF DONE TABLE (0)
3942	015176	020137	001112			CMP	R1,WIDTH		:CHECK IF TOO LARGE
3943	015202	101657				BLOS	2\$:CONTINUE TEST, OK
3944	015204	012700	000033		13\$:	MOV	#33,R0		:CLEAR ALL TABS BEFORE EXITING
3945	015210	104011				PRINTC			
3946	015212	012700	000062			MOV	#62,R0		
3947	015216	104011				PRINTC			
3948	015220	104001				CHAIN			:SELECT TEST OR LOOP ON TEST
3949	015222	000640				BR	TEST34		:LOOP ON TEST
3950									
3951	015224	000001	000002	000004	20\$:	.WORD	1,2,4,8.,16.,32.,64.,128.,132.,0		
3952	015232	000010	000020	000040					
3953	015240	000100	000200	000204					
3954	015246	000000							
3955	015250	000001	000002	000003	30\$:	.WORD	1,2,3,5,9.,18.,36.,71.,73.,0		
3956	015256	000005	000011	000022					
3957	015264	000044	000107	000111					
3958	015272	000000							

CMGD6

```
3959 ;XXXXXXXXXXXXXXXXXX
3960 ;
3961 ;TEST35 - VERTICAL TAB OPTION
3962 ;
3963 ;XXXXXXXXXXXXXXXXXX
3964
3965 015274 000035 TEST35: 35
3966 015276 015274 TEST35
3967 015300 104012 PRTHDR ;PRINT TEST HEADER
3968 015302 104002 1$: TYPEM ;TYPE INSTR
3969 015304 015615 20$
3970 015306 104013 READ ;WAIT FOR KYBD FLAG
3971 015310 012700 000036 MOV #30.,R0 ;DELAY FOR HALF DUPLEX
3972 015314 104003 DELAY
3973 015316 022737 000177 001124 CMP #177,TEMPCH ;CHECK CHAR
3974 015324 001505 BEQ 12$ ;EXIT TEST IF RUBOUT
3975 015326 005004 CLR R4 ;SET LINE COUNT
3976 015330 012700 000033 MOV #33,R0 ;CLEAR VERTICAL TABS
3977 015334 104011 PRINTC
3978 015336 012700 000064 MOV #64,R0
3979 015342 104011 PRINTC
3980 015344 104002 2$: TYPEM ;TYPE REF LINE
3981 015346 015573 15$
3982 015350 005204 INC R4 ;INC LINE COUNT
3983 015352 020427 000013 CMP R4,#13 ;CHECK IT
3984 015356 003013 BGT 35$ ;BRANCH IF DONE REF.
3985 015360 010401 MOV R4,R1 ;GET LF COUNT
3986 015362 104010 3$: LF ;SEND LF
3987 015364 005301 DEC R1 ;DEC LF COUNT
3988 015366 001375 BNE 3$ ;CONTINUE
3989 015370 012700 000033 MOV #33,R0 ;SET TAB FOR THIS LINE
3990 015374 104011 PRINTC
3991 015376 012700 000063 MOV #63,R0
3992 015402 104011 PRINTC
3993 015404 000757 BR 2$ ;CONTINUE
3994 015406 012700 000014 35$: MOV #14,R0 ;SEND FF
3995 015412 104011 PRINTC
3996 015414 104002 TYPEM ;TYPE MESSG
3997 015416 015707 30$
3998 015420 104013 READ ;WAIT FOR KYBD FLAG
3999 015422 012700 000036 MOV #30.,R0 ;DELAY FOR HALF DUPLEX
4000 015426 104003 DELAY
4001 015430 022737 000177 001124 CMP #177,TEMPCH ;CHECK CHAR
4002 015436 001440 BEQ 12$ ;EXIT TEST IF RUBOUT
4003 015440 005004 CLR R4 ;RESET LF COUNT
4004 015442 104002 4$: TYPEM ;TYPE REF LINE
4005 015444 015572 10$
4006 015446 005204 INC R4 ;INC LINE COUNT
4007 015450 020427 000013 CMP R4,#13 ;CHECK IT
4008 015454 003014 BGT 8$ ;BRANCH IF DONE
4009 015456 012700 000013 MOV #13,R0 ;SEND TAB
4010 015462 104011 PRINTC
4011 015464 010401 MOV R4,R1 ;SET FILL COUNT
4012 015466 162701 000020 SUB #16.,R1 ;SUBTRACT 16
4013 015472 003763 BLE 4$ ;SKIP NULLS IF COUNT < 0
4014 015474 005000 5$: CLR R0 ;SET NULL CHAR
```



```
4060 ;XXXXXXXXXXXXXXXXXXXXX
4061 ;
4062 ;CHECK SWITCH REGISTER ROUTINE. CHECKS FOR ^G TO ALLOW
4063 ;CHANGING OF LOC 176. ROUTINE IS ENTERED AT CNTLU FOR START UP PURPOSES.
4064 ;WHEN A ^G IS GIVEN, THE PROGRAM ENTERS AT CKSWR. THE PROGRAM
4065 ;GETS CONTENTS OF SOFT. SWITCH REG, TYPES IT OUT, AND THEN
4066 ;SEEKS NEW DATA FROM OPERATOR. ONCE DATA IS SUPPLIED, IT INSERTS THIS
4067 ;DATA INTO THE SOFT. SWITCH REG. AND RESUMES OPERATION IN THE
4068 ;MAIN PROGRAM. INCORRECT ENTRIES (SUCH AS 8,9,LETTERS) ARE DELETED
4069 ;AND THE PROCESS RESTARTED. TYPING ^U ALLOWS THE PRESENT ENTRY
4070 ;TO BE DELETED AND THE PROCESS RESTARTED.
4071 ;
4072 ;
4073 ;XXXXXXXXXXXXXXXXXXXXX
4074 ;
4075 ;
4076 ;*****
4077 015736 CHGD7:
4078 015736 022737 000176 001244 $CKSWR: CMP #SWREG,SR ;SOFTWARE SW REG PRES?
4079 015744 001125 BNE OUT ;NO, GET OUT
4080 015746 023727 001124 000007 CMP TEMPCH,#7 ;IS IT A ^G
4081 015754 001121 BNE OUT ;NO, GET OUT
4082 015756 010046 MOV R0,-(SP) ;SAVE R0
4083 015760 010446 MOV R4,-(SP) ;SAVE R4
4084 015762 013700 001124 MOV TEMPCH,R0 ;SET UP TO PRINT ^G
4085 015766 104002 TYPEM ;TYPE MESSAGE ROUTINE
4086 015770 020121 CNTG
4087 015772 000402 BR OVER ;GET AROUND SAVE R0,R4
4088 015774 010046 $CNTLU: MOV R0,-(SP) ;SAVE R0
4089 015776 010446 MOV R4,-(SP) ;SAVE R4
4090 016000 104002 OVER: TYPEM ;ALLOW SWR= TO BE TYPED OUT
4091 016002 020130 SWR
4092 016004 017746 163234 MOV @SR,-(SP) ;MOV CONTENTS OF SWR
4093 016010 004737 016222 JSR PC,TYPEM ;OCTAL TYPE OUT ROUTINE
4094 016014 005726 POPSP ;CORRECT STACK POINTER
4095 016016 104002 TYPEM ;ALLOW NEW= TO BE TYPED
4096 016020 020143 NEW
4097 016022 005037 001252 CLR TEMPST ;CLEAR TEMP STORAGE LOC
4098 016026 012737 000007 001254 MOV #7,COUNT ;SET UP TO ACCEPT 7 CHAR
4099 016034 104022 1$: INRDY ;IS CHARACTER THERE?
4100 016036 000776 BR 1$
4101 016040 013700 001124 MOV TEMPCH,R0 ;SET UP FOR PRINT ROUTINE
4102 016044 104011 PRINTC ;PRINT ROUTINE
4103 016046 013737 001124 001250 MOV TEMPCH,TIB ;STORE BUFFER
4104 016054 042737 177600 001250 BIC #177600,TIB ;STRIP OFF GARBAGE
4105 016062 122737 000025 001250 CMPB #25,TIB ;IS IT A ^U
4106 016070 001001 BNE 2$ ;BRANCH IF NOT
4107 016072 000742 3$: BR OVER ;START OVER
4108 016074 122737 000015 001250 2$: CMPB #15,TIB ;IS IT A <CR>
4109 016102 001005 BNE 4$ ;BRANCH IF NOT
4110 016104 104007 SCR_LF
4111 016106 022737 000007 001254 CMP #7,COUNT ;WAS <CR> FIRST CHAR
4112 016114 001033 BNE 7$ ;CHANGE SWREG IF NOT FIRST <CR>
4113 016116 122737 000060 001250 4$: CMPB #60,TIB ;IS IT LESS THAN 0
4114 016124 003004 BGT 5$ ;GO TO ? ROUTINE IF SO
4115 016126 122737 000067 001250 CMPB #67,TIB ;IS IT GREATER THAN 7
```

4116	016134	002003				BGE	6\$:GO TO ? ROUTINE IF SO
4117	016136	104002			5\$:	TYPEM			:SET UP FOR ? TYPEOUT
4118	016140	020153				QUEST			
4119	016142	000716				BR	OVER		:START INPUT STRING OVER
4120	016144	006337	001252		6\$:	ASL	TEMPST		:MULTIPLY BY 10
4121	016150	006337	001252			ASL	TEMPST		
4122	016154	006337	001252			ASL	TEMPST		
4123	016160	142737	000060	001250		BICB	#60,TIB		:CLEAR OF ASCII
4124	016166	153737	001250	001252		BISB	TIB,TEMPST		:MOV CHAR TO TEMPST
4125	016174	005337	001254			DEC	COUNT		:ONLY WANT 6 NUMBERS AND <CR>
4126	016200	001756				BEQ	5\$:IF = 7 TOO MANY NUMBERS
4127	016202	000714				BR	1\$:GET NEXT CHAR
4128	016204	013777	001252	163032	7\$:	MOV	TEMPST,@SR		:CHANGE SWR CONTENTS
4129	016212	012604				MOV	(SP)+,R4		:RESTORE REG 4
4130	016214	012600				MOV	(SP)+,R0		:RESTORE REG 0
4131	016216	104006				CRLF			:CARRIAGE RETURN
4132	016220	000002			OUT:	RTI			:RETURN TO PROGRAM
4133									
4134									
4135									
4136	016222	112737	000001	001256	TYPOC:	MOVB	#1,FILL		:SET THE ZERO FILL SWITCH
4137	016230	112737	000006	001261		MOVB	#6,MODE+1		:SET FOR SIX DIGITS
4138	016236	112737	000005	001262	TYPON:	MOVB	#5,CNT		:SET THE ITERATION COUNT
4139	016244	010046				MOV	R0,-(SP)		:SAVE REG 0
4140	016246	010346				MOV	R3,-(SP)		:SAVE R3
4141	016250	010446				MOV	R4,-(SP)		:SAVE R4
4142	016252	010546				MOV	R5,-(SP)		:SAVE R5
4143	016254	113704	001261			MOVB	MODE+1,R4		:GET THE NUMBER OF DIGITS TO TYPE
4144	016260	005404				NEG	R4		
4145	016262	062704	000006			ADD	#6,R4		:SUBTRACT IT FOR MAX. ALLOWED
4146	016265	110437	001260			MOVB	R4,MODE		:SAVE IT FOR USE
4147	016272	113704	001256			MOVB	FILL,R4		:GET THE ZERO FILL SWITCH
4148	016276	016605	000012			MOV	12(%),R5		:PICKUP THE INPUT NUMBER
4149	016302	005003				CLR	R3		:CLEAR THE OUTPUT WORD
4150	016304	006105			1\$:	ROL	R5		:ROTATE MSB INTO 'C'
4151	016306	000404				BR	3\$:GO DO MSB
4152	016310	006105			2\$:	ROL	R5		:FORM THIS DIGIT
4153	016312	006105				ROL	R5		
4154	016314	006105				ROL	R5		
4155	016316	010503				MOV	R5,R3		
4156	016320	006103			3\$:	ROL	R3		:GET LSB OF THIS DIGIT
4157	016322	105337	001260			DECB	MODE		:TYPE THIS DIGIT?
4158	016326	100012				BPL	7\$:BR IF NO
4159	016330	042703	177770			BIC	#177770,R3		:GET RID OF JUNK
4160	016334	001002				BNE	4\$:TEST FOR 0
4161	016336	005704				TST	R4		:SUPPRESS THIS 0
4162	016340	001403				BEQ	5\$:BR IF YES
4163	016342	005204			4\$:	INC	R4		:DON'T SUPPRESS ANYMORE 0'S
4164	016344	052703	000060			BIS	#60,R3		:MAKE THIS DIGIT ASCII
4165	016350	010300			5\$:	MOV	R3,R0		:SETUP FOR PRINT ROUTINE
4166	016352	104011				PRINTC			
4167	016354	105337	001262		7\$:	DECB	CNT		:COUNT BY 1
4168	016360	003353				BGT	2\$:BR IF MORE TO DO
4169	016362	002402				BLT	6\$:BR IF DONE
4170	016364	005204				INC	R4		:INSURE LAST DIGIT ISN'T A BLANK
4171	016366	000750				BR	2\$:GO DO THE LAST DIGIT

4172	016370	012605	68:	MOV	(SP)+,R5	:RESTORE R5
4173	016372	012604		MOV	(SP)+,R4	:RESTORE R4
4174	016374	012603		MOV	(SP)+,R3	:RESTORE R3
4175	016376	012600		MOV	(SP)+,R0	:RESTORE R0
4176	016400	000207		RTS	PC	:RETURN FROM SUBROUTINE PC
4177						
4178						
4179						
4180						
4181						
4182						

4239	016474	016707		
4240	016476	016707	.WORD	16707
4241	016500	016707	.WORD	16707
4242	016502	000020	.WORD	16707
4243	016502	016707	DH1102:	
4244	016504	016707	.WORD	16707
4245	016506	016707	.WORD	16707
4246	016510	016707	.WORD	16707
4247	016512	016707	.WORD	16707
4248	016514	016707	.WORD	16707
4249	016516	016707	.WORD	16707
4250	016520	016707	.WORD	16707
4251	016522	016707	.WORD	16707
4252	016524	016707	.WORD	16707
4253	016526	016707	.WORD	16707
4254	016530	016707	.WORD	16707
4255	016532	016707	.WORD	16707
4256	016534	016707	.WORD	16707
4257	016536	016707	.WORD	16707
4258	016540	016707	.WORD	16707
4259	016542	000020	DH1103:	
4260	016542	016707	.WORD	16707
4261	016544	016707	.WORD	16707
4262	016546	016707	.WORD	16707
4263	016550	016707	.WORD	16707
4264	016552	016707	.WORD	16707
4265	016554	016707	.WORD	16707
4266	016556	016707	.WORD	16707
4267	016560	016707	.WORD	16707
4268	016562	016707	.WORD	16707
4269	016564	016707	.WORD	16707
4270	016566	016707	.WORD	16707
4271	016570	016707	.WORD	16707
4272	016572	016707	.WORD	16707
4273	016574	016707	.WORD	16707
4274	016576	016707	.WORD	16707
4275	016600	016707	.WORD	16707
4276	016602	000020	DH1104:	
4277	016602	016707	.WORD	16707
4278	016604	016707	.WORD	16707
4279	016606	016707	.WORD	16707
4280	016610	016707	.WORD	16707
4281	016612	016707	.WORD	16707
4282	016614	016707	.WORD	16707
4283	016616	016707	.WORD	16707
4284	016620	016707	.WORD	16707
4285	016622	016707	.WORD	16707
4286	016624	016707	.WORD	16707
4287	016626	016707	.WORD	16707
4288	016630	016707	.WORD	16707
4289	016632	016707	.WORD	16707
4290	016634	016707	.WORD	16707
4291	016636	016707	.WORD	16707
4292	016640	016707	.WORD	16707
4293	016642	000020	DH1105:	
4294	016642	016707	.WORD	16707

4295	016644	016707		.WORD	16707
4296	016646	016707		.WORD	16707
4297	016650	016707		.WORD	16707
4298	016652	016707		.WORD	16707
4299	016654	016707		.WORD	16707
4300	016656	016707		.WORD	16707
4301	016660	016707		.WORD	16707
4302	016662	016707		.WORD	16707
4303	016664	016707		.WORD	16707
4304	016666	016707		.WORD	16707
4305	016670	016707		.WORD	16707
4306	016672	016707		.WORD	16707
4307	016674	016707		.WORD	16707
4308	016676	016707		.WORD	16707
4309	016700	016707		.WORD	16707
4310	016702	000020	DH1106:		
4311	016702	016707		.WORD	16707
4312	016704	016707		.WORD	16707
4313	016706	016707		.WORD	16707
4314	016710	016707		.WORD	16707
4315	016712	016707		.WORD	16707
4316	016714	016707		.WORD	16707
4317	016716	016707		.WORD	16707
4318	016720	016707		.WORD	16707
4319	016722	016707		.WORD	16707
4320	016724	016707		.WORD	16707
4321	016726	016707		.WORD	16707
4322	016730	016707		.WORD	16707
4323	016732	016707		.WORD	16707
4324	016734	016707		.WORD	16707
4325	016736	016707		.WORD	16707
4326	016740	016707		.WORD	16707
4327	016742	000020	DH1107:		
4328	016742	016707		.WORD	16707
4329	016744	016707		.WORD	16707
4330	016746	016707		.WORD	16707
4331	016750	016707		.WORD	16707
4332	016752	016707		.WORD	16707
4333	016754	016707		.WORD	16707
4334	016756	016707		.WORD	16707
4335	016760	016707		.WORD	16707
4336	016762	016707		.WORD	16707
4337	016764	016707		.WORD	16707
4338	016766	016707		.WORD	16707
4339	016770	016707		.WORD	16707
4340	016772	016707		.WORD	16707
4341	016774	016707		.WORD	16707
4342	016776	016707		.WORD	16707
4343	017000	016707		.WORD	16707
4344	017002	000020	DH1110:		
4345	017002	016707		.WORD	16707
4346	017004	016707		.WORD	16707
4347	017006	016707		.WORD	16707
4348	017010	016707		.WORD	16707
4349	017012	016707		.WORD	16707
4350	017014	016707		.WORD	16707

4351	017016	016707		.WORD	16707
4352	017020	016707		.WORD	16707
4353	017022	016707		.WORD	16707
4354	017024	016707		.WORD	16707
4355	017026	016707		.WORD	16707
4356	017030	016707		.WORD	16707
4357	017032	016707		.WORD	16707
4358	017034	016707		.WORD	16707
4359	017036	016707		.WORD	16707
4360	017040	016707		.WORD	16707
4361	017042	000020	DH1111:		
4362	017042	016707		.WORD	16707
4363	017044	016707		.WORD	16707
4364	017046	016707		.WORD	16707
4365	017050	016707		.WORD	16707
4366	017052	016707		.WORD	16707
4367	017054	016707		.WORD	16707
4368	017056	016707		.WORD	16707
4369	017060	016707		.WORD	16707
4370	017062	016707		.WORD	16707
4371	017064	016707		.WORD	16707
4372	017066	016707		.WORD	16707
4373	017070	016707		.WORD	16707
4374	017072	016707		.WORD	16707
4375	017074	016707		.WORD	16707
4376	017076	016707		.WORD	16707
4377	017100	016707		.WORD	16707
4378	017102	000020	DH1112:		
4379	017102	016707		.WORD	16707
4380	017104	016707		.WORD	16707
4381	017106	016707		.WORD	16707
4382	017110	016707		.WORD	16707
4383	017112	016707		.WORD	16707
4384	017114	016707		.WORD	16707
4385	017116	016707		.WORD	16707
4386	017120	016707		.WORD	16707
4387	017122	016707		.WORD	16707
4388	017124	016707		.WORD	16707
4389	017126	016707		.WORD	16707
4390	017130	016707		.WORD	16707
4391	017132	016707		.WORD	16707
4392	017134	016707		.WORD	16707
4393	017136	016707		.WORD	16707
4394	017140	016707		.WORD	16707
4395	017142	000020	DH1113:		
4396	017142	016707		.WORD	16707
4397	017144	016707		.WORD	16707
4398	017146	016707		.WORD	16707
4399	017150	016707		.WORD	16707
4400	017152	016707		.WORD	16707
4401	017154	016707		.WORD	16707
4402	017156	016707		.WORD	16707
4403	017160	016707		.WORD	16707
4404	017162	016707		.WORD	16707
4405	017164	016707		.WORD	16707
4406	017166	016707		.WORD	16707

4407	017170	016707	.WORD	16707
4408	017172	016707	.WORD	16707
4409	017174	016707	.WORD	16707
4410	017176	016707	.WORD	16707
4411	017200	016707	.WORD	16707
4412	017202	000020		
4413	017202	016707	DH1114: .WORD	16707
4414	017204	016707	.WORD	16707
4415	017206	016707	.WORD	16707
4416	017210	016707	.WORD	16707
4417	017212	016707	.WORD	16707
4418	017214	016707	.WORD	16707
4419	017216	016707	.WORD	16707
4420	017220	016707	.WORD	16707
4421	017222	016707	.WORD	16707
4422	017224	016707	.WORD	16707
4423	017226	016707	.WORD	16707
4424	017230	016707	.WORD	16707
4425	017232	016707	.WORD	16707
4426	017234	016707	.WORD	16707
4427	017236	016707	.WORD	16707
4428	017240	016707	.WORD	16707
4429	017242	000020		
4430	017242	016707	DH1115: .WORD	16707
4431	017244	016707	.WORD	16707
4432	017246	016707	.WORD	16707
4433	017250	016707	.WORD	16707
4434	017252	016707	.WORD	16707
4435	017254	016707	.WORD	16707
4436	017256	016707	.WORD	16707
4437	017260	016707	.WORD	16707
4438	017262	016707	.WORD	16707
4439	017264	016707	.WORD	16707
4440	017266	016707	.WORD	16707
4441	017270	016707	.WORD	16707
4442	017272	016707	.WORD	16707
4443	017274	016707	.WORD	16707
4444	017276	016707	.WORD	16707
4445	017300	016707	.WORD	16707
4446	017302	000020		
4447	017302	016707	DH1116: .WORD	16707
4448	017304	016707	.WORD	16707
4449	017306	016707	.WORD	16707
4450	017310	016707	.WORD	16707
4451	017312	016707	.WORD	16707
4452	017314	016707	.WORD	16707
4453	017316	016707	.WORD	16707
4454	017320	016707	.WORD	16707
4455	017322	016707	.WORD	16707
4456	017324	016707	.WORD	16707
4457	017326	016707	.WORD	16707
4458	017330	016707	.WORD	16707
4459	017332	016707	.WORD	16707
4460	017334	016707	.WORD	16707
4461	017336	016707	.WORD	16707
4462	017340	016707	.WORD	16707

4463	017342	000020		
4464	017342	016707		
4465	017344	016707	.WORD	16707
4466	017346	016707	.WORD	16707
4467	017350	016707	.WORD	16707
4468	017352	016707	.WORD	16707
4469	017354	016707	.WORD	16707
4470	017356	016707	.WORD	16707
4471	017360	016707	.WORD	16707
4472	017362	016707	.WORD	16707
4473	017364	016707	.WORD	16707
4474	017366	016707	.WORD	16707
4475	017370	016707	.WORD	16707
4476	017372	016707	.WORD	16707
4477	017374	016707	.WORD	16707
4478	017376	016707	.WORD	16707
4479	017400	016707	.WORD	16707

DH1117:


```
4520 .SBTTL DIAGNOSTIC MESSAGES
4521
4522
4523
4524 017442 007600 055103 040514 STARTM: .ASCII <ACRLF><17>/CZLADD/<ACRLF>
4525 017450 042104 100060
4526 017454 040514 033063 052040 .ASCII /LA36 TERMINAL DIAGNOSTIC/<ACRLF>
4527 017462 051105 044515 040516
4528 017470 020114 044504 043501
4529 017476 047516 052123 041511
4530 017504 200
4531 017505 104 030510 020061 .ASCIZ /DH11 & DJ11 INTERFACES/<ACRLF><12>
4532 017512 020046 045104 030461
4533 017520 044440 052116 051105
4534 017526 040506 042503 100123
4535 017534 000012
4536 017536 005200 047105 020104 ENDPAS: .ASCII <ACRLF><12>/END OF PASS /
4537 017544 043117 050040 051501
4538 017552 020123 040
4539 017555 060 030060 100060 PASMES: .ASCIZ /0000/<ACRLF><12>
4540 017562 000012
4541 017564 030060 042040 030510 DH11S: .ASCIZ /00 DH11'S UNDER TEST/<ACRLF><12>
4542 017572 023461 020123 047125
4543 017600 042504 020122 042524
4544 017606 052123 005200 000
4545 017613 060 020060 045104 DJ11S: .ASCIZ /00 DJ11'S UNDER TEST/<ACRLF><12>
4546 017620 030461 051447 052440
4547 017626 042116 051105 052040
4548 017634 051505 100124 000012
4549 017642 007600 052012 051505 HDRMSG: .ASCIZ <ACRLF><17><12>/TEST #/
4550 017650 020124 000043
4551 017654 030060 020060 047503 HDRO: .ASCIZ /000 COLUMNS/<ACRLF><12>
4552 017662 052514 047115 100123
4553 017670 000012
4554 017672 054524 042520 040440 EO25MA: .ASCII /TYPE ANY PRINTABLE CHARACTER /
4555 017700 054516 050040 044522
4556 017706 052116 041101 042514
4557 017714 041440 040510 040522
4558 017722 052103 051105 040
4559 017727 101 042116 046040 .ASCIZ /AND LISTEN FOR BELL...../
4560 017734 051511 042524 020116
4561 017742 047506 020122 042502
4562 017750 046114 027056 027056
4563 017756 027056 027056 027056
4564 017764 027056 027056 000056
4565 017772 047200 052117 042440 EO25MB: .ASCIZ <ACRLF>/NOT ENOUGH COLUMNS/<ACRLF>
4566 020000 047516 043525 020110
4567 020006 047503 052514 047115
4568 020014 100123 000
4569 020017 200 041505 047510 ECOEND: .ASCIZ <ACRLF>/ECHO TEST TERMINATED/<ACRLF>
4570 020024 052040 051505 020124
4571 020032 042524 046522 047111
4572 020040 052101 042105 000200
4573 020046 020040
4574 020050 020040 020040 EO23M: .ASCII / /
LINE5: .ASCII / / ;MSG FOR TEST E023
4575
```

```
4576 020054 020040 020040 020040 LINE5A: .ASCIZ / /<ACRLF>
4577 020062 020040 000200
4578 020066 007600 051412 046105 MMSG3: .ASCIZ <ACRLF><17><12>/SELECT TEST # /
4579 020074 041505 020124 042524
4580 020102 052123 021440 020040
4581 020110 000
4582
4583 020111 105 042526 116 EVEN: .ASCII /EVEN/
4584 020115 117 042104 040 ODD: .ASCII /ODD /
4585
4586 :*****
4587 :CHGDB
4588 :*****
4589 020121 040 057040 020107 LNTG: .ASCIZ / ^G /
4590 020126 000040
4591 020130 007600 020012 051440 SWR: .ASCIZ <ACRLF><17><12>/ SWR= /
4592 020136 051127 020075 000
4593 020143 040 047040 053505 NEW: .ASCIZ / NEW= /
4594 020150 020075 000
4595 020153 040 037440 020040 QUEST: .ASCIZ / ? /
4596 020160 000
4597 020162 .FVEN
4598
4599
4600
4601 000001 .END
```

ACRLF = 000200	1354#	2262	3667	3749	3753	3757	3829	3839	4044	4048	4524	4526	4531
ACTIV 001222	4536	4539	4541	4545	4549	4551	4565	4569	4576	4578	4591		
ADTEMP 004412	1457#	1619*	1680	1707	1774*	1793*							
BAR 001150	2280	2299#											
BITTAB 001162	1446#	1864*	2046	2053*									
BIT0 = 000001	1451#	1880	1907										
BIT1 = 000002	1335#												
BIT10 = 002000	1336#												
BIT11 = 004000	1345#												
BIT12 = 010000	1346#	1798											
BIT13 = 020000	1347#												
BIT14 = 040000	1348#	2099	2396	2533									
BIT15 = 100000	1349#												
BIT2 = 000004	1350#												
BIT3 = 000010	1337#												
BIT4 = 000020	1338#												
BIT5 = 000040	1339#												
BIT6 = 000100	1340#	1748											
BIT7 = 000200	1341#	2843											
BIT8 = 000400	1342#												
BIT9 = 001000	1343#	1613	1801	2056	2594								
BTOASC = 104015	1344#												
BUFR 012234	1371#	1569	1581	1609	3188								
BYCR 001146	3426	3452	3457	3472#									
CARA 001144	1445#	1862*	2051*										
CHAIN = 104001	1444#	1860*	2050*										
CHAINN 002034	1359#	2651	2679	2749	2819	2854	2914	2978	3045	3094	3113	3144	3191
CHAINY 002062	3225	3251	3277	3366	3469	3510	3539	3665	3726	3821	3948	4027	
CHARAC 006160	1643#	1996											
CHGD10 004572	1616	1661#	1803										
CHGD3 001316	2484	2616#											
CHGD4 002034	2384#												
CHGD5 003476	1495#												
CHGD6 014732	1642#												
CHGD7 015736	2028#												
CHGD9 001502	3878#												
CKSWR = 104024	4077#												
CLEAN = 104017	1541#												
CNT 001262	1378#	1643	2404	2448									
CNTDH 001154	1373#	1504	1665	1760									
CNTG 020121	1474#	4138*	4167*										
CNTLSW 001104	1448#	1834*	1843*	1871	1874*	1891*	1900*						
CNTLU = 104025	4086	4589#											
CNVCTR 004404	1428#	1505*	1645	1755*	1757*	1759*	1769*	1788*	1798	1801			
COUNT 001254	1379#	1548											
CR = 104014	2278*	2290*	2295#										
CRLF = 104006	1471#	4098*	4111	4125*									
CSR 001236	1370#	2338	2806	2814	2849	2866	2907	2971	3005	3022	3039	3177	3781
CURTST 001120	3800	3886	3908	3931									
DELAY = 104003	1364#	1582	1610	2182	2367	2648	2674	2741	2748	2900	2909	2913	2960
	2973	2977	3010	3027	3044	3090	3093	3110	3170	3246	3272	3443	3456
	3534	3557	3560	3573	3599	3614	3630	3647	3657	3820	3938	4025	4035
	4131												
	1463#	1925*	2056*	2057									
	1434#	1668	1775	1823*									
	1361#	1598	1718	1733	1745	1790	1805	2473	2505	2591	2893	2911	2953

FORWD = 104016	1372#	1667	1773															
FOUR = 104023	1377#	3888	3910	3932														
HDRMSG 017642	2355	4549#																
HDRO 017654	1567	2638	4551#															
HERE 010750	3138	3144#																
INRDY = 104022	1376#	2399	2536	3737	4099													
IOSW 001224	1458#	1487*	1489*	1576	1620	1647	1734	2044	2113	2391								
LF = 104010	1366#	2368	2818	2839	2853	2904	2970	3133	3180	3444	3782	3986	4026					
	4036																	
LCFNT 001132	1439#	2834*	2838	2842*	2843													
LINENO 001156	1449#	1781	1876*	1905*	2049	2062	2085	2106										
LINE2 007144	2709	2753#																
LINE3 007666	2836	2869#																
LINE5 020050	3369	4574#																
LINE5A 020054	3384	4576#																
LOGICA 010740	1403	3140#																
LPR 001142	1443#	1858*	1879*															
MACHER 000004	1394#	1518*	1531*	1631*														
MASK 001160	1450#	1881*	1908*	1909	2046	2053	2055											
MESG3 020066	1618	1792	4578#															
MG24 012042	3350	3356	3400#															
MG25 012046	3360	3404#																
MODE 001260	1473#	4137*	4143	4146*	4157*													
MONIC 011324	3285#	3343																
MOVNUM 011770	3344	3357	3361	3384#														
NEW 020143	4096	4593#																
NG 002634	1737	1754	1764	1772	1804#													
NRCRA 001140	1442#	1776	1837*	1838	1842*	1854	1923*	2080	2101									
NXTST 001110	1430#	1615*	1771*	1820	1822*													
ODD 020115	3393	4584#																
OPEN = 000000	1352#	1428	1429	1430	1431	1432	1433	1434	1435	1436	1437	1438	1439					
	1448	1449	1450	1457	1458	1459	1460	1461	1462	1469	1470	1471	1472					
	1473	1474	1475	1945	1946	1947	1948	1957	1958	1966	1967							
	4079	4081	4132#															
OUT 016220	3433	3452#																
OUTPUT 012152	4087	4090#	4107	4119														
OVER 016000	1437#	2125*	2130*	2133*	2135*	3389												
PARITY 001126	3162*	3184*	3186	3194#														
PASCNT 011124	3185	4539#																
PASMES 017555	1438#	2124*	2125	2126*	3371	3377*	3378*	3379*										
PCHAR 001130	1506	2235#																
PFAIL 004216	1355#	4094																
POPSP = 005726	1353#	1526	1538	1651	1679	1702												
POPSP2= 022626	1435#	3241*	3244*	3265*	3270*													
POSI 001122	1771	1934#																
PRGTAB 003246	1367#	2031	2184	2340	2344	2362	2366	2643	2671	2683	2688	2691	2713					
PRINTC= 104011	2732	2739	2745	2800	2813	2846	2848	2858	2860	2863	2884	2888	2889					
	2897	2903	2918	2944	2948	2949	2957	2963	2965	2967	2997	3017	3031					
	3067	3109	3130	3167	3174	3243	3267	3464	3532	3538	3551	3575	3578					
	3580	3583	3585	3591	3593	3595	3598	3605	3607	3610	3613	3622	3624					
	3626	3632	3634	3636	3639	3641	3643	3646	3649	3651	3653	3656	3660					
	3662	3664	3786	3789	3802	3805	3883	3885	3893	3899	3901	3903	3905					
	3921	3923	3927	3934	3945	3947	3977	3979	3990	3992	3995	4010	4015					
	4020	4022	4024	4030	4032	4034	4102	4166										
PRINTJ 005474	2393	2531#																
PRTMDR= 104012	1368#	2636	2661	2707	2796	2833	2880	2940	2993	3062	3106	3127	3163					

		2568*	2569	2584	2588	3218	3337	3340	3348	3351	3430	3432	3436	3442*
		3454	3455*	3458*	3498	3619	3628	3731	3734	3778	3797	3973	4001	4080
		4084	4101	4103										
TEMPST	001252	1470#	4097*	4120*	4121*	4122*	4124*	4128						
TENPWR	004410	2281*	2283	2287	2297#									
TERM	012224	2412	2455	2542	2573	3431	3467#							
TESTC =	104020	1374#	1721	1736										
TEST30	012740	1960	3520#	3521	3540									
TEST31	013122	1961	3571#	3572	3666									
TEST32	013740	1962	3716#	3717	3727									
TEST33	014170	1963	3768#	3769	3822									
TEST34	014724	1964	3874#	3875	3949									
TEST35	015274	1965	3965#	3966	4028									
TIB	001250	1469#	4103*	4104*	4105	4108	4113	4115	4123*	4124				
TIMER	001102	1426#	2217	3736										
TIMES	001264	1475#												
TTYCTL =	104004	1362#	1650											
TTYJ	002112	1679#	2000											
TTYJA	002114	1622	1680#	1796										
TTYJTL =	104005	1363#	1649											
TTY1	002142	1702#	1999											
TTY1A	002144	1621	1703#	1797										
TTY1B	002174	1685	1711#											
TTY1G	002500	1682	1708	1776#	1782	1786								
TTY1H	002540	1787#	1802	1808	2422	2465	2552	2583	3627	3733	3780	3799	4037	
TTY1L	002606	1777	1798#											
TYP	004114	1995	2150#											
TYPE =	104000	1358#	3345	3362	3493	3745								
TYPEM =	104002	1360#	1573	1583	1611	1617	1791	2250	2354	2637	3189	3223	3249	3275
		3364	3467	3490	3506	3508	3524	3527	3576	3586	3596	3608	3611	3615
		3637	3644	3654	3722	3724	3773	3792	3808	3809	3811	3814	3816	3817
		3968	3980	3996	4004	4085	4090	4095	4117					
TYPM	004144	1997	2175#											
TYPOC	016222	4093	4136#											
TYPON	016236	4138#												
T3OSC	013120	3528	3563#											
WIDTH	001112	1431#	1566*	2642	2798	2816	2835	2851	2881	2890	2901	2916	2941	2950
		2961	2995	3012	3029	3165	3172	3241	3265	3488	3548	3558	3881	3912
		3935	3942											
\$BTASC	004320	2008	2278#											
\$CKSWR	015736	2016	4078#											
\$CLFAN	002020	1631#	2010											
\$CNTLU	015774	2017	4088#											
\$CR	004506	2007	2343#											
\$CRLF	004474	2001	2338#											
\$ECHO	003516	2012	2044#	2047										
\$FORWD	002652	1820#	2009											
\$FOUR	003476	2015	2029#											
\$INRDY	003654	2013	2080#	2086										
\$LF	004476	2003	2339#											
\$PRHDR	004516	2005	2354#											
\$PRTC	004572	2004	2385#											
\$READ	003716	2006	2099#	2102	2107									
\$SCRLF	004456	2002	2325#											
\$TESTC	004424	2011	2310#											
.	= 020162	1387#	1391	1392#	1396#	1401#	1405#	1409#	1420#	2616#	2780#	3473#	3759#	3853#

4059# 4597#

. ABS. 020162 000

ERRORS DETECTED: 0

CZLADD.BIN,CZLADD.LST/CRF/SOL/NL:TOC=CZLADD.P11

RUN-TIME: 13 28 1 SECONDS

RUN-TIME RATIO: 369/44=8.2

CORE USED: 7K (13 PAGES)